Au-delà de la régression linéaire

GLM, régularisation et méthodes non paramétriques

Julien Miron

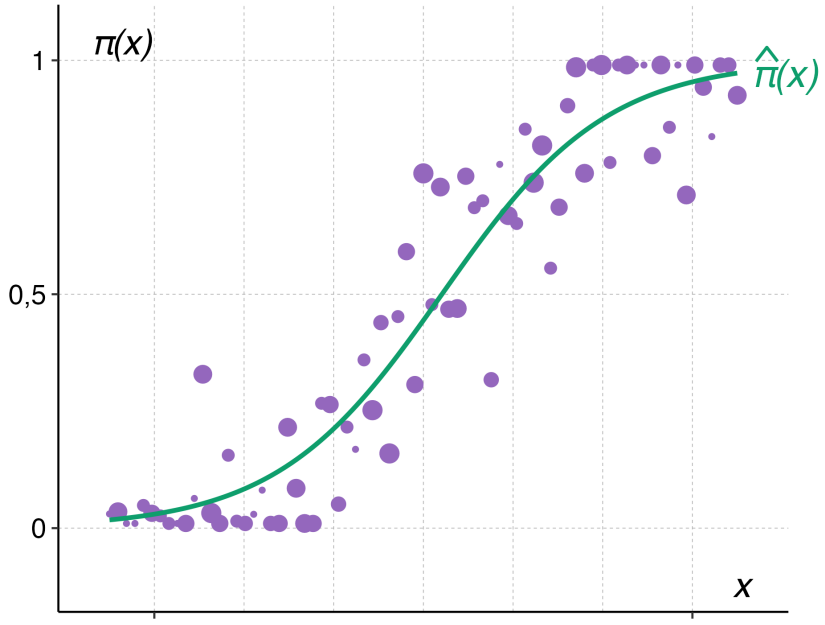


Table des matières

Préface **10**
Remerciements . . . 10
Provenance des données . . . 10
Librairie R du livre . . . 11
Utilisation de l'IA . . . 11

0 Rappels de régression linéaire **12**
0.1 Le modèle linéaire . . . 12
0.2 Estimation par moindres carrés . . . 13
0.3 Inférence statistique . . . 14
0.3.1 Intervalles de confiance et tests sur les coefficients . . . 14
0.3.2 Test F et comparaison de modèles emboîtés . . . 14
0.3.3 Coefficient de détermination . . . 14
0.4 Prédiction . . . 15
0.5 Diagnostics du modèle . . . 15
0.5.1 Résidus . . . 15
0.5.2 Observations aberrantes et influentes . . . 15
0.6 Sélection de variables : AIC et BIC . . . 17
0.7 Limites du modèle linéaire . . . 17
0.8 Exercices . . . 18
0.8.1 Niveau facile . . . 18
0.8.2 Niveau moyen . . . 19
0.8.3 Niveau difficile . . . 19
Références complémentaires . . . 20

1 Simulations de Monte Carlo **21**
1.1 Principe . . . 21
1.2 La recette . . . 22
1.3 Exemple : couverture d'un intervalle de confiance . . . 23
1.4 Précautions . . . 27
1.5 Exercices . . . 27
1.5.1 Niveau facile . . . 28
1.5.2 Niveau moyen . . . 28
1.5.3 Niveau difficile . . . 29
1.5.4 Exercices appliqués . . . 30

Références complémentaires . . . 31

2 Modèles de régression pour données binaires 32
2.1 Données et modèle . . . 34
2.1.1 Données individuelles (Bernoulli) . . . 34
2.1.2 Données groupées (binomiale) . . . 34
2.1.3 Fonctions de lien . . . 35
2.2 Inférence statistique . . . 37
2.2.1 Estimation des paramètres β . . . 37
2.2.2 Séparation des données . . . 39
2.2.3 Information de Fisher et variance des estimateurs . . . 42
2.2.4 Intervalles de confiance pour les β_j . . . 47
2.2.5 Test du score (Rao) . . . 54
2.2.6 Intervalle de confiance pour une probabilité prédite . . . 55
2.3 Sélection de modèles . . . 60
2.3.1 Déviance . . . 60
2.3.2 Comparaison de modèles emboîtés . . . 62
2.3.3 Comparaison de modèles non emboîtés : AIC et BIC . . . 63
2.3.4 Sélection automatique . . . 64
2.4 Validation et diagnostic de modèles . . . 66
2.4.1 Résidus . . . 66
2.4.2 Tests d'adéquation globale . . . 67
2.4.3 Adéquation avec des données individuelles . . . 69
2.4.4 Graphiques et surdispersion . . . 72
2.4.5 Observations aberrantes et valeurs influentes . . . 72
2.5 Spécificité, sensibilité et courbe ROC . . . 78
2.6 Exercices . . . 81
2.6.1 Niveau facile . . . 81
2.6.2 Niveau moyen . . . 82
2.6.3 Niveau difficile . . . 82
2.6.4 Exercices appliqués . . . 83
Références complémentaires . . . 85

3 Modèles linéaires généralisés (GLM) 86
3.1 Famille exponentielle . . . 86
3.1.1 Définition . . . 86
3.1.2 Mise en forme des lois usuelles . . . 88
3.2 Définition d'un modèle linéaire généralisé . . . 91
3.3 Estimation des paramètres . . . 92
3.3.1 Vraisemblance et score . . . 92
3.3.2 Loi asymptotique de $\widehat{\beta}$ . . . 93
3.3.3 Déviance individuelle . . . 93
3.3.4 Estimation de ϕ . . . 95

3.4 Inférence statistique . . . 97
3.4.1 Intervalles de confiance et tests pour β_j . . . 97
3.4.2 Intervalle de confiance pour une valeur prédite . . . 98
3.4.3 Comparaison de modèles emboîtés . . . 99
3.5 Sélection et validation de modèles . . . 99
3.6 Exemple détaillé : la régression de Poisson . . . 100
3.6.1 Variables explicatives *offset* . . . 101
3.6.2 Vraisemblance et score . . . 104
3.6.3 Information de Fisher et variance des estimateurs . . . 105
3.6.4 Déviance et adéquation . . . 106
3.6.5 Surdispersion . . . 107
3.6.6 Comparaison de modèles emboîtés . . . 110
3.6.7 Graphiques et observations aberrantes . . . 111
3.7 Exercices . . . 115
3.7.1 Niveau facile . . . 115
3.7.2 Niveau moyen . . . 115
3.7.3 Niveau difficile . . . 116
3.7.4 Exercices appliqués . . . 117
Références complémentaires . . . 118

4 Extensions des GLM : réponses catégorielles et excès de zéros 119
4.1 Réponses nominales : régression multinomiale . . . 119
4.1.1 Modèle . . . 119
4.1.2 Estimation . . . 120
4.2 Réponses ordinales : modèle à cotes proportionnelles . . . 124
4.2.1 Modèle . . . 124
4.2.2 Estimation et vérification de l'hypothèse . . . 125
4.3 Comptes avec excès de zéros . . . 127
4.3.1 Modèles à zéros excédentaires . . . 127
4.3.2 Modèles à barrière . . . 128
4.4 Exercices . . . 132
4.4.1 Niveau facile . . . 132
4.4.2 Niveau moyen . . . 133
4.4.3 Niveau difficile . . . 134
Références complémentaires . . . 134

5 Prédiction, mesures de performance et validation croisée 135
5.1 Adéquation versus précision de prédiction . . . 135
5.2 Validation croisée . . . 137
5.2.1 Approche par ensemble de validation . . . 138
5.2.2 Validation croisée avec exclusion d'une observation (LOOCV de *leave one out cross validation*) . . . 139
5.2.3 Validation croisée à K blocs . . . 140

5.3 Validation croisée pour les GLM . . . 142
5.3.1 Régression logistique et classification . . . 142
5.3.2 Régression de Poisson . . . 143
5.3.3 Approximation de $VC_{(n)}$ sans réajustement . . . 144
5.3.4 Exemples sur les données de l'Empress of Ireland . . . 145
5.4 Exercices . . . 149
5.4.1 Niveau facile . . . 149
5.4.2 Niveau moyen . . . 150
5.4.3 Niveau difficile . . . 151
5.4.4 Exercices appliqués . . . 152
Références complémentaires . . . 153

6 Sélection de variables et régularisation 154
6.1 Sélection de variables : rappel . . . 154
6.2 Régression régularisée . . . 154
6.2.1 Régression ridge . . . 157
6.2.2 Régression lasso . . . 160
6.2.3 Interprétation géométrique . . . 163
6.2.4 Trajectoires des coefficients . . . 164
6.2.5 Sélection du paramètre λ . . . 165
6.2.6 Elastic net . . . 167
6.3 Sélection de variables par décomposition spectrale . . . 168
6.3.1 Décomposition spectrale de la matrice de corrélation . . . 168
6.3.2 Indices de conditionnement et proportions de variance . . . 169
6.3.3 Méthodes de Jolliffe . . . 170
6.3.4 Lien avec ridge et la régression sur composantes principales . . . 170
6.4 Régularisation en régression logistique . . . 174
6.4.1 Log-vraisemblance pénalisée . . . 175
6.4.2 Estimation . . . 175
6.4.3 Choix de λ . . . 177
6.5 Exercices . . . 182
6.5.1 Niveau facile . . . 182
6.5.2 Niveau moyen . . . 183
6.5.3 Niveau difficile . . . 183
6.5.4 Exercices appliqués . . . 185
Références complémentaires . . . 185

7 Régression non paramétrique : k plus proches voisins et estimateur à noyau 186
7.1 Estimateur des k plus proches voisins (kNN) . . . 187
7.1.1 Définition . . . 187
7.1.2 Compromis biais-variance et choix de k . . . 188
7.1.3 Plusieurs variables explicatives et distance de Mahalanobis . . . 193

7.2 Estimateur à noyau (Nadaraya-Watson) . . . 194
7.2.1 Des voisins aux noyaux . . . 194
7.2.2 Choix du paramètre de fenêtre h . . . 195
7.3 Régression non paramétrique et classification . . . 199
7.4 Exercices . . . 202
7.4.1 Niveau facile . . . 202
7.4.2 Niveau moyen . . . 203
7.4.3 Niveau difficile . . . 204
Références complémentaires . . . 205

8 Régression non linéaire : polynômes, fonctions par morceaux et splines 206
8.1 Régression polynomiale . . . 206
8.2 Régression avec des fonctions constantes par morceaux . . . 214
8.3 Régression avec des splines . . . 216
8.3.1 Splines de régression . . . 216
8.3.2 Base des puissances tronquées . . . 218
8.3.3 B-splines . . . 220
8.3.4 Splines naturelles . . . 222
8.3.5 Splines de lissage (splines pénalisées) . . . 224
8.4 Exercices . . . 227
8.4.1 Exercices appliqués . . . 228
8.4.2 Exercices théoriques . . . 231
Références complémentaires . . . 232

9 Modèles additifs et modèles additifs généralisés (GAM) 234
9.1 Motivation . . . 234
9.2 Estimation par moindres carrés pénalisés . . . 234
9.2.1 Matrice d'influence et degrés de liberté effectifs . . . 235
9.2.2 Choix des paramètres de lissage . . . 235
9.2.3 Inférence . . . 235
9.3 Modèles additifs généralisés . . . 236
9.3.1 Algorithme d'ajustement (moindres carrés pondérés itératifs pénalisés) . . . 237
9.3.2 Choix des paramètres de lissage et comparaison de modèles . . . 237
9.3.3 En pratique avec R . . . 238
9.4 Exercices . . . 246
9.4.1 Exercices appliqués . . . 246
9.4.2 Exercices théoriques . . . 249
Références complémentaires . . . 250

10 Données corrélées : modèles mixtes et GEE 251
10.1 Conséquences de la dépendance . . . 251
10.2 Modèle linéaire mixte . . . 252
10.2.1 Modèle à ordonnée à l'origine aléatoire . . . 252

10.2.2 Forme générale . . . 253
10.2.3 Estimation . . . 254
10.2.4 Inférence sur les effets fixes . . . 255
10.2.5 Prédiction des effets aléatoires . . . 255
10.2.6 Tester une composante de variance . . . 256
10.2.7 Effets intra-groupes et inter-groupes . . . 260
10.2.8 Pentes aléatoires . . . 262
10.2.9 Diagnostics . . . 266
10.3 Effets aléatoires et pénalisation . . . 266
10.4 Modèle linéaire généralisé mixte . . . 270
10.5 Équations d'estimation généralisées (GEE) . . . 275
10.5.1 Choisir la structure de travail . . . 278
10.6 Exercices . . . 286
10.6.1 Niveau facile . . . 286
10.6.2 Niveau moyen . . . 287
10.6.3 Niveau difficile . . . 288
Références complémentaires . . . 289

11 Au-delà des modèles de ce livre **290**
11.1 Au-delà de l'espérance : GAMLSS et régression quantile . . . 290
11.1.1 GAMLSS . . . 290
11.1.2 Régression quantile . . . 291
11.2 Durées de vie : le modèle de Cox . . . 292
11.3 Pour aller plus loin . . . 292
11.4 Vue d'ensemble des modèles . . . 292

12 Ouverture sur la statistique robuste **298**
12.1 Pourquoi les moindres carrés sont fragiles . . . 298
12.2 Deux mesures de robustesse . . . 299
12.3 L'idée centrale : borner la contribution de chaque observation . . . 300
12.3.1 M-estimateurs . . . 300
12.3.2 Extension aux GLM . . . 302
12.4 Robustesse et modélisation . . . 303
Références complémentaires . . . 305

Solutions des exercices **306**
Chapitre 0 : Rappels de régression linéaire . . . 306
Niveau facile . . . 306
Niveau moyen . . . 307
Niveau difficile . . . 309
Chapitre 1 : Simulations de Monte Carlo . . . 311
Niveau facile . . . 311
Niveau moyen . . . 312

Niveau difficile . . . 313
Exercices appliqués . . . 315
Chapitre 2 : Modèles de régression pour données binaires . . . 319
Niveau facile . . . 320
Niveau moyen . . . 321
Niveau difficile . . . 322
Exercices appliqués . . . 324
Chapitre 3 : Modèles linéaires généralisés (GLM) . . . 335
Niveau facile . . . 335
Niveau moyen . . . 336
Niveau difficile . . . 337
Exercices appliqués . . . 340
Chapitre 4 : Extensions des GLM . . . 346
Niveau facile . . . 346
Niveau moyen . . . 347
Niveau difficile . . . 348
Chapitre 5 : Prédiction, mesures de performance et validation croisée . . . 350
Niveau facile . . . 350
Niveau moyen . . . 351
Niveau difficile . . . 352
Exercices appliqués . . . 353
Chapitre 6 : Sélection de variables et régularisation . . . 357
Niveau facile . . . 357
Niveau moyen . . . 358
Niveau difficile . . . 360
Exercices appliqués . . . 361
Chapitre 7 : k plus proches voisins et estimateur à noyau . . . 368
Niveau facile . . . 368
Niveau moyen . . . 369
Niveau difficile . . . 370
Chapitre 8 : Polynômes, fonctions par morceaux et splines . . . 372
Exercices appliqués . . . 372
Exercices théoriques . . . 386
Chapitre 9 : Modèles additifs et modèles additifs généralisés (GAM) . . . 388
Exercices appliqués . . . 388
Exercices théoriques . . . 401
Chapitre 10 : Données corrélées . . . 403
Niveau facile . . . 404
Niveau moyen . . . 405
Niveau difficile . . . 407

Références complémentaires **410**
Dunn et Smyth (2018). *Generalized Linear Models with Examples in R* . . . 410

Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R* . 410
Morris, White et Crowther (2019). « Using simulation studies to evaluate statistical methods » 411
McCullagh et Nelder (1989). *Generalized Linear Models* 411
Faraway (2016). *Extending the Linear Model with R* 411
Agresti (2013). *Categorical Data Analysis* 411
James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R* 412
Wood (2017). *Generalized Additive Models : An Introduction with R* 412
Heritier, Cantoni, Copt et Victoria-Feser (2009). *Robust Methods in Biostatistics* 412
Zeileis, Kleiber et Jackman (2008). « Regression Models for Count Data in R » 412
Agresti (2010). *Analysis of Ordinal Categorical Data* 413
Hastie, Tibshirani et Friedman (2009). *The Elements of Statistical Learning* . . 413
Hastie, Tibshirani et Wainwright (2015). *Statistical Learning with Sparsity : The Lasso and Generalizations* 413
Wasserman (2006). *All of Nonparametric Statistics* 413
Ruppert, Wand et Carroll (2003). *Semiparametric Regression* 413
Fitzmaurice, Laird et Ware (2011). *Applied Longitudinal Analysis* 414
Pinheiro et Bates (2000). *Mixed-Effects Models in S and S-PLUS* 414

Index thématique **418**

Index des fonctions R **420**

Préface

Ce livre porte sur la régression au-delà du modèle linéaire classique. Il commence par des rappels de régression linéaire (chapitre 0), suivis de la simulation de Monte Carlo (Chapitre 1), un outil qu'on réutilise en régression logistique. Viennent ensuite la régression logistique (Chapitre 2), puis les modèles linéaires généralisés (GLM) en général, avec la régression de Poisson comme exemple principal (Chapitre 3), et leurs extensions aux réponses catégorielles et aux comptes avec excès de zéros (Chapitre 4). Suivent la validation croisée (Chapitre 5), puis la sélection de modèles et la régularisation (ridge, lasso, Chapitre 6). En régression non paramétrique, on voit les k plus proches voisins et l'estimateur à noyau (Chapitre 7), puis les splines (Chapitre 8) et les modèles additifs généralisés (GAM, Chapitre 9). Viennent ensuite les modèles pour données corrélées, modèles mixtes et GEE (Chapitre 10). Le livre se termine avec deux chapitres d'ouverture, l'un sur des modèles qui dépassent le cadre du livre (Chapitre 11), l'autre sur la statistique robuste (Chapitre 12). Chaque chapitre se termine par un encadré « À retenir » et par des exercices, dont les solutions sont regroupées en annexe (Solutions des exercices). Suivent des exemples interactifs et des références complémentaires.

Il reste certainement des coquilles. Si vous en trouvez, écrivez-moi un courriel à julien.miron@mat.ulaval.ca : c'est la façon la plus rapide de les faire corriger.

Pour en savoir plus sur moi et sur mes autres projets, vous pouvez consulter mon site personnel sur www.julienmiron.ca.

Remerciements

Ce livre s'appuie en partie sur du matériel de cours développé par Lajmi Lakhal Chaieb (Université Laval) et par Eva Cantoni (Université de Genève). Je remercie aussi Guillaume Blanc et Benjamin Poilane pour leur révision méticuleuse.

Provenance des données

Une partie des jeux de données utilisés dans ce livre provient de donneesbleues.ca.

Librairie R du livre

Les jeux de données propres au livre (`empress_individus`, `empress_agrege`, `arbres` et `qualite_air`) sont aussi offerts dans la librairie R `ulavalreg`, avec quelques fonctions utilisées dans le livre, comme le test de Hosmer-Lemeshow (`hosmer_lemeshow()`) et les résidus groupés (`residus_groupes()`). La librairie évite de télécharger les fichiers CSV et de gérer les chemins d'accès et l'encodage. Elle s'installe à partir de GitHub :

```
# install.packages("remotes")
remotes::install_github("JulienMiron/ulavalreg")
```

Ensuite, `library(ulavalreg)` donne accès aux données, par exemple avec `data(empress_individus)`.

Utilisation de l'IA

Ce livre a été préparé avec l'aide de Claude, un modèle de langage développé par Anthropic (plusieurs versions, de Claude Sonnet 5 à Claude Opus 5.5). Claude m'a surtout aidé à rédiger le chapitre sur les simulations de Monte Carlo, ceux sur les extensions des GLM et sur les données corrélées, les deux chapitres d'ouverture (autres modèles et statistique robuste) ainsi que la majorité des exercices, avec leurs solutions. Il a aussi facilité l'écriture de plusieurs exemples en R, notamment ceux des chapitres sur les k plus proches voisins, les splines et les GAM. Le choix des sujets et l'organisation du livre sont les miens.

La prose de l'ensemble du livre a été retravaillée avec Claude, à partir d'un profil établi sur mes propres textes, pour qu'elle ressemble le plus possible à ma façon d'écrire. Claude a aussi fait deux relectures complètes pour repérer les erreurs. J'ai vérifié et validé l'ensemble du contenu, et j'en assume la responsabilité.

0 Rappels de régression linéaire

Ce chapitre rappelle ce qu'il faut connaître de la régression linéaire pour la suite du livre : le modèle, l'estimation par moindres carrés, l'inférence sur les coefficients, la prédiction, les diagnostics et les critères de sélection. Rien ici n'est nouveau si vous avez suivi un cours de régression linéaire, et le chapitre sert surtout à fixer la notation.

0.1 Le modèle linéaire

On dispose de données $\{(Y_i, X_i),\, i = 1, \dots, n\}$, où Y_i est la variable réponse (continue) et $X_i = (1, x_{i1}, \dots, x_{ip})^\top$ est le vecteur de dimension $p+1$ formé d'un 1 (pour l'ordonnée à l'origine) et des p variables explicatives [1]. Le modèle de régression linéaire multiple s'écrit

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, \qquad i = 1, \dots, n,$$

où $\beta_0, \beta_1, \dots, \beta_p$ sont les coefficients de régression et où les erreurs ϵ_i sont indépendantes et identiquement distribuées selon $\mathcal{N}(0, \sigma^2)$.

Définition — Matrice de design

En écriture matricielle, $Y = X\beta + \epsilon$, où $Y = (Y_1, \dots, Y_n)^\top$, $\beta = (\beta_0, \dots, \beta_p)^\top$ et $\epsilon = (\epsilon_1, \dots, \epsilon_n)^\top$. La matrice X est la matrice de design (*design matrix*) [a], de dimension $n \times (p+1)$, dont la première colonne ne contient que des 1 pour inclure l'ordonnée à l'origine :

$$X = \begin{pmatrix} X_1^\top \\ \vdots \\ X_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}.$$

a. Les termes anglais sont donnés entre parenthèses tout au long du livre : comme la plupart des références en statistique sont en anglais, il est utile de les reconnaître rapidement.

Sous ces hypothèses, on a $\mu_i = E[Y_i | x_{i1}, \dots, x_{ip}] = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ et $Var[Y_i] = \sigma^2$, de sorte que la variance ne dépend pas de i (homoscédasticité).

1. En régression, les variables explicatives sont souvent supposées connues, même si on peut aussi les traiter comme des variables aléatoires. La notation garde les majuscules pour ces vecteurs, même s'ils contiennent des constantes et non des variables aléatoires.

0.2 Estimation par moindres carrés

On estime β en minimisant la somme des carrés des écarts

$$SSE(\beta) = \sum_{i=1}^{n} \left\{ Y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \right\}^2 = (Y - X\beta)^\top (Y - X\beta).$$

Si X est de plein rang ($\text{rang}(X) = p+1$, ce qui exige $n \geq p+1$; l'estimation de σ^2 plus bas demandera $n > p+1$), la solution est explicite :

$$\boxed{\hat{\beta} = (X^\top X)^{-1} X^\top Y}$$

et l'estimateur obtenu suit la loi

$$\hat{\beta} \sim \mathcal{N}\left\{ \beta,\, \sigma^2 (X^\top X)^{-1} \right\}.$$

Comme les erreurs sont normales, l'estimateur des moindres carrés est aussi l'estimateur du maximum de vraisemblance. Il est sans biais et de variance minimale parmi les estimateurs linéaires sans biais (théorème de Gauss-Markov). Notez que le théorème de Gauss-Markov ne demande pas la normalité, qui sert plutôt à l'inférence (plus bas).

> **Définition — Matrice chapeau et leviers**
> Les valeurs prédites sont $\hat{Y} = X\hat{\beta} = HY$, où $H = X(X^\top X)^{-1} X^\top$ est la matrice chapeau (*hat matrix*). Les éléments diagonaux $h_i = [H]_{ii}$ sont les leviers (*leverage*) des observations.

L'estimateur sans biais usuel de σ^2 est

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n - (p+1)} = \frac{e^\top e}{n - (p+1)},$$

où $e_i = Y_i - \hat{Y}_i$ sont les résidus (voir la section sur les diagnostics du modèle, plus bas). On note $SSE = SSE(\hat{\beta}) = \sum_{i=1}^{n} e_i^2 = e^\top e$ la somme des carrés résiduels. L'estimateur du maximum de vraisemblance divise plutôt par n :

$$\hat{\sigma}^2_{MV} = \frac{e^\top e}{n}.$$

Il est biaisé, mais la différence est négligeable dès que n est grand devant p.

0.3 Inférence statistique

0.3.1 Intervalles de confiance et tests sur les coefficients

Un intervalle de confiance à $100 \times (1-\alpha)\%$ pour β_j est

$$\left[\hat{\beta}_j - t_{\alpha/2,\, n-(p+1)}\, se(\hat{\beta}_j)\,;\ \hat{\beta}_j + t_{\alpha/2,\, n-(p+1)}\, se(\hat{\beta}_j)\right],$$

où $se(\hat{\beta}_j) = \sqrt{\widehat{\sigma}^2 \left[(X^\top X)^{-1}\right]_{j+1,j+1}}$ et où $t_{\gamma,\nu}$ est le quantile d'ordre $1-\gamma$ de la loi de Student à ν degrés de liberté. C'est le $(j+1)^{\text{e}}$ élément de la diagonale de $(X^\top X)^{-1}$ qui intervient, parce que l'indice des coefficients commence à 0.

Le test de $H_0 : \beta_j = 0$ contre $H_1 : \beta_j \neq 0$ repose sur la statistique $t_{obs} = \hat{\beta}_j / se(\hat{\beta}_j)$, qui suit une loi $t_{n-(p+1)}$ sous H_0. On rejette H_0 au seuil α si $|t_{obs}| > t_{\alpha/2,\, n-(p+1)}$. La valeur-p de ce test est celle que R affiche dans la colonne `Pr(>|t|)` de `summary(lm(...))`.

0.3.2 Test F et comparaison de modèles emboîtés

> **Définition — Modèles emboîtés**
> Deux modèles sont emboîtés si l'un est complètement inclus dans l'autre, c'est-à-dire si le modèle réduit s'obtient en fixant à 0 certains coefficients du modèle complet.

La comparaison d'un modèle réduit (à $q+1$ paramètres) et d'un modèle complet dans lequel il est emboîté (à $p+1$ paramètres, $q < p$) se fait avec la statistique

$$F_{obs} = \frac{(SSE_{\text{réduit}} - SSE_{\text{complet}})/(p-q)}{SSE_{\text{complet}}/(n-(p+1))},$$

qui suit une loi $F_{p-q,\, n-(p+1)}$ sous l'hypothèse nulle que le modèle réduit est adéquat. Une grande valeur de F_{obs} signifie que les $p-q$ variables retirées expliquaient quelque chose.

0.3.3 Coefficient de détermination

Le coefficient de détermination

$$R^2 = 1 - \frac{SSE}{SST}, \qquad SST = \sum_{i=1}^{n} (Y_i - \bar{Y})^2,$$

mesure la proportion de la variabilité de Y expliquée par le modèle. Mais R^2 ne diminue jamais quand on ajoute une variable, même inutile, ce qui explique qu'en régression multiple, le R^2 ajusté soit préférable :

$$R^2_{adj} = 1 - \frac{SSE/(n-(p+1))}{SST/(n-1)}.$$

0.4 Prédiction

Pour une nouvelle observation $X^* = (1, x_1^*, \dots, x_p^*)^\top$, la valeur prédite est $\hat{Y}^* = X^{*\top}\hat{\beta}$ et $Var[\hat{Y}^*] = \sigma^2 X^{*\top}(X^\top X)^{-1}X^*$. Il faut distinguer deux intervalles :

— l'intervalle de confiance pour la moyenne $E[Y^*|x_1^*, \dots, x_p^*]$:

$$X^{*\top}\hat{\beta} \pm t_{\alpha/2,\, n-(p+1)}\, \hat{\sigma}\sqrt{X^{*\top}(X^\top X)^{-1}X^*};$$

— l'intervalle de prédiction pour une observation individuelle Y^* en X^* :

$$X^{*\top}\hat{\beta} \pm t_{\alpha/2,\, n-(p+1)}\, \hat{\sigma}\sqrt{1 + X^{*\top}(X^\top X)^{-1}X^*}.$$

Le second est toujours plus large que le premier parce qu'en plus de l'incertitude sur $\hat{\beta}$, il tient compte de l'erreur ϵ^* de la nouvelle observation.

0.5 Diagnostics du modèle

0.5.1 Résidus

Définition — Résidus, résidus standardisés et studentisés

Les résidus sont $e_i = Y_i - \hat{Y}_i$ et les résidus standardisés sont $r_i = e_i/(\hat{\sigma}\sqrt{1-h_i})$. Les résidus studentisés sont calculés de la même façon, mais en excluant l'observation i de l'estimation de σ.

On vérifie les hypothèses du modèle avec des graphiques des résidus, un par hypothèse :

— linéarité : résidus contre valeurs ajustées, sans tendance ;
— homoscédasticité : dispersion constante des résidus ;
— normalité : diagramme quantile-quantile ;
— indépendance : résidus dans l'ordre des observations.

Il existe un test formel pour chacune de ces hypothèses, mais le graphique est en général plus informatif : il montre comment l'hypothèse est violée, pas seulement si elle l'est.

0.5.2 Observations aberrantes et influentes

Le levier h_i mesure le potentiel d'influence de l'observation i, c'est-à-dire à quel point ses variables explicatives sont loin de celles des autres. Une valeur qui dépasse $2(p+1)/n$ mérite un examen.

Définition — Distance de Cook

La distance de Cook,

$$D_i = \frac{r_i^2}{p+1} \cdot \frac{h_i}{1-h_i},$$

mesure l'influence globale de l'observation i sur les coefficients estimés. Elle combine le résidu et le levier, parce qu'une observation n'est vraiment influente que si elle a à la fois un grand résidu et un grand levier.

Définition — DFBETA et DFFIT

Les DFBETA et les DFFIT mesurent l'effet du retrait de l'observation i, respectivement sur les coefficients et sur la valeur ajustée $\hat{Y}_i$, dans les unités d'origine :

$$\text{DFBETA}_i = \hat{\beta} - \hat{\beta}_{(-i)} = \frac{(X^\top X)^{-1} X_i \, e_i}{1-h_i},$$

$$\text{DFFIT}_i = \hat{Y}_i - \hat{Y}_{i(-i)} = \frac{h_i \, e_i}{1-h_i},$$

où $\hat{\beta}_{(-i)}$ et $\hat{Y}_{i(-i)}$ sont obtenus sans l'observation i. Notez que DFBETA_i est un vecteur de dimension $p+1$, dont on note DFBETA_{ij} l'élément associé à β_j.

Ces différences dépendent des unités de mesure des variables, ce qui rend difficile de fixer un seuil. Les logiciels rapportent plutôt les versions standardisées.

Définition — DFBETAS et DFFITS

Les DFBETAS et les DFFITS divisent les différences précédentes par l'écart-type correspondant, estimé sans l'observation i :

$$\text{DFBETAS}_{ij} = \frac{\text{DFBETA}_{ij}}{\hat{\sigma}_{(-i)} \sqrt{\left[(X^\top X)^{-1}\right]_{j+1,j+1}}},$$

$$\text{DFFITS}_i = \frac{\text{DFFIT}_i}{\hat{\sigma}_{(-i)} \sqrt{h_i}},$$

où $\hat{\sigma}_{(-i)}$ est l'estimation de σ obtenue sans l'observation i (celle des résidus studentisés).

En R, `dfbeta()` donne les DFBETA (non standardisés), alors que `dfbetas()`, `dffits()` et `influence.measures(...)` donnent les versions standardisées. Il faut donc vérifier laquelle des deux versions un logiciel affiche avant de comparer les valeurs à un seuil.

0.6 Sélection de variables : AIC et BIC

Les critères d'information servent à comparer des modèles, emboîtés ou non :

$$AIC = -2\,\ell(\hat{\beta}, \hat{\sigma}^2_{MV};\, y) + 2\,(p+1),$$
$$BIC = -2\,\ell(\hat{\beta}, \hat{\sigma}^2_{MV};\, y) + \log(n)\,(p+1),$$

où $\ell(\hat{\beta}, \hat{\sigma}^2_{MV};\, y)$ est la log-vraisemblance maximisée [2]. On préfère le modèle qui a le plus petit AIC (ou BIC). Le BIC pénalise plus fortement les modèles complexes dès que $n \geq 8$, parce que $\log(8) > 2$.

Les procédures de sélection automatique *forward* (ajout progressif), *backward* (retrait progressif) et *stepwise* (mixte) parcourent l'espace des modèles en se basant sur ces critères. En R, on utilise la fonction `step`. On revient sur la sélection de modèles au Chapitre 6.

0.7 Limites du modèle linéaire

Le modèle linéaire classique repose sur deux hypothèses restrictives :

1. la variable réponse est continue et normalement distribuée ;
2. la relation entre $E[Y]$ et les variables explicatives est linéaire.

Dans beaucoup d'applications, ni l'une ni l'autre ne tient. Par exemple, la réponse peut être binaire (défaut de paiement, présence d'une maladie) ou un dénombrement (nombre de cas de cancer), et la relation peut être franchement non linéaire. Les chapitres suivants lèvent ces restrictions une à la fois : les modèles linéaires généralisés traitent la première (Chapitre 2 et Chapitre 3), puis les méthodes non paramétriques et additives traitent la seconde (Chapitre 7 à Chapitre 9).

> **Remarque — À retenir**
> — Moindres carrés : si X est de plein rang, $\hat{\beta} = (X^\top X)^{-1} X^\top Y$, qui est aussi l'estimateur du maximum de vraisemblance quand les erreurs sont normales. L'estimateur sans biais de σ^2 divise $e^\top e$ par $n-(p+1)$, celui du maximum de vraisemblance par n ;
> — Confiance ou prédiction : l'intervalle de prédiction pour une nouvelle observation Y^* est toujours plus large que l'intervalle de confiance pour la moyenne $E[Y^*|x^*_1, \ldots, x^*_p]$, parce qu'il tient compte de l'erreur ϵ^* de la nouvelle observation ;

2. En R, `AIC()` et `BIC()` comptent aussi σ^2 parmi les paramètres et utilisent donc $p+2$ au lieu de $p+1$. Les valeurs diffèrent d'une constante, mais le classement des modèles est le même. La fonction `step`, elle, compte $p+1$ paramètres.

— Comparaison de modèles : le test F compare deux modèles emboîtés et, en régression multiple, le R^2 ajusté est préférable au R^2, qui ne diminue jamais quand une variable est ajoutée, même inutile. L'AIC et le BIC comparent aussi des modèles non emboîtés (le plus petit est préféré) et servent à la sélection pas à pas avec `step` ;
— Diagnostics : les hypothèses se vérifient avec un graphique de résidus par hypothèse (linéarité, homoscédasticité, normalité, indépendance). Un levier h_i supérieur à $2(p+1)/n$ mérite un examen, et la distance de Cook combine le résidu et le levier ;
— Versions standardisées : `dfbeta()` donne les DFBETA dans les unités d'origine, alors que `dfbetas()`, `dffits()` et `influence.measures()` donnent les versions standardisées. Vérifiez laquelle est affichée avant de comparer à un seuil.

0.8 Exercices

Les exercices sont classés par niveau de difficulté. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

0.8.1 Niveau facile

Exercice 1. Soit le modèle de régression simple $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, pour $i = 1, \ldots, n$.

a) Écrivez la matrice de design X, puis calculez explicitement $X^\top X$ et $X^\top Y$ en fonction de n, $\bar{x}$, $\bar{Y}$, $\sum x_i^2$ et $\sum x_i Y_i$.
b) À partir de $\hat{\beta} = (X^\top X)^{-1} X^\top Y$, retrouvez les formules usuelles

$$\hat{\beta}_1 = \frac{\sum_i (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_i (x_i - \bar{x})^2} \qquad \text{et} \qquad \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}.$$

Voir la solution

Exercice 2. Lorsque le modèle contient une ordonnée à l'origine, les équations normales s'écrivent

$$X^\top (Y - X\hat{\beta}) = 0.$$

Notons $e_i = Y_i - \hat{Y}_i$ le résidu de l'observation i. Montrez que ces équations impliquent que $\sum_{i=1}^n e_i = 0$ et que $\sum_{i=1}^n x_{ij} e_i = 0$ pour chaque variable explicative j. Qu'en déduisez-vous sur la moyenne des valeurs prédites $\hat{Y}_i$?

Voir la solution

Exercice 3. Considérez l'ajout d'une variable explicative supplémentaire à un modèle de régression linéaire multiple. Montrez algébriquement (à partir de la définition de SSE comme

minimum d'une somme de carrés) que le R^2 ne peut pas diminuer. Expliquez ensuite, sans calcul, pourquoi le R^2 ajusté peut, lui, diminuer.

Voir la solution

0.8.2 Niveau moyen

Exercice 1. Soit $H = X(X^\top X)^{-1}X^\top$ la matrice chapeau (*hat matrix*). Montrez que H est symétrique et idempotente ($H^2 = H$), puis que $I - H$ l'est également. Déduisez-en que $Var[e] = \sigma^2(I - H)$, où $e = Y - \hat{Y}$, et que $Var[e_i] = \sigma^2(1 - h_i)$.

Voir la solution

Exercice 2. En supposant $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, écrivez la vraisemblance du modèle linéaire et dérivez les estimateurs du maximum de vraisemblance $\hat{\beta}_{MV}$ et $\hat{\sigma}^2_{MV}$. Montrez que $\hat{\beta}_{MV}$ coïncide avec l'estimateur des moindres carrés, mais que $\hat{\sigma}^2_{MV}$ est biaisé. Précisez ensuite le lien entre $\hat{\sigma}^2_{MV}$ et l'estimateur sans biais $\hat{\sigma}^2$.

Voir la solution

Exercice 3. Montrez que $E[SSE] = \sigma^2(n-(p+1))$, ce qui justifie que $\hat{\sigma}^2 = SSE/(n-(p+1))$ est un estimateur sans biais de σ^2. (Indice : utilisez le fait que $SSE = Y^\top(I - H)Y$ et la formule de l'espérance d'une forme quadratique.)

Voir la solution

0.8.3 Niveau difficile

Exercice 1. Démontrez le théorème de Gauss-Markov. Montrez que l'estimateur des moindres carrés $\hat{\beta}$ est celui de variance minimale parmi les estimateurs linéaires sans biais de β (BLUE, *best linear unbiased estimator*). (Indice : considérez un estimateur linéaire quelconque $\tilde{\beta} = CY$ sans biais. Écrivez $C = (X^\top X)^{-1}X^\top + D$ et montrez que $DX = 0$.)

Voir la solution

Exercice 2. Montrez que $(n-(p+1))\hat{\sigma}^2/\sigma^2 \sim \chi^2_{n-(p+1)}$ et que cette statistique est indépendante de $\hat{\beta}$. (Indice : utilisez le théorème de Cochran. Exploitez le fait que H et $I - H$ sont des projecteurs orthogonaux de rangs $p + 1$ et $n - (p + 1)$. Notez aussi que $\hat{\beta}$ et e sont des transformations linéaires non corrélées, donc indépendantes, du vecteur gaussien Y.)

Voir la solution

Exercice 3. En partant du rapport de vraisemblance entre le modèle réduit (à $q+1$ paramètres) et le modèle complet (à $p + 1$ paramètres, $q < p$), dérivez la statistique

$$F_{obs} = \frac{(SSE_{\text{réduit}} - SSE_{\text{complet}})/(p-q)}{SSE_{\text{complet}}/(n-(p+1))}$$

et montrez qu'elle suit une loi $F_{p-q,\, n-(p+1)}$ sous l'hypothèse nulle que le modèle réduit est adéquat. Précisez quelles hypothèses de normalité et d'indépendance sont nécessaires.

Voir la solution

Exercice 4. La distance de Cook est définie par

$$D_i = \frac{\sum_{k=1}^{n} (\hat{Y}_k - \hat{Y}_{k(-i)})^2}{(p+1)\hat{\sigma}^2},$$

où $\hat{Y}_{k(-i)}$ est la valeur ajustée en X_k obtenue sans l'observation i. À partir de cette définition, retrouvez la formule

$$D_i = \frac{r_i^2}{p+1} \cdot \frac{h_i}{1-h_i}.$$

Indice : utilisez la formule de mise à jour des moindres carrés lorsqu'une observation est retirée,

$$\widehat{\beta}_{(-i)} = \widehat{\beta} - \frac{(X^\top X)^{-1} X_i e_i}{1-h_i}.$$

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Dunn et Smyth (2018). *Generalized Linear Models with Examples in R*
— Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R*

1 Simulations de Monte Carlo

Ce chapitre utilise les librairies R suivantes.

```
library(ggplot2)
```

Ce chapitre présente la simulation de Monte Carlo. Il s'agit de fabriquer des jeux de données dont le vrai modèle est connu, d'y appliquer la méthode étudiée, puis de regarder ce qu'elle fait sur un grand nombre de répétitions. Cet outil est nécessaire parce que plusieurs résultats de ce livre ne sont connus que de façon asymptotique. C'est le cas de la loi de $\widehat{\beta}$ dans un modèle linéaire généralisé (GLM, Chapitre 3), de la validité des intervalles de Wald et du niveau d'un test du rapport de vraisemblance. Ces résultats disent ce qui se passe quand $n \to \infty$, mais pas si l'approximation est bonne pour le n dont on dispose.

J'illustre d'abord l'outil sur le modèle linéaire, où les résultats exacts sont connus, ce qui permet de vérifier que la simulation fonctionne. On s'en servira ensuite en régression logistique (Chapitre 2).

1.1 Principe

On cherche une quantité de la forme $\theta = E[g(Y)]$, où Y est un jeu de données généré par un modèle entièrement spécifié. La fonction g s'applique à ce jeu de données : un estimateur, l'indicatrice qu'un intervalle contient la vraie valeur, l'indicatrice qu'un test rejette H_0, etc. Cette espérance est rarement calculable explicitement.

Définition — Simulation de Monte Carlo

On génère B jeux de données indépendants $Y^{(1)}, \ldots, Y^{(B)}$ à partir du modèle spécifié, puis on estime θ par la moyenne

$$\widehat{\theta}_B = \frac{1}{B} \sum_{b=1}^{B} g\left(Y^{(b)}\right).$$

Si $E|g(Y)| < \infty$, la loi des grands nombres donne $\widehat{\theta}_B \to \theta$ quand $B \to \infty$.

Il ne faut pas voir $\hat{\theta}_B$ comme la valeur exacte de θ. Comme $\hat{\theta}_B$ est lui-même aléatoire, il comporte une erreur, appelée erreur de Monte Carlo.

Propriété — Erreur de Monte Carlo

L'estimateur $\hat{\theta}_B$ est sans biais pour θ et, si $Var[g(Y)] < \infty$,

$$Var\left[\hat{\theta}_B\right] = \frac{Var\left[g(Y)\right]}{B}.$$

Quand g est une indicatrice, θ est une probabilité (par exemple, la couverture réelle d'un intervalle), $Var[g(Y)] = \theta(1-\theta)$, et l'erreur-type de Monte Carlo s'estime alors par

$$\sqrt{\frac{\hat{\theta}_B\left(1-\hat{\theta}_B\right)}{B}}.$$

Prenons $B = 1000$ répétitions et une couverture réelle de 0,95 : l'erreur-type de Monte Carlo vaut alors $\sqrt{0{,}95 \times 0{,}05/1000} \approx 0{,}007$. Un résultat de 0,94 est donc compatible avec 0,95, alors qu'un résultat de 0,91 ne l'est pas. Notez que pour diviser l'erreur par deux, il faut multiplier B par quatre, parce que l'erreur-type décroît comme $1/\sqrt{B}$.

Un intervalle de confiance pour θ rend cette comparaison explicite. Comme $\hat{\theta}_B$ est une moyenne de B variables indépendantes, il est approximativement normal quand B est grand, et un intervalle de confiance à 95 % pour la couverture réelle est

$$\hat{\theta}_B \pm 1{,}96\sqrt{\frac{\hat{\theta}_B\left(1-\hat{\theta}_B\right)}{B}}.$$

Avec $B = 1000$, un résultat de 0,94 donne l'intervalle [0,925 ; 0,955], qui contient 0,95. Un résultat de 0,91 donne [0,892 ; 0,928], qui ne le contient pas.

Remarque — Erreur de Monte Carlo et erreur statistique

Le nombre de répétitions B contrôle la précision de la simulation, et rien d'autre : l'erreur de Monte Carlo peut être rendue aussi petite qu'on veut. La performance de la méthode étudiée, elle, dépend de la taille d'échantillon n de chaque jeu de données simulé, et B n'y change rien. Il ne faut pas confondre les deux. On augmente B pour mieux estimer la performance d'une méthode à n fixé, et on fait varier n pour voir comment cette performance change.

1.2 La recette

Une étude par simulation suit presque toujours les mêmes étapes :

1. fixer le vrai modèle : valeurs des paramètres, taille d'échantillon n, valeurs des variables explicatives et loi des erreurs ;
2. écrire une fonction qui génère un jeu de données à partir de ce modèle ;
3. répéter B fois : générer un jeu de données, appliquer la méthode étudiée (ajuster le modèle, construire un intervalle, faire un test) et conserver le résultat ;
4. résumer les B résultats par une moyenne, avec son erreur de Monte Carlo.

En R, on fixe la graine du générateur de nombres pseudo-aléatoires avec `set.seed()` avant la simulation, pour que les résultats soient reproductibles. La fonction `replicate()` répète une expression B fois.

Définition — Résumés usuels

Notons $\widehat{\beta}^{(b)}$ l'estimation obtenue sur le b^{e} jeu de données et $[L^{(b)}, U^{(b)}]$ l'intervalle de confiance correspondant. Pour un paramètre dont la vraie valeur est β, on estime

$$\begin{aligned} \text{le biais :} \quad & \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{\beta}^{(b)} - \beta\right), \\ \text{l'erreur quadratique moyenne :} \quad & \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{\beta}^{(b)} - \beta\right)^2, \\ \text{la couverture :} \quad & \frac{1}{B}\sum_{b=1}^{B}\mathbf{1}\left\{L^{(b)} \leq \beta \leq U^{(b)}\right\}. \end{aligned}$$

Un test se résume de la même façon par son taux de rejet. Calculé sur des données générées sous H_0, il estime le niveau réel du test, et sous une alternative, sa puissance.

1.3 Exemple : couverture d'un intervalle de confiance

Considérons le modèle $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ avec $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, où $\beta_0 = 1$, $\beta_1 = 2$ et $\sigma = 3$, et où les x_i sont n valeurs également espacées entre 0 et 1. Dans ce modèle, l'intervalle de confiance du chapitre 0, basé sur le quantile de la loi de Student, a une couverture exactement égale à $1 - \alpha$. On le vérifie par simulation, puis on le compare à l'intervalle où le quantile de Student est remplacé par celui de la loi normale, qui n'est valable qu'asymptotiquement.

Exemple — Distribution de $\hat{\beta}_1$ et couverture avec $n = 10$

On commence par définir la fonction qui génère un jeu de données, puis on répète $B = 2000$ fois l'ajustement du modèle, en conservant $\widehat{\beta}_1$ et son erreur-type.

```
simuler_lineaire <- function(n, beta0 = 1, beta1 = 2, sigma = 3) {
  x <- seq(0, 1, length.out = n)
```

```r
  y <- beta0 + beta1 * x + rnorm(n, mean = 0, sd = sigma)
  data.frame(x = x, y = y)
}

B <- 2000
n <- 10
beta1 <- 2

set.seed(4300)
resultats <- replicate(B, {
  dat <- simuler_lineaire(n, beta1 = beta1)
  coefs <- coef(summary(lm(y ~ x, data = dat)))["x", ]
  c(estimation = coefs[["Estimate"]], erreur_type = coefs[["Std.
   ↪  Error"]])
})
resultats <- as.data.frame(t(resultats))
```

On estime ensuite le biais de $\hat{\beta}_1$:

```r
mean(resultats$estimation) - beta1
```

```
[1] -0.0657941
```

Le biais estimé, $-0{,}066$, est compatible avec zéro, comme le prévoit la théorie. En effet, l'écart-type de $\hat{\beta}_1$ vaut ici $\sigma / \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \approx 2{,}97$, de sorte que l'erreur-type de Monte Carlo du biais est d'environ $2{,}97 / \sqrt{2000} \approx 0{,}066$.
L'histogramme ci-dessous regroupe les estimations de β_1, une par jeu de données simulé. La distribution est centrée sur la vraie valeur et a la forme d'une loi normale, ce qui correspond à la loi théorique $\hat{\beta}_1 \sim \mathcal{N}\left\{\beta_1, \sigma^2 \left[(X^\top X)^{-1}\right]_{2,2}\right\}$.

```r
ggplot(resultats, aes(x = estimation)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30,
                 fill = "grey80", color = "white") +
  geom_vline(xintercept = beta1, linewidth = 1) +
  labs(x = expression(hat(beta)[1]), y = "Densité") +
  theme_minimal(base_size = 13)
```

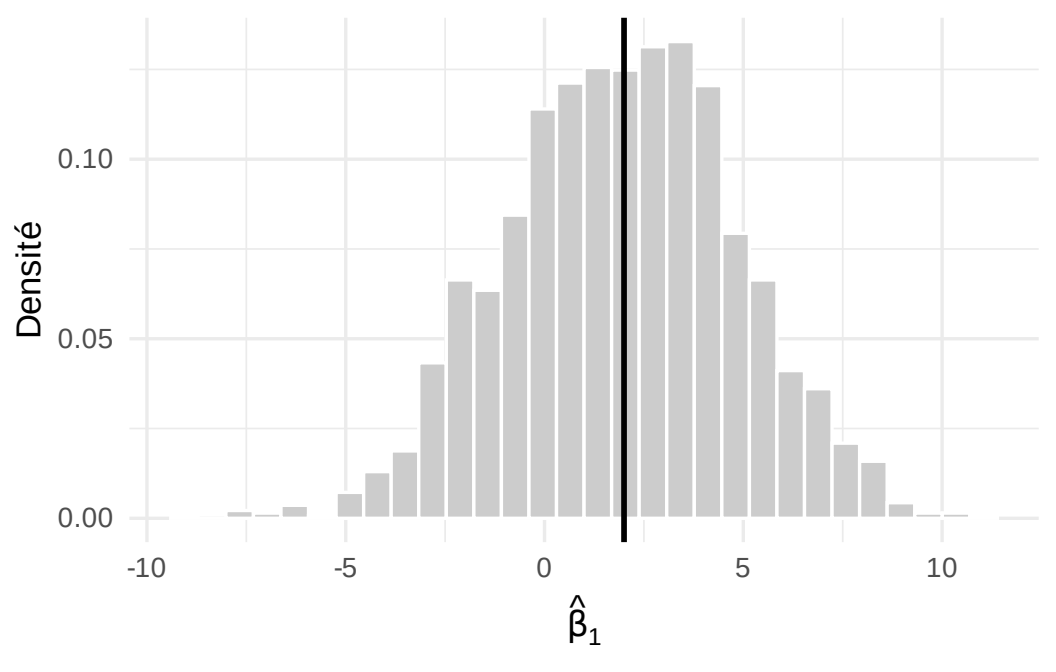


FIGURE 1.1 : Distribution de $\hat{\beta}_1$ sur 2000 jeux de données simulés ($n = 10$). La droite verticale indique la vraie valeur $\beta_1 = 2$.

Pour la couverture, on construit sur chaque jeu de données deux intervalles $\hat{\beta}_1 \pm q\, se(\hat{\beta}_1)$, où q est le quantile d'ordre 0,975, soit de la loi de Student à $n-2$ degrés de liberté, soit de la loi normale. La couverture estimée est la proportion des intervalles qui contiennent $\beta_1 = 2$.

```
couverture <- function(res, beta, quantile) {
  mean(abs(res$estimation - beta) <= quantile * res$erreur_type)
}

c(
  student = couverture(resultats, beta1, qt(0.975, df = n - 2)),
  normale = couverture(resultats, beta1, qnorm(0.975))
)
```

```
student normale
 0.9515  0.9180
```

Avec $B = 2000$, l'erreur-type de Monte Carlo sur ces proportions est d'environ 0,005 pour l'intervalle de Student et 0,006 pour celui basé sur la loi normale. Les intervalles

de confiance à 95 % pour les couvertures réelles sont [0,942 ; 0,961] pour l'intervalle de Student et [0,906 ; 0,930] pour celui basé sur la loi normale. La couverture de l'intervalle de Student est bien compatible avec 0,95, alors que celle de l'intervalle basé sur la loi normale est nettement trop faible. Avec $n = 10$, le quantile normal est trop petit, de sorte que les intervalles sont trop étroits.

Faisons maintenant varier la taille d'échantillon pour voir à partir de quand l'approximation normale devient acceptable.

Exemple — Couverture en fonction de n

```
couverture_n <- function(n, B = 2000, beta1 = 2) {
  res <- replicate(B, {
    dat <- simuler_lineaire(n, beta1 = beta1)
    coefs <- coef(summary(lm(y ~ x, data = dat)))["x", ]
    c(estimation = coefs[["Estimate"]], erreur_type = coefs[["Std.
     ↪  Error"]])
  })
  res <- as.data.frame(t(res))
  data.frame(
    n = n,
    student = couverture(res, beta1, qt(0.975, df = n - 2)),
    normale = couverture(res, beta1, qnorm(0.975))
  )
}

set.seed(4300)
do.call(rbind, lapply(c(5, 10, 20, 50, 100, 500), couverture_n))
```

```
    n student normale
1   5   0.956  0.8690
2  10   0.954  0.9210
3  20   0.951  0.9340
4  50   0.949  0.9420
5 100   0.957  0.9555
6 500   0.953  0.9525
```

La couverture de l'intervalle de Student reste autour de 0,95 quel que soit n, aux erreurs de Monte Carlo près. Celle de l'intervalle basé sur la loi normale s'en approche seulement à mesure que n augmente, ce qui est cohérent avec la convergence de la loi de Student vers la loi normale.

1.4 Précautions

Une simulation ne démontre rien. Elle décrit le comportement d'une méthode dans les configurations simulées, et seulement dans celles-là. Par exemple, une couverture de 0,95 obtenue avec $n = 100$, $\sigma = 3$ et des erreurs normales ne garantit rien avec des erreurs asymétriques, ni pour une autre taille d'échantillon. Il faut donc choisir des configurations représentatives de la situation qui nous intéresse et, dans la mesure du possible, en faire varier plusieurs.

Il faut aussi garder en tête les limites suivantes :

- le choix du vrai modèle est arbitraire : les conclusions valent pour le modèle simulé, qui n'est jamais exactement celui des données réelles ;
- les résultats sont aléatoires : on rapporte toujours l'erreur de Monte Carlo, ou on prend B assez grand pour qu'elle soit négligeable devant les différences qu'on cherche à détecter ;
- le code peut être faux : il faut d'abord le tester sur un cas dont la réponse est connue, comme on l'a fait ici avec l'intervalle de Student.

Remarque — À retenir

- Principe : une quantité $\theta = E[g(Y)]$ (biais, erreur quadratique moyenne, couverture, taux de rejet) s'estime par la moyenne $\hat{\theta}_B$ de g sur B jeux de données simulés à partir d'un modèle entièrement spécifié. Sous H_0, le taux de rejet estime le niveau réel d'un test, et sous une alternative, sa puissance ;
- Erreur de Monte Carlo : $\hat{\theta}_B$ est aléatoire. Pour une proportion, son erreur-type s'estime par $\sqrt{\hat{\theta}_B(1-\hat{\theta}_B)/B}$ et décroît comme $1/\sqrt{B}$, de sorte qu'il faut multiplier B par quatre pour la diviser par deux ;
- B ou n : B contrôle seulement la précision de la simulation, alors que la performance de la méthode dépend de la taille n de chaque jeu de données simulé. Augmentez B pour mieux estimer la performance à n fixé, et faites varier n pour voir comment elle change ;
- Recette en R : fixer le vrai modèle, écrire une fonction qui génère un jeu de données, répéter B fois la méthode avec `replicate()` après avoir fixé la graine avec `set.seed()`, puis résumer par une moyenne et son erreur de Monte Carlo ;
- Précautions : une simulation ne démontre rien, puisque ses conclusions ne valent que pour les configurations simulées. Testez d'abord le code sur un cas dont la réponse est connue, comme l'intervalle de Student dans le modèle linéaire.

1.5 Exercices

Les exercices sont classés par niveau de difficulté, et les exercices appliqués de la fin demandent d'analyser un jeu de données en R. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

1.5.1 Niveau facile

Exercice 1. Soit $\hat{\theta}_B = \frac{1}{B}\sum_{b=1}^{B} g(Y^{(b)})$, où les jeux de données $Y^{(1)}, \ldots, Y^{(B)}$ sont indépendants et générés par le même modèle. Notons $\theta = E[g(Y)]$ et $\tau^2 = Var[g(Y)] < \infty$.

a) Montrez que $\hat{\theta}_B$ est sans biais pour θ et que $Var[\hat{\theta}_B] = \tau^2/B$.
b) Lorsque g est une indicatrice, montrez que $\tau^2 = \theta(1-\theta)$ et que cette variance est au plus $1/4$.

Voir la solution

Exercice 2. On veut estimer la couverture réelle d'un intervalle de confiance dont le niveau nominal est de 95 %.

a) Si la couverture réelle est proche de 0,95, combien de répétitions B faut-il pour que l'erreur-type de Monte Carlo soit d'au plus 0,005 ?
b) Combien en faudrait-il si on ne savait rien de la couverture réelle ?

Voir la solution

Exercice 3. Une simulation de $B = 1\,000$ répétitions donne une couverture estimée de 0,93 pour un intervalle de niveau nominal 95 %. Ce résultat est-il compatible avec une couverture réelle de 0,95 ? Construisez aussi un intervalle de confiance approximatif à 95 % pour la couverture réelle.

Voir la solution

1.5.2 Niveau moyen

Exercice 1. Notons $\hat{\beta}^{(1)}, \ldots, \hat{\beta}^{(B)}$ les estimations obtenues sur B jeux de données simulés, $\bar{\beta} = \frac{1}{B}\sum_b \hat{\beta}^{(b)}$ leur moyenne et β la vraie valeur du paramètre.

a) Montrez que l'erreur quadratique moyenne estimée se décompose exactement en

$$\frac{1}{B}\sum_{b=1}^{B}\left(\hat{\beta}^{(b)} - \beta\right)^2 = \left(\bar{\beta} - \beta\right)^2 + \frac{1}{B}\sum_{b=1}^{B}\left(\hat{\beta}^{(b)} - \bar{\beta}\right)^2,$$

c'est-à-dire en un biais estimé au carré et une variance estimée.
b) Donnez une estimation de l'erreur-type de Monte Carlo du biais estimé $\bar{\beta} - \beta$.

Voir la solution

Exercice 2. Soit T une statistique de test continue dont la fonction de répartition sous H_0, notée F_0, est continue et strictement croissante. On rejette H_0 pour les grandes valeurs de T, avec la valeur-p $p = 1 - F_0(T)$.

a) Montrez que, sous H_0, la valeur-p suit une loi uniforme sur $[0,1]$.
b) Déduisez-en que le taux de rejet calculé par simulation sous H_0 estime bien le niveau réel du test, et donnez son erreur-type de Monte Carlo pour $\alpha = 0{,}05$ et $B = 2\,000$ lorsque le test a le bon niveau.
c) Que devient ce raisonnement lorsque F_0 n'est qu'une approximation asymptotique de la loi de T, comme pour le test de Wald dans un GLM ?

Voir la solution

Exercice 3. On veut comparer la couverture de deux intervalles, A et B, construits sur les mêmes jeux de données simulés. Notons $G_A^{(b)}$ et $G_B^{(b)}$ les indicatrices que chaque intervalle contient la vraie valeur pour le b^{e} jeu de données, et $\widehat{\theta}_A$ et $\widehat{\theta}_B$ les couvertures estimées.

a) Montrez que

$$Var[\widehat{\theta}_A - \widehat{\theta}_B] = \frac{Var[G_A] + Var[G_B] - 2\,Cov[G_A, G_B]}{B}.$$

b) Pourquoi vaut-il mieux évaluer les deux intervalles sur les mêmes jeux de données plutôt que sur des jeux simulés indépendamment pour chaque méthode ? Faites le lien avec la remarque du Chapitre 5 sur la réinitialisation de la graine avant chaque appel de `cv.glm`.

Voir la solution

1.5.3 Niveau difficile

Exercice 1. Pour une proportion binomiale, $Y \sim \text{Bin}(n, \pi)$, l'intervalle de Wald à 95 % est $\hat{\pi} \pm 1{,}96\sqrt{\hat{\pi}(1-\hat{\pi})/n}$, avec $\hat{\pi} = Y/n$.

a) Expliquez pourquoi la couverture réelle de cet intervalle peut se calculer exactement, sans simulation, et écrivez-la sous la forme d'une somme sur les valeurs possibles de Y.
b) Calculez la couverture réelle pour $n = 10$ et $\pi = 0{,}1$.
c) Une simulation de Monte Carlo avec $B = 10\,000$ donnerait-elle une estimation proche de cette valeur ? Avec quelle erreur-type ?

Voir la solution

Exercice 2. On cherche $\theta = E[g(U)] = \int_0^1 g(u)\,du$, avec $U \sim \mathcal{U}(0,1)$ et g une fonction monotone. Considérez deux estimateurs qui utilisent chacun $2B$ évaluations de g :

— l'estimateur usuel $\widehat{\theta}_{\text{ind}} = \frac{1}{2B}\sum_{b=1}^{2B} g(U_b)$, avec des U_b indépendants ;
— l'estimateur antithétique $\widehat{\theta}_{\text{anti}} = \frac{1}{B}\sum_{b=1}^{B} \frac{g(U_b)+g(1-U_b)}{2}$.

a) Montrez que les deux estimateurs sont sans biais et que

$$Var[\hat{\theta}_{\text{anti}}] = \frac{Var[g(U)] + Cov[g(U), g(1-U)]}{2B}, \qquad Var[\hat{\theta}_{\text{ind}}] = \frac{Var[g(U)]}{2B}.$$

b) Montrez que $Cov[g(U), g(1-U)] \leq 0$ lorsque g est monotone. (Indice : si V est une copie indépendante de U, montrez que $Cov[a(U), c(U)] = \frac{1}{2}E[\{a(U) - a(V)\}\{c(U) - c(V)\}]$, puis étudiez le signe du produit lorsque a est croissante et c décroissante.)

c) Calculez le facteur de réduction de variance $Var[\hat{\theta}_{\text{ind}}]/Var[\hat{\theta}_{\text{anti}}]$ pour $g(u) = e^u$.

Voir la solution

1.5.4 Exercices appliqués

Exercice appliqué 1. Couverture des intervalles quand la variance n'est pas constante. Le jeu de données `pipeline` de la librairie `faraway` (Faraway 2025) contient 107 défauts d'un oléoduc de l'Alaska. La profondeur de chaque défaut a d'abord été mesurée sur le terrain par ultrasons (`Field`), puis remesurée en laboratoire (`Lab`), de façon plus précise. On veut prédire la mesure de laboratoire à partir de la mesure sur le terrain.

a) Ajustez la régression linéaire simple de `Lab` sur `Field` et interprétez la pente. Tracez les résidus en fonction des valeurs ajustées, puis calculez l'écart-type des résidus dans chaque quart des valeurs de `Field`. L'hypothèse de variance constante est-elle plausible ? Vérifiez qu'un modèle où l'écart-type de l'erreur est proportionnel à la mesure sur le terrain, $\sigma_i = \gamma x_i$, décrit bien les données, et estimez γ par

$$\hat{\gamma} = \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\frac{e_i}{x_i} \right)^2 \right\}^{1/2}.$$

b) Écrivez une fonction qui simule B jeux de données du modèle $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, en gardant les valeurs de `Field` des données et en prenant les estimations de a) comme vraies valeurs de β_0 et de β_1. L'écart-type de l'erreur est un argument de la fonction. Pour chaque jeu simulé, la fonction ajuste la droite et vérifie si l'intervalle de confiance à 95 % pour β_1 contient la vraie pente, et si les intervalles de prédiction à 95 % pour `Field` $= 10$ et `Field` $= 70$ contiennent une nouvelle observation simulée. Utilisez-la d'abord avec des erreurs normales de variance constante, d'écart-type $\hat{\sigma}$. Quelles couvertures attendez-vous ? Vos estimations de Monte Carlo sont-elles compatibles avec ces valeurs ?

c) Reprenez la simulation avec $\sigma_i = \hat{\gamma} x_i$. Comment les trois couvertures changent-elles ? Montrez ensuite que

$$Var[\hat{\beta}_1] = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sigma_i^2}{S_{xx}^2}, \qquad S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2,$$

et comparez cette valeur à l'erreur-type que calcule typiquement `lm`. Servez-vous de cette comparaison pour approximer la couverture de l'intervalle de β_1. (Indice : $E[SSE] = \sum_i \sigma_i^2(1-h_i)$, où h_i est le levier de l'observation i.)

d) Répétez c) en dupliquant quatre fois chaque valeur de `Field` ($n = 428$). Le problème disparaît-il quand n augmente ? Quelles conséquences pratiques en tirez-vous, et comment corrigeriez-vous l'analyse ?

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Morris, White et Crowther (2019). « Using simulation studies to evaluate statistical methods »

2 Modèles de régression pour données binaires

Ce chapitre utilise les librairies R suivantes.

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(latex2exp)
library(GLMsData)
library(purrr)
library(clinfun)
```

Ce chapitre porte sur la régression quand la variable réponse Y_i suit une loi de Bernoulli ou une loi binomiale. Le modèle linéaire du chapitre 0 suppose une réponse normale, de sorte qu'il ne convient pas à une réponse discrète, qui ne prend qu'un nombre fini (ou dénombrable) de valeurs. Le chapitre présente d'abord le modèle, l'estimation par maximum de vraisemblance et l'inférence sur les coefficients, puis la sélection et le diagnostic de modèles, et enfin la courbe ROC.

Exemple — Jeu de données : le naufrage de l'*Empress of Ireland*

Le 29 mai 1914, le paquebot *Empress of Ireland* a coulé en moins de 15 minutes dans le fleuve Saint-Laurent, près de Rimouski, après une collision avec le charbonnier norvégien *Storstad*. Sur les 1057 passagers à bord, seulement 217 ont survécu. Pour chaque passager, les données donnent sa classe de billet (`classe` : première, deuxième ou troisième), son sexe (`sexe`) et son groupe d'âge (`groupe_age` : adulte ou enfant), ainsi qu'un indicateur de survie.

Ces données sont disponibles sous deux formes, utilisées tout au long du chapitre :

- la version individuelle `empress_individus` (⬇) : une ligne par passager (1057 lignes) et une réponse binaire `survecu` ;
- la version agrégée `empress_agrege` (⬇) : un `effectif` total et un nombre de `survivants` pour chaque combinaison classe-sexe-groupe d'âge (11 lignes). Elle contient aussi une variable numérique auxiliaire `classe_num` (1 = première, 2 = deuxième, 3 = troisième), qui sert d'indicateur de l'éloignement du pont supérieur (et donc des canots de sauvetage).

La version individuelle s'obtient en dépliant la version agrégée avec `uncount()` :

```r
empress_agrege <- data.frame(
  classe     = c("Première", "Première", "Première",
                 "Deuxième", "Deuxième", "Deuxième", "Deuxième",
                 "Troisième", "Troisième", "Troisième", "Troisième"),
  sexe       = c("Homme", "Femme", "Femme",
                 "Homme", "Homme", "Femme", "Femme",
                 "Homme", "Homme", "Femme", "Femme"),
  groupe_age = c("Adulte", "Adulte", "Enfant",
                 "Adulte", "Enfant", "Adulte", "Enfant",
                 "Adulte", "Enfant", "Adulte", "Enfant"),
  effectif   = c(49, 34, 4, 114, 11, 107, 21, 446, 54, 169, 48),
  survivants = c(24, 11, 1,  33,  0,  13,  2, 115,  1,  17,  0)
)

empress_agrege <- empress_agrege |>
  mutate(classe_num = as.numeric(factor(classe, levels = c("Première",
   ↪  "Deuxième", "Troisième"))))

empress_individus <- empress_agrege |>
  uncount(effectif, .remove = FALSE) |>
  group_by(classe, sexe, groupe_age) |>
  mutate(survecu = as.numeric(row_number() <= first(survivants))) |>
  ungroup() |>
  select(classe, sexe, groupe_age, classe_num, survecu)

empress_agrege
```

```
      classe  sexe groupe_age effectif survivants classe_num
1   Première Homme     Adulte       49         24          1
2   Première Femme     Adulte       34         11          1
3   Première Femme     Enfant        4          1          1
4   Deuxième Homme     Adulte      114         33          2
5   Deuxième Homme     Enfant       11          0          2
6   Deuxième Femme     Adulte      107         13          2
7   Deuxième Femme     Enfant       21          2          2
8  Troisième Homme     Adulte      446        115          3
9  Troisième Homme     Enfant       54          1          3
10 Troisième Femme     Adulte      169         17          3
11 Troisième Femme     Enfant       48          0          3
```

Notez qu'aucun garçon de première classe n'était à bord : la combinaison (première, homme, enfant) est absente des données, et ce n'est pas une erreur de saisie.

2.1 Données et modèle

2.1.1 Données individuelles (Bernoulli)

Dans les données individuelles, comme `empress_individus`, la réponse $Y_i \in \{0,1\}$ indique si le passager i a survécu, et les données ont la forme $\{(Y_i, X_i),\ i = 1, \dots, n\}$. Le modèle suppose que Y_i suit une loi de Bernoulli de paramètre $\pi_i \in [0,1]$ (on note x_{ij} la valeur de la j^{e} variable explicative pour l'observation i, et $X_i = (1, x_{i1}, \dots, x_{ip})^\top$) :

$$\begin{aligned} P(Y_i = y_i) &= \pi_i^{y_i}(1-\pi_i)^{1-y_i}, \quad y_i \in \{0,1\}, \\ E[Y_i] &= \mu_i = \pi_i, \\ Var[Y_i] &= \pi_i(1-\pi_i). \end{aligned}$$

Définition — Prédicteur linéaire et fonction de lien

Posons $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ (le prédicteur linéaire). Ce prédicteur est relié à π_i par une fonction de lien g, supposée connue : $\eta_i = g(\pi_i)$, c'est-à-dire

$$\pi_i = P(Y_i = 1) = g^{-1}(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}).$$

2.1.2 Données groupées (binomiale)

Dans les données groupées, comme `empress_agrege`, $Y_i \in \{0, 1, \dots, m_i\}$ est le nombre de survivants (`survivants`) du i^{e} groupe classe-sexe-groupe d'âge. On note m_i l'effectif (`effectif`) de ce groupe, de sorte que $Y_i \sim \text{Bin}(m_i, \pi_i)$:

$$\begin{aligned} P(Y_i = y_i) &= \binom{m_i}{y_i} \pi_i^{y_i}(1-\pi_i)^{m_i - y_i}, \quad y_i \in \{0, 1, \dots, m_i\}, \\ E[Y_i] &= \mu_i = m_i \pi_i, \\ Var[Y_i] &= m_i \pi_i (1-\pi_i) = \frac{\mu_i(m_i - \mu_i)}{m_i}. \end{aligned}$$

Le lien s'écrit de nouveau $\eta_i = g(\pi_i)$, ou encore $\eta_i = g(\mu_i / m_i)$, souvent noté $g(\mu_i)$ par abus de notation. Si $m_1 = \cdots = m_n = 1$, la loi binomiale se réduit à la loi de Bernoulli, qui en est donc un cas particulier : tout ce qui suit couvre les deux situations.

2.1.3 Fonctions de lien

La fonction de lien doit garantir que $\pi_i = g^{-1}(\eta_i) \in [0,1]$. Trois choix sont courants :

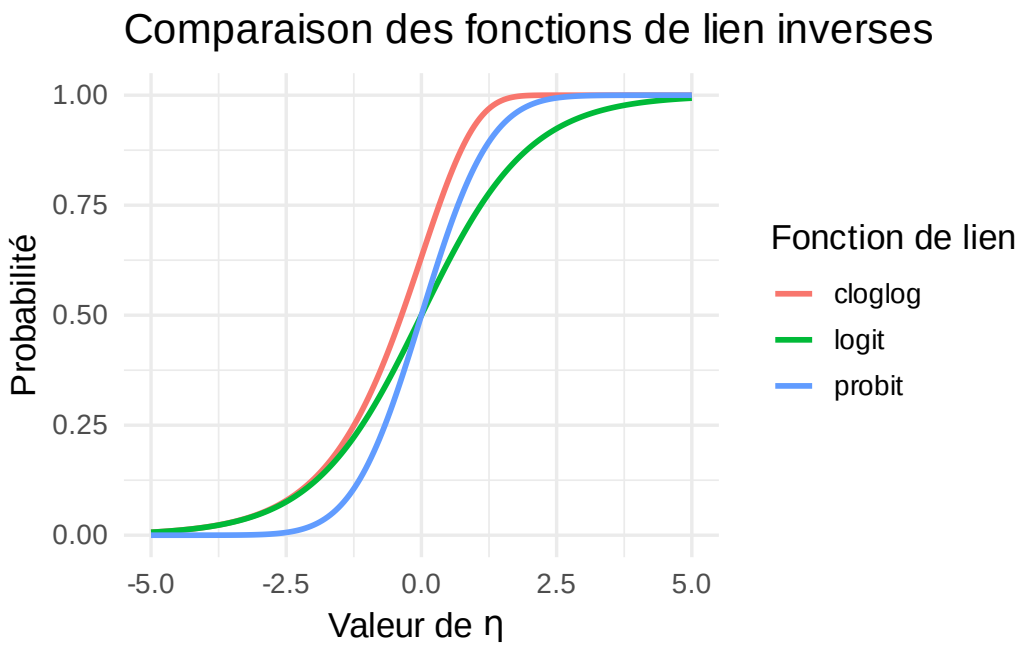
- logit : $g(u) = \text{logit}(u) = \log\Big(\dfrac{u}{1-u}\Big)$, d'où $\pi_i = \dfrac{e^{\eta_i}}{1+e^{\eta_i}}$. En R, il s'agit de `link="logit"` dans `glm`. C'est le lien le plus utilisé en régression binaire, en bonne partie parce que ses coefficients s'interprètent facilement (voir le bloc Interprétation plus bas) ;
- probit : $g(u) = \Phi^{-1}(u)$, où Φ est la fonction de répartition de la $\mathcal{N}(0,1)$, d'où $\pi_i = \Phi(\eta_i)$. En R, l'option correspondante est `link="probit"` ;
- c-log-log : $g(u) = \log\{-\log(1-u)\}$, d'où $\pi_i = 1-e^{-e^{\eta_i}}$. En R, on utilise `link="cloglog"`.

Propriété — g^{-1} est une fonction de répartition

Dans les trois cas, $g^{-1}(\cdot)$ est une fonction de répartition. C'est cette propriété qui garantit que $\pi_i = g^{-1}(\eta_i) \in [0,1]$, quel que soit le lien et quelle que soit la valeur (réelle, non bornée) de η_i.

En pratique, les trois liens donnent souvent des résultats semblables.

Le graphique suivant compare les probabilités $g^{-1}(\eta_i)$ des trois liens en fonction de η_i.



Le cas du lien logit — Interprétation

Avec le lien logit, le prédicteur linéaire s'écrit

$$\eta_i = \log\left(\frac{\pi_i}{1-\pi_i}\right)$$

et donc,

$$e^{\eta_i} = \frac{\pi_i}{1-\pi_i}.$$

Définition — Cote (odds)

Le rapport ci-dessus est la cote (*odds*), notée o_i :

$$o_i = \frac{\pi_i}{1-\pi_i} = \frac{P(Y_i=1)}{P(Y_i=0)}.$$

La cote compare la probabilité que l'événement se produise ($Y_i = 1$) à la probabilité qu'il ne se produise pas ($Y_i = 0$). Par exemple :

- si $\pi_i = 0{,}5$, alors $o_i = 1$: l'événement a autant de chances d'arriver que de ne pas arriver ;
- si $\pi_i = 0{,}75$, alors $o_i = 3$: l'événement a 3 fois plus de chances de se produire que de ne pas se produire.

Passons à l'interprétation du coefficient β_j. Dans le modèle,

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_j x_{ij} + \cdots$$

si x_{ij} augmente d'une unité (les autres variables restant fixes), η_i augmente de β_j.

Propriété — Effet multiplicatif sur la cote

Comme $o_i = e^{\eta_i}$, une augmentation d'une unité de x_{ij} (les autres variables restant fixes) multiplie la cote :

$$o_i^{\text{nouveau}} = e^{\eta_i + \beta_j} = e^{\eta_i} \cdot e^{\beta_j} = o_i^{\text{ancien}} \cdot e^{\beta_j}.$$

Une augmentation d'une unité de x_{ij} multiplie donc la cote par e^{β_j}. Ce facteur e^{β_j} s'appelle le rapport de cotes (*odds ratio*), et trois cas se présentent :

- $e^{\beta_j} > 1$: augmenter x_{ij} augmente la cote (et donc $P(Y_i = 1)$) ;
- $e^{\beta_j} < 1$: augmenter x_{ij} diminue la cote ;
- $e^{\beta_j} = 1$ (c'est-à-dire $\beta_j = 0$) : la j^{e} variable explicative n'a pas d'effet sur la cote.

2.2 Inférence statistique

2.2.1 Estimation des paramètres β

Comme on estime β par maximum de vraisemblance, on part de la vraisemblance. Avec n variables observées $Y_i \sim \text{Bin}(m_i, \pi_i)$, la log-vraisemblance s'écrit

$$\ell(\pi; y) = \sum_{i=1}^{n} \log\left[P(Y_i = y_i)\right] = \sum_{i=1}^{n} \left\{ \log \binom{m_i}{y_i} + y_i \log(\pi_i) + (m_i - y_i)\log(1-\pi_i) \right\}.$$

Cette fonction dépend de β indirectement, par π_i (rappel de la section sur les fonctions de lien) :

$$\pi_i = g^{-1}(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}),$$

où g^{-1} est l'inverse de la fonction de lien (la fonction logistique pour le lien logit).

Il est parfois plus commode de travailler avec μ_i, l'espérance de Y_i (donc $\mu_i = m_i\pi_i$), qu'avec π_i. En substituant $\pi_i = \mu_i/m_i$ dans la log-vraisemblance, on peut regrouper dans une constante K les termes qui ne dépendent pas de β (dont $\log\binom{m_i}{y_i}$), ce qui donne la forme équivalente, plus simple,

$$\ell(\mu; y) = \sum_{i=1}^{n} \left\{ y_i \log(\mu_i) + (m_i - y_i)\log(m_i - \mu_i) \right\} + K,$$

où $\mu = (\mu_1, \ldots, \mu_n)^\top$. Comme K ne dépend pas de β, elle n'affecte pas la maximisation. C'est sous cette forme que le chapitre suivant fera reconnaître l'appartenance du modèle binomial à la famille exponentielle, et elle est aussi plus facile à utiliser pour le calcul de la déviance.

Le cas du lien logit — Log-vraisemblance

Avec le lien logit, $\pi_i = \dfrac{e^{\eta_i}}{1+e^{\eta_i}}$ et $1-\pi_i = \dfrac{1}{1+e^{\eta_i}}$, où

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}.$$

Il vient alors

$$\log(\pi_i) = \eta_i - \log(1+e^{\eta_i}), \qquad \log(1-\pi_i) = -\log(1+e^{\eta_i}),$$

de sorte que

$$y_i \log(\pi_i) + (m_i - y_i)\log(1-\pi_i) = y_i\eta_i - m_i \log(1+e^{\eta_i}).$$

La log-vraisemblance s'écrit donc directement en fonction de β :

$$\ell(\beta; y) = \sum_{i=1}^{n} \left\{ \log \binom{m_i}{y_i} + y_i \eta_i - m_i \log\left(1 + e^{\eta_i}\right) \right\}.$$

Définition — Score

Le score, noté $U(\beta)$, est le vecteur des dérivées partielles de la log-vraisemblance par rapport à β :

$$U(\beta) = \frac{\partial \ell(\beta; y)}{\partial \beta} = \left(\frac{\partial \ell}{\partial \beta_0}, \dots, \frac{\partial \ell}{\partial \beta_p} \right)^\top.$$

L'estimateur du maximum de vraisemblance $\hat{\beta}$ est la solution de l'équation du score $U(\beta) = 0$, c'est-à-dire la valeur qui maximise ℓ. Contrairement à la régression linéaire, il n'y a pas de solution explicite (forme fermée), à cause de la non-linéarité de la fonction de lien g. On résout donc l'équation numériquement, en général par l'algorithme des moindres carrés pondérés itératifs (IRLS, *iteratively reweighted least squares*). Cet algorithme est l'algorithme du score de Fisher (*Fisher scoring*) appliqué à ce problème. Avec le lien logit, il coïncide avec la méthode de Newton-Raphson.

Le cas du lien logit — Estimation

Commençons par la dérivée par rapport à β_j. Il s'agit de dériver $\ell(\beta; y)$ terme à terme par rapport à β_j ($j = 0, 1, \dots, p$), en utilisant $\dfrac{\partial \eta_i}{\partial \beta_j} = x_{ij}$ (avec $x_{i0} \equiv 1$ pour l'ordonnée à l'origine).
Le terme $\log \binom{m_i}{y_i}$ ne dépend pas de β, donc sa dérivée est nulle. Pour le terme $y_i \eta_i$:

$$\frac{\partial}{\partial \beta_j} (y_i \eta_i) = y_i x_{ij}.$$

Le terme $m_i \log(1 + e^{\eta_i})$ se dérive en chaîne :

$$\frac{\partial}{\partial \beta_j} \log\left(1 + e^{\eta_i}\right) = \frac{e^{\eta_i}}{1 + e^{\eta_i}} \cdot \frac{\partial \eta_i}{\partial \beta_j} = \pi_i x_{ij},$$

d'où

$$\frac{\partial}{\partial \beta_j} \left(m_i \log(1 + e^{\eta_i}) \right) = m_i \pi_i x_{ij}.$$

En combinant les trois termes, on obtient

$$\frac{\partial \ell(\beta; y)}{\partial \beta_j} = \sum_{i=1}^{n} (y_i - m_i \pi_i)\, x_{ij}. \tag{2.1}$$

Passons à la forme matricielle (lien logit). Notons X la matrice de design $(n \times (p+1))$, $y = (y_1, \dots, y_n)^\top$ et $\mu = (m_1\pi_1, \dots, m_n\pi_n)^\top$. Le score s'écrit alors

$$U(\beta) = X^\top (y - \mu). \tag{2.2}$$

2.2.2 Séparation des données

Il arrive que l'estimateur du maximum de vraisemblance n'existe pas, ce qui se produit quand les covariables séparent parfaitement les deux groupes.

Définition — Séparation complète et quasi complète

Il y a séparation complète si un hyperplan classe parfaitement les Y_i à partir des covariables, c'est-à-dire s'il existe β tel que $X_i^\top \beta > 0$ chaque fois que $Y_i = 1$ et $X_i^\top \beta < 0$ chaque fois que $Y_i = 0$ (ou l'inverse).
Il y a séparation quasi complète quand cette séparation est presque parfaite : un hyperplan classe alors correctement toutes les observations, sauf un sous-ensemble où $X_i^\top \beta = 0$ exactement.

Remarque — Intuition géométrique : une pente qui explose

Avec un seul prédicteur continu et le lien logit, $\hat{\pi}(x) = g^{-1}(\hat{\beta}_0 + \hat{\beta}_1 x)$, dont la pente est

$$\hat{\pi}'(x) = \hat{\beta}_1\, \hat{\pi}(x)\{1 - \hat{\pi}(x)\},$$

maximale au point d'inflexion ($\hat{\pi}(x) = 1/2$), où elle vaut $\hat{\beta}_1/4$. Sous séparation, l'algorithme pousse $|\hat{\beta}_1| \to \infty$ pour continuer à augmenter la vraisemblance, de sorte que la pente maximale de la courbe augmente elle aussi. La sigmoïde devient de plus en plus abrupte et se rapproche d'une fonction en escalier (0 d'un côté du seuil de séparation, 1 de l'autre). Mais aucune sigmoïde à pente finie ne reproduit ce saut, et c'est pour cela que l'algorithme ne converge pas, plutôt que de donner une pente infinie.
Avec plusieurs covariables, l'image est la même, et l'hyperplan séparateur devient une frontière nette (les probabilités prédites tendent vers 0 ou 1 de chaque côté). Mais « la pente » n'est plus un scalaire : c'est la norme $\|\hat{\beta}\|$ dans la direction séparatrice qui diverge.

En présence de séparation (complète ou quasi complète), l'estimateur du maximum de vraisemblance $\hat{\beta}$ n'existe pas au sens usuel. La vraisemblance continue d'augmenter quand $\|\beta\| \to \infty$ dans la direction qui sépare les données, sans jamais atteindre de maximum fini. En pratique, l'algorithme IRLS ne converge pas, ou alors il converge vers des $\hat{\beta}_j$ énormes, limités seulement

par son critère d'arrêt.

R signale souvent le problème par l'avertissement

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

accompagné de coefficients estimés énormes en valeur absolue et d'erreurs-types démesurées (souvent plus grandes que le coefficient lui-même de plusieurs ordres de grandeur).

Exemple — Illustration de la séparation complète

On génère 10 observations autour de $-2{,}5$ avec $y = 0$ et 10 autour de 2,5 avec $y = 1$. Avec cette graine, les deux groupes ne se chevauchent pas (séparation complète). On ajuste ensuite la régression logistique.

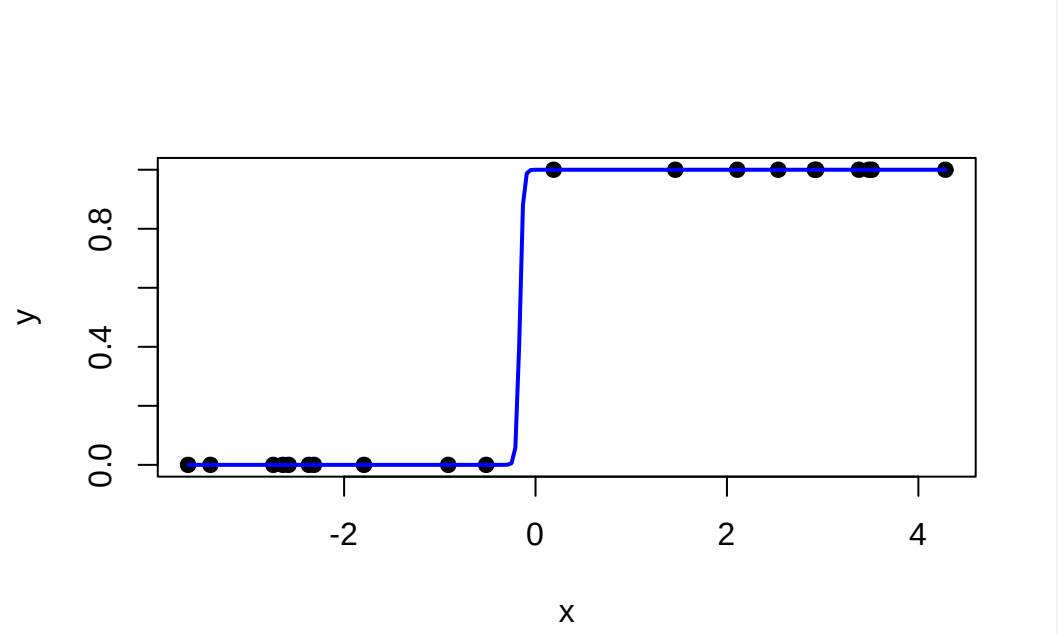
```r
set.seed(2)
x <- c(rnorm(10, mean = -2.5), rnorm(10, mean = 2.5))
y <- c(rep(0, 10), rep(1, 10)) # y=0 pour le premier groupe, y=1 pour le
 ↪  second ;
# avec cette graine, les deux groupes ne se chevauchent pas

modele_separe <- glm(y ~ x, family = binomial)
```

```
Warning: glm.fit: algorithm did not converge

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(modele_separe)$coefficients
```

```
             Estimate Std. Error      z value  Pr(>|z|)
(Intercept)  9.924497   19989.53 0.0004964849 0.9996039
x           60.320086   50086.41 0.0012043204 0.9990391
```

```r
grille_x <- data.frame(x = seq(min(x), max(x), length.out = 200))
grille_x$p_hat <- predict(modele_separe, newdata = grille_x, type =
 ↪  "response")

plot(x, y, pch = 19, ylim = c(0, 1))
lines(grille_x$x, grille_x$p_hat, col = "blue", lwd = 2)
```



La courbe $\hat{\pi}(x)$ ressemble à une fonction en escalier plutôt qu'à une sigmoïde, puisqu'elle passe de 0 à 1 sur un intervalle minuscule autour du seuil de séparation. Ici, n'importe quel seuil entre $-0{,}52$ et $0{,}19$ sépare les deux groupes, et la courbe ajustée passe à $1/2$ vers $x \approx -0{,}16$. C'est parce que la pente $\hat{\beta}_1/4$ diverge sous séparation complète.
Le coefficient de `x` et son erreur-type sont tous deux énormes, ce qui signale une séparation et non un effet réel de cette taille.

Remarque — Que faire en cas de séparation ?

Sous séparation (complète ou quasi complète), aucune méthode d'inférence fondée sur l'estimateur du maximum de vraisemblance ordinaire ne règle le problème, puisque cet estimateur n'est pas fini. Changer de test ou d'intervalle (Wald, rapport de vraisemblance, vraisemblance profilée) ne suffit donc pas : il faut changer d'estimateur ou de données. Cinq approches sont courantes :

- régression logistique pénalisée (correction de Firth) : une pénalité est ajoutée à la vraisemblance (`logistf` en R), ce qui garantit un estimateur fini même sous séparation ;
- approche bayésienne : une loi *a priori* faiblement informative sur les coefficients (par exemple avec `arm::bayesglm`) joue le même rôle qu'une pénalité et donne aussi des estimations finies ;
- retrait ou regroupement de la variable en cause : si une variable catégorielle cause la séparation, la catégorie fautive peut être fusionnée avec une catégorie voisine ;

— régularisation (ridge, lasso, Chapitre 6) : les coefficients sont contraints, ce qui les empêche de diverger ;
— collecte de données : si la séparation vient d'un petit échantillon et non d'une relation déterministe, un échantillon plus grand peut la faire disparaître.

Notez que la séparation a aussi une conséquence sur l'inférence avant même d'être atteinte : les tests de Wald deviennent peu fiables quand les données sont presque séparées, sans séparation au sens strict (voir le phénomène de Hauck-Donner plus loin).

2.2.3 Information de Fisher et variance des estimateurs

La variance de $\hat{\beta}$ s'obtient à partir de la matrice d'information de Fisher I, dont l'élément $(j+1, k+1)$ est

$$I_{jk} = -\left.\frac{\partial^2 \ell}{\partial\beta_j \partial\beta_k}\right|_{\hat{\beta}}.$$

Il s'agit de l'information observée. Avec le lien logit, elle coïncide avec l'information attendue $X^\top W X$ donnée plus bas. Pour un autre lien, les deux diffèrent ; R utilise l'information attendue, et c'est elle qu'on note I dans la suite.

La matrice de variance-covariance de $\hat{\beta}$ est estimée par

$$\hat{\mathrm{V}}(\hat{\beta}) = I^{-1},$$

et la loi asymptotique de $\hat{\beta}$ est approchée par

$$\hat{\beta} \;\dot{\sim}\; \mathcal{N}\{\beta, \hat{\mathrm{V}}(\hat{\beta})\}.$$

Notation : la notation $\dot{\sim}$ signifie que la variable aléatoire suit approximativement la loi indiquée, de façon asymptotique. Ici, sous des conditions de régularité, $\hat{\beta}$ est asymptotiquement normal quand l'information totale croît, par exemple quand $n \to \infty$: la loi de $\hat{\beta}$ converge en distribution vers la loi normale indiquée.

On note $\hat{\mathrm{V}}(\hat{\beta}_j) = [\hat{\mathrm{V}}(\hat{\beta})]_{j+1,j+1}$ la variance estimée de $\hat{\beta}_j$.

Dans le cas binomial, on montre que

$$\hat{\mathrm{V}}(\hat{\beta}) = (X^\top W X)^{-1},$$

où W est une matrice diagonale d'entrées

$$w_i = \frac{(\partial \mu_i / \partial \eta_i)^2}{Var[Y_i]} = \frac{m_i \{(g^{-1})'(\hat{\eta}_i)\}^2}{\hat{\pi}_i(1-\hat{\pi}_i)},$$

avec

$$\hat{\mu}_i = m_i \hat{\pi}_i = m_i \, g^{-1}(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}).$$

Le cas du lien logit — Expression de w_i

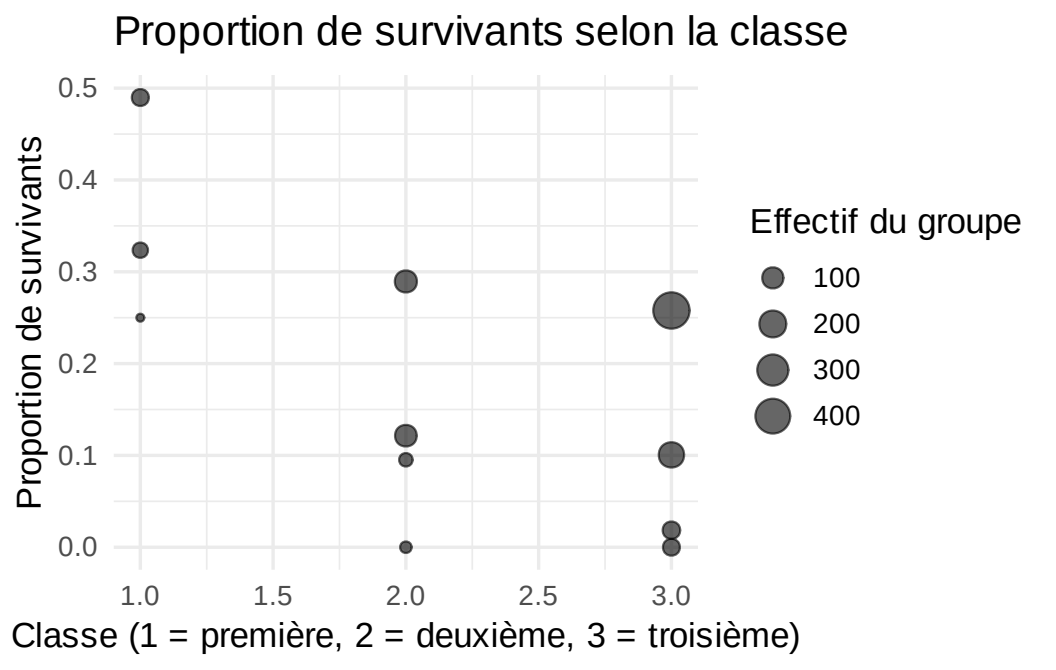
Avec le lien logit, $\hat{\pi}_i = \dfrac{e^{\hat{\eta}_i}}{1+e^{\hat{\eta}_i}}$, où $\hat{\eta}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$. Comme $(g^{-1})'(\eta) = \pi(1-\pi)$, l'entrée w_i se simplifie en

$$w_i = m_i \, \hat{\pi}_i(1-\hat{\pi}_i) = \frac{\hat{\mu}_i(m_i - \hat{\mu}_i)}{m_i}.$$

Exemple — Jeu de données `empress_agrege` : comparaison des liens

On ajuste le modèle avec les trois fonctions de lien et `classe_num` comme seule variable explicative. On regarde d'abord les proportions observées de survivants par classe.

```
empress_obs <- empress_agrege |>
  mutate(prop_observee = survivants / effectif)

ggplot() +
  geom_point(
    data = empress_obs,
    aes(x = classe_num, y = prop_observee, size = effectif),
    color = "black",
    alpha = 0.6
  ) +
  labs(
    x = "Classe (1 = première, 2 = deuxième, 3 = troisième)",
    y = "Proportion de survivants",
    color = "Fonction de lien",
    size = "Effectif du groupe",
    title = "Proportion de survivants selon la classe"
  ) +
  theme_minimal(base_size = 13)
```




```
modele_agrege_classe_logit <- glm(
  cbind(survivants, effectif - survivants) ~ classe_num,
  data = empress_agrege,
  family = binomial(link = "logit")
)
modele_agrege_classe_probit <- glm(
  cbind(survivants, effectif - survivants) ~ classe_num,
  data = empress_agrege,
  family = binomial(link = "probit")
)
modele_agrege_classe_cloglog <- glm(
  cbind(survivants, effectif - survivants) ~ classe_num,
  data = empress_agrege,
  family = binomial(link = "cloglog")
)
summary(modele_agrege_classe_logit)$coefficients
```

```
             Estimate Std. Error    z value   Pr(>|z|)
(Intercept) -0.2727075  0.2876705 -0.9479859 0.34313662
classe_num  -0.4245292  0.1110337 -3.8234261 0.00013161
```

```r
summary(modele_agrege_classe_probit)$coefficients
```

```
             Estimate Std. Error   z value     Pr(>|z|)
(Intercept) -0.1993538 0.17249544 -1.155705 0.2478020066
classe_num  -0.2437527 0.06568542 -3.710911 0.0002065147
```

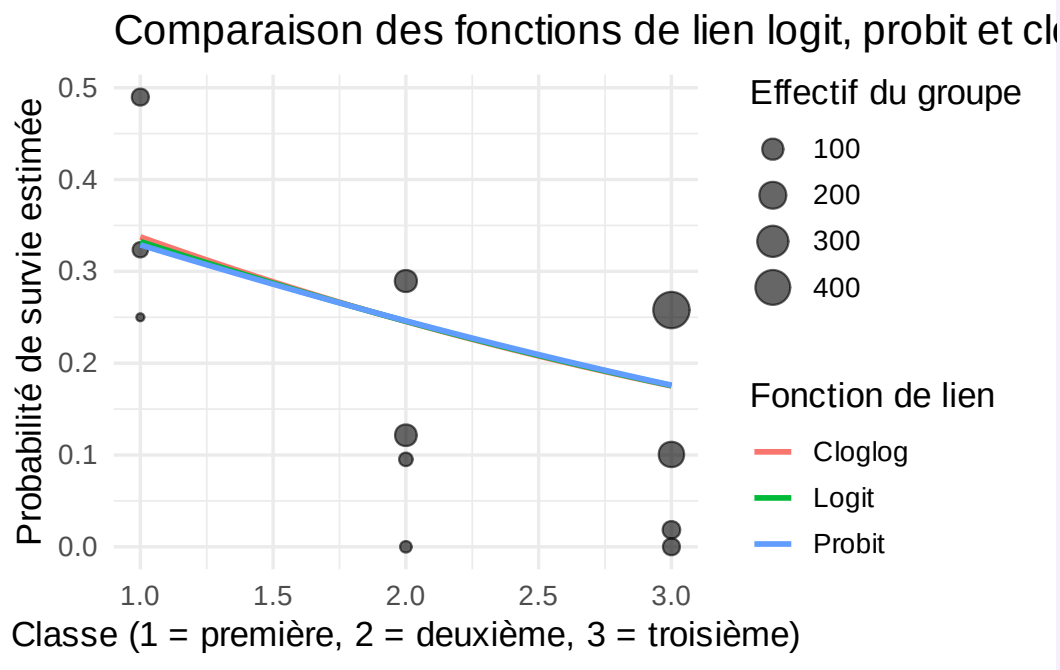
```r
summary(modele_agrege_classe_cloglog)$coefficients
```

```
             Estimate Std. Error   z value     Pr(>|z|)
(Intercept) -0.5073411 0.24167756 -2.099248 3.579504e-02
classe_num  -0.3798464 0.09480604 -4.006563 6.160872e-05
```

La colonne `Std. Error` de `summary()` donne les erreurs-types estimées des coefficients, comme en régression linéaire. Traçons ensuite les probabilités prédites des trois modèles.

```r
grille <- data.frame(
  classe_num = seq(min(empress_agrege$classe_num),
   ↪  max(empress_agrege$classe_num), length.out = 200)
)

predire_modele <- function(modele, nom_lien) {
  grille |>
    mutate(
      p_hat = predict(modele, newdata = grille, type = "response"),
      lien = nom_lien
    )
}

predictions <- bind_rows(
  predire_modele(modele_agrege_classe_logit, "Logit"),
  predire_modele(modele_agrege_classe_probit, "Probit"),
  predire_modele(modele_agrege_classe_cloglog, "Cloglog")
)

empress_obs <- empress_agrege |>
  mutate(prop_observee = survivants / effectif)

ggplot() +
  geom_point(
```

```
  data = empress_obs,
  aes(x = classe_num, y = prop_observee, size = effectif),
  color = "black",
  alpha = 0.6
) +
geom_line(
  data = predictions,
  aes(x = classe_num, y = p_hat, color = lien),
  linewidth = 1
) +
labs(
  x = "Classe (1 = première, 2 = deuxième, 3 = troisième)",
  y = "Probabilité de survie estimée",
  color = "Fonction de lien",
  size = "Effectif du groupe",
  title = "Comparaison des fonctions de lien logit, probit et cloglog"
) +
theme_minimal(base_size = 13)
```



Pour les trois liens, la valeur-p associée à `classe_num` (environ $1{,}3 \times 10^{-4}$ pour le logit, $2{,}1 \times 10^{-4}$ pour le probit et $6{,}2 \times 10^{-5}$ pour le cloglog) est bien sous le seuil de 5%. On rejette donc $H_0 : \beta_1 = 0$ dans les trois cas : l'effet de la classe est significatif quel que soit le lien.

Les trois liens donnent le même sens d'effet (plus le numéro de classe est élevé, plus la probabilité de survie est faible). Les courbes de probabilité prédite $\hat{\pi}$ sont aussi presque confondues, c'est-à-dire qu'à `classe_num` fixé, les probabilités prédites diffèrent de moins de 1 point de pourcentage d'un lien à l'autre.

2.2.4 Intervalles de confiance pour les β_j

Les intervalles de confiance et les tests sur les β_j se construisent par la méthode de Wald, à partir de la loi asymptotique de $\hat{\beta}$. Un intervalle de confiance de niveau $(1-\alpha)$ pour β_j est

$$\left[\hat{\beta}_j - z_{\alpha/2}\sqrt{\hat{\mathrm{V}}(\hat{\beta}_j)}\,;\ \hat{\beta}_j + z_{\alpha/2}\sqrt{\hat{\mathrm{V}}(\hat{\beta}_j)}\right],$$

où z_γ est le quantile de la $\mathcal{N}(0,1)$ défini par $P(Z > z_\gamma) = \gamma$.

Pour tester $H_0 : \beta_j = 0$ contre $H_1 : \beta_j \neq 0$, on rejette H_0 au seuil α si $|z_{obs}| > z_{\alpha/2}$, avec $z_{obs} = \hat{\beta}_j / \sqrt{\hat{\mathrm{V}}(\hat{\beta}_j)}$. La valeur-p est $2P(Z > |z_{obs}|)$.

Exemple — Intervalle de confiance pour β_1 dans l'exemple `empress_agrege`

Avec le lien logit, on construit un intervalle de confiance de niveau 95% pour β_1 (l'effet de `classe_num`) à partir de la sortie de `summary()` :

```
summary(modele_agrege_classe_logit)$coefficients
```

```
             Estimate Std. Error    z value   Pr(>|z|)
(Intercept) -0.2727075  0.2876705 -0.9479859 0.34313662
classe_num  -0.4245292  0.1110337 -3.8234261 0.00013161
```

L'intervalle est $[\hat{\beta}_1 - 1{,}96 \times \sqrt{\hat{\mathrm{V}}(\hat{\beta}_1)}\,;\ \hat{\beta}_1 + 1{,}96 \times \sqrt{\hat{\mathrm{V}}(\hat{\beta}_1)}] = [-0{,}6422\,;\ -0{,}2069]$.
R le calcule aussi directement :

```
confint(modele_agrege_classe_logit)
```

```
Waiting for profiling to be done...
```

```
                2.5 %     97.5 %
(Intercept) -0.8459280  0.2838251
classe_num  -0.6409082 -0.2050072
```

Notez que les deux intervalles ne sont pas identiques.

Remarque — Intervalle de Wald ou intervalle par vraisemblance profilée ?

Il ne faut pas voir `confint(modele)` comme le calcul de l'intervalle de Wald. Appliquée à un objet `glm`, cette fonction calcule plutôt un intervalle par vraisemblance profilée (*profile likelihood*). Cette méthode ne suppose pas une log-vraisemblance parabolique (symétrique) autour de $\hat{\beta}_j$. Elle parcourt sa vraie courbe en faisant varier β_j (en réoptimisant les autres paramètres à chaque valeur), puis inverse le test du rapport de vraisemblance pour trouver les bornes. L'intervalle obtenu n'est donc en général pas symétrique autour de $\hat{\beta}_j$, contrairement à celui de Wald, qui s'obtient avec `confint.default(modele)`.

```
confint.default(modele_agrege_classe_logit)
```

```
                2.5 %     97.5 %
(Intercept) -0.8365313  0.2911162
classe_num  -0.6421512 -0.2069071
```

Les deux méthodes donnent des résultats semblables avec un grand échantillon et des coefficients modérés. Dans de nombreux cas, l'intervalle par vraisemblance profilée est préférable, parce qu'il ne dépend pas de la courbure locale en $\hat{\beta}_j$. Comme le test du rapport de vraisemblance, il n'est pas touché par le phénomène de Hauck-Donner décrit plus bas, et il se comporte mieux avec des données presque séparées et de petits échantillons. Dans ces situations, l'intervalle de Wald peut devenir trompeur : il est alors trop large, au point de contenir 0 alors que l'effet est réel. L'intervalle de Wald garde son utilité pour l'intuition et les calculs à la main. Pour les résultats rapportés, je vous conseille plutôt l'intervalle par vraisemblance profilée.
Notez toutefois que la vraisemblance profilée n'est pas un remède à la séparation. Sous séparation complète ou quasi complète, l'estimateur du maximum de vraisemblance n'est pas fini, au moins une borne de l'intervalle est infinie et `confint()` peut échouer. Il faut alors changer d'estimateur (voir la remarque « Que faire en cas de séparation ? »).

Pour aller plus loin avec la théorie — Construction de l'intervalle par vraisemblance profilée

On définit d'abord la log-vraisemblance profilée de β_j : pour chaque valeur fixée de β_j, on maximise la log-vraisemblance sur les autres paramètres β_{-j} :

$$\ell_p(\beta_j) = \max_{\beta_{-j}} \ell(\beta_j, \beta_{-j}).$$

Les bornes (β_j^L, β_j^U) de l'intervalle de niveau $(1-\alpha)$ sont les deux solutions de l'équation

$$2\left[\ell(\hat{\beta}) - \ell_p(\beta_j)\right] = \chi^2_{1,\alpha}.$$

L'équation est non linéaire en β_j (chaque évaluation de ℓ_p demande une optimisation), de sorte qu'elle n'a pas de solution explicite, sauf cas particuliers : contrairement à l'intervalle de Wald, cet intervalle n'a pas de forme fermée.
Cas particulier avec forme fermée : en régression linéaire avec erreurs normales et variance σ^2 connue, la log-vraisemblance est exactement quadratique en β, si bien que la vraisemblance profilée coïncide avec l'approximation de Wald et que les deux intervalles sont identiques. Ce n'est plus vrai dès qu'on sort de ce cadre, comme en régression logistique. Même en régression linéaire, si σ^2 est estimée, $\ell_p(\beta_j)$ n'est plus quadratique et l'équation ci-dessus ne redonne pas exactement l'intervalle de Student.
Résolution en pratique : `confint()` appliqué à un objet `glm` (fonction de la librairie `stats` depuis R 4.4, auparavant fournie par `MASS`) calcule $\ell_p(\beta_j)$ sur une grille de valeurs de β_j autour de $\hat{\beta}_j$. Il réajuste le modèle (les autres paramètres libres) à chaque valeur, d'où le message « Waiting for profiling to be done… ». Les points sont ensuite interpolés par une spline, et on obtient les bornes par recherche de racine sur cette interpolation, plutôt que par résolution directe de l'équation ci-dessus.

Remarque — Le phénomène de Hauck-Donner

Le test de Wald pour $H_0 : \beta_j = 0$ repose sur la statistique

$$z_{obs} = \frac{\hat{\beta}_j}{\sqrt{\hat{\mathrm{V}}(\hat{\beta}_j)}},$$

où l'erreur-type vient de la courbure de la log-vraisemblance évaluée en $\hat{\beta}_j$, et non sous H_0. Cette dépendance locale a un effet contre-intuitif : si $|\hat{\beta}_j|$ devient grand (par exemple sous séparation, complète ou quasi complète, ou pour un effet fort), la log-vraisemblance s'aplatit en $\hat{\beta}_j$. L'erreur-type $\sqrt{\hat{\mathrm{V}}(\hat{\beta}_j)}$ gonfle alors, ce qui peut faire diminuer z_{obs}, et l'effet paraît non significatif alors qu'il l'est. On peut montrer que cela vient de ce que l'erreur-type croît plus vite que la valeur absolue de l'estimation.
Ce phénomène, documenté par Hauck et Donner (1977), ne touche pas le test du rapport de vraisemblance, qui compare directement la vraisemblance en $\hat{\beta}_j$ à celle en 0, au lieu de s'appuyer sur une approximation locale :

$$G^2 = -2\left[\ell(\text{modèle réduit}) - \ell(\text{modèle complet})\right] \;\dot{\sim}\; \chi^2_1 \quad \text{sous } H_0.$$

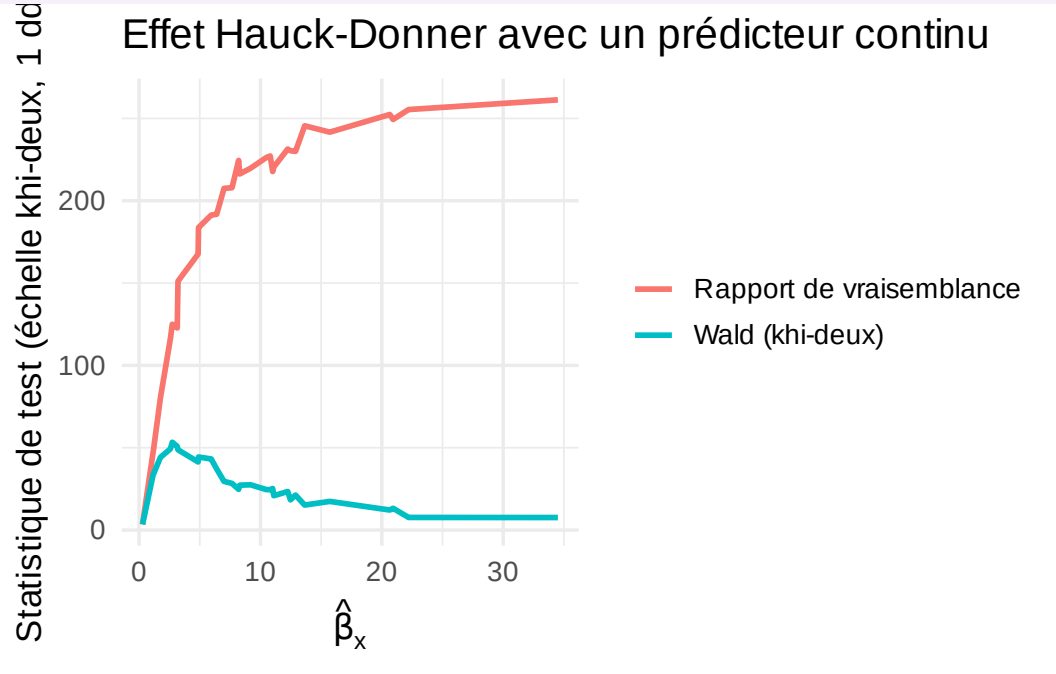
En pratique, la plupart des fonctions de R affichent par défaut le test de Wald, qui coûte moins de calcul et qui est fiable avec un grand échantillon. Méfiez-vous d'un grand coefficient accompagné d'une valeur-p de Wald étonnamment élevée : dans ce cas, comparez avec `anova(modele_reduit, modele_complet, test = "Chisq")` avant de conclure à la non-significativité.

Exemple — Effet Hauck-Donner

On étudie le phénomène par simulation (Chapitre 1) : on génère des données logistiques avec un effet β_1 de plus en plus grand, puis on compare les deux statistiques de test sur chaque jeu de données. La fonction `simuler_logistique` sert aussi plus bas dans cet exemple.

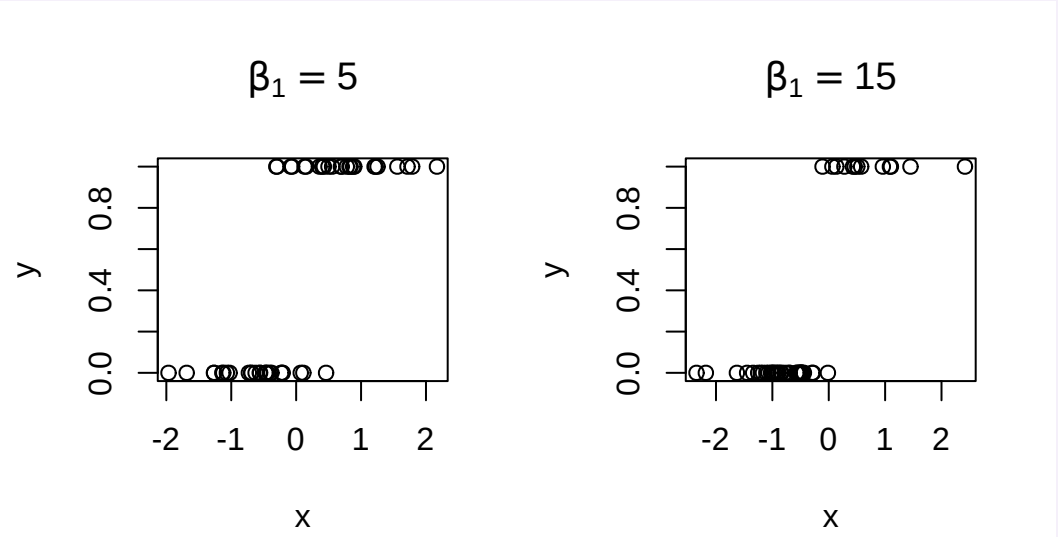
```
simuler_logistique <- function(n, beta0 = 0, beta1 = 1) {
  x <- rnorm(n, mean = 0, sd = 1)
  eta <- beta0 + beta1 * x
  p <- plogis(eta) # équivalent à 1 / (1 + exp(-eta))
  y <- rbinom(n, size = 1, prob = p)
  data.frame(x = x, y = y)
}

set.seed(123)
n <- 200
valeurs_beta1 <- seq(0.5, 15, by = 0.5) # vrai effet croissant (effet de
 ↪  plus en plus fort)

resultats <- map_dfr(valeurs_beta1, function(b1) {
  dat <- simuler_logistique(n, beta0 = 0, beta1 = b1)

  mod_complet <- glm(y ~ x, data = dat, family = binomial)
  mod_reduit <- glm(y ~ 1, data = dat, family = binomial)

  beta_hat <- coef(mod_complet)["x"]
  et_hat <- summary(mod_complet)$coefficients["x", "Std. Error"]
  wald_chisq <- (beta_hat / et_hat)^2
  lr_chisq <- as.numeric(2 * (logLik(mod_complet) - logLik(mod_reduit)))

  data.frame(
    beta1_vrai = b1,
    beta_hat = beta_hat,
    wald_chisq = wald_chisq,
    lr_chisq = lr_chisq
  )
})

resultats |>
  tidyr::pivot_longer(
    c(wald_chisq, lr_chisq),
    names_to = "test",
```

```
  values_to = "statistique"
) |>
mutate(
  test = recode(
    test,
    wald_chisq = "Wald (khi-deux)",
    lr_chisq = "Rapport de vraisemblance"
  )
) |>
ggplot(aes(x = beta_hat, y = statistique, color = test)) +
geom_line(linewidth = 1) +
labs(
  x = expression(hat(beta)[x]),
  y = "Statistique de test (échelle khi-deux, 1 ddl)",
  color = NULL,
  title = "Effet Hauck-Donner avec un prédicteur continu"
) +
theme_minimal(base_size = 13)
```



On compare maintenant les sorties des deux tests sur deux jeux de données de taille $n = 50$: un cas « normal » ($\beta_1 = 5$, pas de séparation) et un cas presque séparé ($\beta_1 = 15$: une seule paire d'observations se chevauche) :

```r
# Cas 1 : beta1 = 5, pas de séparation
set.seed(123)
dat_normal <- simuler_logistique(n = 50, beta0 = 0, beta1 = 5)

# Cas 2 : beta1 = 15, données presque séparées
set.seed(1234)
dat_separe <- simuler_logistique(n = 50, beta0 = 0, beta1 = 15)

par(mfrow = c(1, 2))
plot(dat_normal$x, dat_normal$y, xlab = "x", ylab = "y", main =
 ↪  expression(beta[1] == 5))
plot(dat_separe$x, dat_separe$y, xlab = "x", ylab = "y", main =
 ↪  expression(beta[1] == 15))
```




```r
mod_normal <- glm(y ~ x, data = dat_normal, family = binomial)
summary(mod_normal)$coefficients
```

```
             Estimate Std. Error   z value    Pr(>|z|)
(Intercept) 0.1083352  0.5187463 0.2088405 0.834572762
x           4.7939867  1.4068959 3.4074921 0.000655628
```

```r
mod_separe <- glm(y ~ x, data = dat_separe, family = binomial)
summary(mod_separe)$coefficients
```

```
             Estimate Std. Error   z value  Pr(>|z|)
(Intercept)  1.028036   1.301805 0.7897003 0.4297028
x           14.144188   7.929599 1.7837205 0.0744691
```

Dans le cas normal, la valeur-p de Wald pour `x` est d'environ $6{,}56 \times 10^{-4}$, sous le seuil de 5%, de sorte qu'on rejette $H_0 : \beta_1 = 0$, comme prévu puisque l'effet simulé est réel. Dans le cas presque séparé, elle grimpe à environ 0,074, au-dessus du seuil : le test de Wald ne rejette pas H_0, alors que l'effet simulé est trois fois plus fort ($\beta_1 = 15$ contre $\beta_1 = 5$). C'est le phénomène de Hauck-Donner : le test de Wald devient trompeur quand l'effet est très fort ou que les données sont presque séparées.
Voici les valeurs-p des tests du rapport de vraisemblance pour les deux cas :

```
mod_normal_reduit <- glm(y ~ 1, data = dat_normal, family = binomial)
anova(mod_normal_reduit, mod_normal, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ x
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        49     69.315
2        48     25.222  1   44.093 3.131e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod_separe_reduit <- glm(y ~ 1, data = dat_separe, family = binomial)
anova(mod_separe_reduit, mod_separe, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ x
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        49     57.306
2        48      4.990  1   52.316 4.725e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Par contre, le test du rapport de vraisemblance rejette H_0 dans les deux cas au seuil de 5%, avec une valeur-p $\approx 3{,}13 \times 10^{-11}$ pour le cas normal et une valeur-p $\approx 4{,}72 \times 10^{-13}$ pour le cas presque séparé, toutes deux bien sous 0,05. Il détecte donc l'effet même quand les données sont presque séparées, là où le test de Wald échoue.

2.2.5 Test du score (Rao)

Le test de Wald n'utilise que le modèle complet, et le test du rapport de vraisemblance utilise les deux modèles. Le troisième test classique fondé sur la vraisemblance, le test du score (de Rao, aussi appelé test des multiplicateurs de Lagrange), n'utilise que le modèle réduit.

Définition — Test du score

Partageons $\beta = (\beta_1^\top, \beta_2^\top)^\top$, où β_2 regroupe les d coefficients testés, et notons $\tilde{\beta}$ l'estimateur du maximum de vraisemblance sous $H_0 : \beta_2 = 0$ (les coefficients du modèle réduit, complétés par des zéros). La statistique du score est

$$S = U(\tilde{\beta})^\top \, I(\tilde{\beta})^{-1} \, U(\tilde{\beta}) \;\dot{\sim}\; \chi^2_d \quad \text{sous } H_0,$$

où U est le score du modèle complet (Équation 2.2) et I son information de Fisher. Les composantes de $U(\tilde{\beta})$ associées à β_1 sont nulles, puisque ce sont les équations du modèle réduit : seules celles de β_2 contribuent.

Les trois tests regardent la même log-vraisemblance de trois façons. Le test de Wald mesure l'écart horizontal entre $\hat{\beta}_j$ et la valeur nulle, à l'aide de la courbure en $\hat{\beta}_j$. Le test du rapport de vraisemblance mesure l'écart vertical entre les deux maximums. Le test du score mesure la pente de la log-vraisemblance à la valeur nulle : si H_0 est vraie, cette pente devrait être proche de 0. Comme il est évalué sous H_0, le test du score n'est pas touché par le phénomène de Hauck-Donner. Il se calcule aussi quand le modèle complet est difficile à ajuster, par exemple sous séparation, où $\hat{\beta}$ n'est pas fini. En R, `anova(modele_reduit, modele_complet, test = "Rao")` le calcule.

Voici les valeurs-p des trois tests sur les deux jeux de données simulés de l'exemple précédent :

```
trois_tests <- function(reduit, complet) {
  c(Wald  = summary(complet)$coefficients["x", "Pr(>|z|)"],
    RV    = anova(reduit, complet, test = "Chisq")[2, "Pr(>Chi)"],
    Score = anova(reduit, complet, test = "Rao")[2, "Pr(>Chi)"])
}
tests_hd <- rbind(normal = trois_tests(mod_normal_reduit, mod_normal),
                  presque_separe = trois_tests(mod_separe_reduit,
                   ↪  mod_separe))
tests_hd
```

```
                    Wald           RV        Score
normal         0.000655628 3.131180e-11 7.460944e-08
presque_separe 0.074469098 4.724647e-13 2.060330e-08
```

Dans le cas presque séparé, le test du score rejette H_0 comme le test du rapport de vraisemblance (valeur-p $\approx 2{,}06 \times 10^{-8}$), alors que le test de Wald ne la rejette pas.

Le tableau suivant résume les méthodes d'inférence sur β_j vues dans cette section. Dans tous les cas, on suppose le modèle bien spécifié et les observations indépendantes, et les méthodes reposent sur une approximation asymptotique.

Méthode	Statistique ou intervalle	Approximation utilisée	Limites
Wald	$z_{obs} = \hat{\beta}_j / \sqrt{\hat{\mathrm{V}}(\hat{\beta}_j)}$ et $\hat{\beta}_j \pm z_{\alpha/2}\sqrt{\hat{\mathrm{V}}(\hat{\beta}_j)}$	$\hat{\beta} \dot{\sim} \mathcal{N}\{\beta, \hat{\mathrm{V}}(\hat{\beta})\}$, avec la courbure évaluée en $\hat{\beta}_j$	phénomène de Hauck-Donner ; peu fiable avec des données presque séparées ou un petit échantillon
Rapport de vraisemblance	$G^2 = -2\,[\ell(\text{modèle réduit}) - \ell(\text{modèle complet})]$	$G^2 \dot{\sim} \chi^2_1$ sous H_0	demande d'ajuster les deux modèles ; ne rend pas l'estimateur fini sous séparation
Score (Rao)	$S = U(\tilde{\beta})^\top I(\tilde{\beta})^{-1} U(\tilde{\beta})$, évaluée sous H_0	$S \dot{\sim} \chi^2_1$ sous H_0	ne demande que le modèle réduit ; ne donne pas d'estimation de β_j
Vraisemblance profilée	bornes telles que $2\left[\ell(\hat{\beta}) - \ell_p(\beta_j)\right] = \chi^2_{1,\alpha}$	celle du rapport de vraisemblance	calcul itératif ; au moins une borne infinie sous séparation

2.2.6 Intervalle de confiance pour une probabilité prédite

Étant donné $X^* = (1, x_1^*, \ldots, x_p^*)^\top$, on estime π^* par $\hat{\pi}^* = g^{-1}(X^{*\top}\hat{\beta})$, et on construit un intervalle de confiance pour π^* de deux façons :

1. transformation de l'intervalle sur η^* : un IC pour $\eta^* = X^{*\top}\beta$ est
$$\left[X^{*\top}\hat{\beta} \pm z_{\alpha/2}\sqrt{\hat{\mathrm{V}}(X^{*\top}\hat{\beta})}\right], \qquad \hat{\mathrm{V}}(X^{*\top}\hat{\beta}) = X^{*\top}\hat{\mathrm{V}}(\hat{\beta})\,X^*,$$
et il suffit d'appliquer g^{-1} aux deux bornes. Cet intervalle reste toujours dans $[0,1]$;
2. méthode du delta : $\hat{\mathrm{V}}(\hat{\pi}^*) = h(X^{*\top}\hat{\beta})\,\hat{\mathrm{V}}(X^{*\top}\hat{\beta})$ avec $h(u) = \{\partial g^{-1}(u)/\partial u\}^2$, d'où l'IC approximatif $\left[\hat{\pi}^* \pm z_{\alpha/2}\sqrt{\hat{\mathrm{V}}(\hat{\pi}^*)}\right]$.

En pratique, les deux méthodes donnent des résultats semblables.

Exemple — IC pour la probabilité de survie en deuxième classe — `empress_agrege`

On reprend le modèle `modele_agrege_classe_logit` (`classe_num` seulement) pour calculer un intervalle de confiance à 95% pour la probabilité de survie d'un passager de deuxième classe, soit $X^* = (1, 2)^\top$.

```
X_etoile <- data.frame(classe_num = 2)

# Prédiction sur l'échelle du prédicteur linéaire, avec erreur-type
eta_etoile <- predict(modele_agrege_classe_logit, newdata = X_etoile,
 ↪  type = "link", se.fit = TRUE)
eta_etoile$fit
```

```
       1
-1.121766
```

```
eta_etoile$se.fit
```

```
[1] 0.09448949
```

```
z <- qnorm(0.975)

# Approche 1 : transformation des bornes sur eta*
bornes_eta <- eta_etoile$fit + c(-1, 1) * z * eta_etoile$se.fit
ic_transform <- plogis(bornes_eta)
ic_transform
```

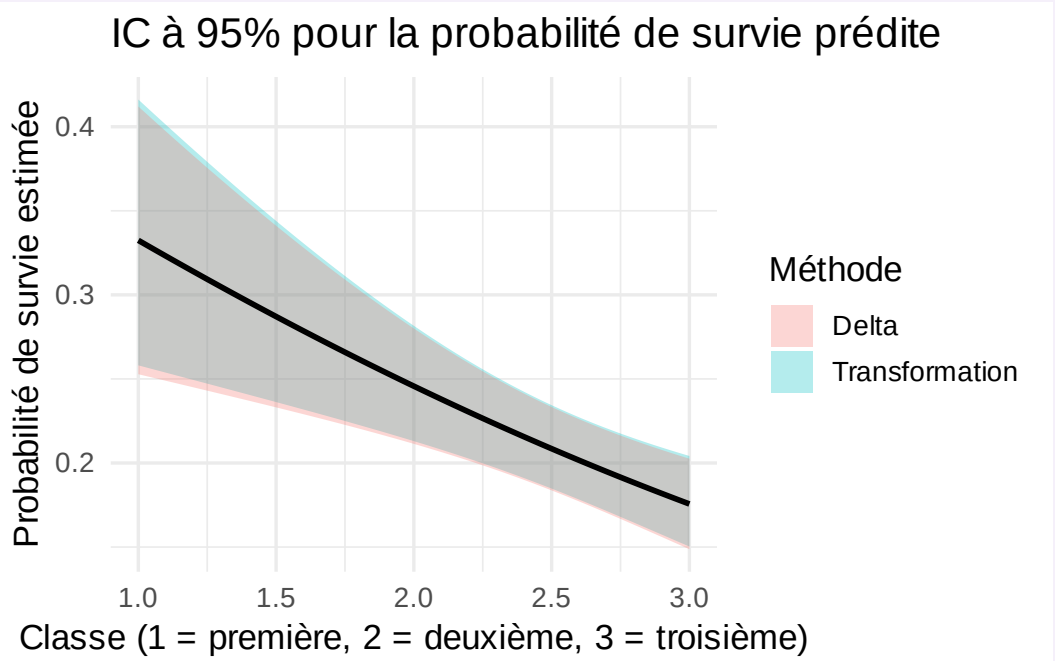
```
[1] 0.2129957 0.2815937
```

```
# Approche 2 : méthode du delta
pi_etoile <- plogis(eta_etoile$fit)
h <- (pi_etoile * (1 - pi_etoile))^2
se_pi <- sqrt(h * eta_etoile$se.fit^2)
ic_delta <- pi_etoile + c(-1, 1) * z * se_pi
ic_delta
```

```
[1] 0.2113627 0.2800050
```

Pour un passager de deuxième classe, la probabilité de survie estimée est $\hat{\pi}^* \approx 0{,}246$. L'intervalle par transformation est [0,213 ; 0,282] et celui par la méthode du delta [0,211 ; 0,280], ce qui les rend presque identiques, comme annoncé.
On étend la comparaison à toute la plage de `classe_num` :

```
grille <- data.frame(classe_num = seq(1, 3, length.out = 200))
pred <- predict(modele_agrege_classe_logit, newdata = grille, type =
 ↪  "link", se.fit = TRUE)

grille <- grille |>
  mutate(
    eta = pred$fit, se_eta = pred$se.fit,
    pi_hat = plogis(eta),
    # approche 1 : transformation des bornes sur eta
    ic_transform_bas  = plogis(eta - z * se_eta),
    ic_transform_haut = plogis(eta + z * se_eta),
    # approche 2 : méthode du delta
    se_pi_delta   = sqrt((pi_hat * (1 - pi_hat))^2 * se_eta^2),
    ic_delta_bas  = pi_hat - z * se_pi_delta,
    ic_delta_haut = pi_hat + z * se_pi_delta
  )

ggplot(grille, aes(x = classe_num)) +
  geom_ribbon(aes(ymin = ic_transform_bas, ymax = ic_transform_haut, fill
   ↪  = "Transformation"),
              alpha = 0.3) +
  geom_ribbon(aes(ymin = ic_delta_bas, ymax = ic_delta_haut, fill =
   ↪  "Delta"),
              alpha = 0.3) +
  geom_line(aes(y = pi_hat), linewidth = 1) +
  labs(
    x = "Classe (1 = première, 2 = deuxième, 3 = troisième)",
    y = "Probabilité de survie estimée",
    fill = "Méthode",
    title = "IC à 95% pour la probabilité de survie prédite"
  ) +
  theme_minimal(base_size = 13)
```



Les deux bandes de confiance se superposent presque sur toute la plage de `classe_num` : avec un échantillon de cette taille et un effet modéré, le choix de la méthode importe peu. Notez que la borne inférieure de l'IC delta descend jusqu'à environ 0,149 en troisième classe (`classe_num` $= 3$) sans devenir négative, mais rien ne le garantit en général. L'IC par transformation reste dans $[0, 1]$ par construction, alors que l'IC delta peut en sortir avec d'autres données ou d'autres valeurs de X^*.

Le cas du lien logit — Méthode du delta

Avec le lien logit, $g^{-1}(u) = \dfrac{e^u}{1+e^u}$, dont la dérivée est

$$\frac{\partial g^{-1}(u)}{\partial u} = \frac{e^u}{(1+e^u)^2} = g^{-1}(u)\left\{1 - g^{-1}(u)\right\}.$$

Ainsi, $h(u) = \left[g^{-1}(u)\{1-g^{-1}(u)\}\right]^2$ et, en $u = X^{*\top}\hat{\beta}$, $h(X^{*\top}\hat{\beta}) = [\hat{\pi}^*(1-\hat{\pi}^*)]^2$. La méthode du delta donne

$$\hat{\mathrm{V}}(\hat{\pi}^*) = [\hat{\pi}^*(1-\hat{\pi}^*)]^2\, \hat{\mathrm{V}}(X^{*\top}\hat{\beta}).$$

Pour aller plus loin avec la théorie — Idée de la méthode du delta

La loi asymptotique de $\hat{\theta} = X^{*\top}\hat{\beta}$ est connue : $\hat{\theta} \overset{\cdot}{\sim} \mathcal{N}(\theta, \hat{\mathrm{V}}(\hat{\theta}))$. Par contre, la loi recherchée est celle de $\hat{\pi}^* = g^{-1}(\hat{\theta})$, une transformation non linéaire de $\hat{\theta}$, et ni la normalité

asymptotique ni la variance ne traversent telles quelles une transformation non linéaire. La méthode du delta remplace g^{-1} par sa tangente autour de θ (développement de Taylor au premier ordre) :

$$g^{-1}(\hat{\theta}) \approx g^{-1}(\theta) + \left.\frac{\partial g^{-1}(u)}{\partial u}\right|_{u=\theta} (\hat{\theta} - \theta).$$

Comme $g^{-1}(\theta) = \pi^*$ est une constante et que le second terme est linéaire en $\hat{\theta}$, la variance se calcule directement :

$$Var[\hat{\pi}^*] \approx \left[\left.\frac{\partial g^{-1}(u)}{\partial u}\right|_{u=\theta}\right]^2 Var[\hat{\theta}],$$

estimée en substituant $\hat{\theta}$ à θ (d'où $\hat{\mathrm{V}}(\hat{\pi}^*) = h(X^{*\top}\hat{\beta})\,\hat{\mathrm{V}}(X^{*\top}\hat{\beta})$, où $h(u) = \{\partial g^{-1}(u)/\partial u\}^2$). Contrairement à la première approche (transformer les bornes de l'IC sur η^*), cette méthode approche directement la variance de $\hat{\pi}^*$ sur son échelle naturelle. Comme elle repose sur une approximation linéaire locale, rien ne garantit que l'intervalle obtenu reste dans $[0, 1]$.

Remarque — Prédictions et comparaisons avec `marginaleffects`

La librairie `marginaleffects` (Arel-Bundock et al., s. d.) fait ces calculs pour la plupart des modèles du livre. Pour un GLM, `predictions` construit par défaut l'intervalle sur l'échelle du prédicteur linéaire, puis transforme ses bornes, comme la première approche. La fonction `avg_comparisons` donne l'effet moyen d'une variable sur la probabilité elle-même, avec son intervalle de confiance (méthode du delta).

```
library(marginaleffects)

# Probabilités de survie prédites et IC à 95 % pour les trois classes
predictions(modele_agrege_classe_logit, newdata = datagrid(classe_num =
 ↪  1:3)) |>
  as.data.frame() |>
  dplyr::select(classe_num, estimate, conf.low, conf.high)
```

```
  classe_num  estimate  conf.low conf.high
1          1 0.3324252 0.2580674 0.4161895
2          2 0.2456839 0.2129957 0.2815937
3          3 0.1756220 0.1503012 0.2041835
```

```
# Variation moyenne de la probabilité de survie pour une classe de plus
```

```r
avg_comparisons(modele_agrege_classe_logit, variables = "classe_num") |>
  as.data.frame() |>
  dplyr::select(term, contrast, estimate, conf.low, conf.high)
```

```
       term contrast    estimate   conf.low   conf.high
1 classe_num       +1 -0.06852344 -0.1033115 -0.03373537
```

Pour la deuxième classe, l'intervalle est celui obtenu plus haut par transformation des bornes. Passer à la classe suivante diminue la probabilité de survie d'environ 6,9 points de pourcentage en moyenne. Contrairement au rapport de cotes $e^{\hat{\beta}_1}$, cette quantité s'exprime directement sur l'échelle des probabilités.

2.3 Sélection de modèles

2.3.1 Déviance

Pour juger si un modèle à $p+1$ paramètres β s'ajuste bien aux données, on le compare au modèle le plus flexible possible, qui donne à chaque observation son propre paramètre, sans structure imposée par les variables explicatives.

Définition — Modèle saturé

Le modèle saturé a n paramètres $\{\tilde{\mu}_1, \ldots, \tilde{\mu}_n\}$ (un par observation, plutôt que $p+1$ pour tout l'échantillon).

On voit facilement que le modèle saturé maximise la log-vraisemblance

$$\ell(\tilde{\mu}_1, \ldots, \tilde{\mu}_n; y) = \sum_{i=1}^{n} \{y_i \log(\tilde{\mu}_i) + (m_i - y_i) \log(m_i - \tilde{\mu}_i)\} + K,$$

en $\{\tilde{\mu}_1, \ldots, \tilde{\mu}_n\} = \{y_1, \ldots, y_n\}$, c'est-à-dire quand chaque paramètre est ajusté exactement sur l'observation correspondante. Le modèle saturé est l'ajustement parfait : il reproduit les données sans aucun résidu, mais il n'a aucune valeur prédictive ou explicative, parce qu'il ne fait que mémoriser les observations. Il sert seulement de référence, à laquelle on compare un modèle plus parcimonieux (avec $p+1 \ll n$ paramètres) pour mesurer ce qui reste à expliquer.

En notant $\tilde{\mu} = (\tilde{\mu}_1, \ldots, \tilde{\mu}_n)^\top$ et $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_n)^\top$, la statistique du rapport de vraisemblance entre le modèle saturé et le modèle considéré est

$$\Lambda = \frac{L(\tilde{\mu}; y)}{L(\hat{\mu}; y)} = \frac{L(y; y)}{L(\hat{\mu}; y)}.$$

Définition — Déviance

La déviance D est deux fois le log de ce rapport :

$$D = 2\left\{\ell(y;y) - \ell(\hat{\mu};y)\right\} = 2\sum_{i=1}^{n}\left\{y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) + (m_i - y_i)\log\left(\frac{m_i - y_i}{m_i - \hat{\mu}_i}\right)\right\} = \sum_{i=1}^{n} d_i,$$

avec la convention $0 \log 0 = 0$ quand $y_i = 0$ ou $y_i = m_i$.

D est la déviance du modèle et d_i la déviance individuelle de l'observation i. Chaque d_i mesure l'écart entre l'ajustement parfait y_i et l'ajustement $\hat{\mu}_i$ du modèle.

Remarque

Le facteur 2 dans la déviance sert à obtenir

$$D \xrightarrow{d} \chi^2_{n-(p+1)} \qquad \text{lorsque } m_i \to \infty,$$

c'est-à-dire que, si le modèle est bien spécifié, la déviance tend en loi vers un khi-deux à $n-(p+1)$ degrés de liberté quand les m_i tendent vers l'infini (n restant fixé).
Notation : la flèche $\xrightarrow{d}$ indique une convergence en distribution (ou en loi). La loi de la quantité à gauche (ici D) se rapproche de celle indiquée à droite (ici $\chi^2_{n-(p+1)}$) quand le paramètre asymptotique (ici m_i) tend vers l'infini. Cela ne veut pas dire que D elle-même converge vers une valeur. C'est aussi une convergence en loi qui justifie la notation $\dot{\sim}$ utilisée plus haut pour $\hat{\beta}$, mais quand $n \to \infty$ et non quand $m_i \to \infty$.
Quand les m_i sont finis mais assez grands (par exemple $m_i\hat{\pi}_i$ et $m_i(1-\hat{\pi}_i)$ d'au moins 5 environ), on utilise

$$D \;\dot{\sim}\; \chi^2_{n-(p+1)}$$

comme approximation, si le modèle est bien spécifié. Avec de petits m_i, cette approximation n'est pas fiable, même si n est grand.

Propriété — Non-négativité de la déviance

Le modèle saturé maximise la vraisemblance parmi tous les modèles possibles, de sorte que $\ell(y;y) \geq \ell(\hat{\mu};y)$ dans tous les cas, d'où $D \geq 0$. Plus D est petit, plus le modèle se rapproche du modèle saturé, donc mieux il s'ajuste aux données.

La déviance joue le rôle de la somme des carrés des résidus en régression linéaire. Elle sert à évaluer l'adéquation du modèle et à comparer des modèles.

2.3.2 Comparaison de modèles emboîtés

Considérons deux modèles emboîtés (définition au chapitre 0), et notons D_{complet} la déviance du modèle complet et $D_{\text{réduit}}$ celle du modèle réduit. La statistique du rapport de vraisemblance $G^2 = D_{\text{réduit}} - D_{\text{complet}}$ sert à les comparer.

Propriété — Distribution asymptotique de $G^2 = D_{\text{réduit}} - D_{\text{complet}}$

Sous H_0 (le modèle réduit suffit), G^2 suit approximativement (asymptotiquement, quand $n \to \infty$) une loi du khi-deux à d degrés de liberté, où d est la différence du nombre de paramètres :

$$G^2 \overset{\cdot}{\sim} \chi^2_d.$$

Contrairement à l'approximation de la loi de D elle-même, ce résultat reste valable quand les m_i sont petits (y compris $m_i = 1$).

Exemple — Ajout du sexe et du groupe d'âge dans l'exemple `empress_agrege`

On teste $H_0 : \beta_{\text{sexe}} = \beta_{\text{âge}} = 0$, où $\beta_{\text{âge}}$ est le coefficient de `groupe_age` (modèle avec `classe_num` seulement contre modèle avec `sexe` et `groupe_age` en plus) :

```
modele_agrege_reduit <- glm(
  cbind(survivants, effectif - survivants) ~ classe_num,
  data = empress_agrege,
  family = binomial
)
modele_agrege_complet <- glm(
  cbind(survivants, effectif - survivants) ~ classe_num + sexe +
   ↪  groupe_age,
  data = empress_agrege,
  family = binomial
)

stat.test <- deviance(modele_agrege_reduit) -
 ↪  deviance(modele_agrege_complet)
ddl.test <- df.residual(modele_agrege_reduit) -
 ↪  df.residual(modele_agrege_complet)
valeurp <- pchisq(stat.test, df = ddl.test, lower.tail = FALSE)
valeurp
```

```
[1] 2.8459e-16
```

Comme la valeur-p $\approx 2{,}85 \times 10^{-16}$ est bien sous le seuil de 5%, on rejette H_0 : le sexe et le groupe d'âge apportent une information que la classe seule ne capture pas. Les coefficients de ce modèle apparaissent plus loin, dans le tableau de l'exemple de diagnostics, et l'effet du sexe y va à l'encontre de l'idée reçue « les femmes et les enfants d'abord ». Ce n'est pas étonnant pour un navire qui a coulé en moins de 15 minutes, sans le temps d'une évacuation ordonnée.

2.3.3 Comparaison de modèles non emboîtés : AIC et BIC

Comme pour le modèle linéaire, on utilise

$$AIC = -2\,\ell(\hat{\mu}; y) + 2(p+1), \qquad BIC = -2\,\ell(\hat{\mu}; y) + \log(n)(p+1).$$

On préfère le modèle qui a le plus petit AIC (ou BIC). En R, ces critères se calculent avec `AIC(modele)` et `AIC(modele, k=log(n))`.

Exemple — Modèles non emboîtés avec `empress_individus`

Comparons deux modèles qui ne sont pas emboîtés l'un dans l'autre : `classe + sexe` et `classe + groupe_age`.

```
modele_indiv_classe_sexe <- glm(survecu ~ classe + sexe, data =
 ↪  empress_individus, family = binomial)

modele_indiv_classe_age <- glm(survecu ~ classe + groupe_age, data =
 ↪  empress_individus, family = binomial)

AIC(modele_indiv_classe_sexe)
```

```
[1] 1022.129
```

```
AIC(modele_indiv_classe_age)
```

```
[1] 1021.349
```

Le modèle 2 (`classe + groupe_age`) a un AIC un peu plus faible que le modèle 1 (`classe + sexe`) et serait donc préféré, mais l'écart est mince (moins d'un point). C'est probablement le signe qu'aucun des deux n'est satisfaisant, chacun omettant une variable importante, comme le confirme la sélection automatique ci-dessous.

2.3.4 Sélection automatique

Les méthodes pas à pas *forward*, *backward* et *stepwise* fonctionnent comme pour le modèle linéaire (fonction `step` avec l'argument `direction`).

Exemple — Sélection automatique sur le jeu de données `empress_individus`

On applique les trois méthodes au modèle additif à trois variables.

```
# Modèle complet : effets additifs des trois variables explicatives
 ↪  disponibles
modele_indiv_complet <- glm(survecu ~ classe + sexe + groupe_age,
                            data = empress_individus, family = binomial)

# Backward : part du modèle complet, retire des termes
modele_indiv_backward <- step(modele_indiv_complet, direction =
 ↪  "backward")
```

```
Start:  AIC=993.66
survecu ~ classe + sexe + groupe_age

             Df Deviance     AIC
<none>            983.66  993.66
- classe      2  1005.84 1011.84
- sexe        1  1013.35 1021.35
- groupe_age  1  1014.13 1022.13
```

```
# Forward : part du modèle vide, ajoute des termes tant que l'AIC diminue
# (au plus jusqu'au modèle complet)
modele_indiv_vide <- glm(survecu ~ 1, data = empress_individus, family =
 ↪  binomial)
modele_indiv_forward <- step(modele_indiv_vide, direction = "forward",
                         scope = formula(modele_indiv_complet))
```

```
Start:  AIC=1075.19
survecu ~ 1

             Df Deviance    AIC
+ groupe_age  1   1031.3 1035.3
+ sexe        1   1040.9 1044.9
```

```
+ classe      2   1051.6 1057.6
<none>            1073.2 1075.2

Step:  AIC=1035.32
survecu ~ groupe_age

         Df Deviance    AIC
+ sexe    1   1005.8 1011.8
+ classe  2   1013.4 1021.4
<none>        1031.3 1035.3

Step:  AIC=1011.84
survecu ~ groupe_age + sexe

         Df Deviance     AIC
+ classe  2   983.66  993.66
<none>       1005.84 1011.84

Step:  AIC=993.66
survecu ~ groupe_age + sexe + classe
```

```r
# Stepwise : part du modèle complet, ajoute et retire au besoin
modele_indiv_final <- step(modele_indiv_complet, direction = "both")
```

```
Start:  AIC=993.66
survecu ~ classe + sexe + groupe_age

             Df Deviance     AIC
<none>            983.66  993.66
- classe      2  1005.84 1011.84
- sexe        1  1013.35 1021.35
- groupe_age  1  1014.13 1022.13
```

```r
# Termes retenus et AIC du modèle choisi par chacune des trois méthodes
selection <- list(backward = modele_indiv_backward, forward =
 ↪  modele_indiv_forward,
                  stepwise = modele_indiv_final)
t(sapply(selection, function(m)
  c(termes = paste(sort(attr(terms(m), "term.labels")), collapse = " +
   ↪  "),
    AIC = format(round(AIC(m), 2)))))
```

```
         termes                          AIC
backward "classe + groupe_age + sexe" "993.66"
forward  "classe + groupe_age + sexe" "993.66"
stepwise "classe + groupe_age + sexe" "993.66"
```

```
summary(modele_indiv_final)$coefficients
```

```
                   Estimate Std. Error   z value     Pr(>|z|)
(Intercept)      -1.8870253  0.2084659 -9.051960 1.404254e-19
classePremière    1.0070485  0.2808246  3.586040 3.357369e-04
classeTroisième  -0.2009025  0.1954994 -1.027637 3.041205e-01
sexeHomme         0.9890825  0.1919365  5.153175 2.561127e-07
groupe_ageEnfant -2.0826480  0.5160819 -4.035499 5.448642e-05
```

Les trois méthodes aboutissent ici au même modèle final (`classe + sexe + groupe_age`), ce qui n'est pas toujours le cas. Chacune ne parcourt qu'un chemin parmi tous les sous-modèles possibles (une variable ajoutée ou retirée à la fois), de sorte que rien ne garantit qu'elles arrivent au même optimum local, encore moins à l'optimum global. Les trois variables sont conservées, parce qu'en retirer une fait augmenter l'AIC.
Notez que les coefficients du modèle final ne sont pas tous significatifs au seuil de 5%. La modalité de référence de `classe` est « Deuxième » (R prend la première par ordre alphabétique), de sorte que `classeTroisième` compare la troisième classe à la deuxième, et `classePremière` la première à la deuxième. La valeur-p de `classeTroisième` est d'environ 0,30, bien au-dessus de 0,05, de sorte qu'on ne rejette pas $H_0 : \beta_{\text{classeTroisième}} = 0$. À l'inverse, `classePremière` (valeur-p $\approx 3{,}36 \times 10^{-4}$), `sexeHomme` (valeur-p $\approx 2{,}56 \times 10^{-7}$) et `groupe_ageEnfant` (valeur-p $\approx 5{,}45 \times 10^{-5}$) sont significatifs. La variable `classe` reste quand même dans le modèle. Ce qui décide de son retrait, c'est la variation de l'AIC quand on retire le facteur en entier (ses deux indicatrices à la fois), ici de 993,66 à 1011,84, et non la significativité d'un seul de ses coefficients. Ce critère n'est pas un test au seuil de 5% : retirer un terme à d paramètres fait baisser l'AIC exactement quand $G^2 < 2d$, ce qui revient à un test du rapport de vraisemblance à un seuil implicite d'environ 0,157 pour $d = 1$ et 0,135 pour $d = 2$.

2.4 Validation et diagnostic de modèles

2.4.1 Résidus

En régression binomiale, on utilise deux types de résidus.

Définition — Résidus de déviance et résidus de Pearson

— Résidus de déviance : $r_{D,i} = \text{signe}(y_i - \hat{\mu}_i)\sqrt{d_i}$, avec d_i la déviance individuelle de l'observation i ;

— résidus de Pearson :

$$r_{P,i} = \frac{y_i - m_i\hat{\pi}_i}{\sqrt{m_i\hat{\pi}_i(1-\hat{\pi}_i)}} = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i(m_i - \hat{\mu}_i)/m_i}}.$$

2.4.2 Tests d'adéquation globale

L'adéquation globale du modèle se teste avec deux statistiques : la déviance D et le khi-deux de Pearson

$$X_P^2 = \sum_{i=1}^{n} r_{P,i}^2 = \sum_{i=1}^{n} \frac{m_i(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i(m_i - \hat{\mu}_i)}.$$

Propriété — Convergence de D et X_P^2

Quand tous les $m_i \to \infty$ (n restant fixé), D et X_P^2 convergent en loi vers une $\chi^2_{n-(p+1)}$ si le modèle est bien spécifié.

L'hypothèse nulle du test est que le modèle est adéquat, de sorte qu'une petite valeur-p mène à rejeter le modèle.

Exemple — Test d'adéquation avec l'exemple `empress_agrege`

Reprenons le modèle `modele_agrege_classe_logit` ajusté plus haut, avec `classe_num` comme seule variable explicative :

```
# Déviance
D <- deviance(modele_agrege_classe_logit)
ddl <- df.residual(modele_agrege_classe_logit)
p.deviance <- pchisq(D, df = ddl, lower.tail = FALSE)

# Khi-deux de Pearson
chi2.pearson <- sum(residuals(modele_agrege_classe_logit, type =
 ↪  "pearson")^2)
p.pearson <- pchisq(chi2.pearson, df = ddl, lower.tail = FALSE)

data.frame(
  Statistique = c("Déviance", "Khi-deux de Pearson"),
  Valeur = c(D, chi2.pearson),
  ddl = ddl,
```

```
  p.valeur = c(p.deviance, p.pearson)
)
```

```
          Statistique   Valeur ddl     p.valeur
1            Déviance 84.93991   9 1.677178e-14
2 Khi-deux de Pearson 68.66520   9 2.780196e-11
```

Les deux valeurs-p (environ $1{,}68 \times 10^{-14}$ pour la déviance et $2{,}78 \times 10^{-11}$ pour le khi-deux de Pearson) sont bien sous le seuil de 5%. On rejette donc l'adéquation du modèle : la classe seule ne suffit pas à expliquer la survie. C'est cohérent avec le test emboîté vu plus haut, où le sexe et le groupe d'âge apportaient une information essentielle.

Remarque — Déviance ou khi-deux de Pearson ?

D et X_P^2 convergent tous deux vers une loi $\chi^2_{n-(p+1)}$ quand les $m_i \to \infty$, mais ils se comportent différemment en échantillon fini. En général, la statistique de Pearson respecte mieux cette approximation que la déviance, surtout quand les m_i sont petits ou modérés (McCullagh et Nelder 1989 ; Agresti 2013).
Avec des données individuelles (Bernoulli, $m_i = 1$), on est dans le cas extrême où les deux tests sont inutilisables. En effet, un des deux termes de d_i s'annule toujours (par la convention $0 \log 0 = 0$), de sorte que

$$d_i = -2\left[y_i \log(\hat{\mu}_i) + (1 - y_i) \log(1 - \hat{\mu}_i)\right].$$

De plus, la log-vraisemblance du modèle saturé vaut alors exactement 0, comme $\tilde{\mu}_i = y_i \in \{0, 1\}$ annule $y_i \log(y_i) + (1 - y_i) \log(1 - y_i)$ par la même convention. La déviance se réduit donc à

$$D = \sum_{i=1}^{n} d_i = -2\,\ell(\hat{\mu}; y),$$

c'est-à-dire une simple transformation de la vraisemblance du modèle, sans point de comparaison, de sorte qu'elle perd son interprétation comme test d'adéquation. X_P^2, lui, reste au moins défini dans ce cas, même si son approximation χ^2 n'est pas fiable non plus. Quand les deux statistiques mènent à la même conclusion (avec des m_i raisonnablement grands), peu importe laquelle on rapporte. Il ne faut pas voir un désaccord marqué entre les deux valeurs-p comme le signe que l'un des deux tests est le bon. C'est plutôt l'approximation asymptotique commune aux deux tests qui devient peu fiable.
Remarquez que cette distinction ne concerne que le test d'adéquation globale d'un seul modèle. Pour comparer des modèles emboîtés, la déviance garde un avantage clair. Sous H_0, les différences de déviances G^2 suivent approximativement une loi χ^2 dès que n est

grand (le nombre de paramètres restant fixé), même quand les m_i sont petits (y compris $m_i = 1$), ce qui n'est pas le cas des différences de statistiques de Pearson.

2.4.3 Adéquation avec des données individuelles

Avec des données de Bernoulli ($m_i = 1$), ni D ni X^2_P ne permettent de tester l'adéquation (voir la remarque précédente). Trois outils regroupent alors les observations selon leur probabilité prédite, pour retrouver des effectifs assez grands dans chaque groupe.

Définition — Test de Hosmer-Lemeshow

On classe les observations en G groupes de tailles à peu près égales selon les probabilités prédites $\hat{\pi}_i$ (souvent $G = 10$, les déciles). Dans le groupe g, de taille n_g, notons O_g le nombre observé de succès et $E_g = \sum_{i \in g} \hat{\pi}_i$ le nombre prédit. La statistique

$$C = \sum_{g=1}^{G} \frac{(O_g - E_g)^2}{E_g \left(1 - E_g/n_g\right)}$$

suit approximativement une loi χ^2_{G-2} si le modèle est bien spécifié (Hosmer et al. 2013).

Le test a des limites. Le résultat dépend du nombre de groupes, la puissance est faible avec un petit échantillon, et un modèle qui n'a que quelques combinaisons distinctes de variables explicatives (comme ceux d'`empress_individus`, qui en ont au plus 11) crée des ex æquo dans les $\hat{\pi}_i$. Dans ce dernier cas, il vaut mieux agréger les données et revenir au test de la Section 2.4.2. Le test est offert par la fonction `hoslem.test` de la librairie `ResourceSelection`, mais il se programme en quelques lignes.

Deux graphiques complètent le test :

- résidus groupés (*binned residuals*) : pour chaque groupe, la moyenne des résidus bruts $y_i - \hat{\pi}_i$ en fonction de la probabilité prédite moyenne, avec une bande de ± 2 erreurs-types, $\pm 2\sqrt{\bar{\pi}_g(1-\bar{\pi}_g)/n_g}$. Si le modèle est bien spécifié, environ 95 % des points tombent dans la bande, sans tendance. La fonction `binnedplot` de la librairie `arm` trace ce graphique (Gelman et Hill 2007) ;
- courbe de calibration : la proportion observée de succès de chaque groupe en fonction de la probabilité prédite moyenne, qui devrait suivre la droite identité. Une probabilité prédite de 0,3 doit correspondre à environ 30 % de succès. Notez qu'un modèle logistique ajusté par maximum de vraisemblance est toujours bien calibré en moyenne sur ses propres données d'entraînement. La calibration s'évalue donc surtout sur des données test (voir le Chapitre 5).

Exemple — Adéquation sur des données simulées

On simule $n = 500$ observations dont la vraie relation est quadratique sur l'échelle logit, $\text{logit}\,\pi(x) = -1{,}5 + 0{,}5\,x^2$, puis on ajuste deux modèles : l'un linéaire en x (mal spécifié), l'autre quadratique.

```
set.seed(4300)
n_sim <- 500
sim_adeq <- tibble(x = runif(n_sim, -3, 3)) |>
  mutate(y = rbinom(n_sim, 1, plogis(-1.5 + 0.5 * x^2)))

m_lin  <- glm(y ~ x, family = binomial, data = sim_adeq)
m_quad <- glm(y ~ x + I(x^2), family = binomial, data = sim_adeq)

# Test de Hosmer-Lemeshow avec G groupes formés selon les probabilités
↪  prédites
hosmer_lemeshow <- function(modele, G = 10) {
  tibble(y = modele$y, p = fitted(modele)) |>
    mutate(groupe = ntile(p, G)) |>
    group_by(groupe) |>
    summarise(n = n(), O = sum(y), E = sum(p), .groups = "drop") |>
    summarise(C = sum((O - E)^2 / (E * (1 - E / n)))) |>
    mutate(ddl = G - 2, valeur_p = pchisq(C, ddl, lower.tail = FALSE))
}
hl <- bind_rows(lineaire = hosmer_lemeshow(m_lin),
                quadratique = hosmer_lemeshow(m_quad), .id = "modele")
hl
```

```
# A tibble: 2 x 4
  modele           C   ddl valeur_p
  <chr>        <dbl> <dbl>    <dbl>
1 lineaire    155.       8 2.24e-29
2 quadratique   6.24     8 6.20e- 1
```

Le test rejette nettement le modèle linéaire ($C \approx 154{,}5$ pour 8 degrés de liberté), mais pas le modèle quadratique (valeur-p $\approx 0{,}62$). Les résidus groupés montrent la forme de l'écart :

```
residus_groupes <- function(modele, G = 20) {
  tibble(r = modele$y - fitted(modele), p = fitted(modele)) |>
    mutate(groupe = ntile(p, G)) |>
    group_by(groupe) |>
```

```
    summarise(p_moy = mean(p), r_moy = mean(r), n = n(), .groups =
     ↪  "drop") |>
    mutate(borne = 2 * sqrt(p_moy * (1 - p_moy) / n))
}
bind_rows(`Linéaire en x` = residus_groupes(m_lin),
          `Quadratique en x` = residus_groupes(m_quad), .id = "modele")
           ↪  |>
  ggplot(aes(p_moy, r_moy)) +
  geom_ribbon(aes(ymin = -borne, ymax = borne), fill = "grey85") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_point() +
  facet_wrap(~ modele) +
  labs(x = "Probabilité prédite moyenne du groupe", y = "Résidu moyen du
   ↪  groupe") +
  theme_minimal(base_size = 13)
```

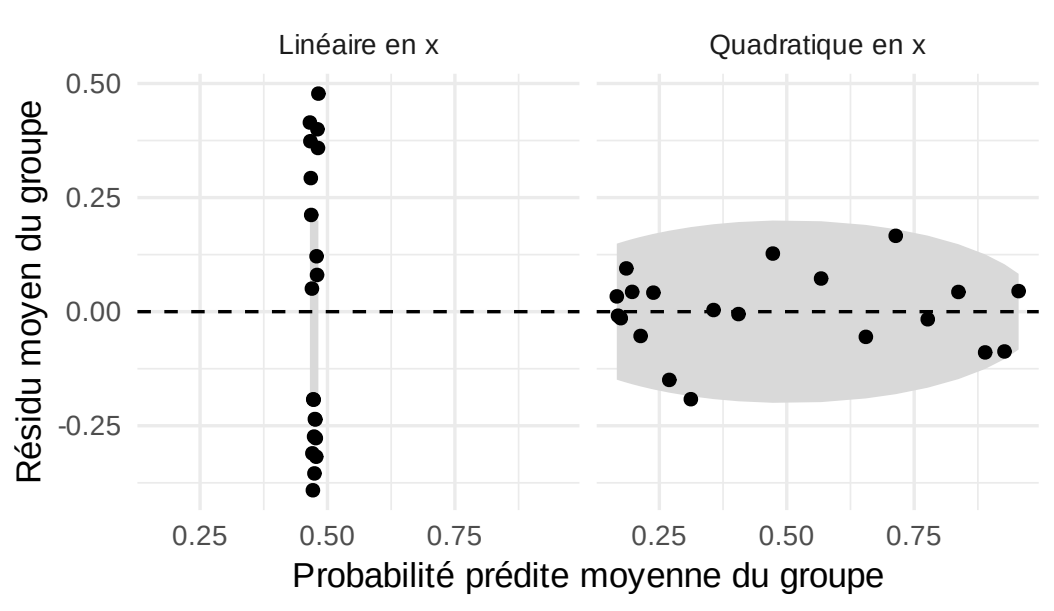


FIGURE 2.1 : Résidus groupés (20 groupes) des modèles linéaire et quadratique ajustés aux données simulées. La bande grise correspond à ± 2 erreurs-types.

Avec le modèle linéaire, plusieurs groupes sortent de la bande et les résidus moyens suivent une tendance : le modèle sous-estime les probabilités pour certaines valeurs prédites et les surestime pour d'autres. Avec le modèle quadratique, les points restent dans la bande, autour de 0.

2.4.4 Graphiques et surdispersion

Le graphique des résidus en fonction du numéro d'observation ne doit montrer aucune tendance. Si plusieurs résidus dépassent 3 en valeur absolue, on soupçonne de la surdispersion.

> **Définition — Surdispersion (overdispersion)**
> La surdispersion (*overdispersion*) survient quand $E[Y_i] = m_i\pi_i$ mais $Var[Y_i] > m_i\pi_i(1-\pi_i)$. Le modèle pose alors $Var[Y_i] = \phi\, m_i\pi_i(1-\pi_i)$, et ϕ est estimé par $\tilde{\phi}_P = X^2_P/(n-(p+1))$.

En cas de surdispersion, il suffit de spécifier `family=quasibinomial()` au lieu de `family=binomial()` dans `glm`.

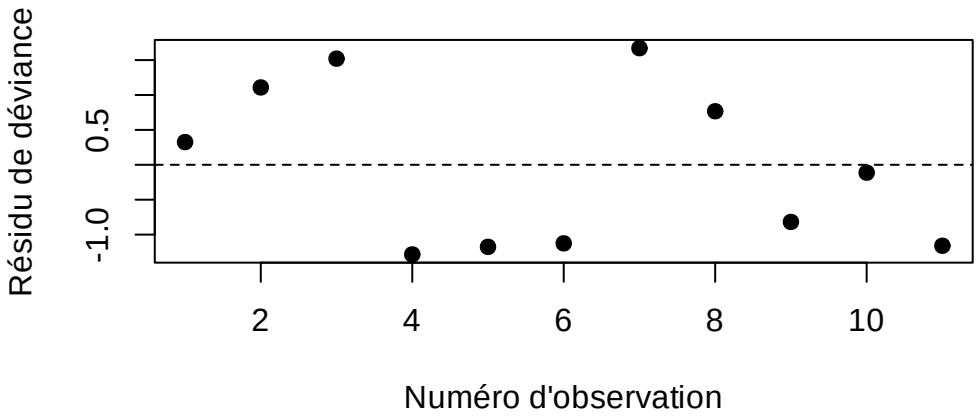
La linéarité du prédicteur se vérifie avec le nuage de points $\{(\tilde{\pi}_i, \hat{\pi}_i)\}$ ($\tilde{\pi}_i$ en abscisse), où $\hat{\pi}_i = g^{-1}(\hat{\eta}_i)$ et $\tilde{\pi}_i = y_i/m_i$. Si le modèle est bien spécifié, les points se placent autour de la droite identité (la première bissectrice).

2.4.5 Observations aberrantes et valeurs influentes

Le principe est le même qu'en régression linéaire (chapitre 0). Les définitions des leviers, des résidus standardisés et des résidus studentisés changent, mais leur interprétation reste la même, tout comme celle de la distance de Cook, des DFBETAS et des DFFITS.

> **Exemple — Diagnostics et surdispersion — données `empress_agrege`**
> On reprend le modèle `modele_agrege_complet` ajusté plus haut (`classe_num + sexe + groupe_age`) sur les données agrégées pour passer en revue les diagnostics de cette section.
> Commençons par les résidus en fonction du numéro d'observation :

```
res_dev <- residuals(modele_agrege_complet, type = "deviance")

plot(res_dev, xlab = "Numéro d'observation", ylab = "Résidu de déviance",
     pch = 19)
abline(h = 0, lty = 2)
abline(h = c(-3, 3), lty = 3, col = "red")
```



Aucun résidu ne dépasse les seuils à ± 3 (pointillés rouges), mais plusieurs approchent quand même $\pm 1{,}5$ pour seulement 11 observations, ce qui reste un signal à vérifier. Estimons ensuite ϕ et passons au modèle quasibinomial :

```
chi2_p <- sum(residuals(modele_agrege_complet, type = "pearson")^2)
n <- nrow(empress_agrege)
p1 <- length(coef(modele_agrege_complet))  # nombre de paramètres, soit
 ↪  p+1
phi_hat <- chi2_p / (n - p1)
phi_hat
```

```
[1] 2.393373
```

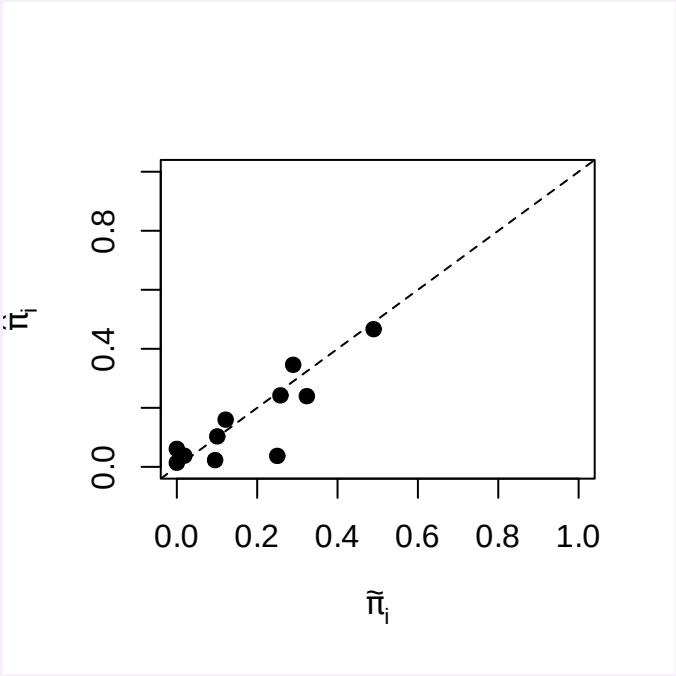
```
modele_agrege_complet_quasi <- glm(cbind(survivants, effectif -
 ↪  survivants) ~ classe_num + sexe + groupe_age,
                  family = quasibinomial, data = empress_agrege)

# Estimations, erreurs-types et valeurs-p des modèles binomial et
 ↪  quasibinomial
coef_bin   <- summary(modele_agrege_complet)$coefficients
coef_quasi <- summary(modele_agrege_complet_quasi)$coefficients
round(cbind(Estimate          = coef_bin[, "Estimate"],
            "SE binomial"     = coef_bin[, "Std. Error"],
            "SE quasi"        = coef_quasi[, "Std. Error"],
```

```
            "Pr(>|z|) binom." = coef_bin[, "Pr(>|z|)"],
            "Pr(>|t|) quasi"  = coef_quasi[, "Pr(>|t|)"]), 4)
```

```
                 Estimate SE binomial SE quasi Pr(>|z|) binom. Pr(>|t|) quasi
(Intercept)       -0.6517      0.3099   0.4794          0.0355         0.2162
classe_num        -0.5034      0.1162   0.1798          0.0000         0.0266
sexeHomme          1.0211      0.1905   0.2947          0.0000         0.0105
groupe_ageEnfant  -2.0991      0.5161   0.7985          0.0000         0.0340
```

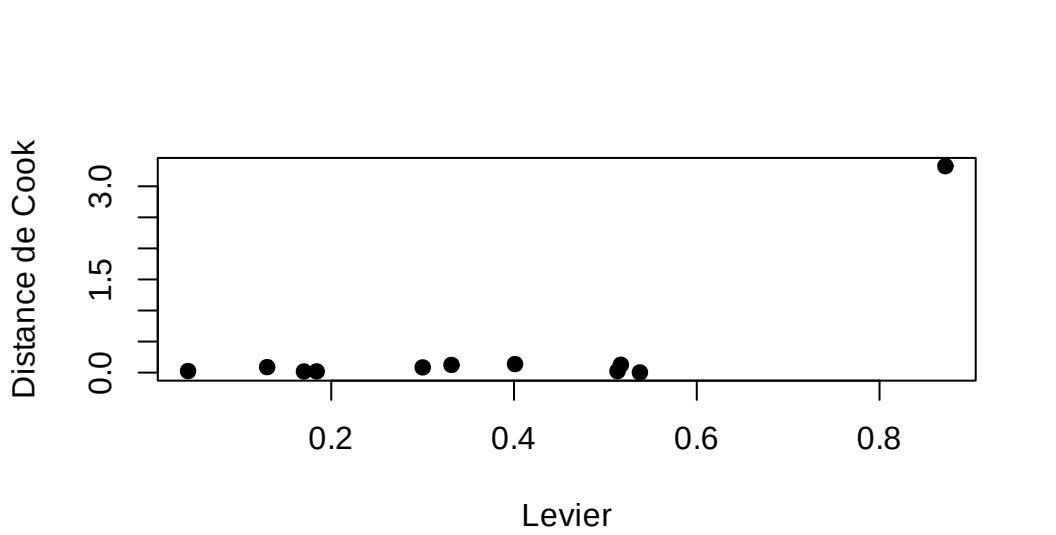
On obtient $\tilde{\phi}_P \approx 2{,}4$, c'est-à-dire une variance observée environ 2,4 fois celle du modèle binomial. Cela pourrait suggérer de la surdispersion, mais il faut lire cette valeur avec prudence. Elle repose sur 11 groupes et 7 degrés de liberté seulement, et elle est gonflée par deux petits groupes d'enfants de sexe féminin (observations 3 et 7, résidus de Pearson d'environ 2,2 et 2,2). Les tests d'adéquation ne concordent d'ailleurs pas : $X^2_P \approx 16{,}8$ donne une valeur-p $\approx 0{,}019$, mais la déviance ($D \approx 13{,}3$) donne une valeur-p $\approx 0{,}064$, au-dessus de 5%. Le chapitre sur les GLM modélise les mêmes données par une régression de Poisson et n'y détecte aucune surdispersion (Section 3.6.5). On ajuste quand même le modèle quasibinomial, pour illustrer la démarche et vérifier que les conclusions résistent à des erreurs-types plus grandes. Ce modèle gonfle les erreurs-types d'un facteur $\sqrt{\tilde{\phi}_P}$, sans changer les estimations des β_j (et `summary()` remplace alors le test z par un test t à $n-(p+1)$ degrés de liberté, comme en régression linéaire, d'où la colonne `Pr(>|t|)`). Malgré ce gonflement, les trois valeurs-p restent sous le seuil de 5% (`classe_num` : valeur-p $\approx 0{,}027$; `sexeHomme` : valeur-p $\approx 0{,}010$; `groupe_ageEnfant` : valeur-p $\approx 0{,}034$). Les trois effets restent donc significatifs même si l'on tient compte d'une éventuelle surdispersion. Vérifions maintenant la linéarité, $\hat{\pi}_i$ contre $\tilde{\pi}_i$:

```
pi_hat   <- fitted(modele_agrege_complet_quasi)
pi_tilde <- empress_agrege$survivants / empress_agrege$effectif

plot(pi_tilde, pi_hat, xlab = expression(tilde(pi)[i]),
     ylab = expression(hat(pi)[i]), pch = 19,
     xlim = c(0, 1), ylim = c(0, 1))
abline(0, 1, lty = 2)
```



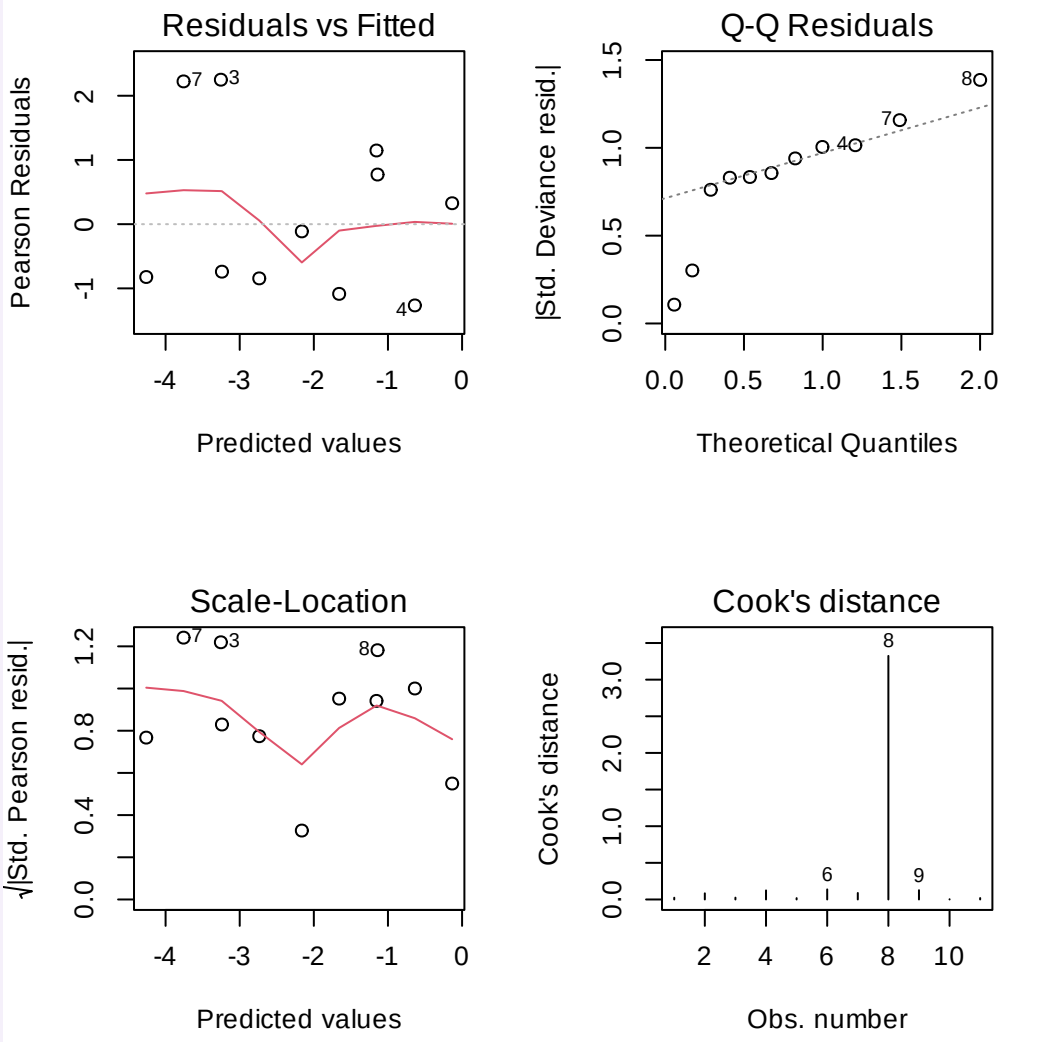
Les points suivent la droite identité dans l'ensemble (corrélation d'environ 0,87), mais la dispersion est notable pour un aussi petit nombre de groupes, surtout pour les petits groupes, dont les proportions observées $\tilde{\pi}_i$ sont très variables.
Restent les observations aberrantes et les valeurs influentes :

```
lev   <- hatvalues(modele_agrege_complet_quasi)
rstd  <- rstandard(modele_agrege_complet_quasi, type = "pearson")
rstu  <- rstudent(modele_agrege_complet_quasi)
cook  <- cooks.distance(modele_agrege_complet_quasi)
dfb   <- dfbetas(modele_agrege_complet_quasi)
dff   <- dffits(modele_agrege_complet_quasi)

plot(lev, cook, xlab = "Levier", ylab = "Distance de Cook", pch = 19)
```



La fonction `plot()` produit quatre graphiques de diagnostic, réunis ici dans une seule figure.

```
par(mfrow = c(2, 2))
plot(modele_agrege_complet_quasi, which = 1:4)
```



— Résidus contre valeurs ajustées (en haut à gauche) : deux points sont nettement au-dessus des autres (observations 3 et 7, les deux petits groupes d'enfants de sexe féminin déjà repérés) ;
— diagramme quantile-quantile (en haut à droite) : il est ici demi-normal, puisque pour un modèle ajusté avec `glm()`, R trace depuis la version 4.3 les valeurs absolues des résidus de déviance standardisés. Il ne montre pas d'écart marqué dans sa partie haute. Les deux plus petits résidus tombent sous la droite, ce qui est peu informatif avec seulement 11 observations ;
— *scale-location* (en bas à gauche) : la dispersion des résidus n'est pas tout à fait

constante ;
— distance de Cook (en bas à droite) : une seule observation est influente.

Le seuil usuel pour un levier élevé est $2(p+1)/n$, soit ici $2 \times 4/11 \approx 0{,}73$. L'observation 8 (troisième classe, hommes adultes) le dépasse, avec un levier de 0,87 et une distance de Cook de 3,32, alors que les distances de Cook des autres observations sont loin derrière (toutes sous 0,14). Ses DFBETAS (jusqu'à 1,9 en valeur absolue) et son DFFITS ($\approx 4{,}0$, contre moins de 0,8 en valeur absolue ailleurs) confirment qu'elle déplace fortement l'ajustement. Ce n'est pas étonnant : ce groupe compte à lui seul 446 des 1057 passagers (42 %), de sorte que le modèle doit reproduire son taux de survie, sinon la déviance globale explose. C'est un effet de poids d'échantillon, pas une anomalie des données.

2.5 Spécificité, sensibilité et courbe ROC

Avec des données individuelles, l'adéquation est difficile à évaluer (voir la remarque Déviance ou khi-deux de Pearson ?), mais la capacité prédictive du modèle peut se mesurer. On ajuste d'abord le modèle et on calcule les probabilités prédites $\hat{\pi}_i = g^{-1}(\hat{\eta}_i)$.

On fixe ensuite un seuil $u \in [0,1]$ (souvent $u = 1/2$), et chaque observation est classée $\hat{y}_i = 1$ si $\hat{\pi}_i > u$, et $\hat{y}_i = 0$ sinon.

Définition — Spécificité et sensibilité

La capacité prédictive se mesure par

$$\begin{aligned} \text{Spécificité (taux de vrais négatifs) :} \quad & Spec = P(\hat{y} = 0 \mid y = 0), \\ \text{Sensibilité (taux de vrais positifs) :} \quad & Sens = P(\hat{y} = 1 \mid y = 1). \end{aligned}$$

À u fixé, elles s'estiment par

$$\widehat{Spec}(u) = \frac{\sum_{i=1}^{n} \mathbf{1}_{\{(\hat{\pi}_i \le u)\,\&\,(y_i = 0)\}}}{\sum_{i=1}^{n} \mathbf{1}_{\{y_i = 0\}}}, \qquad \widehat{Sens}(u) = \frac{\sum_{i=1}^{n} \mathbf{1}_{\{(\hat{\pi}_i > u)\,\&\,(y_i = 1)\}}}{\sum_{i=1}^{n} \mathbf{1}_{\{y_i = 1\}}}.$$

Définition — Courbe ROC et AUC

La courbe ROC est le tracé des points $\{(1 - \widehat{Spec}(u_b),\ \widehat{Sens}(u_b))\}$ obtenus en faisant varier u sur une grille $\{u_1 = 0, u_2, \ldots, u_B = 1\}$. L'aire sous la courbe (AUC) mesure la capacité prédictive. Une AUC de 0,5 correspond à un classement au hasard (la diagonale), et une AUC de 1 à un classement parfait. À titre indicatif (il s'agit d'une convention, pas

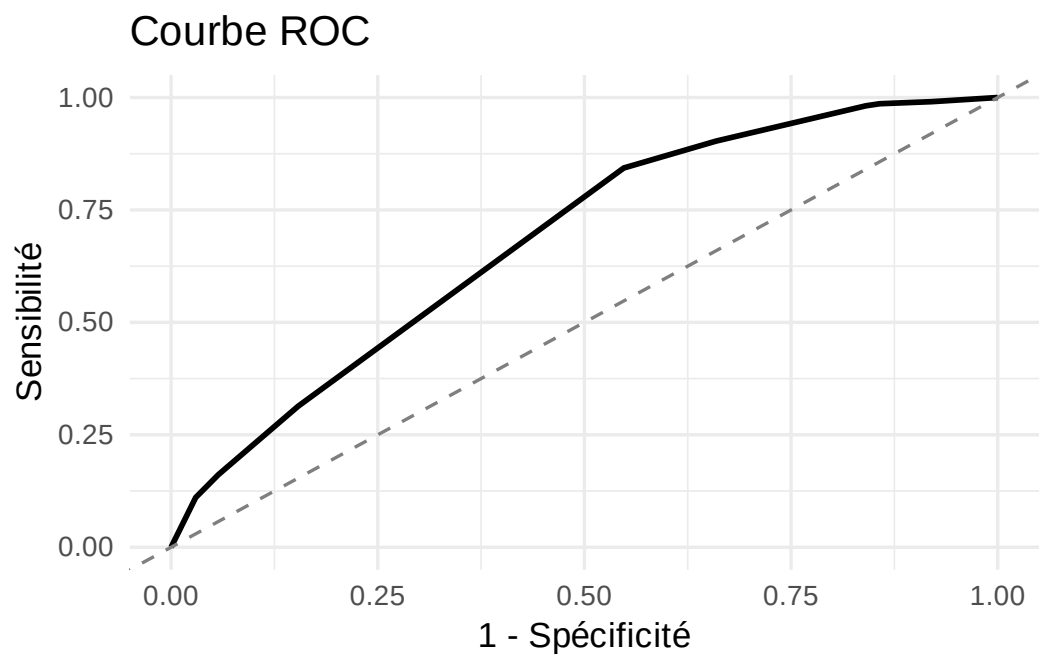
d'un résultat, et d'autres auteurs utilisent d'autres seuils et d'autres qualificatifs) :

$$\text{AUC} \in \begin{cases} [0{,}9,\ 1] & \text{modèle excellent,} \\ [0{,}8,\ 0{,}9) & \text{modèle bon,} \\ [0{,}7,\ 0{,}8) & \text{modèle moyen,} \\ [0{,}6,\ 0{,}7) & \text{modèle mauvais,} \\ [0{,}5,\ 0{,}6) & \text{modèle très faible,} \\ [0,\ 0{,}5) & \text{pire que le hasard (classement inversé).} \end{cases}$$

Exemple — Courbe ROC avec l'exemple `empress_individus`

On reprend `modele_indiv_final`, le modèle retenu par la sélection automatique (`classe + sexe + groupe_age`), et on construit la courbe ROC directement à partir des estimateurs $\widehat{Spec}(u)$ et $\widehat{Sens}(u)$ ci-dessus :

```
pi_hat <- fitted(modele_indiv_final)
y <- empress_individus$survecu

seuils <- seq(0, 1, by = 0.01)
sens <- sapply(seuils, function(u) mean(pi_hat[y == 1] > u))
spec <- sapply(seuils, function(u) mean(pi_hat[y == 0] <= u))

roc <- data.frame(seuil = seuils, sensibilite = sens, specificite = spec)
 ↪  |>
  arrange(1 - specificite)

ggplot(roc, aes(x = 1 - specificite, y = sensibilite)) +
  geom_line(linewidth = 1) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color =
   ↪  "grey50") +
  labs(x = "1 - Spécificité", y = "Sensibilité", title = "Courbe ROC") +
  theme_minimal(base_size = 13)
```




```
# Aire sous la courbe par la règle du trapèze
auc <- sum(diff(1 - roc$specificite) *
             (head(roc$sensibilite, -1) + tail(roc$sensibilite, -1)) / 2)
auc
```

```
[1] 0.6825598
```

L'aire sous la courbe est d'environ 0,68, de sorte que le modèle est mauvais selon l'échelle ci-dessus. La classe, le sexe et le groupe d'âge influencent la survie (les tests précédents le confirment), mais ils sont loin de suffire pour prédire, passager par passager, qui a survécu. Beaucoup d'autres facteurs (emplacement au moment de l'impact, capacité physique, chance) ont joué.

Remarque — À retenir

— Rapport de cotes : avec le lien logit, une augmentation d'une unité de x_{ij} (les autres variables restant fixes) multiplie la cote $o_i = \pi_i/(1-\pi_i)$ par e^{β_j}. C'est en bonne partie pour cette interprétation que le logit est le lien le plus utilisé, même si les liens probit et c-log-log donnent souvent des résultats semblables ;

— Séparation : si les covariables séparent parfaitement (ou presque) les deux groupes,

l'estimateur du maximum de vraisemblance n'existe pas, et R affiche souvent `glm.fit: fitted probabilities numerically 0 or 1 occurred`, avec des coefficients et des erreurs-types énormes. Changer de test ou d'intervalle ne suffit pas, il faut changer d'estimateur (correction de Firth avec `logistf`, régularisation) ou de données ;

— Tests : des trois tests sur β_j (Wald, score de Rao, rapport de vraisemblance), celui de Wald, affiché par défaut par `summary`, devient trompeur avec un effet fort ou des données presque séparées (phénomène de Hauck-Donner). Comparez alors avec `anova(modele_reduit, modele_complet, test = "Chisq")`, et notez que `confint()` donne l'intervalle par vraisemblance profilée, `confint.default()` celui de Wald ;

— Déviance : la différence de déviances $G^2 = D_{\text{réduit}} - D_{\text{complet}}$ compare des modèles emboîtés même avec $m_i = 1$, alors que D et X^2_P ne testent l'adéquation globale que si les m_i sont assez grands. Avec des données individuelles, ces deux tests sont inutilisables (d'où le test de Hosmer-Lemeshow et les résidus groupés), et la courbe ROC et l'AUC mesurent plutôt la capacité prédictive ;

— Surdispersion : $\tilde{\phi}_P = X^2_P/(n-(p+1))$ estime ϕ, et `family = quasibinomial` multiplie les erreurs-types par $\sqrt{\tilde{\phi}_P}$ sans changer les estimations des β_j, avec un test t à la place du test z. Lisez $\tilde{\phi}_P$ avec prudence quand il repose sur peu de degrés de liberté.

2.6 Exercices

Les exercices sont classés par niveau de difficulté, et les exercices appliqués de la fin demandent d'analyser un jeu de données en R. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

2.6.1 Niveau facile

Exercice 1. Pour chacune des trois fonctions de lien présentées au Chapitre 2 (logit, probit, c-log-log), vérifiez que g^{-1} est bien l'inverse de g. Il s'agit de vérifier que $g\{g^{-1}(\eta)\} = \eta$ pour tout $\eta \in \mathbb{R}$.

Voir la solution

Exercice 2. Avec le lien logit, la cote s'écrit $o_i = \pi_i/(1-\pi_i) = e^{\eta_i}$. Calculez o_i et π_i pour $\eta_i = 0$, $\eta_i = \log 2$ et $\eta_i = -\log 3$.

Voir la solution

Exercice 3. Soit $Y_i \sim \text{Bin}(m_i, \pi_i)$, avec $\mu_i = m_i\pi_i$. Montrez que

$$Var[Y_i] = \frac{\mu_i(m_i - \mu_i)}{m_i}$$

est maximale, pour m_i fixé, lorsque $\pi_i = 1/2$. Quel est le lien avec le poids $w_i = m_i\pi_i(1-\pi_i)$ utilisé dans l'estimation par IRLS ?

Voir la solution

2.6.2 Niveau moyen

Exercice 1. En partant de la log-vraisemblance avec lien logit,

$$\ell(\beta; y) = \sum_{i=1}^{n} \left\{ \log \binom{m_i}{y_i} + y_i\eta_i - m_i \log\left(1 + e^{\eta_i}\right) \right\}, \qquad \eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip},$$

dérivez le score $U(\beta) = \partial\ell/\partial\beta$ et montrez qu'il s'écrit, sous forme matricielle, $U(\beta) = X^\top(y - \mu)$.

Voir la solution

Exercice 2. Dérivez une fois de plus le résultat de l'exercice précédent par rapport à β. Montrez que la matrice d'information de Fisher pour le lien logit s'écrit $I(\beta) = X^\top W X$, où $W = \text{diag}(w_1, \ldots, w_n)$ et $w_i = m_i\pi_i(1-\pi_i)$.

Voir la solution

Exercice 3. Avec un seul prédicteur continu x et le lien logit, $\hat{\pi}(x) = g^{-1}(\hat{\beta}_0 + \hat{\beta}_1 x)$. Montrez que

$$\hat{\pi}'(x) = \hat{\beta}_1\, \hat{\pi}(x)\{1 - \hat{\pi}(x)\}$$

et que, si $\hat{\beta}_1 > 0$, cette pente est maximale au point d'inflexion $\hat{\pi}(x) = 1/2$, où elle vaut $\hat{\beta}_1/4$. Que se passe-t-il si $\hat{\beta}_1 < 0$?

Voir la solution

2.6.3 Niveau difficile

Exercice 1. Montrez que, lorsque les $m_i \to \infty$, la déviance individuelle d_i est asymptotiquement équivalente au carré du résidu de Pearson $r^2_{P,i}$, c'est-à-dire que

$$d_i = 2\left\{ y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) + (m_i - y_i)\log\left(\frac{m_i - y_i}{m_i - \hat{\mu}_i}\right)\right\} \approx \frac{(y_i - \hat{\mu}_i)^2\, m_i}{\hat{\mu}_i(m_i - \hat{\mu}_i)} = r^2_{P,i}.$$

Déduisez-en que $D \approx X_P^2$ pour des m_i grands. (Indice : posez $\delta = y_i - \hat{\mu}_i$ et effectuez un développement de Taylor à l'ordre 2 de $\log(1+\delta/\hat{\mu}_i)$ et $\log(1-\delta/(m_i-\hat{\mu}_i))$ autour de $\delta = 0$.)

Voir la solution

Exercice 2. Montrez que, pour deux modèles emboîtés de déviances $D_{\text{réduit}}$ et D_{complet}, la différence de déviances $D_{\text{réduit}} - D_{\text{complet}}$ est exactement égale à la statistique du test du rapport de vraisemblance

$$G^2 = -2\left\{\ell(\text{modèle réduit}) - \ell(\text{modèle complet})\right\}.$$

Expliquez pourquoi le terme associé au modèle saturé n'apparaît pas dans cette différence, alors qu'il intervient dans la définition de chaque déviance prise individuellement.

Voir la solution

Exercice 3. Considérez un modèle logistique à un seul prédicteur x, sans ordonnée à l'origine ($\eta_i = \beta x_i$). Supposez qu'il y a séparation complète, au sens où $y_i = 1$ si $x_i > 0$ et $y_i = 0$ si $x_i < 0$ (aucun x_i n'est nul). Montrez que le score $U(\beta) = \sum_i (y_i - \pi_i) x_i$ est strictement positif pour toute valeur finie de β. Concluez que l'estimateur du maximum de vraisemblance $\hat{\beta}$ n'existe pas, c'est-à-dire que $\ell(\beta)$ n'atteint son supremum qu'à la limite $\beta \to \infty$.

Voir la solution

Exercice 4. Considérez un modèle logistique à un seul prédicteur x (sans ordonnée à l'origine, données individuelles $m_i = 1$). Pour ce modèle, l'information de Fisher est $I(\beta) = \sum_{i=1}^{n} w_i x_i^2$, avec $w_i = \pi_i(1-\pi_i)$. Supposez que les données présentent une séparation complète comme à l'exercice précédent. Supposez aussi que l'algorithme d'estimation s'arrête à une valeur $\hat{\beta}$ grande mais finie (à cause du critère d'arrêt numérique). Montrez que, pour des $x_i \neq 0$ fixés, w_i décroît exponentiellement vite vers 0 quand $\hat{\beta} \to \infty$. Expliquez ensuite pourquoi la statistique de Wald $z_{obs} = \hat{\beta}\sqrt{I(\hat{\beta})}$ tend alors vers 0 plutôt que vers l'infini (phénomène de Hauck-Donner).

Voir la solution

2.6.4 Exercices appliqués

Exercice appliqué 1. Sélection de modèles et données presque séparées. Le jeu de données `nminer` de la librairie `GLMsData` (Dunn et Smyth 2018) décrit 31 transects de deux hectares dans des boisés de buloke (un casuarina indigène) des plaines de Wimmera, dans l'ouest de Victoria (Australie). On s'intéresse à la présence du mineur bruyant (*Manorina melanocephala*), un oiseau indigène agressif qui chasse les espèces plus petites. La variable `Minerab` donne le nombre de mineurs bruyants observés dans le transect. Les variables explicatives décrivent l'habitat : nombre d'eucalyptus dans le transect (`Eucs`), superficie en hectares de la parcelle de végétation où se trouve le transect (`Area`), indicateur de pâturage (`Grazed`; selon l'aide de `GLMsData`, 0 = pâturé et 1 = non pâturé), présence d'arbustes (`Shrubs`), nombre d'arbres de type buloke (`Bulokes`) et nombre de morceaux de bois mort au sol (`Timber`).

a) Construisez la réponse `Detecte`, qui vaut 1 si au moins un mineur bruyant a été observé dans le transect et 0 sinon, et vérifiez qu'elle coïncide avec la variable `Miners` du jeu de données. Ajustez les modèles logistiques `Detecte ~ Area + Grazed + Shrubs + Timber` et `Detecte ~ Eucs + Grazed + Bulokes`. Lequel préférez-vous selon l'AIC ? Pourquoi ne peut-on pas les comparer par une différence de déviances ?

b) Appliquez les sélections *backward*, *forward* et *stepwise* de la fonction `step()`, à partir du modèle additif à six variables (ou du modèle vide pour *forward*). Les trois méthodes retiennent-elles le même modèle ? Est-ce garanti en général ?

c) Réajustez le modèle retenu et examinez les avertissements de R, les coefficients, leurs erreurs-types et les probabilités ajustées. Quels symptômes décrits à la Section 2.2.2 reconnaissez-vous ? Réajustez le modèle avec un critère de convergence plus strict (`control = glm.control(epsilon = 1e-14, maxit = 100)`) pour déterminer si l'estimateur du maximum de vraisemblance existe. Croisez enfin `Detecte` avec l'indicatrice `Eucs >= 12` pour voir d'où vient le problème.

d) Comparez les valeurs-p des tests de Wald de `summary()` à celles des tests du rapport de vraisemblance que donne `drop1(..., test = "Chisq")`. Expliquez l'écart. Une des variables retenues n'est pas significative au seuil de 5 % : pourquoi `step()` la garde-t-elle quand même ? (Indice : à quel seuil implicite correspond l'AIC pour un terme à un paramètre ?)

e) Reprenez la sélection *stepwise* avec le BIC (argument `k = log(n)` de `step()`). Quel modèle obtenez-vous ? Interprétez les rapports de cotes de ce modèle et leurs intervalles de confiance par vraisemblance profilée. Quelles réserves feriez-vous sur ces conclusions ?

Voir la solution

Exercice appliqué 2. Diagnostics et surdispersion. Le jeu de données `troutegg` de la librairie `faraway` (Faraway 2025) compte 20 observations d'une expérience où des boîtes d'œufs de truite ont été enfouies à cinq emplacements d'un cours d'eau (`location`), puis retirées après quatre durées (`period`, en semaines), une boîte par combinaison d'emplacement et de durée. La réponse est le nombre d'œufs survivants (`survive`) parmi les `total` œufs de la boîte. Les variables `location` et `period` sont des facteurs.

a) Calculez la proportion d'œufs survivants dans chaque boîte et tracez-la en fonction de `period`, avec une courbe par emplacement. Décrivez les tendances. Qu'est-ce que ce graphique laisse présager pour un modèle additif en `location` et `period` ?

b) Ajustez le modèle logistique binomial `cbind(survive, total - survive) ~ location + period`. Tracez les résidus de déviance en fonction du numéro d'observation et repérez ceux qui dépassent 3 en valeur absolue. Les conditions de l'approximation χ^2 des tests d'adéquation sont-elles réunies ? Testez l'adéquation du modèle par la déviance et par le khi-deux de Pearson.

c) Estimez le paramètre de dispersion par $\tilde{\phi}_P$ et ajustez le modèle quasibinomial. Comparez, coefficient par coefficient, les erreurs-types et les valeurs-p des deux modèles. Les

conclusions changent-elles ? Testez ensuite l'effet de chaque facteur pris en entier à l'aide du test F (`drop1(..., test = "F")`).

d) Reprenez les diagnostics du chapitre pour le modèle quasibinomial : graphique de $\hat{\pi}_i$ en fonction de $\tilde{\pi}_i$, leviers et distances de Cook. Quelles observations se démarquent ?

e) Que représente concrètement le paramètre de dispersion dans cette expérience ? Proposez une explication de la surdispersion. Peut-on vérifier si une interaction `location:period` suffit à l'expliquer ?

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Dunn et Smyth (2018). *Generalized Linear Models with Examples in R*
— Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R*
— McCullagh et Nelder (1989). *Generalized Linear Models*
— Faraway (2016). *Extending the Linear Model with R*
— Agresti (2013). *Categorical Data Analysis*
— James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R*

3 Modèles linéaires généralisés (GLM)

La régression linéaire et la régression logistique partagent plusieurs concepts : inférence sur les coefficients β, sélection de modèles, diagnostics. Les modèles linéaires généralisés (GLM) les réunissent dans un même cadre, où l'on suppose que la loi de la variable réponse appartient à la famille exponentielle. Cette famille comprend entre autres les lois normale, de Bernoulli, binomiale, de Poisson et gamma, de sorte que les régressions linéaire et logistique des chapitres précédents en sont des cas particuliers.

3.1 Famille exponentielle

Cette section définit d'abord la famille exponentielle, en tire l'espérance et la variance, puis écrit les lois usuelles sous cette forme.

3.1.1 Définition

Définition — Famille exponentielle

La loi de la variable aléatoire Y appartient à la famille exponentielle si sa fonction de probabilité (densité si Y est continue, fonction de masse si Y est discrète) s'écrit

$$f_Y(y) = a(y, \phi) \exp\left\{ \frac{y\theta - b(\theta)}{\phi} \right\},$$

où θ est le paramètre canonique et ϕ le paramètre de dispersion. La fonction $b(\cdot)$ est appelée fonction cumulante.

Cette écriture peut sembler abstraite au premier abord, mais elle a l'avantage de regrouper sous un même formalisme des lois d'apparence différente (continues ou discrètes, bornées ou non). Une fois une loi mise sous cette forme, on obtient directement son espérance et sa variance, sans repasser par le calcul habituel des moments.

Remarque — Forme canonique générale

La forme ci-dessus est celle des GLM : ϕ y apparaît explicitement comme diviseur. En statistique mathématique, on rencontre plus souvent une écriture encore plus générale,

sans référence à ϕ :

$$f_Y(y \mid \theta) = h(y)\, c(\theta)\, \exp\left\{ \sum_{k=1}^{K} w_k(\theta)\, t_k(y) \right\},$$

où $h(y)$ ne dépend pas de θ et $c(\theta)$ est une constante de normalisation qui ne dépend pas de y. Les $t_k(y)$ sont les statistiques suffisantes associées aux paramètres naturels $w_k(\theta)$. Dans le cas à un seul paramètre naturel ($K = 1$), avec $t_1(y) = y$, cette écriture se réduit à

$$f_Y(y \mid \theta) = h(y) \exp\left\{ w(\theta)\, y - A(\theta) \right\}, \qquad A(\theta) = -\log c(\theta).$$

La forme du chapitre s'obtient en posant $w(\theta) = \theta/\phi$ et $A(\theta) = b(\theta)/\phi$, et le paramètre de dispersion ϕ est alors absorbé dans le paramètre naturel et dans $h(y)$ au lieu d'apparaître séparément. Cette forme rappelle aussi que Y est une statistique suffisante pour θ dans un GLM, puisque $t(y) = y$.

Propriété — Moments d'une variable de la famille exponentielle

Si Y appartient à la famille exponentielle, alors

$$E[Y] = \mu = b'(\theta) \qquad \text{et} \qquad Var[Y] = \phi\, b''(\theta) = \phi\, V(\mu),$$

où $V(u) = b''\left[(b')^{-1}(u)\right]$ est appelée fonction de variance.

Pour aller plus loin avec la théorie — Calcul de tous les moments par la fonction génératrice des cumulants

La propriété précédente ne donne que l'espérance et la variance. On obtient tous les moments de Y en passant par sa fonction génératrice des moments (MGF).
Fonction génératrice des moments : pour Y de densité $f_Y(y) = a(y, \phi) \exp\{(y\theta - b(\theta))/\phi\}$,

$$M_Y(t) = E[e^{tY}] = \int a(y, \phi) \exp\left\{ \frac{y(\theta + t\phi) - b(\theta)}{\phi} \right\} dy.$$

Comme f_Y s'intègre à 1 pour tout paramètre canonique valide θ',

$$\int a(y, \phi) \exp\left\{ \frac{y\theta'}{\phi} \right\} dy = \exp\left\{ \frac{b(\theta')}{\phi} \right\}.$$

En posant $\theta' = \theta + t\phi$ dans l'intégrale de $M_Y(t)$, celle-ci se réduit à

$$M_Y(t) = \exp\left\{ \frac{b(\theta + t\phi) - b(\theta)}{\phi} \right\}.$$

Fonction génératrice des cumulants : le logarithme de $M_Y(t)$ est la fonction génératrice des cumulants (CGF) :

$$K(t) = \log M_Y(t) = \frac{b(\theta + t\phi) - b(\theta)}{\phi}.$$

Par définition, les cumulants κ_n de Y sont les dérivées de $K(t)$ évaluées en $t = 0$. On dérive donc $K(t)$ n fois par rapport à t, et comme la dérivation ne porte que sur $\theta + t\phi$, chaque dérivation fait sortir un facteur ϕ :

$$K^{(n)}(t) = \phi^{n-1}\, b^{(n)}(\theta + t\phi) \qquad \Longrightarrow \qquad \kappa_n = K^{(n)}(0) = \phi^{n-1}\, b^{(n)}(\theta).$$

Le cumulant d'ordre n de Y s'obtient donc en dérivant n fois la fonction cumulante $b(\theta)$, puis en multipliant par ϕ^{n-1}, ce qui explique le nom donné à $b(\theta)$. En particulier, on retrouve $\kappa_1 = b'(\theta) = \mu$ et $\kappa_2 = \phi\, b''(\theta) = \phi\, V(\mu)$, comme dans la propriété précédente.
Des cumulants aux moments : les cumulants d'ordre supérieur décrivent la forme de la loi, par les relations usuelles entre cumulants et moments, par exemple :

$$\kappa_3 = E[(Y - \mu)^3], \qquad \kappa_4 = E[(Y - \mu)^4] - 3\kappa_2^2.$$

Le coefficient d'asymétrie vaut $\kappa_3/\kappa_2^{3/2}$ et l'aplatissement en excès vaut κ_4/κ_2^2.

Le paramètre canonique θ est ainsi lié directement à l'espérance de Y (par $\mu = b'(\theta)$). Le paramètre de dispersion ϕ, lui, règle l'amplitude de la variance autour de la fonction de variance $V(\mu)$, qui dépend de la loi choisie. Pour certaines lois (Bernoulli, binomiale, Poisson), ϕ est fixé à 1 et la variance ne dépend que de la moyenne. C'est une contrainte à garder à l'œil lors des diagnostics (voir la surdispersion).

3.1.2 Mise en forme des lois usuelles

Pour identifier θ, ϕ, $a(y, \phi)$ et $b(\theta)$ pour une loi donnée, la démarche est toujours la même : partir de la fonction de probabilité, prendre son logarithme et réarranger les termes pour faire apparaître $\{y\theta - b(\theta)\}/\phi$. Tout ce qui reste (et ne dépend pas de θ) forme $\log a(y, \phi)$.

Exemple — Loi normale

Soit $Y \sim \mathcal{N}(\mu, \sigma^2)$, de densité

$$f_Y(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}.$$

En développant le carré au numérateur de l'exposant,

$$-\frac{(y-\mu)^2}{2\sigma^2} = \frac{y\mu - \mu^2/2}{\sigma^2} - \frac{y^2}{2\sigma^2},$$

on obtient

$$f_Y(y) = \underbrace{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{\frac{-y^2}{2\sigma^2}\right\}}_{a(y,\phi)} \exp\left\{\frac{y\mu - \mu^2/2}{\sigma^2}\right\}.$$

En comparant à la forme générale :

$$\theta = \mu, \qquad \phi = \sigma^2, \qquad b(\theta) = \frac{\theta^2}{2}.$$

On vérifie que

$$b'(\theta) = \theta = \mu, \qquad b''(\theta) = 1.$$

D'où

$$E[Y] = \mu, \qquad Var[Y] = \sigma^2 \times 1 = \sigma^2, \qquad V(\mu) = 1.$$

La variance ne dépend donc pas de la moyenne.

Exemple — Loi binomiale

Soit $Y \sim \text{Bin}(m, \pi)$, de fonction de masse

$$P(Y = y) = \binom{m}{y} \pi^y (1-\pi)^{m-y}, \qquad y = 0, 1, \dots, m.$$

Isolons d'abord le terme en y dans l'exponentielle :

$$\pi^y(1-\pi)^{m-y} = \exp\{y\log(\pi) + (m-y)\log(1-\pi)\} = \exp\left\{y\log\left(\frac{\pi}{1-\pi}\right) + m\log(1-\pi)\right\}.$$

La fonction de masse s'écrit donc

$$P(Y = y) = \binom{m}{y} \exp\left\{y\log\left(\frac{\pi}{1-\pi}\right) + m\log(1-\pi)\right\}, \qquad y = 0, 1, \dots, m.$$

On identifie

$$\theta = \log\left(\frac{\pi}{1-\pi}\right), \qquad \phi = 1, \qquad a(y,\phi) = \binom{m}{y}.$$

Le paramètre canonique θ est ici le logit de π, et c'est pourquoi le lien logit est dit canonique pour la loi binomiale.
Comme $\pi = e^\theta/(1+e^\theta)$, il vient

$$1 - \pi = \frac{1}{1+e^\theta} \qquad \Longrightarrow \qquad \log(1-\pi) = -\log(1+e^\theta).$$

D'où

$$b(\theta) = -m\log(1-\pi) = m\log(1+e^{\theta}).$$

Les dérivées de b donnent bien

$$b'(\theta) = m\pi = E[Y], \qquad b''(\theta) = m\pi(1-\pi) = Var[Y].$$

En substituant $\mu = m\pi$, on obtient

$$V(\mu) = \frac{\mu(m-\mu)}{m}.$$

Exemple — Loi de Poisson

Soit $Y \sim \mathcal{P}(\lambda)$, de fonction de masse

$$P(Y=y) = \frac{\lambda^y e^{-\lambda}}{y!} = \frac{1}{y!}\exp\{y\log(\lambda) - \lambda\}, \qquad y = 0, 1, 2, ...$$

On identifie directement

$$\theta = \log(\lambda), \qquad \phi = 1, \qquad a(y,\phi) = \frac{1}{y!}, \qquad b(\theta) = \lambda = e^{\theta}.$$

Les dérivées de b valent

$$b'(\theta) = e^{\theta} = \lambda, \qquad b''(\theta) = e^{\theta} = \lambda.$$

Il en découle la propriété connue

$$E[Y] = Var[Y] = \lambda, \qquad V(\mu) = \mu.$$

Exemple — Loi gamma

La loi gamma est habituellement paramétrée par sa forme α et son échelle s (ou son taux $1/s$). Afin d'obtenir une forme exponentielle dont le paramètre canonique est lié directement à la moyenne, on la reparamétrise avec

$$\mu = \alpha s \quad \text{(la moyenne)}, \qquad \phi = \frac{1}{\alpha}.$$

Après réarrangement, on obtient

$$\theta = -\frac{1}{\mu}, \qquad b(\theta) = -\log(-\theta).$$

On vérifie que

$$b'(\theta) = -\frac{1}{\theta} = \mu, \qquad b''(\theta) = \frac{1}{\theta^2} = \mu^2.$$

D'où

$$Var[Y] = \phi\mu^2, \qquad V(\mu) = \mu^2.$$

Contrairement aux trois exemples précédents, la variance croît ici avec le carré de la moyenne.

Le tableau suivant résume ces résultats :

loi	θ	ϕ	$b(\theta)$	$V(\mu)$
$\mathcal{N}(\mu, \sigma^2)$	μ	σ^2	$\theta^2/2$	1
$\text{Bin}(m, \pi)$	$\log\{\pi/(1-\pi)\}$	1	$m\log(1+e^\theta)$	$\mu(m-\mu)/m$
$\mathcal{P}(\lambda)$	$\log(\lambda)$	1	e^θ	μ
$\text{Gamma}(\mu, \phi)$	$-1/\mu$	ϕ	$-\log(-\theta)$	μ^2

3.2 Définition d'un modèle linéaire généralisé

Un GLM assemble trois choses : une loi de la famille exponentielle, un prédicteur linéaire et une fonction de lien entre les deux.

Définition — Modèle linéaire généralisé

Un modèle linéaire généralisé (GLM) est composé de trois éléments :

1. Composante aléatoire : Y_i suit une loi de la famille exponentielle de paramètres (θ_i, ϕ). Chaque observation a son propre paramètre canonique θ_i, mais le paramètre de dispersion ϕ est commun à toutes les observations. Ainsi, $E[Y_i] = \mu_i = b'(\theta_i)$ et $Var[Y_i] = \phi\, V(\mu_i)$.
2. Composante systématique : le prédicteur linéaire $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$, qui combine linéairement les variables explicatives.
3. Fonction de lien : une fonction g, monotone et dérivable, qui relie l'espérance μ_i au prédicteur linéaire par $g(\mu_i) = \eta_i$.

Définition — Lien canonique

La fonction de lien est canonique quand elle rend η_i égal au paramètre canonique θ_i, c'est-à-dire quand $g[b'(u)] = u$ pour tout u. C'est le choix qui apparaît naturellement quand la loi est écrite sous forme exponentielle. Par exemple, pour la loi binomiale,

> $\theta = \text{logit}(\pi)$, d'où le lien logit ; pour la loi de Poisson, $\theta = \log(\lambda)$, d'où le lien log.

Les paramètres du modèle sont donc $\beta = (\beta_0, \dots, \beta_p)^\top$ et ϕ. La régression linéaire (loi normale, lien identité), la régression logistique (loi binomiale, lien logit) et la régression de Poisson (loi de Poisson, lien log) en sont des cas particuliers. Rien n'empêche d'utiliser un lien non canonique, et il ne faut pas voir le lien canonique comme un lien privilégié pour l'interprétation : il simplifie seulement certains calculs, comme on le verra pour la régression de Poisson, où le poids IRLS devient $w_i = \widehat{\mu}_i$.

3.3 Estimation des paramètres

On estime β par maximum de vraisemblance, puis ϕ séparément quand la loi ne le fixe pas.

3.3.1 Vraisemblance et score

La log-vraisemblance des données $\{(y_i, X_i)\}_{i=1}^n$ est la somme des contributions individuelles données par la forme exponentielle :

$$\ell(\beta, \phi; y) = \sum_{i=1}^{n} \log[a(y_i, \phi)] + \frac{1}{\phi} \sum_{i=1}^{n} [y_i \theta_i - b(\theta_i)] = \ell_1(\phi) + \frac{1}{\phi}\, \ell_2(\beta).$$

Le paramètre θ_i dépend de β, mais indirectement : par $\mu_i = g^{-1}(\eta_i)$, puis par $\theta_i = (b')^{-1}(\mu_i)$. Le paramètre ϕ, lui, n'apparaît que comme facteur devant $\ell_2(\beta)$ et dans le terme $\ell_1(\phi)$, qui ne dépend pas de β. L'estimateur du maximum de vraisemblance $\widehat{\beta}$ maximise donc $\ell_2(\beta)$ seul, quelle que soit la valeur de ϕ, ce qui permet d'estimer β et ϕ séparément.

Comme en régression logistique, ce problème de maximisation n'a généralement pas de solution analytique, parce que la fonction de lien g rend la relation entre β et l'espérance μ_i non linéaire. On résout donc l'équation du score numériquement par l'algorithme IRLS (*Iteratively Reweighted Least Squares*), qui est le Fisher scoring appliqué à ce problème. À chaque itération, l'algorithme ajuste une régression linéaire pondérée dont les poids et la réponse de travail (*working response*) sont recalculés à partir des valeurs ajustées courantes, jusqu'à convergence.

3.3.2 Loi asymptotique de $\hat{\beta}$

Propriété — Loi asymptotique de l'estimateur du maximum de vraisemblance
L'estimateur $\hat{\beta}$ obtenu par maximisation de $\ell_2(\beta)$ a pour loi asymptotique

$$\hat{\beta} \;\dot{\sim}\; \mathcal{N}\left\{\beta,\ \phi\,(X^\top W X)^{-1}\right\},$$

où X est la matrice de design, $W = \mathrm{diag}\{w_1, \dots, w_n\}$ et

$$w_i = \frac{1}{V(\hat{\mu}_i)\left[g'(\hat{\mu}_i)\right]^2},$$

avec $\hat{\mu}_i = g^{-1}(\hat{\eta}_i)$ et $\hat{\eta}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$.

Cette forme englobe les résultats des chapitres précédents. En régression linéaire ($V(\mu) = 1$, $g'(\mu) = 1$), $w_i = 1$ pour tout i et on retrouve la variance usuelle des moindres carrés. En régression logistique ($m_i = 1$), la simplification donne $w_i = \hat{\mu}_i(1 - \hat{\mu}_i)$.

3.3.3 Déviance individuelle

Comme au chapitre sur la régression logistique, il s'agit de comparer l'ajustement μ d'un modèle à l'ajustement parfait y, mais cette fois pour n'importe quelle loi de la famille exponentielle. Notons

$$t(y, \mu) = y\theta - b(\theta),$$

de sorte que la partie de la log-vraisemblance individuelle qui dépend de θ (donc de μ, puisque $\mu = b'(\theta)$) est $t(y, \mu)/\phi$. Dans $t(y, y)$, θ est évalué en $(b')^{-1}(y)$. La déviance individuelle est deux fois la différence entre t évalué en l'ajustement parfait et t évalué en μ :

$$d(y, \mu) = 2\left\{t(y, y) - t(y, \mu)\right\}.$$

Comme la log-vraisemblance vaut t/ϕ plus un terme qui ne dépend pas de μ, la déviance standardisée $d(y, \mu)/\phi$ est deux fois l'écart de log-vraisemblance entre le modèle saturé et le modèle :

$$\frac{d(y, \mu)}{\phi} = 2\left\{\ell(y; y) - \ell(\mu; y)\right\}.$$

Dans le cas binomial du chapitre précédent, $\phi = 1$ et les deux coïncident.

Propriété — Non-négativité de la déviance individuelle

Dérivons $d(y,\mu)$ par rapport à θ (le paramètre canonique associé à μ) :

$$\frac{\partial d(y,\mu)}{\partial \theta} = -2\{y - b'(\theta)\} \qquad \text{et} \qquad \frac{\partial^2 d(y,\mu)}{\partial \theta^2} = 2\,b''(\theta) = 2\,V(\mu) > 0.$$

La dérivée première s'annule en $\mu = y$ et la dérivée seconde est toujours positive, de sorte que la déviance $d(y,\mu)$ atteint son minimum en $\mu = y$, où elle vaut 0. Par conséquent, $d(y,\mu) \geq 0$ pour tout μ, avec égalité si et seulement si $\mu = y$.

Pour la loi normale, $d(y,\mu) = (y-\mu)^2$: la déviance individuelle généralise donc le carré du résidu de la régression linéaire.

Propriété — Loi asymptotique de la déviance

On montre que, lorsque la dispersion est petite (ϕ petit, ou par exemple μ grand pour la loi de Poisson et m grand pour la loi binomiale), $d(Y,\mu)/\phi$ suit approximativement une loi χ^2_1, où μ est la vraie moyenne de Y. Il ne s'agit pas d'une approximation quand $n \to \infty$: elle porte sur chaque observation. Considérons maintenant des variables indépendantes $\{Y_1, \ldots, Y_n\}$ de paramètres (μ_i, ϕ), avec un paramètre de dispersion ϕ commun à toutes les observations. La déviance et la déviance standardisée, calculées aux vraies moyennes μ_i, sont

$$D(y,\mu) = \sum_{i=1}^{n} d(y_i,\mu_i) \qquad \text{et} \qquad D^*(y,\mu) = \frac{D(y,\mu)}{\phi} \;\dot{\sim}\; \chi^2_n.$$

Remarque — Le cas normal

Pour la loi normale, avec les vraies moyennes μ_i,

$$D^*(y,\mu) = \frac{D(y,\mu)}{\phi} = \frac{\sum_{i=1}^{n}(y_i-\mu_i)^2}{\sigma^2} \sim \chi^2_n,$$

et ce n'est pas une approximation : c'est la loi exacte de la déviance standardisée.
Notez que ce résultat porte sur la déviance calculée autour des vraies moyennes, qui sont inconnues en pratique. Si on les remplace par les valeurs ajustées $\hat{\mu}_i$ d'un modèle linéaire normal (lien identité) à $p+1$ coefficients, la loi reste exacte, mais elle perd un degré de liberté par coefficient estimé :

$$\frac{D(y,\hat{\mu})}{\sigma^2} = \frac{SSE}{\sigma^2} \sim \chi^2_{n-(p+1)}.$$

Le tableau suivant résume le statut de ces lois. Le mot « exacte » y est réservé au modèle normal avec lien identité. Pour les autres GLM, les lois du khi-deux de la déviance et de X^2_P

sont des approximations, valides quand la dispersion est petite (les μ_i assez grands pour la loi de Poisson, les m_i assez grands pour la loi binomiale). Un grand nombre d'observations n ne suffit pas : avec des données de Bernoulli, par exemple, l'approximation ne s'applique pas, peu importe la valeur de n. Les tests d'adéquation globale des GLM non normaux sont donc asymptotiques. La dernière ligne, qui compare deux modèles emboîtés, est différente : son approximation est justifiée quand $n \to \infty$, même avec des données de Bernoulli.

Situation	Statistique	Statut	Degrés de liberté
Une observation, vraie moyenne μ	$d(Y, \mu)/\phi$	exacte pour la loi normale ; sinon approximative (petite dispersion)	1
n observations, vraies moyennes μ_i	$D(y, \mu)/\phi$	exacte pour la loi normale ; sinon approximative (petite dispersion)	n
Modèle saturé ($\hat{\mu}_i = y_i$, n paramètres)	$D(y, y)/\phi$	nulle par construction : c'est la référence	0
Modèle ajusté avec $p+1$ coefficients	$D(y, \hat{\mu})/\phi$ ou X^2_P/ϕ	exacte pour la loi normale avec lien identité ; sinon approximative (petite dispersion)	$n-(p+1)$
Modèles emboîtés à $q+1$ et $p+1$ coefficients	$(D_{\text{réduit}} - D_{\text{complet}})/\phi$	exacte pour la loi normale avec lien identité ; sinon approximative ($n \to \infty$ ou petite dispersion)	$p-q$

Chaque coefficient estimé retire un degré de liberté. Le modèle saturé, qui a autant de paramètres que d'observations, n'en laisse aucun.

3.3.4 Estimation de ϕ

Si ϕ est fixé par la loi (Bernoulli, binomiale, Poisson), l'estimation s'arrête à $\widehat{\beta}$. Sinon (loi normale, gamma), il faut aussi estimer ϕ. Comme l'estimateur du maximum de vraisemblance de ϕ est biaisé en petits échantillons, on lui préfère des estimateurs par la méthode des moments au lieu de chercher directement le maximum en ϕ de $\ell(\widehat{\beta}, \phi; y) = \ell_1(\phi) + \ell_2(\widehat{\beta})/\phi$.

Ces estimateurs reposent sur deux statistiques qui généralisent la statistique du khi-deux de Pearson et la déviance de la régression logistique :

$$X_P^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)} \qquad \text{et} \qquad D(y, \hat{\mu}) = \sum_{i=1}^{n} d(y_i, \hat{\mu}_i).$$

Divisées par ϕ, ces deux statistiques suivent approximativement une loi $\chi^2_{n-(p+1)}$ (voir le tableau précédent : les $p+1$ coefficients estimés retirent $p+1$ degrés de liberté), d'où les estimateurs de Pearson et de déviance :

$$\tilde{\phi}_P = \frac{X_P^2}{n-(p+1)}, \qquad \tilde{\phi}_D = \frac{D(y, \hat{\mu})}{n-(p+1)}.$$

Remarque — Lequel choisir ?

L'estimateur de Pearson a un biais négligeable et reste raisonnablement robuste à une mauvaise spécification de la loi. C'est celui que la fonction `glm` de R utilise par défaut. L'estimateur de déviance a en général une variance plus faible, mais il devient instable quand des valeurs de y sont près des bornes de leur support, par exemple y près de 0 pour la loi gamma.

Pour aller plus loin avec la théorie — Estimateurs de ϕ par la méthode des moments

Il s'agit de construire un estimateur de ϕ à partir des deux statistiques

$$X_P^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}, \qquad D(y, \hat{\mu}) = \sum_{i=1}^{n} d(y_i, \hat{\mu}_i),$$

sachant que chacune, divisée par ϕ, suit approximativement une loi du khi-deux :

$$\frac{X_P^2}{\phi} \;\dot{\sim}\; \chi^2_{n-(p+1)}, \qquad \frac{D(y, \hat{\mu})}{\phi} \;\dot{\sim}\; \chi^2_{n-(p+1)}.$$

Espérance d'une loi du khi-deux : une variable de loi χ^2_d a pour espérance d. Appliqué aux deux quantités ci-dessus, ce résultat donne

$$E\left[\frac{X_P^2}{\phi}\right] \approx n-(p+1), \qquad E\left[\frac{D(y, \hat{\mu})}{\phi}\right] \approx n-(p+1).$$

Isoler ϕ : le paramètre ϕ est une constante, pas une variable aléatoire. Il sort donc de l'espérance dans chacune des deux expressions :

$$\frac{E[X_P^2]}{\phi} \approx n-(p+1), \qquad \frac{E[D(y, \hat{\mu})]}{\phi} \approx n-(p+1).$$

En résolvant pour ϕ,

$$\phi \approx \frac{E[X_P^2]}{n-(p+1)}, \qquad \phi \approx \frac{E[D(y,\hat{\mu})]}{n-(p+1)}.$$

Substitution empirique : la méthode des moments remplace une espérance théorique, inconnue, par sa valeur observée. En remplaçant $E[X_P^2]$ par X_P^2 et $E[D(y,\hat{\mu})]$ par $D(y,\hat{\mu})$, on obtient les deux estimateurs :

$$\tilde{\phi}_P = \frac{X_P^2}{n-(p+1)}, \qquad \tilde{\phi}_D = \frac{D(y,\hat{\mu})}{n-(p+1)}.$$

La division par $n-(p+1)$ plutôt que par n donne aussi un estimateur approximativement sans biais.

3.4 Inférence statistique

On reprend les outils d'inférence de la régression logistique, en distinguant le cas où ϕ est fixé de celui où il est estimé.

3.4.1 Intervalles de confiance et tests pour β_j

Cas où ϕ est fixé : comme en régression logistique, les intervalles de confiance et les tests sur β reposent sur la normalité asymptotique de $\hat{\beta}$. Un intervalle de confiance de niveau $(1-\alpha)$ pour β_j est

$$\left[\hat{\beta}_j - z_{\alpha/2}\sqrt{\phi\, v_j}\,;\ \hat{\beta}_j + z_{\alpha/2}\sqrt{\phi\, v_j}\right],$$

où $v_j = [(X^\top W X)^{-1}]_{j+1,j+1}$. Le quantile z_γ de la $\mathcal{N}(0,1)$ est défini par $P(Z > z_\gamma) = \gamma$. Le test de $H_0 : \beta_j = 0$ se fait comme au chapitre précédent, avec $z_{obs} = \hat{\beta}_j / \sqrt{\phi\, v_j}$.

Cas où ϕ est inconnu : on remplace ϕ par son estimateur $\tilde{\phi}_P$ et les quantiles de la $\mathcal{N}(0,1)$ par ceux de la loi de Student à $n-(p+1)$ degrés de liberté. La loi de Student tient ainsi compte de l'incertitude ajoutée par l'estimation de ϕ :

$$\left[\hat{\beta}_j - t_{\alpha/2,\,n-(p+1)}\sqrt{\tilde{\phi}_P\, v_j}\,;\ \hat{\beta}_j + t_{\alpha/2,\,n-(p+1)}\sqrt{\tilde{\phi}_P\, v_j}\right].$$

Pour aller plus loin avec la théorie — Les intervalles de confiance

Le cas ϕ inconnu regroupe en fait deux situations de nature différente.

Pour la gaussienne, la gamma ou l'inverse gaussienne, ϕ est un vrai paramètre de la loi, libre et estimé. L'intervalle de confiance basé sur la loi de Student à $n-(p+1)$ degrés de liberté est alors exact dans le cas gaussien avec lien identité (la régression linéaire). Avec un autre lien, et pour la gamma et l'inverse gaussienne, c'est une approximation, mais la loi de Student tient compte de l'incertitude due à l'estimation de ϕ.
Les modèles quasi-Poisson et quasi-binomial (cas de surdispersion) relèvent d'une autre situation. Les lois de Poisson et binomiale imposent $\phi = 1$, de sorte que ce ne sont pas de vraies familles exponentielles à dispersion libre. Quand on observe de la surdispersion, il suffit de relâcher cette contrainte en introduisant un ϕ libre, estimé par $\tilde{\phi}_P$, sans changer la fonction de variance. On obtient alors le même intervalle

$$\left[\hat{\beta}_j - t_{\alpha/2,\, n-(p+1)}\sqrt{\tilde{\phi}_P\, v_j}\,;\ \hat{\beta}_j + t_{\alpha/2,\, n-(p+1)}\sqrt{\tilde{\phi}_P\, v_j}\right],$$

mais sa justification repose sur une approximation asymptotique (la quasi-vraisemblance), et non sur la théorie exacte de la vraisemblance. En pratique, le résultat est identique, mais la garantie théorique est un peu plus faible que pour une vraie famille exponentielle à dispersion libre.

3.4.2 Intervalle de confiance pour une valeur prédite

Soit une observation future $X^* = (1, x_1^*, \ldots, x_p^*)^\top$, avec $\eta^* = {X^*}^\top\beta$ et $\mu^* = g^{-1}(\eta^*)$. Comme $\hat{\eta}^* = {X^*}^\top\hat{\beta}$ est une combinaison linéaire de $\hat{\beta}$, sa variance s'obtient directement :

$$\hat{\mathrm{V}}(\hat{\eta}^*) = {X^*}^\top\hat{\mathrm{V}}(\hat{\beta})\,X^*, \qquad \hat{\mathrm{V}}(\hat{\beta}) = \phi\,(X^\top W X)^{-1},$$

où ϕ est remplacé par $\tilde{\phi}_P$ s'il est inconnu. On en tire l'intervalle de confiance pour η^*,

$$\left[{X^*}^\top\hat{\beta} \pm z_{\alpha/2}\sqrt{\hat{\mathrm{V}}(\hat{\eta}^*)}\right],$$

(avec $t_{\alpha/2,\, n-(p+1)}$ au lieu de $z_{\alpha/2}$ si ϕ est estimé) et, en appliquant g^{-1} aux deux bornes, un intervalle de confiance pour μ^*. Comme en régression logistique, on construit l'intervalle sur l'échelle du prédicteur linéaire, puis on transforme ses bornes, de sorte que l'intervalle final respecte le domaine de μ^* (par exemple $[0,1]$ pour une proportion). La méthode du delta appliquée directement à $\hat{\mu}^*$ ne le garantit pas.

3.4.3 Comparaison de modèles emboîtés

Un modèle réduit se compare à un modèle complet (le réduit étant emboîté dans le complet) par différence de déviances, comme en régression logistique. Notons $D_{\text{réduit}}$ et D_{complet} leurs déviances.

Propriété — Cas ϕ fixé

$$G^2 = \frac{D_{\text{réduit}} - D_{\text{complet}}}{\phi}$$

suit approximativement une loi χ^2_d si le modèle réduit est adéquat, où d est la différence du nombre de paramètres entre les deux modèles. On rejette H_0 au seuil α si $G^2 > \chi^2_{d,\alpha}$.

Propriété — Cas ϕ inconnu

Comme on remplace ϕ par $\tilde{\phi}_P$ (calculé sur le modèle complet), la statistique se compare approximativement à une loi F plutôt qu'à une loi χ^2, ce qui tient compte de l'incertitude sur ϕ :

$$F_{obs} = \frac{(D_{\text{réduit}} - D_{\text{complet}})/d}{\tilde{\phi}_P} \;\dot{\sim}\; F_{d,\, n-(p+1)} \quad \text{sous } H_0,$$

où $p+1$ est le nombre de coefficients du modèle complet. La loi F n'est exacte que dans le modèle linéaire normal (loi normale, lien identité).

Le test du score de la Section 2.2.5 s'applique aussi à tous les GLM : `anova(modele_reduit, modele_complet, test = "Rao")`. Comme il n'utilise que le modèle réduit, il est utile quand le modèle complet est difficile à ajuster.

3.5 Sélection et validation de modèles

Les procédures pas à pas *backward*, *forward* et *stepwise* du chapitre précédent restent valides pour tout GLM, avec les critères

$$AIC = -2\,\ell(\hat{\beta}, \hat{\phi}; y) + 2k, \qquad BIC = -2\,\ell(\hat{\beta}, \hat{\phi}; y) + \log(n)\,k,$$

où $k = p+1$ si ϕ est fixé par la loi et $k = p+2$ si ϕ est estimé (il compte alors comme un paramètre). Ici, $\hat{\phi}$ est l'estimateur du maximum de vraisemblance de ϕ, et non $\tilde{\phi}_P$: la fonction `AIC` de R utilise SSE/n pour la loi normale (c'est bien le maximum de vraisemblance) et, pour la loi gamma, $D(y, \hat{\mu})/n$, qui n'en est qu'une approximation.

Les outils de validation des chapitres sur les régressions linéaire et logistique se généralisent aussi à l'ensemble des GLM : résidus de Pearson et de déviance, levier, DFBETAS, DFFITS et distance de Cook. On substitue la déviance individuelle d_i au carré du résidu e_i^2 (et la déviance D à la somme des carrés résiduels SSE) là où il le faut.

3.6 Exemple détaillé : la régression de Poisson

En régression de Poisson, la variable réponse $Y_i \in \mathbb{N}$ est un dénombrement (nombre de survivants dans un groupe, nombre de nouveaux cas d'une maladie, nombre de sinistres, etc.), et le modèle suppose que $Y_i \sim \mathcal{P}(\mu_i)$:

$$P(Y_i = y_i) = e^{-\mu_i} \frac{\mu_i^{y_i}}{y_i!}, \qquad E[Y_i] = Var[Y_i] = \mu_i.$$

Exemple — Rappel : le jeu de données `empress`

Les exemples de cette section reprennent les données du naufrage de l'*Empress of Ireland* (29 mai 1914), présentées au chapitre sur la régression logistique. On utilise ici la version agrégée `empress_agrege` : une ligne par strate classe × sexe × groupe d'âge (11 strates pour les 1057 passagers, parce qu'aucun garçon ne voyage en première classe), avec le nombre de passagers m_i (`effectif`), le nombre de survivants y_i (`survivants`) et la variable numérique `classe_num` (1, 2 ou 3). La version individuelle `empress_individus` a servi au chapitre sur la régression logistique et sert aussi au chapitre sur la validation croisée.

```
library(dplyr)
library(ggplot2)

niveaux.classe <- c("Première", "Deuxième", "Troisième")

empress.agr <- read.csv("Jeux de données/empress_agrege.csv") |>
  mutate(classe = factor(classe, levels = niveaux.classe))
empress.agr
```

```
      classe  sexe groupe_age effectif survivants classe_num
1   Première Homme     Adulte       49         24          1
2   Première Femme     Adulte       34         11          1
3   Première Femme     Enfant        4          1          1
4   Deuxième Homme     Adulte      114         33          2
5   Deuxième Homme     Enfant       11          0          2
```

```
6   Deuxième Femme     Adulte      107         13          2
7   Deuxième Femme     Enfant       21          2          2
8  Troisième Homme     Adulte      446        115          3
9  Troisième Homme     Enfant       54          1          3
10 Troisième Femme     Adulte      169         17          3
11 Troisième Femme     Enfant       48          0          3
```

Avec la fonction de lien canonique $g(\mu_i) = \log(\mu_i)$, on a

$$\mu_i = e^{\eta_i}, \qquad \eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip},$$

c'est-à-dire

$$\mu_i = e^{\beta_0} \times \left\{e^{\beta_1}\right\}^{x_{i1}} \times \cdots \times \left\{e^{\beta_p}\right\}^{x_{ip}}.$$

Remarque — Interprétation des coefficients

Si x_{ij} augmente d'une unité, les autres variables étant fixées, μ_i est multiplié par e^{β_j}. L'effet est donc multiplicatif sur l'échelle de la réponse, comme le rapport de cotes e^{β_j} l'est sur la cote en régression logistique. Si $\beta_j > 0$, alors $e^{\beta_j} > 1$ et μ_i croît avec x_{ij} ; si $\beta_j < 0$, μ_i décroît.

3.6.1 Variables explicatives *offset*

On montre d'abord quand un compte binomial se comporte comme un compte de Poisson, puis on en tire la variable offset.

Pour aller plus loin avec la théorie — Approximation de la loi binomiale par la loi de Poisson

Soit $Y_i \sim \text{Bin}(m_i, \pi_i)$. Si $m_i \to \infty$ et $\pi_i \to 0$ de sorte que $\mu_i = m_i\pi_i$ reste fixe (ou converge vers une constante), alors la loi de Y_i converge vers une loi de Poisson de moyenne μ_i :

$$P(Y_i = y) = \binom{m_i}{y} \pi_i^y (1-\pi_i)^{m_i - y} \;\longrightarrow\; e^{-\mu_i} \frac{\mu_i^y}{y!}.$$

Intuition : quand π_i est petit, l'événement $\{Y_i = y\}$ correspond à y « succès » improbables parmi un grand nombre d'essais, et le nombre d'essais m_i n'a alors presque plus d'importance en soi. Ce qui détermine la loi de Y_i, c'est le produit $\mu_i = m_i\pi_i$, c'est-à-dire le nombre de succès attendu. Deux strates avec des (m_i, π_i) différents mais le même μ_i produisent donc des distributions de Y_i presque identiques. Par exemple, $m_i = 10\,000$, $\pi_i = 0{,}001$

donne le même μ_i que $m_i = 1000,\ \pi_i = 0{,}01$.
Repère pratique : l'approximation est jugée satisfaisante quand m_i est grand (disons $m_i \geq 20$) et π_i petit (disons $\pi_i \leq 0{,}05$), ce qui garde le produit $\mu_i = m_i\pi_i$ modéré. C'est le cas de la strate des garçons de troisième classe d'`empress` : $m = 54$ et $\tilde{\pi} = 1/54 \approx 0{,}019$, d'où $\mu \approx 1$. En revanche, ce ne l'est pas pour les hommes adultes de première classe ($m = 49$, $\tilde{\pi} = 24/49 \approx 0{,}49$), une strate sur laquelle les diagnostics reviennent plus loin.
Conséquence pour la modélisation : dans ce régime, la loi de Y_i ne dépend pratiquement que du produit $\mu_i = m_i\pi_i$, et seule cette combinaison compte. Il est donc plus naturel, et plus stable numériquement, de modéliser μ_i comme un compte de Poisson plutôt que π_i comme une proportion binomiale. Il faut aussi tenir compte du fait que μ_i dépend mécaniquement de m_i : une strate plus grande produit plus de cas, même à risque individuel égal. On relie donc les covariables au taux μ_i/m_i, et non à μ_i directement, ce qui mène à la variable offset $\log(m_i)$, introduite plus bas.

Quand la probabilité individuelle de l'événement est petite (par exemple $\tilde{\pi} = 1/54 \approx 0{,}019$ pour les garçons de troisième classe), les nombres de cas se comportent plus comme un dénombrement rare que comme une proportion binomiale. La régression de Poisson devient alors une solution de rechange naturelle à la régression logistique, et on modélise l'espérance du taux Y_i/m_i :

$$\log\left(\frac{\mu_i}{m_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} \quad \Longleftrightarrow \quad \log(\mu_i) = \beta_0 + \log(m_i) + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}.$$

Définition — Variable offset

Le terme $\log(m_i)$ dans l'équation ci-dessus est une variable explicative dont le coefficient est fixé à 1 au lieu d'être estimé : c'est une variable offset. Elle sert quand l'espérance de Y_i est proportionnelle à une autre quantité connue (une population exposée, une durée d'exposition, un effort d'échantillonnage). On veut alors modéliser un taux plutôt qu'un compte brut.

Exemple — Ajustement sur `empress`

On modélise le nombre de survivants de chaque strate, avec l'effectif de la strate comme offset et des effets additifs de la classe, du sexe et du groupe d'âge.

```
modele.additif <- glm(survivants ~ offset(log(effectif)) + classe + sexe
 ↪  + groupe_age,
                      family = poisson, data = empress.agr)
summary(modele.additif)$coefficients
```

```
                  Estimate Std. Error   z value     Pr(>|z|)
```

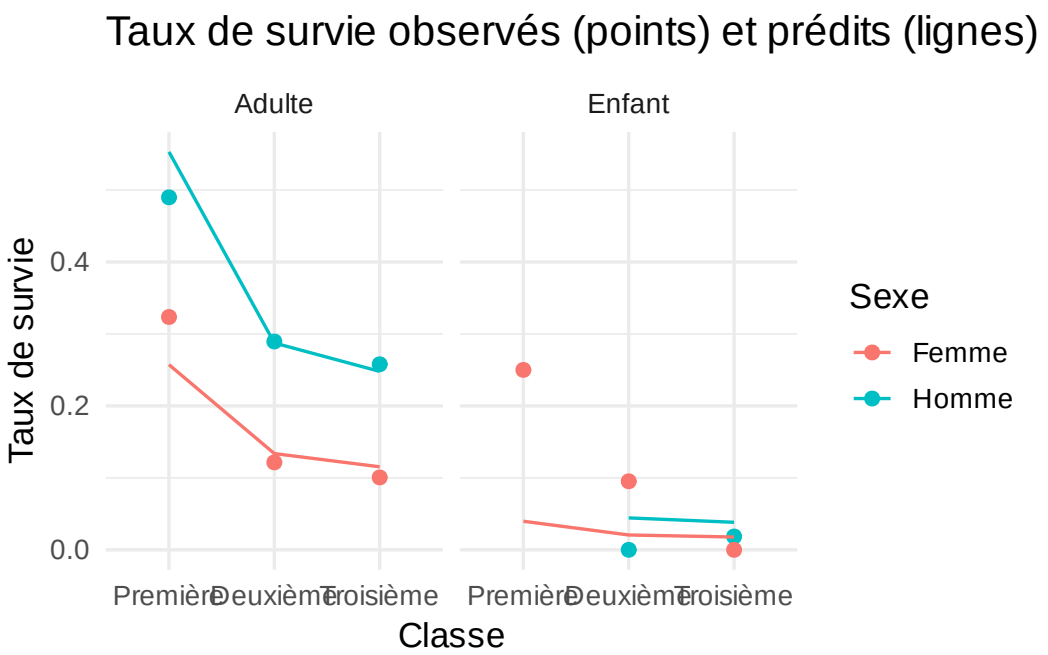
```
(Intercept)      -1.3576348  0.2108504 -6.438852 1.203801e-10
classeDeuxième   -0.6542217  0.2208236 -2.962644 3.050087e-03
classeTroisième  -0.8020421  0.1888224 -4.247599 2.160736e-05
sexeHomme         0.7651304  0.1714048  4.463880 8.048867e-06
groupe_ageEnfant -1.8689502  0.5061053 -3.692809 2.217905e-04
```

```
exp(coef(modele.additif))
```

```
     (Intercept)   classeDeuxième  classeTroisième        sexeHomme
       0.2572685        0.5198465        0.4484123        2.1492747
groupe_ageEnfant
       0.1542855
```

Les autres variables étant fixées, le taux de survie des hommes est environ 2,15 fois celui des femmes, et celui des enfants environ 0,15 fois celui des adultes. Les passagers de deuxième et de troisième classe ont des taux de survie environ 0,52 et 0,45 fois celui de la première classe. L'effet du sexe va à l'encontre du principe « les femmes et les enfants d'abord » : le naufrage, en pleine nuit et en quelques minutes, a laissé peu de temps pour organiser l'évacuation.
La figure suivante compare les taux de survie observés et prédits dans chaque strate :

```
empress.agr |>
  mutate(taux_predit = predict(modele.additif, type = "response") /
   ↪  effectif,
         taux_observe = survivants / effectif) |>
  ggplot(aes(x = classe, color = sexe)) +
  geom_point(aes(y = taux_observe), size = 2) +
  geom_line(aes(y = taux_predit, group = sexe)) +
  facet_wrap(~ groupe_age) +
  labs(x = "Classe", y = "Taux de survie", color = "Sexe",
       title = "Taux de survie observés (points) et prédits (lignes)") +
  theme_minimal(base_size = 13)
```



Les taux prédits suivent d'assez près les taux observés chez les adultes, alors que les écarts sont plus marqués chez les enfants, où les effectifs sont petits (4 filles seulement en première classe).

3.6.2 Vraisemblance et score

Avant de substituer le lien log, la log-vraisemblance s'écrit

$$\ell(\mu; y) = \sum_{i=1}^{n} \{y_i \log(\mu_i) - \mu_i\} - \sum_{i=1}^{n} \log(y_i!),$$

où le dernier terme ne dépend pas de β, de sorte qu'on peut l'ignorer pour l'estimation.

Le cas du lien log — Log-vraisemblance et score sous le lien log

Avec $\mu_i = e^{\eta_i}$, on a $\log(\mu_i) = \eta_i$, de sorte que

$$\ell(\beta; y) = \sum_{i=1}^{n} \{y_i \eta_i - e^{\eta_i}\} + K,$$

où $K = -\sum_i \log(y_i!)$ ne dépend pas de β.

Dérivée par rapport à β_j : notons $\dfrac{\partial \eta_i}{\partial \beta_j} = x_{ij}$ (avec $x_{i0} \equiv 1$). Le terme $y_i \eta_i$ se dérive directement en $y_i x_{ij}$, et pour le second terme, on utilise la règle de dérivation en chaîne :

$$\frac{\partial}{\partial \beta_j} e^{\eta_i} = e^{\eta_i} \cdot \frac{\partial \eta_i}{\partial \beta_j} = \mu_i x_{ij}.$$

D'où

$$\frac{\partial \ell(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} (y_i - \mu_i) x_{ij},$$

et, sous forme matricielle,

$$U(\beta) = X^\top (y - \mu).$$

Cette expression a la même forme que le score de la régression binomiale (Équation 2.2). Seule la relation entre μ_i et η_i change d'un lien à l'autre.

Comme en régression logistique, l'équation $U(\beta) = 0$ n'a pas de solution analytique. Elle se résout numériquement par IRLS (Fisher scoring), ce que fait `glm` avec `family = poisson`.

3.6.3 Information de Fisher et variance des estimateurs

La loi asymptotique de $\hat{\beta}$ est

$$\hat{\beta} \;\dot{\sim}\; \mathcal{N}\{\beta, (X^\top W X)^{-1}\},$$

où $W = \mathrm{diag}(w_i)$.

Le cas du lien log — Expression de w_i sous le lien log

Comme $Var[Y_i] = \mu_i$ et $d\mu_i/d\eta_i = \mu_i$ (lien canonique), le poids IRLS se simplifie en

$$w_i = \frac{(d\mu_i/d\eta_i)^2}{Var[Y_i]} = \frac{\mu_i^2}{\mu_i} = \mu_i,$$

évalué en $\hat{\mu}_i = e^{\hat{\eta}_i}$. Comme pour la binomiale, $\phi = 1$ ici : le modèle de Poisson usuel n'estime aucun paramètre de dispersion.

3.6.4 Déviance et adéquation

Pour la loi de Poisson, on obtient la déviance individuelle en appliquant la définition générale $d(y,\mu) = 2\{t(y,y) - t(y,\mu)\}$, qui coïncide ici avec $2\{\ell(y;y) - \ell(\mu;y)\}$ puisque $\phi = 1$.

Le cas du lien log — Dérivation de la déviance

En fonction de μ, la contribution d'une observation à la log-vraisemblance est $\ell_i(\mu) = y\log(\mu) - \mu$, sans la constante $-\log(y!)$, qui s'annule dans la différence. Le modèle saturé pose $\tilde{\mu} = y$, d'où

$$\ell_i(y) - \ell_i(\mu) = \{y\log(y) - y\} - \{y\log(\mu) - \mu\} = y\log\left(\frac{y}{\mu}\right) - (y-\mu),$$

avec la convention $y\log(y) \equiv 0$ si $y = 0$. D'où

$$d(y,\mu) = 2\left\{y\log\left(\frac{y}{\mu}\right) - (y-\mu)\right\}.$$

L'adéquation globale se vérifie avec

$$X^2_P = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i} \qquad \text{et} \qquad D = \sum_{i=1}^{n} d(y_i, \hat{\mu}_i),$$

qui suivent toutes deux approximativement une $\chi^2_{n-(p+1)}$ si le modèle convient et si les μ_i ne sont pas trop petits.

Exemple — Adéquation sur `empress`

```
mu_hat <- fitted(modele.additif)
n <- nrow(empress.agr)
p <- length(coef(modele.additif)) - 1

X2 <- sum((empress.agr$survivants - mu_hat)^2 / mu_hat)
D  <- deviance(modele.additif)

c(X2 = X2, D = D, dl = n - (p + 1))
```

```
       X2         D        dl
13.564715  9.884635  6.000000
```

```
1 - pchisq(X2, df = n - (p + 1))
```

```
[1] 0.03489506
```

```
1 - pchisq(D,  df = n - (p + 1))
```

```
[1] 0.1295951
```

```
round(mu_hat, 2)
```

```
    1     2     3     4     5     6     7     8     9    10    11
27.09  8.75  0.16 32.77  0.49 14.31  0.43 110.58  2.07 19.50  0.85
```

Remarque — Adéquation et taille des μ_i

Les deux statistiques ne mènent pas à la même conclusion au seuil de 5 % : la déviance ne rejette pas le modèle (valeur-p $\approx 0{,}13$), mais la statistique de Pearson le rejette (valeur-p $\approx 0{,}035$). L'écart vient de la taille des $\hat{\mu}_i$: quatre strates ont une valeur ajustée inférieure à 1, ce qui place l'approximation χ^2 hors de sa zone de validité. La statistique X_P^2 y est plus sensible, comme elle divise par $\hat{\mu}_i$. Par exemple, les filles de première classe ($y = 1$, $\hat{\mu} \approx 0{,}16$) et de deuxième classe ($y = 2$, $\hat{\mu} \approx 0{,}43$) fournissent à elles seules environ les trois quarts de X_P^2. Notez qu'avec seulement $n - (p + 1) = 6$ degrés de liberté, aucune des deux valeurs-p ne doit être prise au pied de la lettre.

3.6.5 Surdispersion

En pratique, il arrive souvent que $Var[Y_i] > E[Y_i]$, c'est-à-dire que la variance observée dans les données dépasse ce que prévoit la loi de Poisson (qui impose $\phi = 1$). C'est la surdispersion (*overdispersion*). Deux stratégies courantes permettent d'en tenir compte.

Remarque — Approche quasi-Poisson

Le modèle spécifie seulement les deux premiers moments de Y_i, sans supposer de loi précise : $E[Y_i] = \mu_i$ et $Var[Y_i] = \phi\,\mu_i$. On obtient les mêmes estimateurs ponctuels $\widehat{\beta}$ qu'avec la régression de Poisson usuelle. Seule $Var[\widehat{\beta}]$ change : elle est multipliée par $\tilde{\phi}_P = X_P^2/(n - (p + 1))$, ce qui élargit les intervalles de confiance et rend les tests plus conservateurs. En R, il suffit d'utiliser `family = quasipoisson`. Cette approche donne des intervalles de confiance et des tests valides sur β, mais elle ne repose sur aucune vraie

fonction de vraisemblance, de sorte que l'AIC et le BIC n'y sont pas définis.

Définition — Loi binomiale négative

La variable Y_i suit une loi binomiale négative de paramètres μ_i et α si

$$P(Y_i = y_i) = \frac{\Gamma(y_i + 1/\alpha)}{\Gamma(y_i+1)\Gamma(1/\alpha)} \left(\frac{1/\alpha}{\mu_i + 1/\alpha}\right)^{1/\alpha} \left(\frac{\mu_i}{\mu_i + 1/\alpha}\right)^{y_i}, \quad y_i = 0, 1, 2, \ldots, \ \alpha > 0.$$

On montre que $E[Y_i] = \mu_i$ et $Var[Y_i] = \mu_i + \alpha\mu_i^2$. La variance dépasse donc l'espérance dès que $\alpha > 0$, et on retrouve la loi de Poisson à la limite $\alpha \to 0$. Avec $\mu_i = e^{X_i^\top \beta}$, les coefficients β gardent la même interprétation multiplicative qu'en régression de Poisson. En R, la fonction `glm.nb` de la librairie `MASS` ajuste ce modèle en estimant β et α conjointement par maximum de vraisemblance. Remarquez qu'elle rapporte le paramètre `theta` $= 1/\alpha$ et non α (à ne pas confondre avec le paramètre canonique θ).

Remarque — Quasi-Poisson ou binomiale négative : la forme de la variance

Les deux approches ne supposent pas la même relation entre la variance et la moyenne :

$$\text{quasi-Poisson : } Var[Y_i] = \phi\,\mu_i, \qquad \text{binomiale négative : } Var[Y_i] = \mu_i + \alpha\mu_i^2.$$

La surdispersion est donc proportionnelle à la moyenne dans le premier cas et croît comme son carré dans le second. Pour de petites valeurs de μ_i, les deux modèles se ressemblent. Pour de grandes valeurs, ils pondèrent les observations différemment. Avec le lien log, $\partial\mu_i/\partial\eta_i = \mu_i$, et les poids IRLS valent

$$w_i = \frac{\mu_i}{\phi} \quad \text{(quasi-Poisson)}, \qquad w_i = \frac{\mu_i}{1+\alpha\mu_i} \quad \text{(binomiale négative)}.$$

Le poids quasi-Poisson croît sans limite avec μ_i, alors que le poids binomial négatif plafonne à $1/\alpha$. La binomiale négative accorde donc relativement moins de poids aux grands comptes, ce qui explique que ses estimations de β puissent s'écarter de celles de la régression de Poisson (que le modèle quasi-Poisson reproduit exactement). Pour choisir entre les deux, il est utile de tracer le carré des résidus bruts, $(y_i - \hat{\mu}_i)^2$, en fonction de $\hat{\mu}_i$, regroupés par classes de $\hat{\mu}_i$: une tendance linéaire appuie le modèle quasi-Poisson, une tendance quadratique la binomiale négative (Ver Hoef et Boveng 2007).

Propriété — Test de surdispersion

La régression de Poisson est le cas particulier $\alpha = 0$ de la binomiale négative, de sorte qu'on teste $H_0 : \alpha = 0$ par le rapport de vraisemblance des deux modèles emboîtés :

$Q_{obs} = 2\{\ell_{NB}(\hat{\beta}_{NB}, \hat{\alpha}) - \ell_P(\hat{\beta}_P)\}$. Dans cette statistique, ℓ_P est la log-vraisemblance du modèle de Poisson, évaluée en son estimateur $\hat{\beta}_P$, et ℓ_{NB} celle du modèle binomial négatif, évaluée en ses estimateurs $(\hat{\beta}_{NB}, \hat{\alpha})$; les deux estimateurs de β diffèrent en général. Il ne faut pas prendre la différence des déviances, parce que les deux modèles n'ont pas le même modèle saturé : la déviance de chacun est calculée par rapport à sa propre loi. Comme $\alpha < 0$ est impossible, la valeur $\alpha = 0$ est à la frontière de l'espace des paramètres, et la statistique de test ne suit donc pas une χ^2_1 sous H_0, mais un mélange

$$Q \sim \tfrac{1}{2}\chi^2_0 + \tfrac{1}{2}\chi^2_1 \quad \text{sous } H_0,$$

où χ^2_0 est la loi dégénérée en 0. Si $Q_{obs} > 0$, la valeur-p est donc $P(\chi^2_1 > Q_{obs})/2$, c'est-à-dire la moitié de ce que donnerait une χ^2_1 usuelle. Si $Q_{obs} = 0$, la valeur-p vaut 1, puisque $P(Q \geq 0) = 1$ sous H_0.

Exemple — Test de surdispersion sur `empress`

```
library(MASS)

modele.nb <- glm.nb(survivants ~ offset(log(effectif)) + classe + sexe +
 ↪  groupe_age,
                    data = empress.agr)
modele.nb$theta
```

```
[1] 25506.2
```

```
modele.nb$th.warn  # avertissement de convergence (masqué ci-dessus)
```

```
[1] "alternation limit reached"
```

```
# différence brute des log-vraisemblances maximisées (binomiale négative
 ↪  - Poisson)
diff.logLik <- as.numeric(logLik(modele.nb) - logLik(modele.additif))
diff.logLik
```

```
[1] -0.003318994
```

```r
# Q ne peut pas être négatif : on le ramène à 0 si glm.nb s'arrête un peu
 ↪  avant le maximum
Q.obs <- max(2 * diff.logLik, 0)
valeur.p <- if (Q.obs > 0) (1 - pchisq(Q.obs, df = 1)) / 2 else 1
c(Q.obs = Q.obs, valeur.p = valeur.p)
```

```
   Q.obs valeur.p
       0        1
```

L'estimation de θ est grande (environ 25 506), de sorte que $\hat{\alpha} = 1/\hat{\theta}$ est pratiquement nul : le maximum de vraisemblance est à la frontière $\alpha = 0$. Les avertissements de `glm.nb`, masqués ici parce qu'ils se répètent près de trente fois (« iteration limit reached », puis « alternation limit reached », conservé dans `modele.nb$th.warn`), viennent de là : l'algorithme, qui alterne entre l'estimation de β et celle de θ, fait croître $\hat{\theta}$ sans fin et s'arrête à sa limite d'itérations. Il s'arrête donc un peu avant le maximum, et sa log-vraisemblance est même très légèrement inférieure à celle du modèle de Poisson (différence de $-0{,}0033$). Le `max(…, 0)` force alors $Q_{obs} = 0$, et la valeur-p vaut 1. On ne détecte aucune surdispersion et le modèle de Poisson usuel convient.

3.6.6 Comparaison de modèles emboîtés

Comme en régression logistique, on compare deux modèles emboîtés par différence de déviances, plutôt que d'examiner l'adéquation d'un seul modèle.

Exemple — Effet de la classe sur `empress`

Testons d'abord l'effet de la classe, puis vérifions si cet effet se résume à une tendance log-linéaire en remplaçant le facteur `classe` par la variable numérique `classe_num`. Ce second modèle est emboîté dans le premier, puisqu'il n'a qu'un seul coefficient pour la classe au lieu de deux.

```r
modele.sans.classe <- update(modele.additif, . ~ . - classe)
anova(modele.sans.classe, modele.additif, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: survivants ~ sexe + groupe_age + offset(log(effectif))
Model 2: survivants ~ offset(log(effectif)) + classe + sexe + groupe_age
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1         8    25.2806
```

```
2         6     9.8846  2   15.396 0.0004537 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
modele.classe.num <- update(modele.additif, . ~ . - classe + classe_num)
anova(modele.classe.num, modele.additif, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: survivants ~ sexe + groupe_age + classe_num + offset(log(effectif))
Model 2: survivants ~ offset(log(effectif)) + classe + sexe + groupe_age
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         7    12.0837
2         6     9.8846  1   2.1991   0.1381
```

Le premier test rejette nettement l'absence d'effet de la classe (différence de déviances d'environ 15,4 à 2 degrés de liberté, valeur-p $\approx 0{,}0005$). Le second ne rejette pas le modèle à tendance log-linéaire (différence d'environ 2,2 à 1 degré de liberté, valeur-p $\approx 0{,}14$). Avec ce modèle, chaque classe inférieure multiplierait le taux de survie par un même facteur. Les coefficients du modèle additif, qui valent $-0{,}65$ pour la deuxième classe et $-0{,}80$ pour la troisième, suggèrent plutôt un écart concentré entre la première classe et les deux autres. Avec seulement 11 strates, les données ne permettent pas de trancher.

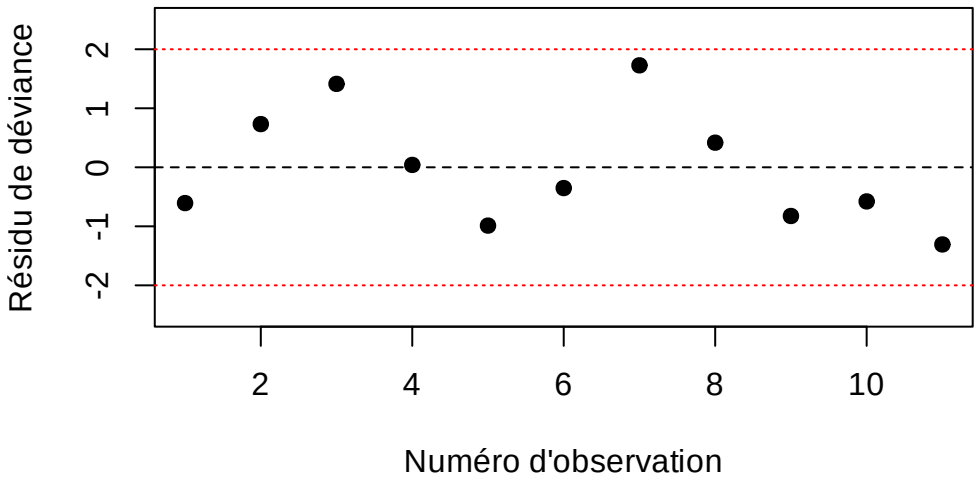
3.6.7 Graphiques et observations aberrantes

Comme en régression logistique, on trace les résidus de déviance en fonction du numéro d'observation pour repérer de la surdispersion ou des observations mal ajustées. Les leviers, la distance de Cook et les DFBETAS servent ensuite à repérer les observations influentes.

Exemple — Diagnostics sur `empress`

Résidus en fonction du numéro d'observation :

```r
res_dev <- residuals(modele.additif, type = "deviance")

plot(res_dev, xlab = "Numéro d'observation", ylab = "Résidu de déviance",
     pch = 19, ylim = c(-2.5, 2.5))
abline(h = 0, lty = 2)
abline(h = c(-2, 2), lty = 3, col = "red")
```



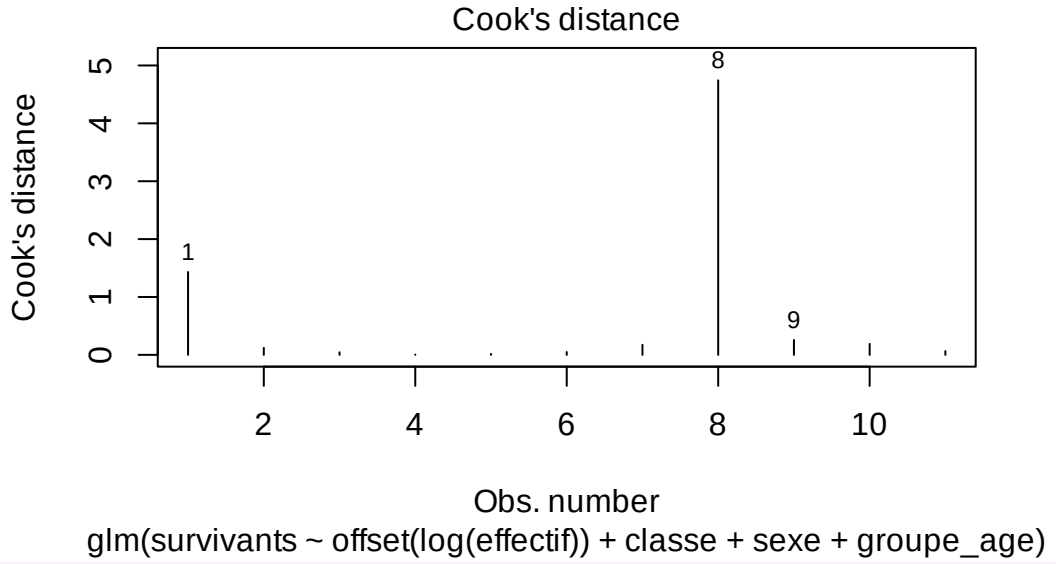
Les lignes pointillées rouges sont à ± 2, un seuil plus strict que le ± 3 souvent utilisé. Avec beaucoup d'observations, environ 5 % des résidus dépassent ± 2 par hasard, d'où le seuil ± 3 ; avec seulement $n = 11$ strates, on s'attend à au plus un résidu au-delà de ± 2 par hasard, et le seuil ± 2 suffit pour repérer les strates mal ajustées. Aucun résidu de déviance ne le dépasse. Les plus grands (environ 1,7 et 1,4) correspondent aux filles de deuxième et de première classe, déjà repérées dans l'analyse de l'adéquation.
Estimation de ϕ :

```
chi2_p <- sum(residuals(modele.additif, type = "pearson")^2)
phi_hat <- chi2_p / (n - (p + 1))
phi_hat
```

```
[1] 2.260786
```

On obtient $\tilde{\phi}_P \approx 2{,}3$, ce qui pourrait suggérer de la surdispersion, mais cette valeur est gonflée par les deux mêmes strates à petit $\hat{\mu}_i$ et repose sur 6 degrés de liberté seulement. Il faut donc la lire avec le test de surdispersion, qui ne détectait aucune surdispersion.
Observations aberrantes et valeurs influentes :

```
plot(modele.additif, which = 4)
```




```
h <- hatvalues(modele.additif)
round(h, 3)
```

```
    1     2     3     4     5     6     7     8     9    10    11
0.801 0.389 0.046 0.775 0.133 0.503 0.119 0.917 0.526 0.564 0.227
```

```
which(h > 2 * (p + 1) / n)
```

```
8
8
```

La strate des hommes adultes de troisième classe (observation 8) se détache nettement, avec une distance de Cook d'environ 4,7 et un levier d'environ 0,92. Elle regroupe 446 des 1057 passagers et 115 des 217 survivants, et comme le poids IRLS vaut $w_i = \hat{\mu}_i$, elle pèse lourd dans l'estimation. La strate des hommes adultes de première classe (observation 1, distance de Cook d'environ 1,4) vient ensuite. C'est aussi celle où le taux de survie ($\approx 0{,}49$) s'éloigne le plus du régime où l'approximation de Poisson se justifie.
Le seuil usuel pour un levier élevé est $2(p+1)/n$, qui vaut ici $10/11 \approx 0{,}91$. Avec 5 paramètres pour 11 observations, le levier moyen $(p+1)/n \approx 0{,}45$ est déjà élevé, et ce repère ne discrimine plus grand-chose, et il vaut mieux regarder les valeurs qui se détachent nettement des autres.

Remarque — Résidus simulés avec `DHARMa`

Pour un GLM non normal, les graphiques de résidus usuels sont difficiles à lire : les résidus sont discrets, leur dispersion dépend de $\hat{\mu}_i$, et des motifs en bandes apparaissent même quand le modèle est correct. La librairie `DHARMa` (Hartig, s. d.) contourne ce problème par simulation. Elle simule de nombreux jeux de réponses à partir du modèle ajusté, puis situe chaque observation dans la loi simulée qui lui correspond. Si le modèle est correct, ces résidus « quantiles » suivent une loi uniforme sur $[0, 1]$, quel que soit le GLM (ou le GLMM, Chapitre 10).

```
library(DHARMa)
res <- simulateResiduals(modele.additif)
plot(res)               # diagramme quantile-quantile uniforme et résidus
↪  contre prédictions
testDispersion(res)     # test de surdispersion par simulation
testZeroInflation(res)  # test d'excès de zéros
```

Ce code n'est pas exécuté ici : avec seulement 11 strates, les tests de `DHARMa` auraient peu de puissance. L'outil devient utile avec un grand nombre d'observations, en particulier pour vérifier la surdispersion et l'excès de zéros (Section 4.3).

Remarque — Poisson ou binomial ?

Plusieurs taux de survie d'`empress` sont modérés, de sorte que le modèle binomial `glm(cbind(survivants, effectif - survivants) ~ classe + sexe + groupe_age, family = binomial)` serait plus naturel pour ces données. Il mène aux mêmes conclusions qualitatives (effets de la classe, du sexe et du groupe d'âge de mêmes signes), mais ses coefficients s'interprètent comme des rapports de cotes et non comme des rapports de taux. J'ai choisi cet exemple pour illustrer l'offset et les outils propres à la régression de Poisson.

Remarque — À retenir

- Famille exponentielle : $E[Y] = \mu = b'(\theta)$ et $Var[Y] = \phi\, V(\mu)$, avec $V(\mu) = 1$, $\mu(m-\mu)/m$, μ et μ^2 pour les lois normale, binomiale, de Poisson et gamma. Pour les lois de Bernoulli, binomiale et de Poisson, $\phi = 1$, de sorte que la variance ne dépend que de la moyenne ;
- Dispersion fixée ou estimée : si ϕ est fixé, les intervalles reposent sur la loi normale et $G^2 = (D_{\text{réduit}} - D_{\text{complet}})/\phi$ se compare à une χ^2_d. Si ϕ est estimé par $\tilde{\phi}_P = X^2_P/(n-(p+1))$ (l'estimateur que `glm` utilise par défaut), la loi de Student et la loi F prennent le relais ;
- Régression de Poisson : avec le lien log, une augmentation d'une unité de x_{ij}

multiplie μ_i par e^{β_j}. Pour modéliser un taux, ajoutez $\log(m_i)$ comme variable offset (coefficient fixé à 1), par exemple `offset(log(effectif))` dans la formule de `glm` ;
— Quasi-Poisson : le modèle pose $Var[Y_i] = \phi\,\mu_i$ (variance linéaire en la moyenne), avec `family = quasipoisson`. Les estimations sont celles du modèle de Poisson, seule $Var[\widehat{\beta}]$ est multipliée par $\widetilde{\phi}_P$, et l'AIC n'est pas défini ;
— Binomiale négative : $Var[Y_i] = \mu_i + \alpha\mu_i^2$ (variance quadratique en la moyenne), ajustée par `glm.nb` de `MASS`, qui rapporte `theta` $= 1/\alpha$. Le test de $H_0 : \alpha = 0$ se fait par le rapport de vraisemblance (pas par la différence des déviances), et, si $Q_{obs} > 0$, sa valeur-p est la moitié de celle d'une χ^2_1, parce que $\alpha = 0$ est à la frontière.

3.7 Exercices

Les exercices sont classés par niveau de difficulté, et les exercices appliqués de la fin demandent d'analyser un jeu de données en R. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

3.7.1 Niveau facile

Exercice 1. Soit $Y \sim \text{Exp}(\lambda)$ (loi exponentielle de taux $\lambda > 0$), de densité $f(y) = \lambda e^{-\lambda y}$, $y > 0$. Mettez cette densité sous la forme de la famille exponentielle et identifiez θ, ϕ, $a(y, \phi)$ et $b(\theta)$. Retrouvez ensuite $E[Y] = 1/\lambda$ et $Var[Y] = 1/\lambda^2$ à l'aide de $b'(\theta)$ et $b''(\theta)$.

Voir la solution

Exercice 2. La loi binomiale a pour fonction $b(\theta) = m\log(1 + e^\theta)$, où $\theta = \text{logit}(\pi)$. Calculez $b'(\theta)$ et $b''(\theta)$. En substituant ensuite $\pi = \mu/m$ et $1 - \pi = (m - \mu)/m$, retrouvez la fonction de variance $V(\mu) = \mu(m - \mu)/m$.

Voir la solution

Exercice 3. Pour la loi de Poisson, $b(\theta) = e^\theta$. Un lien g est dit canonique s'il satisfait $g\{b'(u)\} = u$ pour tout u. Montrez que le lien canonique de la loi de Poisson est bien $g(\mu) = \log(\mu)$.

Voir la solution

3.7.2 Niveau moyen

Exercice 1. Admettez les deux identités générales de la théorie de la vraisemblance, appliquées ici au paramètre canonique θ : $E\left[\dfrac{\partial \log f}{\partial \theta}\right] = 0$ et $Var\left[\dfrac{\partial \log f}{\partial \theta}\right] = -E\left[\dfrac{\partial^2 \log f}{\partial \theta^2}\right]$. En partant

de $\log f(y) = \log a(y,\phi) + (y\theta - b(\theta))/\phi$, utilisez ces deux identités pour retrouver $E[Y] = b'(\theta)$ et $Var[Y] = \phi\, b''(\theta)$.

Voir la solution

Exercice 2. Partez de la log-vraisemblance individuelle $\ell_i(\theta_i) = \log a(y_i,\phi) + (y_i\theta_i - b(\theta_i))/\phi$, où θ_i dépend de β via $\mu_i = g^{-1}(\eta_i)$, puis $\theta_i = (b')^{-1}(\mu_i)$. Dérivez le score général

$$\frac{\partial \ell}{\partial \beta_j} = \frac{1}{\phi}\sum_{i=1}^{n} \frac{y_i - \mu_i}{V(\mu_i)\, g'(\mu_i)}\, x_{ij}$$

à l'aide de la règle de dérivation en chaîne $\dfrac{\partial \ell_i}{\partial \beta_j} = \dfrac{\partial \ell_i}{\partial \theta_i}\cdot\dfrac{\partial \theta_i}{\partial \mu_i}\cdot\dfrac{\partial \mu_i}{\partial \eta_i}\cdot\dfrac{\partial \eta_i}{\partial \beta_j}$. Montrez ensuite que, pour le lien canonique, cette expression se simplifie en $U(\beta) = X^\top(y-\mu)/\phi$.

Voir la solution

Exercice 3. Pour la loi gamma, le lien canonique est $g(\mu) = -1/\mu$ et la fonction de variance est $V(\mu) = \mu^2$. Vérifiez que $V(\mu)\, g'(\mu) = 1$, conformément à la propriété générale établie à l'exercice précédent pour tout lien canonique.

Voir la solution

3.7.3 Niveau difficile

Exercice 1. Le Chapitre 3 affirme, sans démonstration, que la loi asymptotique de $\widehat{\beta}$ a pour matrice de variance $\phi\,(X^\top W X)^{-1}$, où $w_i = 1/\{V(\hat{\mu}_i)[g'(\hat{\mu}_i)]^2\}$. En partant du score général

$$U_j(\beta) = \frac{1}{\phi}\sum_{i=1}^{n} \frac{y_i - \mu_i}{V(\mu_i)\, g'(\mu_i)}\, x_{ij}$$

dérivé à l'exercice 2 du niveau moyen, montrez que $I(\beta) = -E\left[\dfrac{\partial U}{\partial \beta^\top}\right] = \dfrac{1}{\phi} X^\top W X$. Utilisez le fait que $E[Y_i - \mu_i] = 0$ pour annuler les termes où la dérivée tombe sur $c_i = 1/\{V(\mu_i)g'(\mu_i)\}$ (information de Fisher *espérée*, plutôt qu'observée).

Voir la solution

Exercice 2. Démontrez formellement la convergence de la loi binomiale vers la loi de Poisson. Si $m \to \infty$ et $\pi \to 0$ de sorte que $\mu = m\pi$ reste fixe, montrez que, pour tout y fixé,

$$\binom{m}{y}\pi^y(1-\pi)^{m-y} \longrightarrow \frac{e^{-\mu}\mu^y}{y!}.$$

Voir la solution

Exercice 3. Le test de surdispersion de $H_0 : \alpha = 0$ (Poisson) contre $H_1 : \alpha > 0$ (binomiale négative) utilise la statistique du rapport de vraisemblance $Q = 2\{\ell_{NB}(\hat{\beta}_{NB}, \hat{\alpha}) - \ell_P(\hat{\beta}_P)\}$, où ℓ_{NB} et ℓ_P sont les log-vraisemblances maximisées des modèles binomial négatif et de Poisson (et non une différence de déviances, voir la Section 3.6.5). Sous H_0, sa loi asymptotique est le mélange $\frac{1}{2}\chi^2_0 + \frac{1}{2}\chi^2_1$ plutôt qu'une χ^2_1 usuelle. Esquissez un argument qui justifie ce résultat. Dans ce qui suit, $\ell(\alpha)$ désigne la log-vraisemblance profilée, maximisée en β à α fixé, de sorte que $\ell(0) = \ell_P(\hat{\beta}_P)$ et $\ell(\hat{\alpha}) = \ell_{NB}(\hat{\beta}_{NB}, \hat{\alpha})$. Sachez que :

— (i) l'estimateur du maximum de vraisemblance *non contraint* de α, $\hat{\alpha}_{MV}$, serait asymptotiquement $\mathcal{N}(0, v)$ sous H_0 (par la théorie usuelle de la vraisemblance) ;
— (ii) α est en réalité contraint à $[0, \infty)$, de sorte que l'estimateur contraint est $\hat{\alpha}^+ = \max(\hat{\alpha}_{MV}, 0)$;
— (iii) pour un paramètre scalaire non contraint, $2\{\ell(\hat{\alpha}_{MV}) - \ell(0)\} \dot{\sim} \chi^2_1$ sous H_0.

Voir la solution

Exercice 4. Considérez, avec le lien logit, une observation future dont le prédicteur linéaire estimé est $\hat{\eta}^* = 3$ avec $\widehat{\mathrm{V}}(\hat{\eta}^*) = 4$. Utilisez $z_{0,025} = 1{,}96$.

a) Construisez l'intervalle de confiance à 95% pour η^*, puis transformez ses bornes par $g^{-1}(\eta) = e^\eta/(1+e^\eta)$ pour obtenir un intervalle pour la probabilité $\pi^* = g^{-1}(\eta^*)$. Vérifiez qu'il est contenu dans $[0, 1]$.
b) À l'aide de la méthode du delta, montrez que $\widehat{\mathrm{V}}(\hat{\pi}^*) \approx [g^{-1\prime}(\hat{\eta}^*)]^2\, \widehat{\mathrm{V}}(\hat{\eta}^*)$. Calculez cette variance, puis construisez l'intervalle de Wald *directement* sur $\hat{\pi}^*$. Que remarquez-vous ?

Voir la solution

3.7.4 Exercices appliqués

Exercice appliqué 1. Trisomie 21 et âge de la mère. Le jeu de données `downs` de la librairie `GLMsData` (Dunn et Smyth 2018) donne le nombre de naissances vivantes (`Births`) et le nombre de bébés atteints de trisomie 21 (`DS`) en Colombie-Britannique, de 1961 à 1970, pour 30 groupes de mères classées selon l'âge. La variable `Age` est l'âge moyen des mères du groupe. Dans tout l'exercice, utilisez l'âge centré, `age.c = Age - 30`.

a) Calculez le taux observé de trisomie 21 pour 1000 naissances dans chaque groupe et décrivez sa variation avec l'âge. Pourquoi faut-il une variable offset pour modéliser le nombre de cas ? Chaque naissance est une épreuve de Bernoulli : pourquoi un modèle de Poisson est-il quand même raisonnable ici ?
b) Ajustez le modèle de Poisson `DS ~ age.c + offset(log(Births))`. L'approximation χ^2 de la déviance et du khi-deux de Pearson est-elle raisonnable ? Testez l'adéquation du modèle.

c) Ajoutez un terme quadratique en `age.c` et comparez les deux modèles par un test du rapport de vraisemblance. Tracez les taux observés pour 1000 naissances et les deux courbes de taux prédits, sur une échelle logarithmique. Décrivez ce que le modèle linéaire ne parvient pas à reproduire. Le modèle quadratique passe-t-il les tests d'adéquation ?

d) Estimez ϕ pour le modèle quadratique et faites le test de surdispersion de la Section 3.6.5 à l'aide du modèle binomial négatif. Que concluez-vous ? Avec le modèle quasi-Poisson, calculez le taux estimé pour 1000 naissances et son intervalle de confiance à 95 % pour des mères de 25, 35, 40 et 45 ans. Interprétez ces résultats.

e) Ajustez le modèle logistique binomial correspondant, `cbind(DS, Births - DS) ~ age.c + I(age.c^2)`, et comparez ses coefficients à ceux du modèle de Poisson quadratique. Expliquez la ressemblance. Lequel des deux modèles choisiriez-vous pour présenter les résultats ?

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Dunn et Smyth (2018). *Generalized Linear Models with Examples in R*
— Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R*
— McCullagh et Nelder (1989). *Generalized Linear Models*
— Faraway (2016). *Extending the Linear Model with R*
— Agresti (2013). *Categorical Data Analysis*
— James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R*
— Wood (2017). *Generalized Additive Models : An Introduction with R*
— Heritier, Cantoni, Copt et Victoria-Feser (2009). *Robust Methods in Biostatistics*

4 Extensions des GLM : réponses catégorielles et excès de zéros

Ce chapitre utilise les librairies R suivantes.

```
library(ggplot2)
library(MASS)
library(pscl)
library(ordinal)
library(dplyr)
```

Les GLM du Chapitre 3 couvrent les réponses binaires, binomiales et de comptage. Ce chapitre traite trois situations courantes qui sortent de ce cadre, tout en réutilisant ses outils (vraisemblance, score, déviance, tests du rapport de vraisemblance). D'abord, la réponse peut prendre plus de deux catégories sans ordre naturel, ce qui mène à la régression multinomiale. Ensuite, ces catégories peuvent être ordonnées, et le modèle à cotes proportionnelles en tient compte avec beaucoup moins de paramètres. Enfin, des données de comptage peuvent contenir plus de zéros que ne le prévoit la loi de Poisson, et les modèles à zéros excédentaires et à barrière séparent ces zéros du reste des comptes.

4.1 Réponses nominales : régression multinomiale

4.1.1 Modèle

Supposons que la réponse Y_i prend une valeur parmi K catégories $\{1, \dots, K\}$, sans ordre naturel : un mode de transport, une espèce, un parti politique. Notons $\pi_{ik} = P(Y_i = k \mid X_i)$, avec $\sum_{k=1}^{K} \pi_{ik} = 1$, et $y_{ik} = \mathbf{1}\{Y_i = k\}$ l'indicatrice de la catégorie observée.

Définition — Régression logistique multinomiale

On choisit une catégorie de référence, disons la première, et on pose, pour $k = 2, \dots, K$,

$$\log \frac{\pi_{ik}}{\pi_{i1}} = X_i^\top \beta_k,$$

où β_k est un vecteur de $p+1$ coefficients propre à la catégorie k. En posant $\beta_1 = 0$, on obtient

$$\pi_{ik} = \frac{\exp(X_i^\top \beta_k)}{\sum_{l=1}^{K} \exp(X_i^\top \beta_l)}, \qquad k = 1, \dots, K.$$

Le modèle compte $(K-1)(p+1)$ paramètres.

Avec $K = 2$, on retrouve exactement la régression logistique du Chapitre 2, avec la catégorie 2 comme « succès ». Le choix de la catégorie de référence ne change ni les probabilités ajustées ni la vraisemblance. Il change seulement les coefficients rapportés, puisque pour deux catégories quelconques k et l,

$$\log \frac{\pi_{ik}}{\pi_{il}} = X_i^\top (\beta_k - \beta_l).$$

Propriété — Interprétation des coefficients

Si x_j augmente d'une unité, les autres variables restant fixes, le rapport π_{ik}/π_{i1} est multiplié par $e^{\beta_{kj}}$. On parle d'un rapport de cotes relatif à la catégorie de référence. Il ne faut pas voir $\beta_{kj} > 0$ comme une hausse de π_{ik} : la probabilité de la catégorie k peut diminuer si celle d'une troisième catégorie augmente encore plus vite. Pour décrire l'effet d'une variable sur les probabilités elles-mêmes, on trace plutôt les $\hat{\pi}_{ik}$ en fonction de x_j.

4.1.2 Estimation

Les observations étant indépendantes, la log-vraisemblance est celle d'une loi multinomiale à un essai :

$$\ell(\beta_2, \dots, \beta_K) = \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log \pi_{ik}.$$

Le score par rapport à β_k a la même forme qu'en régression logistique,

$$\frac{\partial \ell}{\partial \beta_k} = \sum_{i=1}^{n} (y_{ik} - \pi_{ik}) X_i, \qquad k = 2, \dots, K,$$

et on le résout numériquement par Newton-Raphson ou par un algorithme de quasi-Newton. Les erreurs-types, les tests de Wald et les tests du rapport de vraisemblance s'obtiennent comme pour un GLM. En R, la fonction `multinom` de la librairie `nnet` ajuste ce modèle. Attention : `mgcv` contient aussi une fonction `multinom`, qui n'a rien à voir, de sorte qu'il vaut mieux écrire `nnet::multinom` lorsque les deux librairies sont chargées.

Remarque — Indépendance des alternatives non pertinentes

Le rapport π_{ik}/π_{il} ne dépend que de β_k et de β_l, et pas des autres catégories. Ajouter ou retirer une catégorie ne change donc pas les cotes relatives des autres. Cette propriété, appelée indépendance des alternatives non pertinentes, est parfois irréaliste. L'exemple classique est celui d'un choix entre l'auto et un autobus bleu : si on ajoute un autobus rouge identique au bleu, il devrait prendre ses clients surtout à l'autobus bleu, alors que le modèle multinomial le fait prendre aux deux modes en proportion de leurs probabilités. Quand cette hypothèse est douteuse, on utilise des modèles qui la relâchent (logit emboîté, probit multinomial), qui dépassent le cadre de ce livre.

Exemple — Strate du couvert sur `arbres`

Au Chapitre 7 et au Chapitre 9, la strate du couvert a été réduite à deux catégories (strate inférieure ou non). Le jeu de données en distingue en fait quatre, dont seulement quatre arbres dominants, qu'on regroupe avec les codominants. Il reste trois catégories : codominant (l'arbre reçoit la lumière directement), intermédiaire et opprimé (l'arbre pousse entièrement sous le couvert des autres). On prend la strate codominante, la plus fréquente, comme référence, et on modélise la strate en fonction du diamètre.

```
arbres <- read.csv("Jeux de données/arbres.csv") |>
  mutate(
    espece = factor(species_fr),
    strate = factor(if_else(canopy_stratum == "Dominant", "Codominant",
     ↪  canopy_stratum),
                    levels = c("Codominant", "Intermédiaire", "Opprimé"))
  )
count(arbres, strate)
```

```
         strate   n
1    Codominant 139
2 Intermédiaire  48
3       Opprimé  13
```

La fonction `multinom` prend comme référence le premier niveau du facteur réponse. C'est pourquoi l'argument `levels` place « Codominant » en premier : les deux lignes de `coef(mod.multi)` correspondent alors, dans l'ordre, aux comparaisons « intermédiaire contre codominant » et « opprimé contre codominant ». Sans cet argument, `factor` classerait les niveaux par ordre alphabétique, et la référence changerait avec les noms des catégories.

```
mod.multi <- nnet::multinom(strate ~ diameter_cm, data = arbres, trace =
 ↪  FALSE)
summary(mod.multi)
```

```
Call:
nnet::multinom(formula = strate ~ diameter_cm, data = arbres,
    trace = FALSE)

Coefficients:
              (Intercept) diameter_cm
Intermédiaire    2.123572  -0.2291611
Opprimé          5.902037  -0.6867160

Std. Errors:
              (Intercept) diameter_cm
Intermédiaire   0.7419523  0.05680835
Opprimé         2.4002965  0.22387031

Residual Deviance: 260.6292
AIC: 268.6292
```

```
# rapports de cotes pour 5 cm de diamètre de plus
exp(5 * coef(mod.multi)[, "diameter_cm"])
```

```
Intermédiaire       Opprimé
    0.3179677     0.0322712
```

Les deux pentes sont négatives : plus un arbre est gros, moins il a de chances d'être sous le couvert plutôt que codominant. Pour 5 cm de diamètre de plus, la cote « intermédiaire contre codominant » est multipliée par $e^{5\times(-0{,}229)} \approx 0{,}32$, et la cote « opprimé contre codominant » par environ 0,03. L'effet est donc plus marqué pour la strate opprimée, ce qui est cohérent avec l'ordre naturel des strates, même si le modèle multinomial l'ignore. Le test du rapport de vraisemblance de l'effet du diamètre compare ce modèle au modèle sans variable explicative. Sa statistique vaut 48,6, à comparer à une χ^2_2 (deux coefficients nuls sous H_0), d'où une valeur-p négligeable.

```
grille <- data.frame(diameter_cm = seq(9, 42, length.out = 200))
probas <- predict(mod.multi, newdata = grille, type = "probs")
courbes <- data.frame(diameter_cm = rep(grille$diameter_cm, 3),
                      strate = factor(rep(colnames(probas), each =
                       ↪  nrow(grille)),
                                      levels = levels(arbres$strate)),
                      pi = as.vector(probas))
ggplot(courbes, aes(diameter_cm, pi, colour = strate)) +
```

```
  geom_line(linewidth = 0.9) +
  labs(x = "Diamètre (cm)", y = "Probabilité estimée", colour = NULL) +
  theme_minimal(base_size = 13)
```

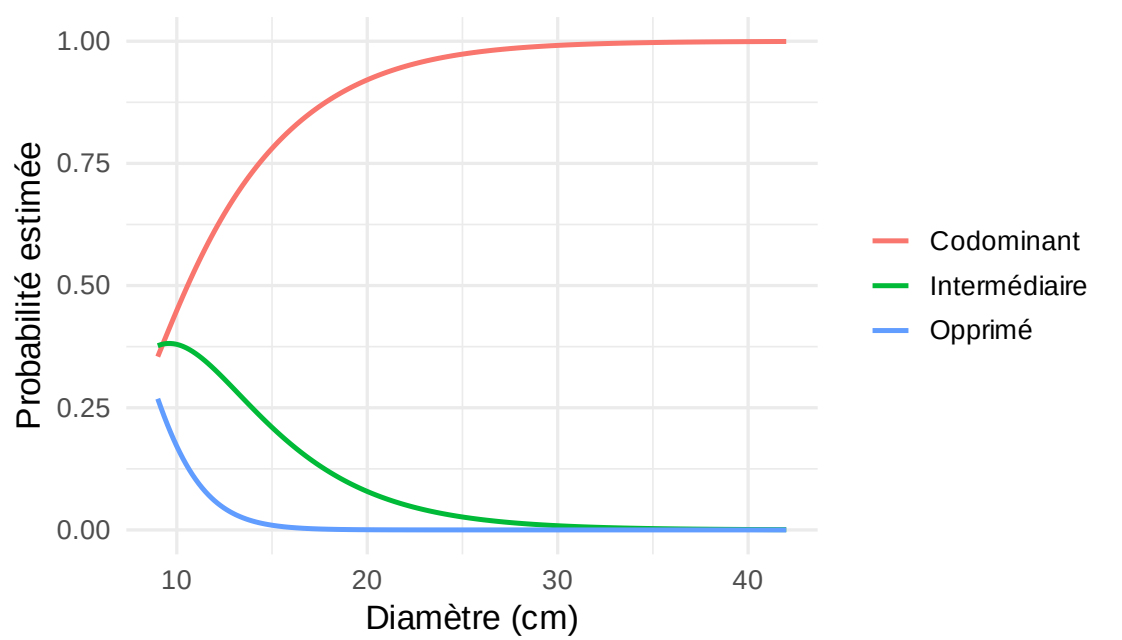


FIGURE 4.1 : Probabilités estimées de chaque strate en fonction du diamètre (régression multinomiale). Les trois courbes somment à 1 pour chaque diamètre.

La probabilité d'être codominant passe d'environ 0,35 pour les plus petits arbres à presque 1 au-delà de 25 cm. La probabilité d'être opprimé n'est appréciable que sous 12 cm environ. La librairie `marginaleffects` (voir le Chapitre 2) résume ces courbes en un effet moyen par catégorie, avec un intervalle de confiance :

```
marginaleffects::avg_comparisons(mod.multi, variables = list(diameter_cm
 ↪  = 5)) |>
  as.data.frame() |>
  dplyr::select(group, estimate, conf.low, conf.high)
```

```
          group    estimate   conf.low   conf.high
1    Codominant  0.18750682  0.1349418  0.24007186
2 Intermédiaire -0.12604787 -0.1773965 -0.07469923
3       Opprimé -0.06145895 -0.0927305 -0.03018741
```

Pour 5 cm de diamètre de plus, la probabilité d'être codominant augmente en moyenne d'environ 19 points de pourcentage. Les trois variations somment à 0, puisque les probabilités somment à 1.

Remarque — Pourquoi ne pas ajouter l'espèce ?
Aucun bouleau à papier de l'échantillon n'est opprimé. Dans un modèle multinomial avec l'espèce, le logit « opprimé contre codominant » du bouleau tend alors vers $-\infty$: c'est la séparation du Chapitre 2, appliquée à une seule des $K-1$ équations. Avec 13 arbres opprimés seulement, les $2 \times 3 = 6$ coefficients d'espèce supplémentaires sont de toute façon mal estimés. Le modèle ordinal de la section suivante règle ce problème en mettant les catégories en commun.

4.2 Réponses ordinales : modèle à cotes proportionnelles

Lorsque les catégories sont ordonnées (opprimé $<$ intermédiaire $<$ codominant, faible $<$ moyen $<$ élevé, stades d'une maladie), le modèle multinomial gaspille de l'information : il traite les catégories comme interchangeables et estime $(K-1)(p+1)$ paramètres. Le modèle à cotes proportionnelles exploite l'ordre.

4.2.1 Modèle

Définition — Modèle à cotes proportionnelles
Pour une réponse ordinale $Y_i \in \{1, \dots, K\}$, on modélise les probabilités cumulées :

$$\operatorname{logit} P(Y_i \le k \mid x_i) = \alpha_k - x_i^\top \beta, \qquad k = 1, \dots, K-1,$$

avec des seuils $\alpha_1 < \alpha_2 < \cdots < \alpha_{K-1}$ et un vecteur β commun à tous les seuils. Le vecteur x_i ne contient pas de 1, puisque les seuils jouent le rôle des ordonnées à l'origine. Le modèle compte $K-1+p$ paramètres. Les probabilités de chaque catégorie s'obtiennent par différence :

$$\pi_{ik} = P(Y_i \le k \mid x_i) - P(Y_i \le k-1 \mid x_i),$$

avec $P(Y_i \le 0) = 0$ et $P(Y_i \le K) = 1$.

Le signe de $x_i^\top \beta$ suit la convention de la fonction `polr` de `MASS` : un coefficient positif fait monter la réponse vers les catégories supérieures. Certains logiciels écrivent $\alpha_k + x_i^\top \beta$, ce qui change le signe des coefficients rapportés, et il faut vérifier la convention avant d'interpréter.

Propriété — Cotes proportionnelles

Pour tout seuil k, la cote de $\{Y_i > k\}$ contre $\{Y_i \leq k\}$ vaut $\exp(x_i^\top \beta - \alpha_k)$. Si x_j augmente d'une unité, cette cote est multipliée par e^{β_j}, quel que soit le seuil. Les courbes des logits cumulés en fonction de x_j sont donc parallèles, et c'est ce qui donne son nom au modèle.

Propriété — Interprétation par une variable latente

Supposons qu'une variable continue non observée $Y_i^* = x_i^\top \beta + \epsilon_i$, où ϵ_i suit la loi logistique standard, détermine la catégorie observée : $Y_i = k$ si $\alpha_{k-1} < Y_i^* \leq \alpha_k$, avec $\alpha_0 = -\infty$ et $\alpha_K = +\infty$. Alors

$$P(Y_i \leq k) = P(Y_i^* \leq \alpha_k) = P(\epsilon_i \leq \alpha_k - x_i^\top \beta) = \frac{e^{\alpha_k - x_i^\top \beta}}{1 + e^{\alpha_k - x_i^\top \beta}},$$

ce qui est exactement le modèle à cotes proportionnelles. Avec une erreur normale plutôt que logistique, on obtient le modèle probit ordinal (`polr(..., method = "probit")`).

Cette interprétation est souvent la plus naturelle. La position dans le couvert, par exemple, résulte d'une compétition continue pour la lumière, que l'inventaire forestier découpe en trois classes.

4.2.2 Estimation et vérification de l'hypothèse

Les paramètres $(\alpha_1, \ldots, \alpha_{K-1}, \beta)$ sont estimés par maximum de vraisemblance, avec la log-vraisemblance $\sum_i \sum_k y_{ik} \log \pi_{ik}$ comme pour le modèle multinomial. En R, on utilise `polr` de `MASS` (avec `Hess = TRUE` pour obtenir les erreurs-types) ou `clm` de la librairie `ordinal`.

L'hypothèse de cotes proportionnelles se vérifie en la relâchant pour une variable à la fois : le modèle à cotes non proportionnelles remplace β_j par β_{jk}, propre à chaque seuil. Les deux modèles sont emboîtés et se comparent par un test du rapport de vraisemblance à $K-2$ degrés de liberté (par variable). La fonction `clm` le permet avec l'argument `nominal`. On peut aussi comparer le modèle à cotes proportionnelles au modèle multinomial par l'AIC, mais pas par un test du rapport de vraisemblance, parce que ces deux modèles ne sont pas emboîtés.

Exemple — Strate du couvert, modèle ordinal

La fonction `polr` exige un facteur ordonné, de la catégorie inférieure à la catégorie supérieure. L'ordre des niveaux fixe le sens des coefficients : avec « Opprimé » en premier et « Codominant » en dernier, un coefficient positif fait monter l'arbre vers la strate codominante. Inverser l'ordre changerait le signe de tous les coefficients.

```
arbres <- arbres |>
```

```
  mutate(strate.ord = factor(strate, levels = c("Opprimé",
   ↪  "Intermédiaire", "Codominant"),
                             ordered = TRUE))
mod.ord1 <- polr(strate.ord ~ diameter_cm, data = arbres, Hess = TRUE)
summary(mod.ord1)
```

```
Call:
polr(formula = strate.ord ~ diameter_cm, data = arbres, Hess = TRUE)

Coefficients:
             Value Std. Error t value
diameter_cm 0.2832     0.0588   4.817

Intercepts:
                          Value  Std. Error t value
Opprimé|Intermédiaire     0.9604 0.7491     1.2821
Intermédiaire|Codominant 3.0270 0.7482     4.0459

Residual Deviance: 264.2776
AIC: 270.2776
```

```
c(multinomial = AIC(mod.multi), ordinal = AIC(mod.ord1))
```

```
multinomial     ordinal
   268.6292    270.2776
```

Le coefficient du diamètre est positif : un arbre plus gros tend vers les strates supérieures. Pour 5 cm de plus, la cote d'être dans une strate supérieure à k plutôt que dans une strate inférieure ou égale à k est multipliée par $e^{5\times 0{,}283} \approx 4{,}1$, pour $k =$ opprimé comme pour $k =$ intermédiaire. Le modèle n'utilise que 3 paramètres, contre 4 pour le modèle multinomial, et les deux ont un AIC semblable. Le test de l'hypothèse de cotes proportionnelles (`clm` avec `nominal = ~ diam.c`, où `diam.c` est le diamètre centré, ce qui évite des seuils qui se croisent hors du domaine des données) donne une valeur-p de 0,09, de sorte que les données ne la contredisent pas nettement.
Le modèle ordinal permet maintenant d'ajouter l'espèce sans difficulté, parce que ses coefficients sont communs aux deux seuils :

```
mod.ord2 <- polr(strate.ord ~ diameter_cm + espece, data = arbres, Hess =
 ↪  TRUE)
round(summary(mod.ord2)$coefficients, 3)
```

```
                          Value Std. Error t value
diameter_cm               0.293      0.061   4.784
especeÉpinette noire     -0.757      0.488  -1.551
especeÉrable rouge       -0.195      0.516  -0.378
especeSapin baumier      -0.951      0.472  -2.013
Opprimé|Intermédiaire     0.512      0.831   0.617
Intermédiaire|Codominant  2.640      0.830   3.182
```

```
exp(coef(mod.ord2))
```

```
         diameter_cm especeÉpinette noire   especeÉrable rouge
           1.3408536            0.4692671           0.8227925
 especeSapin baumier
           0.3862823
```

Le facteur `espece` a pour référence son premier niveau, le bouleau à papier (premier dans l'ordre alphabétique), de sorte que chaque coefficient d'espèce compare cette espèce au bouleau. À diamètre égal, l'épinette noire et le sapin baumier tendent vers les strates inférieures par rapport au bouleau à papier (rapports de cotes inférieurs à 1). On retrouve, avec une réponse à trois catégories, l'ordre des espèces du GAM logistique du Chapitre 9. Le fait qu'aucun bouleau ne soit opprimé ne pose plus de problème : l'information sur le bouleau vient surtout du seuil entre intermédiaire et codominant.

4.3 Comptes avec excès de zéros

Dans plusieurs applications de comptage, une partie des observations ne peut pas produire d'événement. Une personne qui ne fume pas déclare zéro cigarette quoi qu'il arrive, et un site où l'espèce est absente donne un compte nul peu importe l'effort d'échantillonnage. Ces zéros s'ajoutent à ceux que produit la loi de Poisson par hasard, de sorte que les données contiennent plus de zéros que prévu.

4.3.1 Modèles à zéros excédentaires

Définition — Modèle de Poisson à zéros excédentaires (ZIP)

Avec probabilité π_i, l'observation i est un zéro structurel. Avec probabilité $1-\pi_i$, elle suit une loi de Poisson d'espérance μ_i. Ainsi,

$$P(Y_i = 0) = \pi_i + (1-\pi_i)e^{-\mu_i}, \qquad P(Y_i = y) = (1-\pi_i)\frac{e^{-\mu_i}\mu_i^y}{y!}, \quad y = 1, 2, ...$$

Les deux composantes ont chacune leur prédicteur linéaire :

$$\text{logit}(\pi_i) = z_i^\top \gamma, \qquad \log \mu_i = x_i^\top \beta.$$

Le modèle binomial négatif à zéros excédentaires (ZINB) remplace la loi de Poisson par la loi binomiale négative du Chapitre 3.

Propriété — Moments du modèle ZIP

$$E[Y_i] = (1-\pi_i)\mu_i, \qquad Var[Y_i] = (1-\pi_i)\mu_i(1+\pi_i\mu_i).$$

Dès que $\pi_i > 0$, la variance dépasse l'espérance : les zéros excédentaires créent à eux seuls de la surdispersion.

Cette propriété a une conséquence pratique importante. Surdispersion et excès de zéros vont souvent ensemble, et un modèle binomial négatif ordinaire, qui prévoit lui aussi plus de zéros que la loi de Poisson, suffit parfois à reproduire les zéros observés. Il ne faut donc pas conclure à des zéros structurels sur la seule base du nombre de zéros comparé à la loi de Poisson.

4.3.2 Modèles à barrière

Définition — Modèle à barrière (hurdle)

Un premier modèle binaire décide si Y_i franchit la barrière du zéro, et un second modèle, tronqué en zéro, décrit les comptes positifs :

$$P(Y_i = 0) = 1 - p_i, \qquad P(Y_i = y) = p_i \, \frac{f(y;\mu_i)}{1 - f(0;\mu_i)}, \quad y = 1, 2, \dots,$$

où $\text{logit}(p_i) = z_i^\top \gamma$ et $f(\cdot;\mu_i)$ est une loi de Poisson ou binomiale négative d'espérance μ_i avant troncature, avec $\log \mu_i = x_i^\top \beta$.

Les deux modèles diffèrent par la nature des zéros. Dans un modèle à zéros excédentaires, un zéro peut être structurel ou provenir de la loi de comptage. Dans un modèle à barrière, tous les zéros viennent de la partie binaire, et les comptes positifs sont décrits séparément. Comme la vraisemblance du modèle à barrière se factorise en une partie binaire et une partie tronquée, les

deux parties s'estiment séparément. En particulier, si la partie binaire contient une ordonnée à l'origine, le modèle reproduit exactement le nombre de zéros observés (voir les exercices).

En R, les fonctions `zeroinfl` et `hurdle` de la librairie `pscl` ajustent ces modèles. La formule `y ~ x1 + x2 | z1` utilise `x1` et `x2` dans la partie de comptage et `z1` dans la partie binaire ; `y ~ .` utilise toutes les variables dans les deux parties. L'argument `dist` choisit la loi de comptage (`"poisson"` ou `"negbin"`).

Exemple — Publications des doctorants en biochimie

Le jeu de données `bioChemists` de la librairie `pscl` décrit 915 doctorants en biochimie aux États-Unis. La réponse `art` est le nombre d'articles publiés pendant les trois dernières années du doctorat. Les variables explicatives sont le sexe (`fem`), l'état civil (`mar`), le nombre d'enfants de moins de 6 ans (`kid5`), le prestige du programme (`phd`) et le nombre d'articles du directeur de recherche pendant la même période (`ment`).

```
data(bioChemists, package = "pscl")
table(bioChemists$art)
```

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  16  19
275 246 178  84  67  27  17  12   1   2   1   1   2   1   1
```

```
c(moyenne = mean(bioChemists$art), variance = var(bioChemists$art))
```

```
 moyenne variance
1.692896 3.709742
```

Près du tiers des doctorants n'ont rien publié, et la variance est plus de deux fois plus grande que la moyenne. On compare cinq modèles qui utilisent toutes les variables explicatives, en calculant pour chacun le nombre de zéros attendu, $\sum_i \hat{P}(Y_i = 0)$.

```
m.pois  <- glm(art ~ ., data = bioChemists, family = poisson)
m.nb    <- glm.nb(art ~ ., data = bioChemists)
m.zip   <- zeroinfl(art ~ . | ., data = bioChemists, dist = "poisson")
m.zinb  <- zeroinfl(art ~ . | ., data = bioChemists, dist = "negbin")
m.hurdle <- hurdle(art ~ . | ., data = bioChemists, dist = "negbin")

zeros <- c(
  Poisson = sum(dpois(0, fitted(m.pois))),
```

```
  NB      = sum(dnbinom(0, size = m.nb$theta, mu = fitted(m.nb))),
  ZIP     = sum(predict(m.zip, type = "prob")[, 1]),
  ZINB    = sum(predict(m.zinb, type = "prob")[, 1]),
  Barriere.NB = sum(predict(m.hurdle, type = "prob")[, 1])
)
aic <- c(AIC(m.pois), AIC(m.nb), AIC(m.zip), AIC(m.zinb), AIC(m.hurdle))
round(rbind(zeros.attendus = zeros, AIC = aic), 1)
```

```
               Poisson     NB    ZIP   ZINB Barriere.NB
zeros.attendus   191.4  277.8  273.2  285.4       275.0
AIC             3314.1 3135.9 3233.5 3126.0      3131.2
```

Le modèle de Poisson prévoit environ 191 zéros, contre 275 observés. Le modèle ZIP corrige le nombre de zéros, mais son AIC reste très élevé, parce que la loi de Poisson sous-estime aussi la dispersion des comptes positifs. Le modèle binomial négatif ordinaire, lui, reproduit déjà presque exactement le nombre de zéros et améliore nettement l'AIC : l'essentiel de l'excès de zéros apparent est en fait de la surdispersion. Le modèle à barrière reproduit exactement les zéros observés, comme prévu. Le ZINB a le plus petit AIC, environ 10 points de moins que le modèle binomial négatif, pénalité des six paramètres supplémentaires comprise. Il y a donc des zéros structurels, mais leur rôle est secondaire : le modèle binomial négatif reproduisait déjà le nombre de zéros.

```
summary(m.zinb)
```

```
Call:
zeroinfl(formula = art ~ . | ., data = bioChemists, dist = "negbin")

Pearson residuals:
    Min      1Q  Median      3Q     Max
-1.2942 -0.7601 -0.2909  0.4448  6.4155

Count model coefficients (negbin with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.4167465  0.1435966   2.902  0.00371 **
femWomen    -0.1955068  0.0755926  -2.586  0.00970 **
marMarried   0.0975826  0.0844520   1.155  0.24789
kid5        -0.1517325  0.0542061  -2.799  0.00512 **
phd         -0.0007001  0.0362697  -0.019  0.98460
ment         0.0247862  0.0034927   7.097 1.28e-12 ***
```

```
Log(theta)   0.9763565  0.1354695   7.207 5.71e-13 ***

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.19169    1.32282  -0.145  0.88478
femWomen     0.63593    0.84892   0.749  0.45379
marMarried  -1.49947    0.93867  -1.597  0.11017
kid5         0.62843    0.44278   1.419  0.15582
phd         -0.03771    0.30801  -0.122  0.90254
ment        -0.88229    0.31623  -2.790  0.00527 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 2.6548
Number of iterations in BFGS optimization: 43
Log-likelihood: -1550 on 13 Df
```

Les deux parties du ZINB s'interprètent séparément. Dans la partie de comptage, chaque article supplémentaire du directeur multiplie le nombre attendu d'articles du doctorant (parmi ceux qui ne sont pas des zéros structurels) par $e^{0,0248} \approx 1{,}025$, et le fait d'être une femme le multiplie par environ 0,82. Dans la partie binaire, seul le nombre d'articles du directeur est significatif : chaque article du directeur multiplie par $e^{-0,88} \approx 0{,}41$ la cote d'être un zéro structurel, c'est-à-dire un doctorant qui ne publie pas du tout. Un directeur très actif réduit donc la probabilité de ne rien publier et augmente le nombre d'articles de ceux qui publient.

Remarque — Comment choisir ?

Les modèles de Poisson, binomial négatif, ZIP, ZINB et à barrière ne sont pas tous emboîtés : le modèle de Poisson est un cas limite des modèles binomial négatif et à zéros excédentaires, mais les modèles à zéros excédentaires et à barrière ne sont pas emboîtés entre eux, et la valeur $\pi_i = 0$ est à la frontière de l'espace des paramètres (comme pour le test de surdispersion). En pratique, on compare les modèles par l'AIC ou par validation croisée (Chapitre 5) et on examine la répartition des comptes prédits, pas seulement le nombre de zéros. Le choix entre zéros excédentaires et barrière relève surtout de l'interprétation : y a-t-il une sous-population qui ne peut pas produire d'événement (zéros excédentaires), ou un seul mécanisme qui décide d'abord de franchir le zéro (barrière) ?

Remarque — À retenir

— Régression multinomiale : chaque catégorie autre que la référence a son vecteur β_k,

soit $(K-1)(p+1)$ paramètres, et $e^{\beta_{kj}}$ multiplie le rapport π_{ik}/π_{i1}. Un $\beta_{kj} > 0$ n'implique pas une hausse de π_{ik}, d'où l'intérêt de tracer les $\hat{\pi}_{ik}$ (ajustement avec `nnet::multinom`, qui prend le premier niveau du facteur comme référence) ;
— Cotes proportionnelles : logit $P(Y_i \leq k \mid x_i) = \alpha_k - x_i^\top \beta$, avec un β commun à tous les seuils ($K-1+p$ paramètres), de sorte que e^{β_j} multiplie la cote de $\{Y_i > k\}$ quel que soit k. Avec `polr` de `MASS`, l'ordre des niveaux du facteur ordonné fixe le signe des coefficients, et la convention de signe varie d'un logiciel à l'autre ;
— Vérification et comparaison : l'hypothèse de cotes proportionnelles se teste en la relâchant pour une variable à la fois (`clm` de `ordinal`, argument `nominal`), par un test du rapport de vraisemblance à $K-2$ degrés de liberté. Les modèles ordinal et multinomial ne sont pas emboîtés et se comparent par l'AIC ;
— Zéros excédentaires ou barrière : dans un modèle ZIP, un zéro peut être structurel ou venir de la loi de Poisson, alors que dans un modèle à barrière, tous les zéros viennent de la partie binaire. En R, `zeroinfl` et `hurdle` de `pscl`, avec une formule `y ~ x1 + x2 | z1` (partie de comptage, puis partie binaire) et l'argument `dist` ;
— Piège : les zéros excédentaires créent à eux seuls de la surdispersion, et un modèle binomial négatif ordinaire suffit parfois à reproduire les zéros observés. Ne concluez donc pas à des zéros structurels sur la seule base du nombre de zéros comparé à la loi de Poisson, et comparez plutôt les modèles par l'AIC ou par validation croisée.

4.4 Exercices

Les exercices sont classés par niveau de difficulté. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

4.4.1 Niveau facile

Exercice 1. Considérez la régression multinomiale avec K catégories et la catégorie 1 comme référence.

a) Montrez que, pour $K = 2$, le modèle est exactement la régression logistique de $\mathbf{1}\{Y_i = 2\}$.
b) On change de catégorie de référence et on prend la catégorie l. Exprimez les nouveaux coefficients $\beta_k^{(l)}$ en fonction des β_k de départ. Les probabilités ajustées changent-elles ?

Voir la solution

Exercice 2. Un compte suit un modèle ZIP avec $\pi = 0{,}2$ et $\mu = 2$.

a) Calculez $P(Y = 0)$ et $E[Y]$.
b) Calculez la probabilité d'un zéro sous une loi de Poisson de même espérance. Commentez.

Voir la solution

Exercice 3. Un modèle à cotes proportionnelles pour une réponse à trois catégories ($1 < 2 < 3$) a pour seuils $\alpha_1 = 1$ et $\alpha_2 = 3$ et pour coefficient $\beta = 0{,}3$ d'une variable x.

a) Calculez les probabilités des trois catégories pour $x = 10$ et pour $x = 20$.
b) Vérifiez que la cote de $\{Y > k\}$ contre $\{Y \leq k\}$ est multipliée par le même facteur entre $x = 10$ et $x = 20$, que k vaille 1 ou 2.

Voir la solution

4.4.2 Niveau moyen

Exercice 1. Pour la régression multinomiale, la log-vraisemblance s'écrit $\ell = \sum_i \sum_k y_{ik} \log \pi_{ik}$, avec $\sum_k y_{ik} = 1$.

a) Montrez que $\ell = \sum_i \left\{ \sum_{k=2}^{K} y_{ik} X_i^\top \beta_k - \log\left(1 + \sum_{l=2}^{K} e^{X_i^\top \beta_l}\right) \right\}$.
b) Déduisez-en le score $\partial \ell / \partial \beta_k = \sum_i (y_{ik} - \pi_{ik}) X_i$. Faites le lien avec le score de la régression logistique.
c) Montrez que, si le modèle contient une ordonnée à l'origine, le nombre prédit d'observations dans chaque catégorie, $\sum_i \hat{\pi}_{ik}$, est égal au nombre observé.

Voir la solution

Exercice 2. Démontrez les moments du modèle ZIP donnés au Chapitre 4 :

$$E[Y] = (1 - \pi)\mu, \qquad Var[Y] = (1 - \pi)\mu(1 + \pi\mu).$$

(Indice : écrivez $Y = (1 - S)\, W$, où $S \sim \text{Bernoulli}(\pi)$ est indépendant de $W \sim \mathcal{P}(\mu)$.) Montrez ensuite que le rapport $Var[Y]/E[Y]$ dépasse 1 dès que $\pi > 0$.

Voir la solution

Exercice 3. Considérez le modèle à barrière, avec $P(Y_i = 0) = 1 - p_i$ et $P(Y_i = y) = p_i \, f(y; \mu_i) / \{1 - f(0; \mu_i)\}$ pour $y \geq 1$, où $\text{logit}(p_i) = z_i^\top \gamma$ et $\log \mu_i = x_i^\top \beta$.

a) Montrez que la log-vraisemblance se sépare en une partie qui ne dépend que de γ et une partie qui ne dépend que de β. Identifiez chacune.
b) Déduisez-en que, si z_i contient une ordonnée à l'origine, le nombre de zéros prédit, $\sum_i (1 - \hat{p}_i)$, est exactement le nombre de zéros observé.

Voir la solution

4.4.3 Niveau difficile

Exercice 1. On reprend l'interprétation par une variable latente du modèle à cotes proportionnelles : $Y_i^* = x_i^\top \beta + \sigma\epsilon_i$, où ϵ_i suit la loi logistique standard, et $Y_i = k$ si $\alpha_{k-1} < Y_i^* \leq \alpha_k$.

a) Montrez que $(\beta, \alpha_1, \ldots, \alpha_{K-1}, \sigma)$ et $(c\beta, c\alpha_1, \ldots, c\alpha_{K-1}, c\sigma)$ donnent la même loi pour Y_i, quel que soit $c > 0$. Pourquoi peut-on fixer $\sigma = 1$?
b) Montrez de même qu'on ne peut pas inclure une ordonnée à l'origine dans x_i en plus des seuils.
c) En régression multinomiale, le rapport π_{ik}/π_{il} ne dépend pas des autres catégories (indépendance des alternatives non pertinentes). Cette propriété vaut-elle pour le modèle à cotes proportionnelles ? Justifiez à l'aide de π_{i1}/π_{i3} pour $K = 3$.

Voir la solution

Exercice 2 (appliqué). On vérifie numériquement la factorisation de l'exercice 3 du niveau moyen sur les données `bioChemists`.

a) Ajustez le modèle à barrière de Poisson `hurdle(art ~ ., data = bioChemists, dist = "poisson")`. Ajustez ensuite la régression logistique de `art > 0` sur toutes les variables et comparez ses coefficients à ceux de la partie binaire.
b) Écrivez la log-vraisemblance de la loi de Poisson tronquée en zéro pour les doctorants qui ont publié au moins un article, maximisez-la avec `optim` et comparez les estimations à celles de la partie de comptage.
c) Vérifiez que le nombre de zéros prédit est égal au nombre observé.

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— McCullagh et Nelder (1989). *Generalized Linear Models*
— Faraway (2016). *Extending the Linear Model with R*
— Agresti (2013). *Categorical Data Analysis*
— Zeileis, Kleiber et Jackman (2008). « Regression Models for Count Data in R »
— Agresti (2010). *Analysis of Ordinal Categorical Data*

5 Prédiction, mesures de performance et validation croisée

Ce chapitre utilise les librairies R suivantes.

```
library(boot)
library(dplyr)
```

Ce chapitre porte sur le choix d'un modèle pour la prédiction. On distingue d'abord l'adéquation aux données observées de la précision de prédiction pour une observation future. On estime ensuite cette précision par validation croisée (ensemble de validation, exclusion d'une observation, K blocs). La dernière partie traite des fonctions de perte adaptées aux GLM, ainsi que d'une approximation qui évite de réajuster le modèle n fois.

5.1 Adéquation versus précision de prédiction

Cette section oppose deux critères de choix de modèle : l'ajustement aux données observées et la précision de la prédiction pour une nouvelle observation. On les compare sur un jeu de données forestier.

> **Exemple — Jeu de données arbres**
>
> Le jeu de données `arbres` est un extrait de 200 arbres des placettes-échantillons permanentes de l'inventaire forestier du Québec. Ces arbres ont été mesurés entre 2015 et 2025 dans une trentaine de régions écologiques. L'extrait retient les quatre espèces les plus fréquentes (bouleau à papier, épinette noire, érable rouge et sapin baumier), à raison de 50 arbres par espèce et d'un seul arbre par placette. On y trouve notamment le diamètre à hauteur de poitrine (`diameter_cm`), la hauteur (`height_m`), l'espèce (`species_fr`) et la position de l'arbre dans le couvert forestier (`canopy_stratum`).
> Seuls les arbres d'au moins 9 cm de diamètre sont mesurés à l'inventaire. Comme le nombre d'arbres par espèce a été fixé, l'extrait n'est pas représentatif de la composition de la forêt.

L'objectif est de prédire la hauteur H_i d'un arbre à partir de son diamètre D_i. Comme la relation hauteur-diamètre est allométrique, on la modélise sur l'échelle log-log. Pour la suite,

$e(i)$ désigne l'espèce de l'arbre i et $s(i)$, sa position dans le couvert forestier (sa strate de couvert). Le choix se fait parmi cinq modèles de complexité croissante :

$$
\begin{aligned}
\text{Modèle 1 :}\ & \log H_i = \beta_0 + \beta_1 \log D_i + \epsilon_i, \\
\text{Modèle 2 :}\ & \log H_i = \beta_{0,e(i)} + \beta_1 \log D_i + \epsilon_i, \\
\text{Modèle 3 :}\ & \log H_i = \beta_{0,e(i)} + \beta_{1,e(i)} \log D_i + \epsilon_i, \\
\text{Modèle 4 :}\ & \log H_i = \beta_{0,e(i)} + \beta_{1,e(i)} \log D_i + \delta_{s(i)} + \epsilon_i, \\
\text{Modèle 5 :}\ & \log H_i = \sum_{k=0}^{3} \beta_{k,e(i)} \left(\log D_i\right)^k + \delta_{s(i)} + \epsilon_i.
\end{aligned}
$$

Le modèle 2 permet une ordonnée à l'origine par espèce et le modèle 3, en plus, une pente par espèce. Le modèle 4 ajoute l'effet de la strate, alors que le modèle 5 remplace chaque droite par un polynôme cubique propre à l'espèce. Ces modèles comptent respectivement 2, 5, 8, 10 et 18 paramètres de moyenne (avec $\delta_{\text{Codominant}} = 0$ comme contrainte d'identifiabilité).

> **Remarque — Pourquoi ces modèles sont-ils linéaires ?**
> Les cinq modèles s'écrivent $Y = X\beta + \epsilon$ avec $Y_i = \log H_i$. Il ne faut pas voir « linéaire » comme une propriété des variables : les modèles sont linéaires dans les paramètres, même s'ils ne le sont pas dans les variables. Un effet propre à une espèce ou à une strate correspond à des colonnes d'indicatrices et une interaction, à des produits de colonnes. Les termes $(\log D_i)^k$ du modèle 5 sont des colonnes supplémentaires. Sur l'échelle originale, le modèle 1 équivaut à
>
> $$H_i = e^{\beta_0} D_i^{\beta_1} e^{\epsilon_i},$$
>
> soit une loi de puissance avec une erreur multiplicative, ce qui en fait un modèle intrinsèquement linéaire. En revanche, le modèle $H_i = \beta_0 D_i^{\beta_1} + \epsilon_i$, dont l'erreur est additive sur l'échelle originale, n'est pas linéaire. Avec des erreurs normales, c'est par contre un GLM normal avec lien log, puisque $E[H_i] = \exp(\log \beta_0 + \beta_1 \log D_i)$. On l'ajuste donc avec `glm(height_m ~ log(diameter_cm), family = gaussian(link = "log"))`, dont l'estimateur coïncide avec celui des moindres carrés non linéaires.

```
# Seulement 4 arbres dominants : on les regroupe avec les codominants
arbres <- read.csv("Jeux de données/arbres.csv") |>
  mutate(
    espece = factor(species_fr),
    strate = factor(if_else(canopy_stratum == "Dominant", "Codominant",
     ↪  canopy_stratum),
                    levels = c("Codominant", "Intermédiaire", "Opprimé"))
  )
```

```r
formules <- list(
  M1 = log(height_m) ~ log(diameter_cm),
  M2 = log(height_m) ~ log(diameter_cm) + espece,
  M3 = log(height_m) ~ log(diameter_cm) * espece,
  M4 = log(height_m) ~ log(diameter_cm) * espece + strate,
  M5 = log(height_m) ~ poly(log(diameter_cm), 3) * espece + strate
)
modeles <- lapply(formules, lm, data = arbres)

# Erreur d'entraînement : SSE / n
sapply(modeles, function(m) mean(residuals(m)^2))
```

```
        M1         M2         M3         M4         M5
0.03932338 0.03401792 0.03092351 0.02850735 0.02727676
```

Pour chaque modèle, on peut calculer $SSE = \sum_i (y_i - \hat{y}_i)^2$, l'AIC ou le BIC, ou encore faire des tests pour modèles emboîtés. Ici, $\hat{y}_i$ est la valeur ajustée par le modèle estimé sur toutes les données. La SSE (ou le R^2) mesure seulement l'ajustement aux données observées. L'AIC, le BIC et les tests pour modèles emboîtés ajoutent une pénalité ou un contrôle pour la complexité. Sur les arbres, l'erreur d'entraînement SSE/n diminue d'un modèle à l'autre jusqu'au modèle 5, comme c'est toujours le cas pour des modèles emboîtés ajustés par moindres carrés.

Supposons plutôt qu'on veuille prédire la hauteur H_0 d'un nouvel arbre de diamètre D_0. On cherche alors le modèle qui minimise $E[(Y_0 - \hat{Y}_0)^2]$, avec $Y_0 = \log H_0$. Les deux critères, adéquation et précision de la prédiction, sont différents et peuvent se contredire. Par exemple, avec n points distincts, un polynôme de degré $n-1$ a une adéquation parfaite ($SSE = 0$), mais il ne sert à rien pour prédire de nouvelles valeurs : c'est le surajustement.

Il faut ainsi distinguer deux taux d'erreur :

- le taux d'erreur d'entraînement : évalué par SSE/n, il mesure l'adéquation aux données observées ;
- le taux d'erreur test : il mesure la précision de la prédiction pour une observation future.

Le taux d'erreur test n'est pas observable directement. Les méthodes de validation croisée de la section suivante l'estiment à partir des seules données disponibles.

5.2 Validation croisée

Cette section présente trois façons d'estimer le taux d'erreur test avec les données disponibles : l'ensemble de validation, l'exclusion d'une observation et les K blocs.

5.2.1 Approche par ensemble de validation

On partage le jeu de données en deux parties : l'échantillon d'entraînement sert à estimer les paramètres, et l'échantillon test sert à estimer le taux d'erreur test. Avec n_1 observations d'entraînement et $n_2 = n - n_1$ observations test,

$$\widehat{Err}_{test} = \frac{1}{n_2} \sum_{i=n_1+1}^{n} \left(y_i - \hat{y}_{i,\text{ent}}\right)^2,$$

où $\hat{y}_{i,\text{ent}}$ est la prédiction de l'observation i calculée avec les paramètres estimés sur le seul échantillon d'entraînement.

Cette approche a deux défauts. D'abord, la partition est arbitraire, ce qui introduit de la variabilité (défaut (i)). Ensuite, on n'estime β qu'avec une fraction des données, ce qui augmente la variance de $\hat{\beta}$ (défaut (ii)). Le premier défaut s'atténue en répétant la partition aléatoirement un grand nombre de fois, puis en prenant la moyenne des erreurs obtenues.

Exemple — Variabilité de l'ensemble de validation sur `arbres`

Répétons 1000 fois une partition aléatoire en deux moitiés de 100 arbres, en notant chaque fois le modèle qui a la plus petite erreur test.

```
set.seed(4300)
choix <- replicate(1000, {
  i <- sample(nrow(arbres), 100)
  err <- sapply(formules, function(f) {
    m <- lm(f, data = arbres[i, ])
    mean((log(arbres$height_m[-i]) - predict(m, arbres[-i, ]))^2)
  })
  names(which.min(err))
})
table(factor(choix, levels = names(formules)))
```

```
 M1  M2  M3  M4  M5
 15  95 170 664  56
```

Le modèle 4 est choisi le plus souvent, mais pas toujours : selon la partition tirée, chacun des autres modèles peut l'emporter. En pratique, cette approche repose sur une seule partition, qui aurait pu mener à un autre choix. C'est le défaut (i).

5.2.2 Validation croisée avec exclusion d'une observation (LOOCV de *leave one out cross validation*)

Pour remédier au deuxième défaut, il suffit d'exclure une seule observation à la fois. Pour $i = 1, \ldots, n$, on estime β avec toutes les observations sauf la i^{e}, puis on calcule $\hat{y}_{i,(-i)}$. Le critère est

$$VC_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{y}_{i,(-i)} \right)^2 .$$

Propriété — Formule explicite pour la régression linéaire

$$VC_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2 ,$$

où $\hat{y}_i$ est la valeur ajustée par le modèle estimé sur toutes les données et h_i, le levier de l'observation i (défini au chapitre 0, dans les rappels sur la matrice chapeau). Il n'est donc pas nécessaire d'ajuster n régressions : un seul ajustement suffit.

En R, la fonction `cv.glm` de la librairie `boot` calcule $VC_{(n)}$ en réajustant réellement le modèle n fois. Comme elle exige un objet `glm`, un modèle linéaire doit être ajusté avec `glm` et la famille gaussienne, qui est la famille par défaut.

Exemple — LOOCV sur `arbres`

```
loocv <- function(m) mean((residuals(m) / (1 - hatvalues(m)))^2)
vc.n <- sapply(modeles, loocv)
vc.n
```

```
        M1         M2         M3         M4         M5
0.04009088 0.03576446 0.03349013 0.03175003 0.03303419
```

Vérifions que la formule explicite donne exactement le résultat obtenu en ajustant 200 régressions. Notez que les modèles sont créés avec `do.call`. Comme `cv.glm` réévalue l'appel enregistré dans l'objet, un appel créé directement par `lapply(formules, glm, ...)` ne fonctionnerait pas, parce qu'il ferait référence à une fonction `FUN` qui n'existe plus au moment du réajustement.

```
modeles.glm <- lapply(formules, function(f)
  do.call("glm", list(formula = f, data = quote(arbres))))
```

```
c(formule = unname(vc.n["M4"]), cv.glm = cv.glm(arbres,
↪  modeles.glm$M4)$delta[1])
```

```
   formule     cv.glm
0.03175003 0.03175003
```

$VC_{(n)}$ est minimal pour le modèle 4 ($VC_{(n)} \approx 0{,}0318$). Contrairement à l'erreur d'entraînement, $VC_{(n)}$ augmente du modèle 4 au modèle 5, dont les huit paramètres supplémentaires améliorent l'ajustement mais détériorent la prédiction. C'est du surajustement.

5.2.3 Validation croisée à K blocs

Pour d'autres modèles, notamment les GLM, le calcul exact de $VC_{(n)}$ demande l'ajustement de n modèles. Ce calcul peut être coûteux, même s'il existe une approximation (voir la Section 5.3.3). On utilise alors la validation croisée à K blocs, en pratique avec $K = 5$ ou $K = 10$. Le jeu de données est partagé en K parties $A_1, \ldots, A_K$ de tailles à peu près égales. Pour $k = 1, \ldots, K$, on estime β avec toutes les données sauf A_k, et on calcule

$$MSE_k = \frac{1}{n_k} \sum_{i \in A_k} \left\{ y_i - \hat{y}_{i,(-k)} \right\}^2,$$

où n_k est la taille de A_k. On estime le taux d'erreur test par

$$VC_{(K)} = \frac{1}{K} \sum_{k=1}^{K} MSE_k.$$

La fonction `cv.glm` donne $VC_{(K)}$ au moyen de l'argument `K`.

Exemple — Validation croisée à K blocs sur `arbres`

```
vc.K <- sapply(c(5, 10), function(K)
  sapply(modeles.glm, function(m) {
    set.seed(4300)   # même partition pour tous les modèles
    cv.glm(arbres, m, K = K)$delta[1]
  }))
colnames(vc.K) <- c("VC_5", "VC_10")
cbind(VC_n = vc.n, vc.K)
```

```
         VC_n       VC_5      VC_10
M1 0.04009088 0.04004830 0.04007688
M2 0.03576446 0.03560323 0.03588449
M3 0.03349013 0.03368676 0.03348481
M4 0.03175003 0.03196452 0.03209201
M5 0.03303419 0.03345063 0.03320921
```

Les trois critères désignent le modèle 4. Le partage en blocs est aléatoire, mais si on répète l'expérience avec d'autres graines, le choix reste plus stable qu'avec un ensemble de validation unique. En fait, chaque observation sert ici exactement une fois au test, et chaque modèle est ajusté sur 80 % ou 90 % des données plutôt que sur la moitié.
Il ne faut pas oublier de réinitialiser la graine avant chaque appel. Sans cette précaution, chaque modèle serait évalué sur une partition différente, de sorte qu'une partie des écarts entre modèles viendrait du hasard de la partition plutôt que des modèles eux-mêmes.

Remarque — Sur quelle échelle mesure-t-on l'erreur ?

Ici, l'erreur est mesurée sur l'échelle de $\log H$, et les cinq modèles peuvent être comparés parce qu'ils ont la même réponse. Pour comparer un modèle sur $\log H$ à un modèle sur H, il faut calculer les deux erreurs sur la même échelle. Par exemple, on ramène les prédictions du modèle log-log à l'échelle originale avec $\exp(\hat{y}_{i,(-k)})$ avant de calculer MSE_k. Attention : $\exp(\hat{y}_{i,(-k)})$ n'estime pas l'espérance de H_i, mais sa médiane conditionnelle.

Remarque — Fuite d'information et observations dépendantes

La validation croisée n'estime bien l'erreur test que si chaque bloc test reste inconnu du modèle jusqu'à son évaluation. Toute étape qui utilise les données fait partie de l'ajustement : standardisation des variables, imputation des valeurs manquantes, sélection de variables, réglage d'un hyperparamètre (comme λ au Chapitre 6 ou k au Chapitre 7). Chacune de ces étapes doit être refaite à l'intérieur de chaque échantillon d'entraînement, avec les seules données d'entraînement. Par exemple, sélectionner les variables sur toutes les données, puis évaluer le modèle retenu par validation croisée, sous-estime l'erreur test, parce que les blocs test ont déjà servi au choix des variables. On parle de fuite d'information (*data leakage*).
Pour la même raison, quand la validation croisée sert à choisir un modèle ou un hyperparamètre, la plus petite valeur de $VC_{(K)}$ est une estimation optimiste de l'erreur test du modèle retenu. Pour estimer cette erreur sans biais, il faut un échantillon test mis de côté dès le départ, ou une validation croisée emboîtée (une validation croisée interne pour le choix, à l'intérieur de chaque bloc d'une validation croisée externe pour l'évaluation).
La validation croisée suppose aussi que les observations sont indépendantes. Avec des données groupées (plusieurs mesures par sujet, par école ou par placette) ou longitudinales,

il faut répartir les groupes, et non les lignes, entre les blocs (voir le Chapitre 10). Sinon, le modèle voit à l'entraînement des observations très semblables à celles du bloc test, et l'erreur est sous-estimée. Pour une prédiction dans le temps, il faut respecter l'ordre chronologique : on ajuste sur le passé et on évalue sur la période qui suit. Le jeu de données `arbres` ne retient qu'un arbre par placette, de sorte que le premier problème ne se pose pas ici.

5.3 Validation croisée pour les GLM

Les critères $VC_{(n)}$ et $VC_{(K)}$ ont été définis avec l'erreur quadratique, qui est la perte naturelle pour une réponse normale. Dans les GLM des chapitres précédents, la réponse est binaire, binomiale ou un dénombrement, et sa variance dépend de sa moyenne. L'erreur quadratique reste utilisable, mais d'autres fonctions de perte sont souvent mieux adaptées. Avec une fonction de perte $L(y, \mu)$, le critère de validation croisée à K blocs devient

$$VC_{(K)} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in A_k} L\left(y_i, \hat{\mu}_{i,(-k)}\right),$$

où $\hat{\mu}_{i,(-k)} = g^{-1}\left(X_i^\top \hat{\beta}_{(-k)}\right)$ est la moyenne prédite pour l'observation i par le modèle ajusté sans le bloc A_k. On retrouve $VC_{(n)}$ avec $K = n$.

5.3.1 Régression logistique et classification

Avec une réponse binaire, on prédit la classe de l'observation i par $\hat{c}_i = \mathbf{1}_{\{\hat{\pi}_i > u\}}$, pour un seuil u (souvent $1/2$), et le taux d'erreur test devient le taux de mauvaise classification. Ce n'est pas la seule perte possible.

Définition — Fonctions de perte pour une réponse binaire

Pour $y \in \{0, 1\}$ et une probabilité prédite $\hat{\pi}$:

$$\begin{aligned} \text{Mauvaise classification} &: L(y, \hat{\pi}) = \mathbf{1}_{\{y \neq \mathbf{1}_{\{\hat{\pi} > u\}}\}}, \\ \text{Score de Brier} &: L(y, \hat{\pi}) = (y - \hat{\pi})^2, \\ \text{Déviance (log-perte)} &: L(y, \hat{\pi}) = -2\left[y \log \hat{\pi} + (1 - y) \log(1 - \hat{\pi})\right]. \end{aligned}$$

La troisième perte est exactement la déviance individuelle d_i d'une observation de Bernoulli (voir la déviance).

Remarque — Quelle perte choisir ?

Le taux de mauvaise classification ne retient que le côté du seuil u où tombe $\hat{\pi}$, de sorte qu'une prédiction $\hat{\pi} = 0{,}51$ et une prédiction $\hat{\pi} = 0{,}99$ sont traitées de la même façon. De plus, ce critère est discontinu, il dépend du choix arbitraire de u et il est instable quand n est petit.

Le score de Brier et la déviance sont des règles de score propres, c'est-à-dire que leur espérance est minimale quand $\hat{\pi}$ égale la vraie probabilité π. Ils évaluent donc la qualité des probabilités prédites elles-mêmes, pas seulement la classification qui en découle, et, contrairement au taux de mauvaise classification, ils ne dépendent d'aucun seuil u. La déviance pénalise lourdement les erreurs commises avec assurance : elle devient infinie si $\hat{\pi}_{i,(-k)} \in \{0, 1\}$ alors que y_i prend l'autre valeur, ce qui arrive quand un échantillon d'entraînement présente une séparation. Cette pénalité sévère est voulue. Une probabilité prédite de 0 affirme que l'événement est impossible, et l'observer quand même est une erreur grave.

En calcul numérique, plusieurs logiciels tronquent toutefois $\hat{\pi}$ dans $[\delta, 1 - \delta]$ pour un petit $\delta > 0$, ce qui évite le logarithme de zéro (le code plus bas prend $\delta = 10^{-8}$). Notez que la perte d'une observation mal classée avec assurance dépend alors du choix de δ : elle vaut environ $-2 \log \delta \approx 37$ pour $\delta = 10^{-8}$. Quelques observations de ce type peuvent dominer le critère, et il vaut mieux les examiner que de se fier à la valeur tronquée.

La sensibilité, la spécificité, la courbe ROC et l'AUC (Section 2.5) sont d'autres mesures de performance. Comme l'AUC ne dépend d'aucun seuil, elle sert entre autres de critère de validation croisée en classification (voir Chapitre 7). Par contre, elle mesure seulement la capacité du modèle à ordonner les observations, et non la calibration des probabilités.

Remarque — Données groupées

Avec des données binomiales groupées, chaque ligne regroupe m_i essais, de sorte qu'exclure une ligne revient à exclure tout un groupe. On évalue alors la prédiction pour une nouvelle combinaison des variables explicatives, pas pour un nouvel individu. Les deux versions du jeu de données `empress`, individuelle (`empress_individus`) et agrégée (`empress_agrege`), illustrent cette différence dans les exemples plus bas. Remarquez aussi que la fonction `cv.glm` passe à la fonction de coût les proportions y_i/m_i, sans pondération par m_i : il faut en tenir compte si on veut une perte pondérée.

5.3.2 Régression de Poisson

Définition — Fonctions de perte pour un dénombrement

Pour $y \in \mathbb{N}$ et une moyenne prédite $\hat{\mu} > 0$:

$$\text{Erreur quadratique} : L(y,\hat{\mu}) = (y-\hat{\mu})^2,$$
$$\text{Perte de Pearson} : L(y,\hat{\mu}) = \frac{(y-\hat{\mu})^2}{\hat{\mu}},$$
$$\text{Déviance de Poisson} : L(y,\hat{\mu}) = 2\left[y\log\left(\frac{y}{\hat{\mu}}\right) - (y-\hat{\mu})\right],$$

avec la convention $0\log 0 = 0$.

Remarque — Particularités du cas Poisson

Comme $Var[Y_i] = \mu_i$, l'erreur quadratique est dominée par les observations dont le dénombrement est élevé. La perte de Pearson et la déviance, elles, standardisent implicitement chaque erreur par sa variance.
En présence d'une variable offset, la prédiction d'une observation exclue utilise sa propre exposition :

$$\hat{\mu}_{i,(-k)} = m_i \exp\left(X_i^\top \hat{\beta}_{(-k)}\right).$$

En R, je vous conseille d'écrire l'offset dans la formule (`offset(log(m))`) : `predict` l'applique alors correctement aux nouvelles données.
Enfin, comme les modèles de Poisson et quasi-Poisson donnent les mêmes $\hat{\mu}_i$, la validation croisée ne peut pas les départager. Le choix du paramètre de dispersion relève de l'inférence (voir la surdispersion). Un modèle de Poisson et un modèle binomial négatif, en revanche, se comparent à condition d'utiliser la même perte pour les deux.

5.3.3 Approximation de $VC_{(n)}$ sans réajustement

Hormis le modèle linéaire normal, un GLM n'a pas de formule exacte pour $VC_{(n)}$, mais il existe une approximation fondée sur un seul ajustement.

Propriété — Approximation de l'erreur de validation croisée

Pour un GLM avec lien canonique (régression logistique, régression de Poisson, modèle linéaire normal ; le paramètre de dispersion ϕ se simplifie dans le pas de Newton), on a

$$y_i - \hat{\mu}_{i,(-i)} \approx \frac{y_i - \hat{\mu}_i}{1-h_i}, \qquad \text{d'où} \qquad VC_{(n)} \approx \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i-\hat{\mu}_i}{1-h_i}\right)^2,$$

où h_i est le i^{e} élément diagonal de la matrice chapeau

$$H = W^{1/2}X\left(X^\top W X\right)^{-1}X^\top W^{1/2},$$

obtenu en R avec `hatvalues()`.

Pour aller plus loin avec la théorie — Idée de la dérivation

Sous le lien canonique, le score s'écrit

$$U(\beta) = \sum_{j=1}^{n} X_j(y_j - \mu_j),$$

et le poids de l'observation j vaut $w_j = \partial\mu_j/\partial\eta_j$. Posons $J = X^\top W X$. Comme le score complet s'annule en $\hat{\beta}$, le score sans l'observation i, évalué en $\hat{\beta}$, vaut

$$U_{(-i)}(\hat{\beta}) = -X_i(y_i - \hat{\mu}_i).$$

On fait une seule itération de Newton-Raphson à partir de $\hat{\beta}$. Avec la formule de Sherman-Morrison, on obtient

$$\hat{\beta}_{(-i)} \approx \hat{\beta} - \left(J - w_i X_i X_i^\top\right)^{-1} X_i(y_i - \hat{\mu}_i) = \hat{\beta} - \frac{J^{-1}X_i\,(y_i - \hat{\mu}_i)}{1 - h_i}.$$

Il en découle que

$$\hat{\eta}_{i,(-i)} \approx \hat{\eta}_i - \frac{h_i}{w_i}\,\frac{y_i - \hat{\mu}_i}{1 - h_i}.$$

Il reste à linéariser g^{-1} :

$$\hat{\mu}_{i,(-i)} \approx \hat{\mu}_i + w_i\left(\hat{\eta}_{i,(-i)} - \hat{\eta}_i\right) = \hat{\mu}_i - h_i\,\frac{y_i - \hat{\mu}_i}{1 - h_i},$$

ce qui donne le résultat. Dans le modèle linéaire normal, les deux approximations (un seul pas de Newton et la linéarisation de g^{-1}) deviennent exactes, et on retrouve la formule explicite de la LOOCV.

5.3.4 Exemples sur les données de l'Empress of Ireland

Cette section applique les pertes précédentes aux données de l'Empress of Ireland. La fonction `cv.glm` accepte un argument `cost`, soit une fonction `cost(y, yhat)` où `yhat` est la moyenne prédite (échelle de la réponse). Par défaut, `cost` est l'erreur quadratique, c'est-à-dire le score de Brier pour une réponse binaire.

```
niveaux.classe <- c("Première", "Deuxième", "Troisième")

empress <- read.csv("Jeux de données/empress_individus.csv") |>
  mutate(classe = factor(classe, levels = niveaux.classe))

empress.agr <- read.csv("Jeux de données/empress_agrege.csv") |>
```

```
  mutate(classe = factor(classe, levels = niveaux.classe))
```

Exemple — Régression logistique sur `empress_individus`

Quatre modèles sont comparés pour la probabilité de survie d'un passager :

- `m1` : la classe seule ;
- `m2` : les effets additifs de la classe, du sexe et du groupe d'âge ;
- `m3` : le même modèle, avec la classe traitée comme variable numérique (tendance linéaire sur l'échelle logit) ;
- `m4` : un modèle avec interaction entre la classe et le sexe.

```
m1 <- glm(survecu ~ classe, data = empress, family = binomial)
m2 <- glm(survecu ~ classe + sexe + groupe_age, data = empress, family =
 ↪  binomial)
m3 <- glm(survecu ~ classe_num + sexe + groupe_age, data = empress,
 ↪  family = binomial)
m4 <- glm(survecu ~ classe * sexe + groupe_age, data = empress, family =
 ↪  binomial)

# y : réponse observée, p : probabilité prédite
cout.classif <- function(y, p) mean(y != (p > 0.5))
cout.brier   <- function(y, p) mean((y - p)^2)
cout.dev     <- function(y, p) {
  p <- pmin(pmax(p, 1e-8), 1 - 1e-8)   # évite log(0) en cas de
 ↪  séparation
  -2 * mean(y * log(p) + (1 - y) * log(1 - p))
}

modeles.logit <- list(m1 = m1, m2 = m2, m3 = m3, m4 = m4)
couts.logit   <- list(classif = cout.classif, brier = cout.brier,
 ↪  deviance = cout.dev)

# VC_(10) : même graine avant chaque appel pour que tous les modèles
# soient évalués sur la même partition en blocs
vc.logit <- sapply(couts.logit, function(cout)
  sapply(modeles.logit, function(m) {
    set.seed(4300)
    cv.glm(empress, m, cost = cout, K = 10)$delta[1]
  }))
vc.logit
```

```
     classif     brier  deviance
```

```
m1 0.2052980 0.1600676 0.9997842
m2 0.2090823 0.1517565 0.9392754
m3 0.2052980 0.1521921 0.9413214
m4 0.2100284 0.1518934 0.9396020
```

La colonne du taux de mauvaise classification montre ses limites. Seuls 217 des 1057 passagers ont survécu, et aucune strate classe × sexe × âge n'atteint une proportion de survie de 50 % (la plus élevée, 49,0 %, est celle des hommes adultes de première classe). Avec le seuil $u = 1/2$, `m1` et `m3` prédisent toujours le décès, et leur taux de mauvaise classification est exactement la proportion de survivants (0,2053). Le modèle additif `m2` ne reproduit pas les proportions observées : il prédit $\hat{\pi} \approx 0{,}527$ pour les hommes adultes de première classe, donc la survie, ce qui augmente légèrement son taux. Il en va de même pour `m4` dans certains blocs. Le score de Brier et la déviance, eux, évaluent les probabilités prédites elles-mêmes. Ils écartent nettement `m1`, alors que `m2`, `m3` et `m4` sont presque à égalité.
Avec $n = 1057$, le calcul exact de $VC_{(n)}$ demande autant d'ajustements. Comparons-le à l'approximation de la Section 5.3.3, qui n'en demande qu'un seul :

```
h <- hatvalues(m2)
approx <- mean(((empress$survecu - fitted(m2)) / (1 - h))^2)

temps <- system.time(exact <- cv.glm(empress, m2)$delta[1])

c(approx = approx, exact = exact)
```

```
   approx     exact
0.1520801 0.1520766
```

```
temps["elapsed"]
```

```
elapsed
  4.542
```

Exemple — Régression de Poisson sur les données agrégées `empress_agrege`

La réponse est maintenant le nombre de survivants Y_i de chacune des 11 strates. L'effectif m_i de la strate sert d'offset, de sorte que $\mu_i = m_i \exp(X_i^\top \beta)$ et que le terme $\exp(X_i^\top \beta)$ s'interprète comme un taux de survie.

```r
p1 <- glm(survivants ~ classe + offset(log(effectif)),
          family = poisson, data = empress.agr)
p2 <- glm(survivants ~ classe + sexe + groupe_age +
 ↪  offset(log(effectif)),
          family = poisson, data = empress.agr)
p3 <- glm(survivants ~ classe_num + sexe + groupe_age +
 ↪  offset(log(effectif)),
          family = poisson, data = empress.agr)

cout.dev.pois <- function(y, mu) 2 * mean(ifelse(y > 0, y * log(y / mu),
 ↪  0) - (y - mu))
cout.pearson  <- function(y, mu) mean((y - mu)^2 / mu)
cout.eq       <- function(y, mu) mean((y - mu)^2)

modeles.pois <- list(p1 = p1, p2 = p2, p3 = p3)
couts.pois   <- list(deviance = cout.dev.pois, pearson = cout.pearson,
 ↪  quadratique = cout.eq)

# VC_(n) sur les strates : chaque étape exclut une strate complète
sapply(couts.pois, function(cout)
  sapply(modeles.pois, function(m) cv.glm(empress.agr, m, cost =
   ↪  cout)$delta[1]))
```

```
    deviance   pearson quadratique
p1 20.958223 29.565991    774.9233
p2  4.467199  5.709436    215.2962
p3  4.589782  6.085684    237.1694
```

La colonne quadratique est dominée par la strate des hommes adultes de troisième classe. Cette strate compte 446 passagers et donc le plus grand $\hat{\mu}_i$. Les colonnes déviance et Pearson atténuent ce poids, mais cette strate y compte encore pour environ la moitié du critère ou plus.

Deux mises en garde s'imposent. D'abord, l'unité exclue est ici une strate entière. On évalue donc la capacité du modèle à prédire une combinaison classe × sexe × âge à partir des autres, ce qui n'est pas la question de l'exemple précédent (voir la remarque sur les données groupées). Pour la même raison, un modèle avec interactions ne peut pas toujours être évalué ainsi : si on retire la seule strate qui permet d'estimer un coefficient d'interaction, ce coefficient n'est plus estimable. Ensuite, la loi de Poisson n'est qu'une approximation de la loi binomiale quand les proportions sont petites. Comme plusieurs taux de survie sont ici modérés (près de 50 % chez les hommes adultes de première classe), un modèle binomial serait plus naturel. L'exemple sert à illustrer l'offset et le choix de la perte, pas à recommander ce modèle.

Remarque — À retenir

- Entraînement ou test : pour des modèles emboîtés ajustés par moindres carrés, l'erreur d'entraînement SSE/n diminue toujours avec la complexité, alors que la précision de prédiction se mesure par le taux d'erreur test, que la validation croisée estime. Un modèle peut s'ajuster parfaitement et ne servir à rien pour prédire (surajustement) ;
- LOOCV et K blocs : en régression linéaire, $VC_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i-\hat{y}_i}{1-h_i}\right)^2$ ne demande qu'un ajustement. La fonction `cv.glm` de `boot`, qui exige un objet `glm` (famille gaussienne pour un modèle linéaire), donne $VC_{(K)}$ avec l'argument `K` ($K = 5$ ou $K = 10$ en pratique), et la graine doit être réinitialisée avant chaque appel pour que les modèles soient évalués sur la même partition ;
- Fuite d'information : toute étape qui utilise les données (standardisation, imputation, sélection de variables, choix d'un hyperparamètre) doit être refaite dans chaque échantillon d'entraînement. Le minimum de $VC_{(K)}$ est optimiste pour le modèle retenu, et avec des données groupées (plusieurs mesures par sujet ou par placette), ce sont les groupes, et non les lignes, qui sont répartis entre les blocs ;
- Pertes pour une réponse binaire : le taux de mauvaise classification dépend du seuil u et est instable quand n est petit, alors que le score de Brier et la déviance sont des règles de score propres, qui évaluent les probabilités prédites elles-mêmes. Une perte se passe à `cv.glm` par l'argument `cost`, une fonction `cost(y, yhat)` où `yhat` est la moyenne prédite ;
- Poisson et offset : les modèles de Poisson et quasi-Poisson donnent les mêmes $\hat{\mu}_i$, de sorte que la validation croisée ne peut pas les départager. Écrivez l'offset dans la formule (`offset(log(m))`) pour que `predict` l'applique aux nouvelles données.

5.4 Exercices

Les exercices sont classés par niveau de difficulté, et les exercices appliqués de la fin demandent d'analyser un jeu de données en R. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

5.4.1 Niveau facile

Exercice 1. Une régression linéaire simple $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ ajustée sur $n = 5$ observations, avec $x_i = i$, donne les résidus suivants :

i	1	2	3	4	5
e_i	1,0	−1,4	0,8	−1,4	1,0

a) Vérifiez que ces résidus sont cohérents avec les équations normales, puis calculez les leviers à l'aide de la formule $h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_j (x_j - \bar{x})^2}$, valable en régression linéaire simple.

b) Calculez l'erreur d'entraînement SSE/n et le critère $VC_{(n)}$. Quelles observations contribuent le plus à $VC_{(n)}$?

Voir la solution

Exercice 2. Considérez le modèle sans variable explicative, $Y_i = \beta_0 + \epsilon_i$, pour lequel $\hat{Y}_i = \bar{Y}$.

a) Montrez que $h_i = 1/n$ pour tout i.

b) Calculez directement $Y_i - \bar{Y}_{(-i)}$, où $\bar{Y}_{(-i)}$ est la moyenne des observations autres que la i^{e}, et vérifiez qu'il est égal à $(Y_i - \bar{Y})/(1 - h_i)$.

c) Déduisez-en que $VC_{(n)} = \frac{n}{n-1} s^2$, où $s^2 = \frac{1}{n-1} \sum_i (Y_i - \bar{Y})^2$.

Voir la solution

Exercice 3. Soit $Y \sim \text{Bernoulli}(\pi)$ et $p \in [0, 1]$ une probabilité prédite.

a) Montrez que l'espérance du score de Brier vaut $E[(Y - p)^2] = \pi(1 - \pi) + (\pi - p)^2$. Déduisez-en que le score de Brier est une règle de score propre.

b) Pour le taux de mauvaise classification avec le seuil $u = 1/2$, calculez $E[\mathbf{1}\{Y \neq \mathbf{1}_{\{p > 1/2\}}\}]$ en fonction de π et de p. Pourquoi cette perte n'est-elle pas strictement propre ?

Voir la solution

5.4.2 Niveau moyen

Exercice 1. Démontrez la formule explicite de la LOOCV en régression linéaire,

$$Y_i - \hat{Y}_{i,(-i)} = \frac{Y_i - \hat{Y}_i}{1 - h_i}.$$

Utilisez la formule de mise à jour donnée à l'exercice 4 du niveau difficile du chapitre 0 :

$$\hat{\beta}_{(-i)} = \hat{\beta} - \frac{(X^\top X)^{-1} X_i e_i}{1 - h_i}.$$

Voir la solution

Exercice 2. Cet exercice vérifie que les pertes du Chapitre 5 pour les GLM sont propres.

a) Soit $Y \sim \text{Bernoulli}(\pi)$. Montrez que l'espérance de la log-perte $L(Y, p) = -2[Y \log p + (1 - Y) \log(1 - p)]$ est minimale en $p = \pi$.

b) Soit $Y \sim \mathcal{P}(\mu)$. Montrez que l'espérance de la déviance de Poisson

$$L(Y, \hat{\mu}) = 2\left[Y \log\left(\frac{Y}{\hat{\mu}}\right) - (Y - \hat{\mu})\right]$$

est minimale en $\hat{\mu} = \mu$.

Voir la solution

Exercice 3. En régression linéaire, on compare le critère $VC_{(n)}$ à l'erreur d'entraînement SSE/n.

a) Montrez que $VC_{(n)} \geq SSE/n$, avec égalité seulement si tous les résidus des observations de levier non nul sont nuls.
b) On remplace chaque h_i par leur moyenne $\bar{h} = (p+1)/n$ (c'est l'idée de la validation croisée généralisée, Chapitre 9). Montrez que le critère obtenu vaut

$$GCV = \frac{SSE/n}{\{1-(p+1)/n\}^2} \approx \frac{SSE}{n}\left(1 + \frac{2(p+1)}{n}\right)$$

lorsque $(p+1)/n$ est petit. Interprétez le facteur $1 + 2(p+1)/n$.

Voir la solution

5.4.3 Niveau difficile

Exercice 1. Le Chapitre 5 donne l'approximation de $VC_{(n)}$ pour un GLM avec lien canonique, en invoquant la formule de Sherman-Morrison : si J est inversible et $1 - v^\top J^{-1} u \neq 0$,

$$(J - uv^\top)^{-1} = J^{-1} + \frac{J^{-1}uv^\top J^{-1}}{1 - v^\top J^{-1} u}.$$

a) Démontrez cette formule en multipliant le membre de droite par $J - uv^\top$.
b) Avec $J = X^\top W X$, montrez que le i^{e} élément diagonal de $H = W^{1/2} X J^{-1} X^\top W^{1/2}$ est $h_i = w_i X_i^\top J^{-1} X_i$.
c) Déduisez-en que

$$\left(J - w_i X_i X_i^\top\right)^{-1} X_i = \frac{J^{-1} X_i}{1 - h_i},$$

l'égalité utilisée dans l'idée de la dérivation.

Voir la solution

Exercice 2. Considérez le modèle linéaire $Y = X\beta + \epsilon$, avec $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ et $p+1$ paramètres. Supposons qu'on observe de nouvelles réponses Y_i^0 aux mêmes valeurs X_i, indépendantes des Y_i et de même loi.

a) Montrez que l'espérance de l'erreur d'entraînement vaut $E[SSE/n] = \sigma^2\{1-(p+1)/n\}$.

b) Montrez que l'erreur de prédiction des nouvelles réponses vaut

$$E\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i^0 - \hat{Y}_i)^2\right] = \sigma^2\left(1 + \frac{p+1}{n}\right).$$

c) Déduisez-en que l'optimisme de l'erreur d'entraînement, c'est-à-dire la différence entre les deux espérances, vaut $2\sigma^2(p+1)/n$. Faites le lien avec l'exercice 3 du niveau moyen et avec le critère $C_p = SSE/n + 2\hat{\sigma}^2(p+1)/n$.

Voir la solution

5.4.4 Exercices appliqués

Exercice appliqué 1. Choisir un modèle de dépistage du diabète. Les jeux de données `Pima.tr` et `Pima.te` de la librairie `MASS` décrivent des femmes pimas d'au moins 21 ans, vivant près de Phoenix (Arizona), qui ont passé un test de dépistage du diabète. `Pima.tr` (200 femmes) sert d'échantillon d'entraînement et `Pima.te` (332 femmes) d'échantillon test. La variable `type` vaut `"Yes"` si la femme est diabétique selon les critères de l'Organisation mondiale de la santé. Les variables explicatives sont le nombre de grossesses (`npreg`), la glycémie lors d'un test de tolérance au glucose (`glu`), la pression diastolique (`bp`), l'épaisseur du pli cutané au triceps (`skin`), l'indice de masse corporelle (`bmi`), une fonction de prédisposition familiale au diabète (`ped`) et l'âge (`age`).

a) Construisez la réponse binaire `diabete` (1 si `type` vaut `"Yes"`, 0 sinon). Ajustez sur l'échantillon d'entraînement quatre modèles logistiques emboîtés : `M1` avec `glu` seulement ; `M2` avec `glu`, `bmi`, `ped` et `age` ; `M3` avec les sept variables ; et `M4` avec les sept variables et toutes leurs interactions d'ordre 2. Comparez leurs AIC et leur déviance d'entraînement par observation. Pourquoi cette dernière ne permet-elle pas de choisir le modèle ?

b) Calculez $VC_{(10)}$ pour chaque modèle avec la fonction `cv.glm` de la librairie `boot` et trois des pertes du Chapitre 5 : le taux de mauvaise classification au seuil 1/2, le score de Brier et la déviance. Utilisez la même partition en blocs pour tous les modèles (fixez la graine avec `set.seed()` juste avant chaque appel). Quel modèle chaque perte retient-elle ? Commentez le comportement de `M4`.

c) Refaites le calcul de $VC_{(10)}$ avec la déviance pour 20 partitions différentes (graines 1 à 20). Le choix du modèle est-il stable ?

d) Évaluez enfin les quatre modèles ajustés en a) sur l'échantillon test, avec les trois pertes, et comparez les résultats à ceux de la validation croisée. Pourquoi l'erreur test du modèle retenu estime-t-elle mieux son erreur de prédiction que le minimum de $VC_{(10)}$? Interprétez le modèle que vous retenez.

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R*
— James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R*
— Hastie, Tibshirani et Friedman (2009). *The Elements of Statistical Learning*

6 Sélection de variables et régularisation

Ce chapitre porte sur la régression régularisée (ridge et lasso). Cette approche ajoute une pénalité au critère d'ajustement pour réduire la variance des estimateurs, au prix d'un biais. Le chapitre commence par un rappel des outils de sélection de variables déjà vus, traite ensuite le cas linéaire, présente une sélection de variables fondée sur la décomposition spectrale de la matrice de corrélation, puis étend la régularisation à la régression logistique.

6.1 Sélection de variables : rappel

Deux familles d'outils pour choisir les variables d'un modèle ont été vues jusqu'ici :

- les tests pour modèles emboîtés : test F en régression linéaire, différence de déviances pour les GLM ;
- les critères d'information AIC et BIC, qu'on combine aux procédures pas à pas *forward*, *backward* et *stepwise*. Ils servent pour des modèles non emboîtés, ou quand le nombre de sous-modèles (2^p) est trop grand pour une recherche exhaustive.

La régression régularisée est une troisième approche, qui fait l'estimation et, dans le cas du lasso, la sélection de variables en même temps.

6.2 Régression régularisée

Considérons le modèle de régression

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2).$$

L'estimateur des moindres carrés $\widehat{\beta}$ est sans biais et, sous normalité des erreurs, de variance minimale parmi les estimateurs sans biais, mais rien ne garantit qu'il a la plus petite erreur quadratique moyenne :

$$E\left[(\beta_j - \widehat{\beta}_j)^2\right] = \text{biais}(\widehat{\beta}_j)^2 + Var[\widehat{\beta}_j].$$

La régression régularisée ajoute une pénalité au critère des moindres carrés : on minimise

$$Q_\lambda(\beta) = Q(\beta) + \lambda\, P(\beta_1, \ldots, \beta_p),$$

où $Q(\beta) = \sum_{i=1}^{n} \{Y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})\}^2$ est la somme des carrés résiduels, P est une fonction de pénalité connue et $\lambda \geq 0$ est le coefficient de pénalité. La pénalité réduit la variance des estimateurs, mais au prix d'un biais. Avec un bon choix de P et de λ, l'erreur quadratique moyenne diminue, ce qui rend les prédictions plus précises. C'est le compromis biais-variance.

Remarque

L'ordonnée à l'origine β_0 n'est pas pénalisée. En général, les variables explicatives sont standardisées avant l'ajustement, parce que la pénalité dépend de leur échelle. Par exemple, une variable mesurée en mètres plutôt qu'en kilomètres a un coefficient divisé par 1000, ce qui change sa pénalité.

Exemple — Jeu de données `Hitters`

Ces données décrivent les joueurs des ligues majeures de baseball. Elles viennent de la librairie `ISLR` (la librairie `ISLR2` de la deuxième édition de James et al. (2021) contient le même jeu), qui les a tirées de la collection StatLib de l'université Carnegie Mellon. À l'origine, elles ont été préparées pour une séance d'affiches de la section *Statistical Graphics* de l'American Statistical Association en 1988. Le jeu compte 322 joueurs et 20 variables, dont la réponse `Salary`, le salaire annuel au jour d'ouverture de la saison 1987, en milliers de dollars. Les 19 variables explicatives forment quatre groupes :

- statistiques offensives de la saison 1986 : présences au bâton (`AtBat`), coups sûrs (`Hits`), circuits (`HmRun`), points marqués (`Runs`), points produits (`RBI`) et buts sur balles (`Walks`) ;
- statistiques offensives cumulées sur la carrière : `CAtBat`, `CHits`, `CHmRun`, `CRuns`, `CRBI` et `CWalks`, plus le nombre d'années dans les ligues majeures (`Years`) ;
- statistiques défensives de la saison 1986 : retraits (`PutOuts`), assistances (`Assists`) et erreurs (`Errors`) ;
- variables catégorielles à deux modalités : la ligue à la fin de 1986 (`League`, `A` pour américaine ou `N` pour nationale), la division à la fin de 1986 (`Division`, `E` pour est ou `W` pour ouest) et la ligue au début de 1987 (`NewLeague`).

Comme le salaire est manquant pour 59 joueurs, on retire ces joueurs, ce qui laisse $n = 263$ observations. La fonction `model.matrix` transforme les trois variables catégorielles en indicatrices (`LeagueN`, `DivisionW` et `NewLeagueN`), d'où une matrice `x` à 19 colonnes. Cette matrice ne contient pas la colonne de 1 : `glmnet` ajoute lui-même l'ordonnée à l'origine.

```
library(ISLR)

dim(Hitters)
```

```
[1] 322  20
```

```r
sum(is.na(Hitters$Salary))
```

```
[1] 59
```

```r
Hitters <- na.omit(Hitters)
x <- model.matrix(Salary ~ ., Hitters)[, -1]
y <- Hitters$Salary

dim(x)
```

```
[1] 263  19
```

```r
summary(y)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   67.5   190.0   425.0   535.9   750.0  2460.0
```

```r
round(cor(Hitters[, c("Years", "CAtBat", "CHits", "CRuns", "CRBI")]), 2)
```

```
       Years CAtBat CHits CRuns CRBI
Years   1.00   0.92  0.90  0.88 0.86
CAtBat  0.92   1.00  1.00  0.98 0.95
CHits   0.90   1.00  1.00  0.98 0.95
CRuns   0.88   0.98  0.98  1.00 0.95
CRBI    0.86   0.95  0.95  0.95 1.00
```

Deux choses sont à retenir pour la suite. D'abord, les variables de carrière sont fortement corrélées entre elles, parce qu'elles croissent toutes avec l'ancienneté du joueur. Cette colinéarité gonfle la variance des estimateurs des moindres carrés, ce qui est justement la situation où la régularisation est utile. Ensuite, la distribution du salaire est asymétrique à droite : la moyenne dépasse nettement la médiane.

> **Remarque**
> À cause de cette asymétrie, on modélise souvent log(`Salary`) plutôt que `Salary`. Je garde ici le salaire sur son échelle d'origine, comme dans James et al. (2021), de sorte que les erreurs test des exemples sont comparables à celles de cet ouvrage.

6.2.1 Régression ridge

La régression ridge minimise

$$Q_\lambda^{(R)}(\beta) = \sum_{i=1}^{n} \left\{ Y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \right\}^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$

En écriture matricielle, avec la matrice de design X de dimension $n \times (p+1)$ dont la première colonne est formée de 1, $Q_\lambda^{(R)}(\beta) = (Y - X\beta)^\top (Y - X\beta) + \lambda \beta^\top \tilde{I}_{p+1} \beta$, où $\tilde{I}_{p+1} = \text{diag}(0, 1, \ldots, 1)$ est la matrice diagonale de taille $p+1$ qui a des 1 sur la diagonale, sauf un 0 en première position puisque β_0 n'est pas pénalisé. La solution est explicite :

$$\boxed{\hat{\beta}_\lambda^{(R)} = (X^\top X + \lambda \tilde{I}_{p+1})^{-1} X^\top Y}$$

dont découlent l'espérance et la variance :

$$E[\hat{\beta}_\lambda^{(R)}] = (X^\top X + \lambda \tilde{I}_{p+1})^{-1} X^\top X \beta,$$
$$Var[\hat{\beta}_\lambda^{(R)}] = \sigma^2 (X^\top X + \lambda \tilde{I}_{p+1})^{-1} X^\top X (X^\top X + \lambda \tilde{I}_{p+1})^{-1}.$$

L'estimateur est donc biaisé dès que $\lambda > 0$ (sauf dans le cas trivial où $\beta_1 = \cdots = \beta_p = 0$). Quand $\lambda = 0$, on retrouve l'estimateur des moindres carrés : $\hat{\beta}_\lambda^{(R)} = \hat{\beta}$. À l'autre extrême, quand $\lambda \to \infty$, tous les $\hat{\beta}_{j,\lambda}^{(R)}$ $(j \geq 1)$ tendent vers 0 et l'ordonnée à l'origine tend vers $\bar{Y}$. Ce phénomène s'appelle le rétrécissement (*shrinkage*).

Exemple — Régression ridge sur `Hitters`

On ajuste la régression ridge avec `glmnet(x, y, alpha = 0)` sur une grille de 100 valeurs de λ, de 10^{10} à 10^{-2}. La fonction `glmnet` standardise les variables explicatives avant l'ajustement (argument `standardize = TRUE` par défaut), mais elle retourne les coefficients sur l'échelle d'origine des variables.

```
library(glmnet)

grille.lambda <- 10^seq(10, -2, length.out = 100)
reg.ridge <- glmnet(x, y, alpha = 0, lambda = grille.lambda)
```

La valeur retenue, $\lambda = e^{5,44}$, est choisie par validation croisée (la méthode est présentée à la Section 6.2.5). Notez qu'elle s'entend au sens de `glmnet` (voir la remarque sur l'échelle de λ dans `glmnet`).

```
reg.mc <- lm(Salary ~ ., data = Hitters)
```

```r
coef.mc <- coef(reg.mc)[-1]
coef.ridge <- as.vector(coef(reg.ridge, s = lambda.ridge))[-1]

# Écarts types (diviseur n, comme glmnet), pour passer aux coefficients
↪  standardisés
sdx <- apply(x, 2, function(v) sqrt(mean((v - mean(v))^2)))

knitr::kable(
  data.frame(Variable = colnames(x),
             "Moindres carrés" = coef.mc,
             "Ridge" = coef.ridge,
             check.names = FALSE),
  digits = 2, row.names = FALSE
)
```

Variable	Moindres carrés	Ridge
AtBat	-1.98	0.04
Hits	7.50	0.98
HmRun	4.33	0.24
Runs	-2.38	1.10
RBI	-1.04	0.87
Walks	6.23	1.76
Years	-3.49	0.43
CAtBat	-0.17	0.01
CHits	0.13	0.06
CHmRun	-0.17	0.44
CRuns	1.45	0.13
CRBI	0.81	0.13
CWalks	-0.81	0.04
LeagueN	62.60	26.12
DivisionW	-116.85	-89.21
PutOuts	0.28	0.19
Assists	0.37	0.04
Errors	-3.36	-1.72
NewLeagueN	-24.76	7.76

La pénalité porte sur les coefficients des variables standardisées, $\beta_j s_j$, où s_j est l'écart type de la j^{e} variable explicative. La somme de leurs carrés (sans l'ordonnée à l'origine) vaut 17 099 pour ridge, contre 742 797 pour les moindres carrés. Sur l'échelle d'origine, la comparaison serait trompeuse : elle serait dominée par les trois indicatrices, dont les coefficients sont de loin les plus grands. Aucun coefficient n'est exactement nul : ridge rétrécit les coefficients, mais il n'en sélectionne aucun.
Il ne faut pas voir le rétrécissement comme une réduction de chaque coefficient pris séparément, car il porte sur le vecteur des coefficients standardisés pris dans son ensemble. Avec des variables corrélées comme les variables de carrière, un coefficient peut donc augmenter en valeur absolue, ou même changer de signe. En fait, ridge répartit l'effet entre les variables corrélées au lieu de laisser des coefficients de signes opposés se compenser.
Pour mesurer la précision de prédiction, on sépare les données en deux moitiés, l'une pour l'entraînement et l'autre pour le test. Pour que la comparaison soit honnête, on choisit λ par validation croisée sur l'échantillon d'entraînement seulement. Les observations test ne doivent servir ni à estimer les coefficients ni à choisir λ.

```
set.seed(1)
entrainement <- sample(nrow(x), nrow(x) / 2)
test <- -entrainement
y.test <- y[test]

# Modèle nul : on prédit par la moyenne de l'échantillon d'entraînement
err.nul <- mean((mean(y[entrainement]) - y.test)^2)

# Moindres carrés
mc.ent <- lm(y[entrainement] ~ x[entrainement, ])
err.mc <- mean((cbind(1, x[test, ]) %*% coef(mc.ent) - y.test)^2)

# Ridge, lambda choisi par validation croisée sur l'entraînement (voir
 ↪  plus bas)
set.seed(4300)
cv.ridge.ent <- cv.glmnet(x[entrainement, ], y[entrainement],
                          alpha = 0, lambda = grille.lambda)
ridge.ent <- glmnet(x[entrainement, ], y[entrainement],
                    alpha = 0, lambda = grille.lambda)
err.ridge <- mean((predict(ridge.ent, s = cv.ridge.ent$lambda.min,
                           newx = x[test, ]) - y.test)^2)

c("Modèle nul" = err.nul, "Moindres carrés" = err.mc, "Ridge" =
 ↪  err.ridge)
```

```
     Modèle nul Moindres carrés           Ridge
```

```
      224669.9        168593.3        140081.9
```

Sur l'échantillon d'entraînement, la validation croisée retient $\lambda = e^{5,72}$. L'erreur quadratique moyenne test vaut alors 168 593 pour les moindres carrés et 140 082 pour ridge, soit une diminution de 16,9 %. Comme point de référence, le modèle nul, qui ignore toutes les variables explicatives et prédit le même salaire à tout le monde, donne 224 670.

6.2.2 Régression lasso

La régression lasso minimise

$$Q_\lambda^{(L)}(\beta) = \sum_{i=1}^{n} \left\{ Y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \right\}^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

Il n'y a pas d'expression explicite pour $\widehat{\beta}_\lambda^{(L)}$, mais des algorithmes d'optimisation efficaces le calculent (en R : `glmnet(x, y, alpha = 1)`).

La propriété clé du lasso est la suivante : pour les grandes valeurs de λ, certains coefficients $\widehat{\beta}_{j,\lambda}^{(L)}$ valent exactement 0. Le lasso fait donc une sélection de variables automatique. Comme pour ridge, quand λ croît, le biais croît et la variance diminue.

Exemple — Lasso sur `Hitters`

On ajuste le lasso avec `glmnet(x, y, alpha = 1)` sur la même grille, et $\lambda = e^{0,98}$ est lui aussi choisi par validation croisée (Section 6.2.5).

```
reg.lasso <- glmnet(x, y, alpha = 1, lambda = grille.lambda)

coef.lasso <- as.vector(coef(reg.lasso, s = lambda.lasso))[-1]
names(coef.lasso) <- colnames(x)
round(coef.lasso[coef.lasso != 0], 2)
```

```
    AtBat      Hits     Walks     Years    CHmRun     CRuns      CRBI    CWalks
    -1.56      5.69      4.75     -9.52      0.52      0.66      0.39     -
0.53
  LeagueN DivisionW   PutOuts   Assists    Errors
    32.11   -119.26      0.27      0.17     -2.06
```

```
nuls <- paste0("`", colnames(x)[coef.lasso == 0], "`")
liste.nuls <- paste(paste(head(nuls, -1), collapse = ", "), "et",
 ↪  tail(nuls, 1))
```

Le lasso annule 6 coefficients, ceux de `HmRun`, `Runs`, `RBI`, `CAtBat`, `CHits` et `NewLeagueN`, de sorte que le modèle retenu ne garde que 13 variables. Notez que le lasso ne traite pas les variables fortement corrélées comme ridge. Parmi les statistiques de carrière, par exemple, il annule `CAtBat` et `CHits`, presque parfaitement corrélées à `CRuns` (corrélation de 0,98), alors que ridge répartit l'effet entre toutes ces variables.
On calcule l'erreur test sur la même partition que pour ridge, en choisissant encore une fois λ par validation croisée sur l'échantillon d'entraînement.

```
set.seed(4300)
cv.lasso.ent <- cv.glmnet(x[entrainement, ], y[entrainement],
                          alpha = 1, lambda = grille.lambda)
lasso.ent <- glmnet(x[entrainement, ], y[entrainement],
                    alpha = 1, lambda = grille.lambda)
err.lasso <- mean((predict(lasso.ent, s = cv.lasso.ent$lambda.min,
                           newx = x[test, ]) - y.test)^2)
err.lasso
```

```
[1] 144217.6
```

L'erreur test du lasso vaut 144 218, contre 140 082 pour ridge et 168 593 pour les moindres carrés.
La Figure 6.1 montre l'erreur test en fonction de $\log \lambda$ pour les deux méthodes. Quand λ est petit, les deux estimateurs sont proches des moindres carrés : la variance domine. Quand λ est grand, tous les coefficients sont nuls ou presque et l'erreur rejoint celle du modèle nul : le biais domine. Entre les deux, la courbe en U illustre le compromis biais-variance. Le lasso rejoint le modèle nul pour une valeur finie de λ, celle où tous ses coefficients s'annulent, alors que l'erreur de ridge ne s'en approche qu'asymptotiquement.

```
library(ggplot2)

courbe <- function(ajustement, methode) {
  data.frame(
    log.lambda = log(ajustement$lambda),
    erreur = colMeans((predict(ajustement, newx = x[test, ]) -
     ↪  y.test)^2),
    methode = methode
  )
}
erreurs <- rbind(courbe(ridge.ent, "Ridge"), courbe(lasso.ent, "Lasso"))
choix <- data.frame(
  methode = c("Ridge", "Lasso"),
```

```
  log.lambda = log(c(cv.ridge.ent$lambda.min, cv.lasso.ent$lambda.min))
)

ggplot(erreurs, aes(log.lambda, erreur, colour = methode)) +
  geom_line(linewidth = 0.8) +
  geom_vline(data = choix, aes(xintercept = log.lambda, colour =
   ↪  methode),
             linetype = "longdash") +
  geom_hline(yintercept = err.mc, linetype = "dashed") +
  geom_hline(yintercept = err.nul, linetype = "dotted") +
  coord_cartesian(xlim = c(-5, 12)) +
  labs(x = expression(log(lambda)), y = "Erreur quadratique moyenne
   ↪  test",
       colour = NULL) +
  theme_minimal(base_size = 13)
```

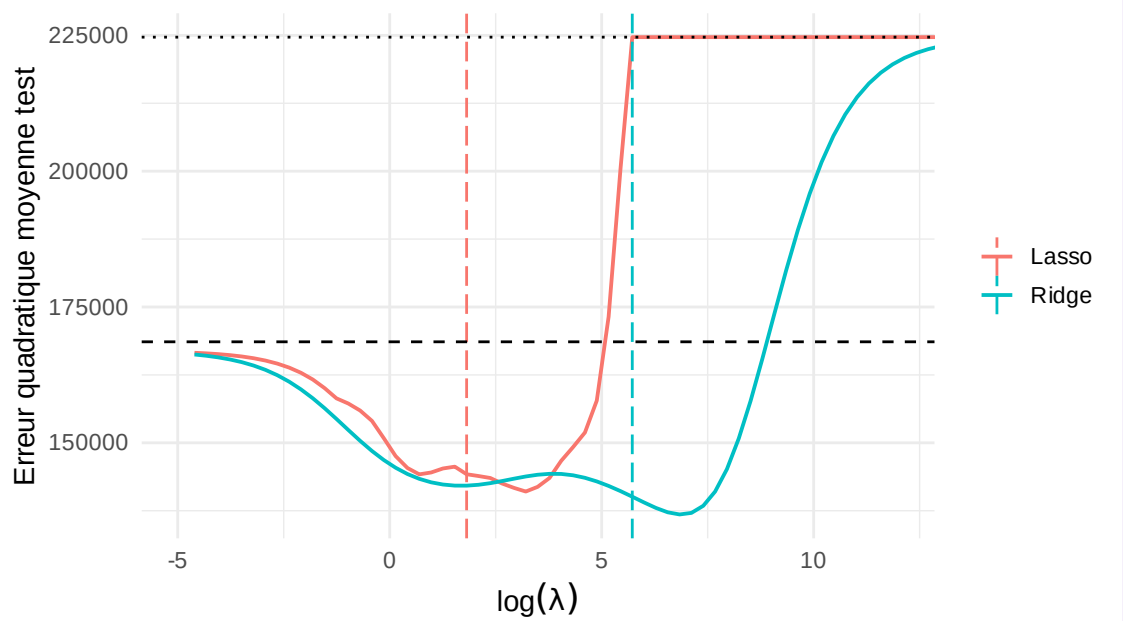


FIGURE 6.1 : Erreur quadratique moyenne test en fonction de $\log \lambda$ pour ridge et le lasso sur les données `Hitters`. Les lignes verticales indiquent les valeurs de λ choisies par validation croisée sur l'échantillon d'entraînement ; les lignes horizontales donnent l'erreur des moindres carrés (tirets) et celle du modèle nul (pointillés).

6.2.3 Interprétation géométrique

Propriété — Formulation sous contrainte

À chaque $\lambda \geq 0$ correspond une valeur $s \geq 0$ (qui dépend des données) telle que l'estimateur ridge est solution de

$$\min_{\beta} \sum_{i=1}^{n} \left\{ Y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right\}^2 \quad \text{sous la contrainte} \quad \sum_{j=1}^{p} \beta_j^2 \leq s,$$

et que l'estimateur lasso est solution du même problème sous la contrainte

$$\sum_{j=1}^{p} |\beta_j| \leq s.$$

Une grande valeur de λ correspond à une petite valeur de s.

Quand les variables explicatives sont centrées, la somme des carrés résiduels $Q(\beta)$ s'écrit, à une constante près, $(\beta - \widehat{\beta})^\top X^\top X(\beta - \widehat{\beta})$. Ses courbes de niveau sont donc des ellipses centrées en l'estimateur des moindres carrés $\widehat{\beta}$. Avec $p = 2$, on peut représenter le problème dans le plan (β_1, β_2) (Figure 6.2), où la région admissible est un disque pour ridge et un losange pour le lasso. La solution est le point où l'ellipse de plus petit niveau touche la région admissible.

Le losange a des coins sur les axes, et quand le contact se fait en un coin, un des coefficients est exactement nul. Le disque n'a pas de coin : le contact se fait presque toujours hors des axes, ce qui explique que ridge rétrécit les coefficients sans jamais les annuler. En dimension p, la région du lasso est un polyèdre dont les sommets et les arêtes sont sur des sous-espaces de coordonnées, et c'est pourquoi plusieurs coefficients peuvent s'annuler en même temps.

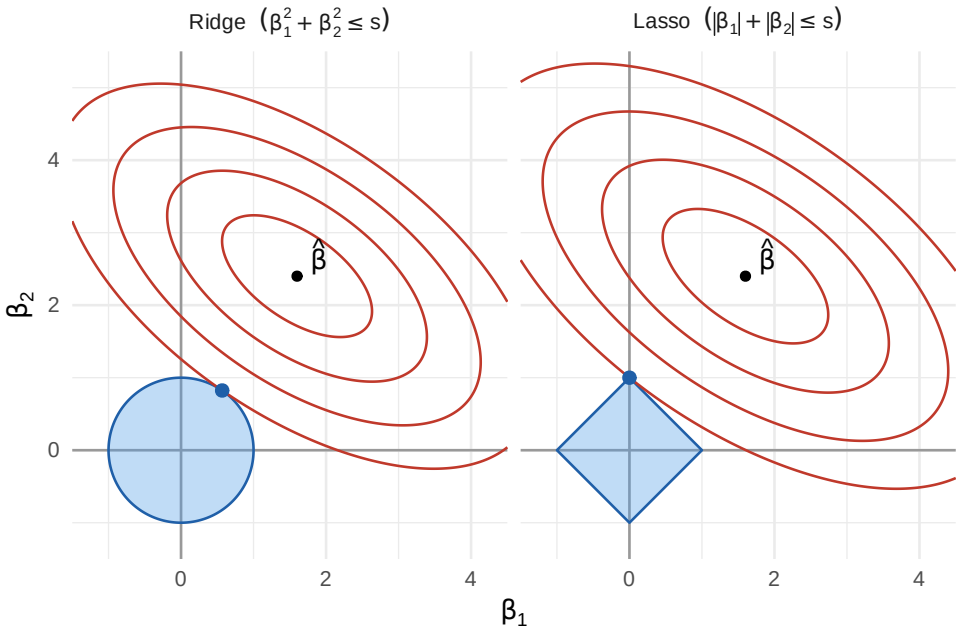


FIGURE 6.2 : Courbes de niveau de la somme des carrés résiduels (ellipses centrées en $\hat{\beta}$) et régions admissibles de ridge (disque) et du lasso (losange). Le point bleu est l'estimateur régularisé : le lasso atteint un coin du losange, où $\hat{\beta}_1 = 0$.

6.2.4 Trajectoires des coefficients

Une autre représentation usuelle trace les estimations $\hat{\beta}_{j,\lambda}$ en fonction de $\log \lambda$ (Figure 6.3). Pour ridge, toutes les trajectoires tendent vers 0 sans l'atteindre, de sorte que l'axe du haut, qui donne le nombre de coefficients non nuls, reste à 19. Les trajectoires du lasso, elles, atteignent 0 l'une après l'autre à mesure que λ augmente, et le nombre de variables retenues diminue en même temps.

```
library(glmnet)

grille.lambda <- 10^seq(10, -2, length.out = 100)

reg.ridge <- glmnet(x, y, alpha = 0, lambda = grille.lambda)
reg.lasso <- glmnet(x, y, alpha = 1, lambda = grille.lambda)

par(mfrow = c(1, 2))
plot(reg.ridge, xvar = "lambda")
title("Ridge", line = 2.5)
```

```r
plot(reg.lasso, xvar = "lambda")
title("Lasso", line = 2.5)
```

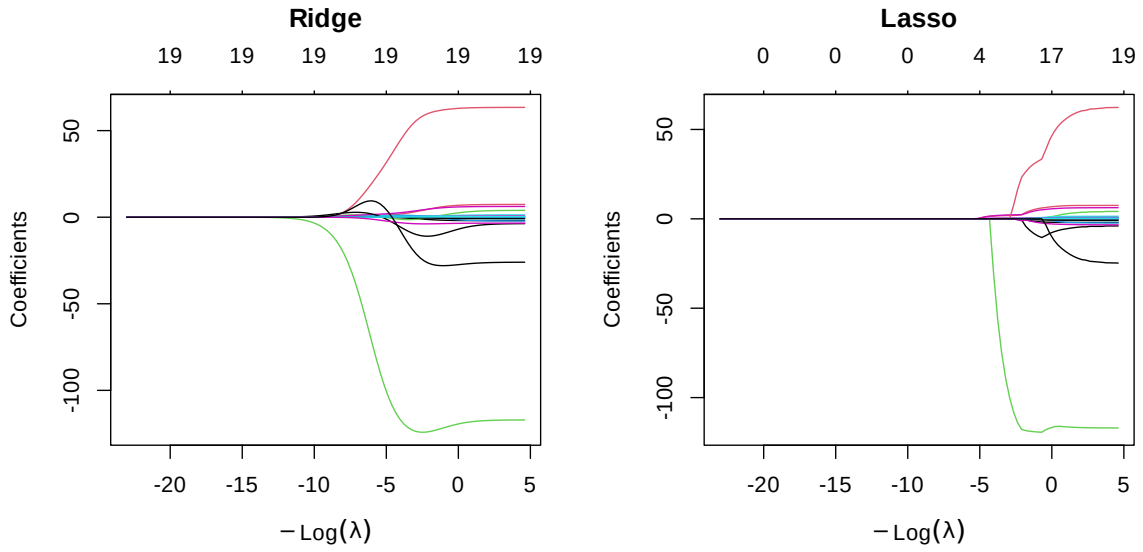


FIGURE 6.3 : Trajectoires des coefficients estimés en fonction de $\log \lambda$ pour les données `Hitters`. Chaque courbe correspond à une variable explicative ; l'axe du haut donne le nombre de coefficients non nuls.

6.2.5 Sélection du paramètre λ

On estime l'erreur test par validation croisée (Chapitre 5) pour une grille de valeurs de λ, puis on retient la valeur qui minimise cette erreur. C'est ainsi qu'ont été choisies les valeurs de λ des exemples précédents. En R, avec les objets `x`, `y` et `grille.lambda` définis plus haut :

```r
set.seed(4300)
cv.ridge <- cv.glmnet(x, y, alpha = 0, lambda = grille.lambda)
set.seed(4300)
cv.lasso <- cv.glmnet(x, y, alpha = 1, lambda = grille.lambda)

c(Ridge = cv.ridge$lambda.min, Lasso = cv.lasso$lambda.min)
```

```
     Ridge      Lasso
231.012970   2.656088
```

Sur les données `Hitters`, on obtient $\lambda = e^{5,44}$ pour ridge et $\lambda = e^{0,98}$ pour le lasso.

La fonction `cv.glmnet` retourne aussi `lambda.1se`, la plus grande valeur de λ dont l'erreur de validation croisée est à moins d'une erreur-type du minimum (la règle d'une erreur-type, détaillée plus bas à la section sur le choix de λ en régression logistique). Le tableau suivant compare les deux choix.

```
resume.lambda <- function(cv, methode) {
  i <- match(c(cv$lambda.min, cv$lambda.1se), cv$lambda)
  data.frame(
    "Méthode" = methode,
    "Règle" = c("lambda.min", "lambda.1se"),
    "log(lambda)" = log(cv$lambda[i]),
    "Erreur VC" = cv$cvm[i],
    "Erreur-type" = cv$cvsd[i],
    "Coefficients non nuls" = cv$nzero[i],
    check.names = FALSE
  )
}
tab.lambda <- rbind(resume.lambda(cv.ridge, "Ridge"), resume.lambda(cv.lasso,
 ↪  "Lasso"))
knitr::kable(tab.lambda, digits = 2, row.names = FALSE)
```

Méthode	Règle	log(lambda)	Erreur VC	Erreur-type	Coefficients non nuls
Ridge	lambda.min	5.44	118250.3	23182.04	19
Ridge	lambda.1se	7.95	137711.5	23344.85	19
Lasso	lambda.min	0.98	118435.6	22952.23	13
Lasso	lambda.1se	4.61	140563.7	22328.09	5

Pour le lasso, `lambda.1se` retient 5 variables au lieu de 13, au prix d'une erreur de validation croisée plus élevée de 18,7 %. Cette hausse est plus petite que l'erreur-type de l'erreur minimale, de sorte que les données ne permettent pas vraiment de distinguer les deux modèles. La règle d'une erreur-type préfère alors le plus simple. Avec ridge, aucun coefficient n'est exactement nul, et `lambda.1se` donne seulement un rétrécissement plus fort. Le choix entre les deux règles est donc un compromis : `lambda.min` vise la plus petite erreur estimée, `lambda.1se` un modèle plus parcimonieux, plus stable et plus facile à interpréter, pour une erreur comparable.

Notez que la validation croisée sert ici à choisir λ. L'erreur minimale qu'elle rapporte n'est pas une estimation indépendante de la performance du modèle retenu : comme λ a été choisi pour la minimiser, elle est optimiste. Pour estimer la performance finale, il faut des données qui n'ont servi ni à l'ajustement ni au choix de λ, comme l'échantillon test de l'exemple sur la régression ridge (voir aussi la remarque sur la fuite d'information au Chapitre 5).

6.2.6 Elastic net

Le lasso et ridge ont chacun un défaut quand des variables explicatives sont fortement corrélées. Le lasso tend à en retenir une seule, choisie de façon instable d'un échantillon à l'autre, alors que ridge les garde toutes, sans en sélectionner aucune. L'elastic net (Zou et Hastie 2005) combine les deux pénalités.

Définition — Elastic net

Pour $\lambda \geq 0$ et $\alpha \in [0,1]$, l'estimateur elastic net minimise

$$Q^{(EN)}_{\lambda,\alpha}(\beta) = \sum_{i=1}^{n} \left\{ Y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \right\}^2 + \lambda \left\{ \frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right\}.$$

Le cas $\alpha = 1$ donne le lasso, et le cas $\alpha = 0$ donne ridge (avec $\lambda/2$ à la place de λ).

La partie L_1 de la pénalité met certains coefficients exactement à 0, comme le lasso. La partie L_2 rend le critère strictement convexe et tend à donner des coefficients semblables à des variables corrélées, qui entrent ou sortent du modèle ensemble. C'est l'argument `alpha` de `glmnet`, avec la même convention que dans la remarque sur l'échelle de λ. En pratique, on choisit α dans une courte grille de valeurs, en gardant les mêmes blocs de validation croisée pour toutes les valeurs (argument `foldid`), puis λ par validation croisée pour chaque α.

Exemple — Elastic net sur `Hitters`

```
set.seed(4300)
blocs.hitters <- sample(rep(1:10, length.out = nrow(x)))
alphas <- c(0, 0.25, 0.5, 0.75, 1)
cv.en <- lapply(alphas, function(a) cv.glmnet(x, y, alpha = a, foldid =
 ↪  blocs.hitters))

tab.en <- data.frame(
  alpha = alphas,
  "log(lambda.min)" = sapply(cv.en, function(cv) log(cv$lambda.min)),
  "Erreur VC" = sapply(cv.en, function(cv) min(cv$cvm)),
  "Coefficients non nuls" = sapply(cv.en, function(cv)
   ↪  cv$nzero[match(cv$lambda.min, cv$lambda)]),
  check.names = FALSE
)
knitr::kable(tab.en, digits = 2, row.names = FALSE)
```

alpha	log(lambda.min)	Erreur VC	Coefficients non nuls
0.00	5.38	118244.0	19
0.25	4.23	118426.3	11
0.50	1.58	118578.2	14
0.75	1.27	118384.3	13
1.00	1.08	118444.5	13

Les erreurs de validation croisée sont proches d'une valeur de α à l'autre (l'écart entre la plus grande et la plus petite est d'environ 0,3 %), ce qui est fréquent. Le choix de α porte alors surtout sur la parcimonie : ridge ($\alpha = 0$) garde les 19 variables, alors que les autres valeurs de α en retiennent de 11 à 14. Comme λ est choisi séparément pour chaque α, ce nombre ne varie pas forcément de façon monotone avec α.

Remarque — Variables catégorielles : le group lasso

Une variable catégorielle à K modalités devient $K-1$ indicatrices dans `model.matrix`. Le lasso traite ces indicatrices séparément, de sorte qu'il peut en garder certaines et en annuler d'autres, ce qui découpe le facteur d'une façon qui dépend de la modalité de référence. Le group lasso (Yuan et Lin 2006) pénalise plutôt la norme de chaque groupe de coefficients, $\lambda \sum_g \sqrt{p_g}\, \|\beta_g\|_2$, où β_g regroupe les p_g coefficients du facteur g. Il retient ou retire donc un facteur en entier. Les librairies `grpreg` et `gglasso` l'implémentent ; ce livre ne le couvre pas davantage.

6.3 Sélection de variables par décomposition spectrale

Les méthodes vues jusqu'ici choisissent les variables selon leur capacité à expliquer la réponse. Une autre approche part d'un constat plus structurel. Quand certaines variables explicatives sont presque des combinaisons linéaires des autres, la matrice $X^\top X$ est mal conditionnée, les variances des estimateurs explosent et une partie de l'information est redondante. La décomposition spectrale de la matrice de corrélation des variables explicatives permet de repérer ces redondances, puis de retirer les variables qui en sont responsables.

6.3.1 Décomposition spectrale de la matrice de corrélation

Notons R la matrice de corrélation $p \times p$ des variables explicatives. Comme elle est symétrique et semi-définie positive, elle s'écrit

$$R = V \Lambda V^\top = \sum_{k=1}^{p} \lambda_k \, v_k v_k^\top,$$

où $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_p)$ est la matrice diagonale des valeurs propres de R, ordonnées de sorte que $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$, et où la k^{e} colonne $v_k = (v_{1k}, \dots, v_{pk})^\top$ de V est un vecteur propre normé associé à λ_k, c'est-à-dire $R v_k = \lambda_k v_k$. Les vecteurs propres sont orthonormés, de sorte que $V^\top V = V V^\top = I_p$. Enfin, comme la trace de R vaut p, la somme des valeurs propres vaut aussi p.

Soit z_i le vecteur des variables standardisées de l'observation i. La combinaison $v_k^\top z_i$ est la k^{e} composante principale, et sa variance empirique vaut λ_k. Une valeur propre proche de 0 signifie donc que

$$\sum_{j=1}^{p} v_{jk}\, z_{ij} \approx 0 \qquad \text{pour tout } i,$$

autrement dit qu'il existe une quasi-dépendance linéaire entre les variables dont les coefficients v_{jk} sont grands en valeur absolue.

Propriété — VIF et valeurs propres

Le facteur d'inflation de la variance de la variable j s'écrit

$$\mathrm{VIF}_j = \left(R^{-1}\right)_{jj} = \sum_{k=1}^{p} \frac{v_{jk}^2}{\lambda_k}.$$

L'identité découle directement de $R^{-1} = V \Lambda^{-1} V^\top$. Elle montre qu'un VIF élevé vient d'une ou de quelques petites valeurs propres, et que seules les variables fortement chargées sur les vecteurs propres correspondants en souffrent.

6.3.2 Indices de conditionnement et proportions de variance

Définition — Indices de conditionnement et proportions de décomposition de la variance

L'indice de conditionnement associé à la k^{e} valeur propre est

$$\kappa_k = \sqrt{\frac{\lambda_1}{\lambda_k}}, \qquad k = 1, \dots, p,$$

et le nombre de conditionnement de R est κ_p.
La proportion de la variance de $\hat{\beta}_j$ attribuable à la k^{e} composante est

$$\phi_{jk} = \frac{v_{jk}^2 / \lambda_k}{\mathrm{VIF}_j}, \qquad \sum_{k=1}^{p} \phi_{jk} = 1.$$

Belsley et al. (1980) proposent la lecture suivante. Un indice κ_k entre 10 et 30 signale une colinéarité modérée ; au-delà de 30, elle est forte. Lorsqu'au moins deux variables ont une proportion ϕ_{jk} supérieure à 0,5 sur une même composante d'indice élevé, ces variables sont impliquées dans la quasi-dépendance.

On en tire une règle de sélection itérative : tant que $\kappa_p > 30$, retirer la variable qui a la plus grande proportion ϕ_{jp} sur la dernière composante, puis recalculer la décomposition.

> **Remarque**
>
> Belsley, Kuh et Welsch travaillent en fait avec la matrice X non centrée (colonne de 1 incluse), dont les colonnes sont ramenées à une norme unitaire. La version présentée ici, fondée sur R, ignore les colinéarités qui impliquent l'ordonnée à l'origine. C'est la version la plus courante en pratique.

6.3.3 Méthodes de Jolliffe

Jolliffe (1972) et Jolliffe (1973) choisissent un sous-ensemble de variables directement à partir des vecteurs propres. Soit q le nombre de valeurs propres supérieures à un seuil λ_0 ; pour une matrice de corrélation, Jolliffe recommande $\lambda_0 = 0{,}7$.

- Méthode B2 (retrait) : pour $k = p, p-1, \dots, q+1$, retirer la variable non encore retirée dont le coefficient $|v_{jk}|$ est le plus grand. Les variables retirées sont celles qui portent les quasi-dépendances.
- Méthode B4 (rétention) : pour $k = 1, \dots, q$, retenir la variable non encore retenue dont le coefficient $|v_{jk}|$ est le plus grand. Chaque variable retenue représente une direction importante de variation.

Les deux méthodes conservent q variables.

> **Remarque — Des méthodes non supervisées**
>
> La variable réponse n'intervient jamais dans ces méthodes. Elles réduisent la redondance entre les variables explicatives, mais rien ne garantit que les variables retirées soient peu utiles pour prédire Y. Une variable de faible variance, donc associée à une petite valeur propre, peut très bien être la plus prédictive.

6.3.4 Lien avec ridge et la régression sur composantes principales

Soit Z la matrice $n \times p$ des variables centrées et standardisées, et $Z = UDV^\top$ sa décomposition en valeurs singulières. Comme $Z^\top Z = (n-1)R$, la matrice V est la même que plus haut et $d_k^2 = (n-1)\lambda_k$. Avec Y centré, les valeurs prédites de trois méthodes s'écrivent (voir l'exercice 1 du niveau difficile pour ridge)

$$\text{moindres carrés :} \quad \hat{Y} = \sum_{k=1}^{p} u_k u_k^\top Y,$$

$$\text{ridge :} \quad \hat{Y}_\lambda^{(R)} = \sum_{k=1}^{p} \frac{d_k^2}{d_k^2 + \lambda}\, u_k u_k^\top Y,$$

$$\text{composantes principales :} \quad \hat{Y}_q^{(CP)} = \sum_{k=1}^{q} u_k u_k^\top Y.$$

Ridge rétrécit surtout les directions de petite valeur propre, précisément celles que signalent les indices de conditionnement. La régression sur composantes principales, elle, les élimine complètement. Aucune des deux ne retire de variables, puisque chaque composante combine toutes les variables. Les règles de Belsley et de Jolliffe sont en ce sens la version « sélection de variables » de la même idée.

Exemple — Décomposition spectrale sur `Hitters`

On considère les 16 variables explicatives numériques (les trois variables catégorielles sont laissées de côté).

```
xnum <- Hitters[, sapply(Hitters, is.numeric)]
xnum <- xnum[, names(xnum) != "Salary"]
p.num <- ncol(xnum)

R.hit <- cor(xnum)
dec <- eigen(R.hit)
lambda.vp <- dec$values
V <- dec$vectors
rownames(V) <- colnames(xnum)

# Valeurs propres et indices de conditionnement
round(lambda.vp, 3)
```

```
 [1] 7.250 4.110 1.728 0.870 0.697 0.512 0.252 0.185 0.132 0.098 0.061 0.055
[13] 0.028 0.014 0.005 0.001
```

```
kappa.vp <- sqrt(lambda.vp[1] / lambda.vp)
round(kappa.vp, 1)
```

```
 [1]  1.0  1.3  2.0  2.9  3.2  3.8  5.4  6.3  7.4  8.6 10.9 11.5 16.0 22.4 38.5
[16] 77.7
```

```r
# VIF
vif <- diag(solve(R.hit))
round(sort(vif, decreasing = TRUE), 1)
```

```
  CHits  CAtBat   CRuns    CRBI  CHmRun    Hits   AtBat  CWalks    Runs     RBI
  495.7   250.1   158.7   131.2    46.3    30.1    22.5    19.7    15.1    11.7
  Years   HmRun   Walks Assists  Errors PutOuts
    9.1     7.6     4.1     2.7     2.2     1.2
```

```r
# Proportions de décomposition de la variance (trois dernières
 ↪  composantes)
phi <- sweep(V^2, 2, lambda.vp, "/")
phi <- phi / rowSums(phi)
round(phi[, (p.num - 2):p.num], 2)
```

```
        [,1] [,2] [,3]
AtBat   0.06 0.11 0.10
Hits    0.07 0.01 0.26
HmRun   0.01 0.10 0.07
Runs    0.12 0.03 0.20
RBI     0.05 0.08 0.03
Walks   0.04 0.00 0.07
Years   0.06 0.16 0.04
CAtBat  0.01 0.42 0.57
CHits   0.00 0.00 0.99
CHmRun  0.12 0.30 0.50
CRuns   0.17 0.20 0.61
CRBI    0.19 0.36 0.43
CWalks  0.11 0.00 0.33
PutOuts 0.01 0.00 0.00
Assists 0.00 0.01 0.03
Errors  0.00 0.00 0.01
```

Les trois plus petites valeurs propres valent 0,014, 0,005 et 0,001, et le nombre de conditionnement atteint $\kappa_{16} = 77,7$. Les VIF les plus élevés sont considérables : `CHits` (495,7), `CAtBat` (250,1), `CRuns` (158,7), `CRBI` (131,2). Sur la dernière composante, `CHits`, `CRuns` et `CAtBat` ont une proportion de variance supérieure à 0,5 (de 0,57 à 0,99). Les statistiques de carrière sont donc presque liées linéairement, ce qui n'a rien d'étonnant : les coups sûrs, les points et les présences au bâton s'accumulent ensemble au fil des saisons.

Exemple — Sélection par décomposition spectrale sur `Hitters`

Les trois fonctions suivantes appliquent la règle fondée sur κ_p et les méthodes B2 et B4 de Jolliffe. On compare ensuite les sous-ensembles retenus par l'erreur quadratique moyenne de validation croisée *leave-one-out*, calculée directement à partir des leviers (Chapitre 5).

```
retrait_conditionnement <- function(X, seuil = 30) {
  repeat {
    e <- eigen(cor(X))
    m <- ncol(X)
    if (sqrt(e$values[1] / e$values[m]) <= seuil) break
    phi <- sweep(e$vectors^2, 2, e$values, "/")
    phi <- phi / rowSums(phi)
    X <- X[, -which.max(phi[, m])]
  }
  colnames(X)
}

jolliffe_B2 <- function(V, lambda, lambda0 = 0.7) {
  retirees <- character(0)
  for (k in rev(which(lambda < lambda0))) {
    a <- abs(V[, k]); a[retirees] <- -Inf
    retirees <- c(retirees, names(which.max(a)))
  }
  setdiff(rownames(V), retirees)
}

jolliffe_B4 <- function(V, lambda, lambda0 = 0.7) {
  retenues <- character(0)
  for (k in which(lambda >= lambda0)) {
    a <- abs(V[, k]); a[retenues] <- -Inf
    retenues <- c(retenues, names(which.max(a)))
  }
  retenues
}

eqm_loo <- function(vars) {
  ajust <- lm(Salary ~ ., data = data.frame(Salary = y, xnum[, vars, drop
 ↪  = FALSE]))
  mean((residuals(ajust) / (1 - hatvalues(ajust)))^2)
}

ensembles <- list(
```

```
  complet         = colnames(xnum),
  conditionnement = retrait_conditionnement(xnum),
  B2              = jolliffe_B2(V, lambda.vp),
  B4              = jolliffe_B4(V, lambda.vp)
)
ensembles[-1]
```

```
$conditionnement
 [1] "AtBat"   "Hits"    "HmRun"   "Runs"    "RBI"     "Walks"   "Years"
 [8] "CHmRun"  "CRuns"   "CRBI"    "CWalks"  "PutOuts" "Assists" "Errors"

$B2
[1] "RBI"     "CRBI"    "PutOuts" "Errors"

$B4
[1] "CRBI"    "AtBat"   "Assists" "PutOuts"
```

```
eqm.spec <- sapply(ensembles, eqm_loo)
round(eqm.spec)
```

```
        complet conditionnement              B2              B4
         121272          118477          125684          124158
```

La règle fondée sur $\kappa_p > 30$ retire `CAtBat` et `CHits`, et le nombre de conditionnement tombe à 20,7. Seulement 4 valeurs propres dépassent $\lambda_0 = 0{,}7$ (la suivante vaut 0,697, juste sous le seuil), de sorte que les méthodes de Jolliffe ne conservent que 4 variables : `RBI`, `CRBI`, `PutOuts` et `Errors` pour B2, et `CRBI`, `AtBat`, `Assists` et `PutOuts` pour B4. L'erreur *leave-one-out* vaut 121 272 pour le modèle complet, 118 477 après le retrait ciblé, 125 684 pour B2 et 124 158 pour B4. Retirer seulement les variables redondantes améliore légèrement la prédiction. La sélection agressive de Jolliffe, qui ignore la réponse, la détériore. Notez aussi la sensibilité au seuil : avec une cinquième valeur propre à peine sous 0,7, un changement minime de λ_0 ajouterait une variable aux deux sous-ensembles.

6.4 Régularisation en régression logistique

Cette section étend ridge et le lasso aux GLM, avec la régression logistique comme cas type. La régularisation y sert à deux choses. D'abord, comme en régression linéaire, elle réduit la variance des estimateurs, ce qui est utile quand le nombre de variables explicatives est grand

par rapport à l'information disponible. En régression logistique, cette information dépend surtout du nombre d'événements du groupe le plus rare. Ensuite, elle règle le problème de la séparation (Section 2.2.2) : l'estimateur du maximum de vraisemblance n'existe pas en présence de séparation, mais l'estimateur régularisé existe toujours, pour $\lambda > 0$ (voir la propriété « Existence sous séparation » plus bas).

6.4.1 Log-vraisemblance pénalisée

On remplace le critère des moindres carrés par moins deux fois la log-vraisemblance, et on minimise

$$Q_\lambda(\beta) = -2\,\ell(\beta; y) + \lambda\, P(\beta_1, \dots, \beta_p),$$

avec $P(\beta_1, \dots, \beta_p) = \sum_{j=1}^{p} \beta_j^2$ pour ridge et $P(\beta_1, \dots, \beta_p) = \sum_{j=1}^{p} |\beta_j|$ pour le lasso. On note $Q_\lambda^{(R)}$ et $Q_\lambda^{(L)}$ ce critère avec la pénalité ridge et avec la pénalité lasso. Comme la déviance (Section 2.3.1) vaut $D(\beta) = 2\{\ell(y; y) - \ell(\beta; y)\}$, où $\ell(y; y)$ ne dépend pas de β, cela revient à minimiser la déviance pénalisée $D(\beta) + \lambda\, P(\beta_1, \dots, \beta_p)$.

> **Remarque**
> En régression linéaire normale, $-2\,\ell(\beta; y) = \sigma^{-2} \sum_{i=1}^{n} (Y_i - X_i^\top \beta)^2 + \text{constante}$. Le critère ci-dessus généralise donc exactement celui de la section précédente, la constante σ^2 étant absorbée dans λ.

> **Le cas du lien logit — Log-vraisemblance pénalisée**
> Avec le lien logit et $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$, le critère à minimiser s'écrit
>
> $$Q_\lambda(\beta) = -2 \sum_{i=1}^{n} \{y_i\, \eta_i - m_i \log(1 + e^{\eta_i})\} + \lambda\, P(\beta_1, \dots, \beta_p),$$
>
> où le terme $\log \binom{m_i}{y_i}$ est omis puisqu'il ne dépend pas de β (pour des données individuelles, $m_i = 1$).

6.4.2 Estimation

Contrairement au cas linéaire, l'estimateur ridge n'a pas de forme explicite, mais l'algorithme IRLS s'adapte directement.

Propriété — IRLS pénalisé (ridge)

Soit $\beta^{(t)}$ l'estimation courante, $\pi_i^{(t)}$ les probabilités correspondantes, $\mu_i^{(t)} = m_i\pi_i^{(t)}$, $W^{(t)} = \operatorname{diag}\left\{m_i\pi_i^{(t)}(1-\pi_i^{(t)})\right\}$ et $z^{(t)} = X\beta^{(t)} + (W^{(t)})^{-1}(y-\mu^{(t)})$ la réponse de travail (*working response*). Une itération de Newton-Raphson appliquée à $Q_\lambda^{(R)}$ donne

$$\beta^{(t+1)} = \left(X^\top W^{(t)} X + \lambda \tilde{I}_{p+1}\right)^{-1} X^\top W^{(t)} z^{(t)}.$$

C'est la formule de l'estimateur ridge linéaire, appliquée à la réponse de travail $z^{(t)}$ avec les poids $W^{(t)}$.

Dans le cas du lasso, la pénalité n'est pas dérivable en 0, de sorte que l'itération de Newton ne s'applique pas telle quelle. À chaque itération, la fonction `glmnet` remplace $-2\,\ell$ par son approximation quadratique (la même somme de carrés pondérés que dans IRLS, calculée avec la réponse de travail). Elle minimise ensuite par descente de coordonnées le problème lasso pondéré qui en résulte.

Propriété — Existence sous séparation

Supposons qu'on observe au moins un succès et au moins un échec. Pour tout $\lambda > 0$, le critère ridge $Q_\lambda^{(R)}$ a un unique minimum fini et le critère lasso $Q_\lambda^{(L)}$ a au moins un minimum fini, même en présence de séparation complète ou quasi complète.

L'argument est court. Comme $-2\,\ell(\beta; y) \geq 0$, la pénalité empêche $\beta_1, \ldots, \beta_p$ de tendre vers l'infini. Et comme on observe au moins un succès et au moins un échec, $-2\,\ell$ tend vers l'infini quand $|\beta_0| \to \infty$ avec les autres coefficients bornés. L'unicité, pour ridge, vient de ce que la matrice $X^\top W X + \lambda \tilde{I}_{p+1}$ est définie positive.

Remarque — Inférence

Les estimateurs régularisés sont biaisés, et la sélection faite par le lasso invalide les erreurs-types, valeurs-p et intervalles de confiance usuels calculés sur le modèle retenu. C'est pourquoi `glmnet` ne fournit pas d'erreurs-types. La régularisation est d'abord un outil de prédiction et de sélection, et l'inférence après sélection demande des méthodes propres à ce problème, que ce livre ne couvre pas.

Remarque — Échelle de λ dans `glmnet`

La fonction `glmnet` n'utilise pas la même échelle pour λ que les critères de ce chapitre.

Avec `family = "gaussian"` (valeur par défaut), elle minimise

$$\frac{1}{2n}\sum_{i=1}^{n}\left\{Y_i-(\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip})\right\}^2+\lambda\left\{\frac{1-\alpha}{2}\sum_{j=1}^{p}\beta_j^2+\alpha\sum_{j=1}^{p}|\beta_j|\right\},$$

et avec `family = "binomial"`, elle minimise

$$-\frac{1}{n}\,\ell(\beta;y)+\lambda\left\{\frac{1-\alpha}{2}\sum_{j=1}^{p}\beta_j^2+\alpha\sum_{j=1}^{p}|\beta_j|\right\}.$$

Dans les deux cas, la pénalité porte sur les coefficients des variables explicatives standardisées (avec `standardize = TRUE`, la valeur par défaut). Pour ridge avec `family = "gaussian"`, la réponse est aussi standardisée en interne, ce qui revient en plus à diviser λ par l'écart type de y. Les valeurs de λ rapportées par `glmnet` ne sont donc pas sur la même échelle que celles des critères $Q_\lambda^{(R)}$, $Q_\lambda^{(L)}$ et Q_λ, et seul l'ordre des valeurs compte pour l'interprétation.
En pratique, une valeur numérique de λ ne se compare donc pas d'une source à l'autre (une formule théorique, `glmnet`, un autre logiciel, ou même `glmnet` avec `standardize = FALSE`) sans tenir compte de la convention de chacune : facteur $1/(2n)$ ou $1/n$ devant la perte, pénalité sur les coefficients standardisés ou non. Dans toutes ces conventions, l'ordonnée à l'origine β_0 n'est pas pénalisée.

6.4.3 Choix de λ

Comme en régression linéaire, on choisit λ par validation croisée (Chapitre 5), mais avec une fonction de perte adaptée à une réponse binaire. Les choix possibles sont la déviance binomiale (`type.measure = "deviance"`, valeur par défaut), le taux de mauvaise classification (`"class"`) ou l'aire sous la courbe ROC (`"auc"`). La fonction `cv.glmnet` retourne deux valeurs :

- `lambda.min` : la valeur qui minimise l'erreur de validation croisée ;
- `lambda.1se` : la plus grande valeur de λ dont l'erreur est à moins d'une erreur-type de ce minimum. Elle donne un modèle plus régularisé (plus parcimonieux dans le cas du lasso) dont l'erreur estimée reste comparable à l'erreur minimale, compte tenu de l'incertitude de la validation croisée. Il s'agit d'une règle heuristique (la règle d'une erreur-type), pas d'un test.

Exemple — Régularisation sur `empress_individus`

Le modèle ajusté contient toutes les interactions entre la classe, le sexe et le groupe d'âge, ce qui le rend saturé : il compte 11 paramètres estimables pour 11 cellules non vides. Deux cellules ne comptent aucun survivant : les garçons de deuxième classe (0 sur 11) et les

filles de troisième classe (0 sur 48). Il y a donc séparation quasi complète et l'estimateur du maximum de vraisemblance n'existe pas. De plus, la cellule « première classe, homme, enfant » est vide, si bien qu'une colonne de la matrice de design est redondante. R l'écarte : c'est le coefficient de `classeTroisième:sexeHomme:groupe_ageEnfant`, qui vaut `NA` dans la table des coefficients plus bas et que `summary` n'affiche pas.

```
niveaux.classe <- c("Première", "Deuxième", "Troisième")

empress <- read.csv("Jeux de données/empress_individus.csv") |>
  dplyr::mutate(classe = factor(classe, levels = niveaux.classe))

empress.agr <- read.csv("Jeux de données/empress_agrege.csv") |>
  dplyr::mutate(classe = factor(classe, levels = niveaux.classe))

m.sature <- glm(survecu ~ classe * sexe * groupe_age,
                data = empress, family = binomial)
round(summary(m.sature)$coefficients, 2)
```

```
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                                -0.74       0.37   -
2.01     0.04
classeDeuxième                             -1.24       0.47   -
2.63     0.01
classeTroisième                            -1.45       0.45   -
3.25     0.00
sexeHomme                                   0.70       0.46    1.50     0.13
groupe_ageEnfant                           -0.36       1.21   -
0.30     0.77
classeDeuxième:sexeHomme                    0.38       0.59    0.65     0.51
classeTroisième:sexeHomme                   0.44       0.54    0.81     0.42
classeDeuxième:groupe_ageEnfant             0.09       1.45    0.06     0.95
classeTroisième:groupe_ageEnfant          -15.01     571.03   -
0.03     0.98
sexeHomme:groupe_ageEnfant                 12.46     571.03    0.02     0.98
classeDeuxième:sexeHomme:groupe_ageEnfant -28.86    1322.47   -
0.02     0.98
```

Les coefficients des cellules sans survivant sont grands en valeur absolue, avec des erreurs-types démesurées. On ajuste ensuite les versions ridge et lasso, en choisissant λ par validation croisée à 10 blocs sur la déviance. Pour que la comparaison soit juste, ridge et le lasso utilisent les mêmes blocs (argument `foldid`).

```r
x.emp <- model.matrix(survecu ~ classe * sexe * groupe_age, empress)[,
 ↪  -1]
y.emp <- empress$survecu

# Mêmes 10 blocs pour ridge et le lasso
set.seed(4300)
blocs <- sample(rep(1:10, length.out = nrow(x.emp)))
cv.ridge.emp <- cv.glmnet(x.emp, y.emp, family = "binomial", alpha = 0,
                          foldid = blocs)
cv.lasso.emp <- cv.glmnet(x.emp, y.emp, family = "binomial", alpha = 1,
                          foldid = blocs)

par(mfrow = c(1, 2))
plot(cv.ridge.emp)
title("Ridge", line = 2.5)
plot(cv.lasso.emp)
title("Lasso", line = 2.5)
```

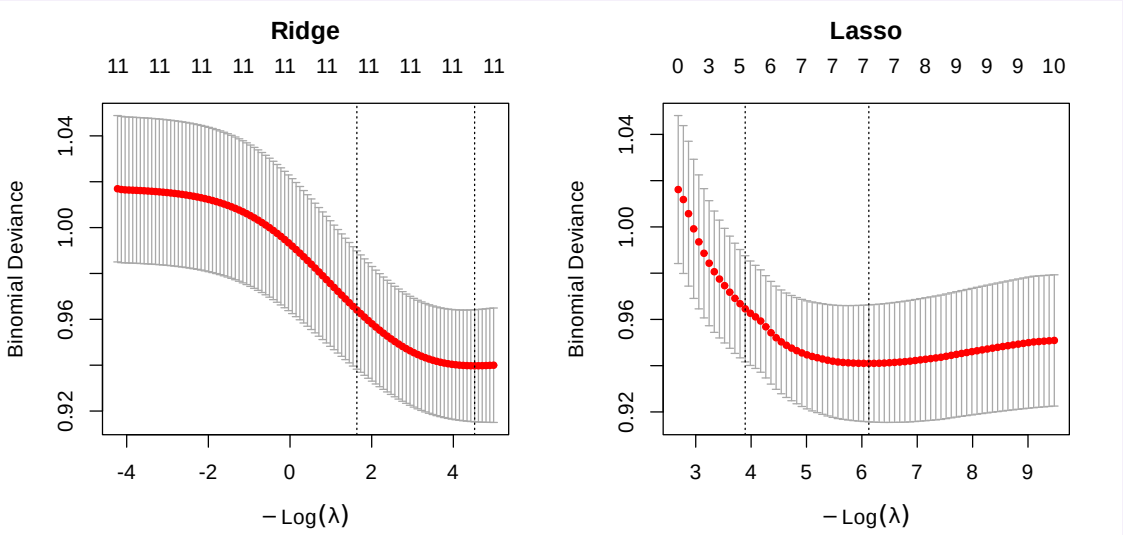


FIGURE 6.4 : Déviance binomiale estimée par validation croisée en fonction de $\log \lambda$ pour ridge (gauche) et le lasso (droite). Les lignes pointillées indiquent `lambda.min` et `lambda.1se`.

Les coefficients régularisés restent finis :

```r
coefs <- cbind(
```

```
  MV    = coef(m.sature),
  Ridge = as.vector(coef(cv.ridge.emp, s = "lambda.min")),
  Lasso = as.vector(coef(cv.lasso.emp, s = "lambda.min"))
)
round(coefs, 3)
```

```
                                                MV  Ridge  Lasso
(Intercept)                                 -0.738 -1.038 -1.021
classeDeuxième                              -1.241 -0.779 -0.875
classeTroisième                             -1.453 -1.028 -1.055
sexeHomme                                    0.697  0.853  0.987
groupe_ageEnfant                            -0.361 -0.617 -0.333
classeDeuxième:sexeHomme                     0.384  0.054  0.000
classeTroisième:sexeHomme                    0.437  0.148  0.015
classeDeuxième:groupe_ageEnfant              0.088  0.048  0.000
classeTroisième:groupe_ageEnfant           -15.014 -1.175 -2.324
sexeHomme:groupe_ageEnfant                  12.462 -0.606  0.000
classeDeuxième:sexeHomme:groupe_ageEnfant  -28.857 -1.824 -2.584
classeTroisième:sexeHomme:groupe_ageEnfant      NA -0.334  0.000
```

On compare enfin les probabilités de survie estimées dans chaque cellule aux proportions observées, calculées à partir de la version agrégée `empress_agrege`.

```
x.agr <- model.matrix(~ classe * sexe * groupe_age, empress.agr)[, -1]

probas <- data.frame(
  empress.agr[, c("classe", "sexe", "groupe_age")],
  Observee = empress.agr$survivants / empress.agr$effectif,
  MV    = predict(m.sature, newdata = empress.agr, type = "response"),
  Ridge = as.vector(predict(cv.ridge.emp, newx = x.agr,
                            s = "lambda.min", type = "response")),
  Lasso = as.vector(predict(cv.lasso.emp, newx = x.agr,
                            s = "lambda.min", type = "response"))
)
knitr::kable(probas, digits = 3, row.names = FALSE)
```

classe	sexe	groupe_age	Observee	MV	Ridge	Lasso
Première	Homme	Adulte	0.490	0.490	0.454	0.492
Première	Femme	Adulte	0.324	0.324	0.262	0.265

Première	Femme	Enfant	0.250	0.250	0.160	0.205
Deuxième	Homme	Adulte	0.289	0.289	0.287	0.287
Deuxième	Homme	Enfant	0.000	0.000	0.020	0.021
Deuxième	Femme	Adulte	0.121	0.121	0.140	0.131
Deuxième	Femme	Enfant	0.095	0.095	0.084	0.097
Troisième	Homme	Adulte	0.258	0.258	0.256	0.255
Troisième	Homme	Enfant	0.019	0.019	0.022	0.023
Troisième	Femme	Adulte	0.101	0.101	0.112	0.111
Troisième	Femme	Enfant	0.000	0.000	0.021	0.009

Pour les 9 cellules qui comptent au moins un survivant, l'ajustement de `glm` reproduit les proportions observées, puisque le modèle est saturé. Dans les deux cellules sans survivant, les probabilités ajustées ne sont pas nulles, mais de l'ordre de $2{,}4 \times 10^{-8}$ (la table les arrondit à 0) : elles tendent vers 0 au fil des itérations, puisque l'estimateur du maximum de vraisemblance n'existe pas. Le modèle prédit donc une survie pratiquement impossible dans ces cellules, ce qui est peu plausible vu les effectifs. Les estimateurs régularisés rapprochent les probabilités des cellules peu informatives de celles des cellules voisines : dans les deux cellules sans survivant, ils donnent des probabilités petites (de 0,009 à 0,021), mais strictement positives.

Remarque — À retenir

- Compromis biais-variance : la pénalité $\lambda\, P(\beta_1, \dots, \beta_p)$ réduit la variance des estimateurs au prix d'un biais, ce qui peut diminuer l'erreur quadratique moyenne. L'ordonnée à l'origine n'est pas pénalisée, et les variables sont standardisées avant l'ajustement, parce que la pénalité dépend de leur échelle (`glmnet` le fait par défaut et retourne les coefficients sur l'échelle d'origine) ;
- Ridge ou lasso : ridge ($\lambda \sum_{j=1}^{p} \beta_j^2$, `alpha = 0`) rétrécit les coefficients sans en annuler aucun et répartit l'effet entre variables corrélées, alors que le lasso ($\lambda \sum_{j=1}^{p} |\beta_j|$, `alpha = 1`) en annule certains et fait une sélection de variables. L'elastic net combine les deux pénalités ;
- Choix de λ : `cv.glmnet` retourne `lambda.min`, qui minimise l'erreur de validation croisée, et `lambda.1se`, la plus grande valeur dont l'erreur est à moins d'une erreur-type de ce minimum (un modèle plus régularisé). L'erreur minimale est optimiste, et la performance finale s'estime sur des données qui n'ont servi ni à l'ajustement ni au choix de λ ;
- Régression logistique : le critère devient la déviance pénalisée (`family = "binomial"`), et l'estimateur régularisé existe pour tout $\lambda > 0$, même sous séparation complète ou quasi complète, pourvu qu'il y ait au moins un succès et un échec ;

— Inférence : les estimateurs régularisés sont biaisés et la sélection du lasso invalide les erreurs-types et les valeurs-p usuelles, de sorte que `glmnet` n'en fournit pas. Notez aussi que les valeurs de λ de `glmnet` ne sont pas sur la même échelle que celles des critères du chapitre.

6.5 Exercices

Les exercices sont classés par niveau de difficulté, et les exercices appliqués de la fin demandent d'analyser un jeu de données en R. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

Dans plusieurs exercices de cette section, on suppose que les variables explicatives sont centrées et que la réponse l'est aussi, de sorte qu'on peut omettre l'ordonnée à l'origine. On note alors X la matrice $n \times p$ des variables explicatives (sans colonne de 1) et $\hat{\beta} = (X^\top X)^{-1} X^\top Y$ l'estimateur des moindres carrés. Le plan est dit orthonormé si $X^\top X = I_p$.

6.5.1 Niveau facile

Exercice 1. En annulant le gradient du critère

$$Q_\lambda^{(R)}(\beta) = (Y - X\beta)^\top (Y - X\beta) + \lambda\, \beta^\top \beta,$$

retrouvez l'estimateur ridge $\hat{\beta}_\lambda^{(R)} = (X^\top X + \lambda I_p)^{-1} X^\top Y$. Pourquoi la matrice $X^\top X + \lambda I_p$ est-elle toujours inversible lorsque $\lambda > 0$, même si $p > n$?

Voir la solution

Exercice 2. Supposez le plan orthonormé ($X^\top X = I_p$). Montrez que $\hat{\beta}_\lambda^{(R)} = \hat{\beta}/(1+\lambda)$. Interprétez ce résultat en termes de rétrécissement.

Voir la solution

Exercice 3. Considérez la régression ridge à une variable explicative centrée x, sans ordonnée à l'origine, de sorte que $\hat{\beta}_\lambda^{(R)} = \sum_i x_i Y_i / (\sum_i x_i^2 + \lambda)$.

a) On mesure la variable dans une autre unité, $\tilde{x}_i = c\, x_i$ avec $c > 0$. Calculez le nouvel estimateur ridge $\tilde{\beta}_\lambda^{(R)}$ et comparez-le à $\hat{\beta}_\lambda^{(R)}/c$.
b) Quelle propriété de l'estimateur des moindres carrés ($\lambda = 0$) perd-on ? Expliquez pourquoi on standardise les variables explicatives avant d'appliquer ridge ou le lasso.

Voir la solution

6.5.2 Niveau moyen

Exercice 1. Supposez le plan orthonormé. Montrez que l'estimateur lasso, qui minimise

$$Q_\lambda^{(L)}(\beta) = \|Y - X\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

s'obtient coordonnée par coordonnée par seuillage doux :

$$\hat{\beta}_{j,\lambda}^{(L)} = \text{signe}(\hat{\beta}_j) \left(|\hat{\beta}_j| - \frac{\lambda}{2} \right)_+ .$$

(Indice : montrez d'abord que $\|Y - X\beta\|^2 = \|Y - X\hat{\beta}\|^2 + \|\beta - \hat{\beta}\|^2$.) Comparez avec ridge (exercice 2 du niveau facile) et expliquez pourquoi seul le lasso sélectionne des variables.

Voir la solution

Exercice 2. Supposez le plan orthonormé, de sorte que $\hat{\beta}_j \sim \mathcal{N}(\beta_j, \sigma^2)$. Considérez l'estimateur ridge $\hat{\beta}_j/(1+\lambda)$ d'un coefficient.

a) Calculez son biais, sa variance et son erreur quadratique moyenne $EQM(\lambda)$.
b) Montrez que $EQM(\lambda)$ est minimale en $\lambda^* = \sigma^2/\beta_j^2$, et que $EQM(\lambda^*) = \sigma^2\beta_j^2/(\beta_j^2+\sigma^2) < \sigma^2 = EQM(0)$.
c) Interprétez λ^*. Pourquoi ce résultat ne permet-il pas de choisir λ en pratique ?

Voir la solution

Exercice 3. Avec l'ordonnée à l'origine non pénalisée et la matrice $\tilde{I}_{p+1} = \text{diag}(0, 1, \ldots, 1)$ du Chapitre 6, considérez les données augmentées

$$\tilde{Y} = \begin{pmatrix} Y \\ 0_p \end{pmatrix}, \qquad \tilde{X} = \begin{pmatrix} X \\ \sqrt{\lambda}\, E \end{pmatrix},$$

où X est la matrice de design $n \times (p+1)$ et $E = (0_p \;\; I_p)$ est la matrice $p \times (p+1)$ obtenue en retirant la première ligne de I_{p+1}. Montrez que les moindres carrés ordinaires sur $(\tilde{X}, \tilde{Y})$ donnent l'estimateur ridge. Interprétez les p observations ajoutées.

Voir la solution

6.5.3 Niveau difficile

Exercice 1. Soit $X = UDV^\top$ la décomposition en valeurs singulières de la matrice $n \times p$ des variables explicatives centrées (supposée de rang p), où U est $n \times p$ avec $U^\top U = I_p$, V est $p \times p$ orthogonale et $D = \text{diag}(d_1, \ldots, d_p)$, avec $d_1 \geq \cdots \geq d_p > 0$.

a) Montrez que

$$\hat{\beta}_\lambda^{(R)} = \sum_{j=1}^{p} \frac{d_j}{d_j^2 + \lambda}\, v_j u_j^\top Y \qquad \text{et} \qquad X\hat{\beta}_\lambda^{(R)} = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}\, u_j u_j^\top Y,$$

où u_j et v_j sont les j^{es} colonnes de U et V.

b) Déduisez-en que les degrés de liberté effectifs de ridge valent $\mathrm{Tr}(H_\lambda) = \sum_j d_j^2/(d_j^2+\lambda)$, qui décroissent de p à 0 quand λ croît de 0 à l'infini.

c) Montrez que $\|\hat{\beta}_\lambda^{(R)}\|^2$ est une fonction décroissante de λ.

d) Quelles directions sont le plus rétrécies ? Faites le lien avec les variables de carrière colinéaires des données `Hitters`.

Voir la solution

Exercice 2. Démontrez la propriété « IRLS pénalisé (ridge) » du Chapitre 6. Pour le critère $Q_\lambda^{(R)}(\beta) = -2\,\ell(\beta; y) + \lambda\,\beta^\top \tilde{I}_{p+1}\beta$ avec le lien logit :

a) calculez le gradient et la matrice hessienne de $Q_\lambda^{(R)}$, à l'aide des exercices 1 et 2 du niveau moyen du chapitre 2 ;

b) montrez qu'une itération de Newton-Raphson à partir de $\beta^{(t)}$ donne

$$\beta^{(t+1)} = \left(X^\top W^{(t)} X + \lambda \tilde{I}_{p+1}\right)^{-1} X^\top W^{(t)} z^{(t)}, \qquad z^{(t)} = X\beta^{(t)} + (W^{(t)})^{-1}(y - \mu^{(t)}).$$

Voir la solution

Exercice 3. Supposez que les deux premières colonnes de X sont identiques ($x_1 = x_2 \neq 0$), les autres variables étant quelconques.

a) Montrez que, pour tout $\lambda > 0$, l'estimateur ridge vérifie $\hat{\beta}_{1,\lambda}^{(R)} = \hat{\beta}_{2,\lambda}^{(R)}$. (Indice : le minimum de $Q_\lambda^{(R)}$ est unique.)

b) Montrez que si $(\hat{\beta}_1, \hat{\beta}_2, \ldots)$ minimise le critère lasso avec $\hat{\beta}_1 + \hat{\beta}_2 = s \neq 0$, alors tout vecteur obtenu en remplaçant $(\hat{\beta}_1, \hat{\beta}_2)$ par $(a, s-a)$, avec a entre 0 et s, le minimise aussi.

c) Comparez avec la discussion de `CHits` et `CRuns` dans l'exemple sur `Hitters`.

Voir la solution

6.5.4 Exercices appliqués

Exercice appliqué 1. Prédire le pourcentage de gras corporel. Le jeu de données `fat` de la librairie `faraway` (Faraway 2025) décrit 252 hommes. Leur pourcentage de gras corporel, `brozek`, est calculé par l'équation de Brozek à partir de la densité du corps, mesurée par pesée hydrostatique, une méthode précise mais lourde. Les autres variables sont faciles à mesurer : l'âge (`age`, en années), le poids (`weight`, en livres), la taille (`height`, en pouces) et dix circonférences en centimètres (cou, poitrine, abdomen, hanches, cuisse, genou, cheville, biceps, avant-bras et poignet). On veut prédire `brozek` à partir de ces mesures.

a) Repérez les deux observations impossibles (une taille et un pourcentage de gras invraisemblables) et retirez-les. Retirez aussi `siri`, `density` et `free`, qui sont calculées à partir de la pesée hydrostatique, ainsi que `adipos`, calculée à partir du poids et de la taille. Ajustez par les moindres carrés la régression de `brozek` sur les 13 variables restantes. Examinez les corrélations entre le poids et les circonférences du tronc et des cuisses, puis les coefficients estimés et leurs valeurs-p. Que constatez-vous ?

b) Séparez les données au hasard en un échantillon d'entraînement de 166 hommes et un échantillon test de 84 hommes. Sur l'échantillon d'entraînement, choisissez λ par validation croisée à 10 blocs pour ridge et pour le lasso avec la fonction `cv.glmnet`, sur la grille `lambda = 10^seq(2, -4, length.out = 100)` et avec les mêmes blocs pour les deux méthodes (argument `foldid`). Comparez les erreurs quadratiques moyennes sur l'échantillon test du modèle nul, des moindres carrés, de ridge et du lasso (avec `lambda.min` et `lambda.1se`). Où se situe `lambda.min` dans la grille ? Commentez.

c) Quelles variables le lasso retient-il avec `lambda.1se` ? Interprétez leurs coefficients. Peut-on conclure que les variables écartées n'ont pas d'effet sur le pourcentage de gras ?

d) Reprenez b) en ajoutant toutes les interactions d'ordre 2 (formule `brozek ~ .^2`). Combien de variables obtenez-vous ? Comparez les erreurs test et tracez les courbes de validation croisée de ridge et du lasso en fonction de $\log \lambda$. Qu'est-ce qui change par rapport à b) ? Quelle conclusion pratique tirez-vous des deux analyses ?

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R*
— James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R*
— Hastie, Tibshirani et Friedman (2009). *The Elements of Statistical Learning*
— Hastie, Tibshirani et Wainwright (2015). *Statistical Learning with Sparsity : The Lasso and Generalizations*

7 Régression non paramétrique : k plus proches voisins et estimateur à noyau

Ce chapitre utilise les librairies R suivantes.

```
library(FNN)
library(ggplot2)
library(dplyr)
```

Ce chapitre présente deux estimateurs non paramétriques d'une fonction de régression, les k plus proches voisins et l'estimateur à noyau, qui ne postulent ni l'un ni l'autre de forme fonctionnelle. Les exemples reprennent le jeu de données `arbres` du Chapitre 5. La réponse Y est la hauteur de l'arbre (`height_m`, en m) et la variable explicative X est son diamètre à hauteur de poitrine (`diameter_cm`, en cm). Le nuage de points montre que la relation entre X et Y n'est pas linéaire : la hauteur croît rapidement avec le diamètre pour les petits arbres, puis de plus en plus lentement. Au Chapitre 5, on avait contourné le problème en passant à l'échelle log-log. Ici, on laisse plutôt les données donner sa forme à la courbe.

```
arbres <- read.csv("Jeux de données/arbres.csv")
D <- arbres$diameter_cm
H <- arbres$height_m
n <- length(D)

ggplot(arbres, aes(diameter_cm, height_m)) +
  geom_point(alpha = 0.6) +
  labs(x = "Diamètre (cm)", y = "Hauteur (m)") +
  theme_minimal(base_size = 13)
```

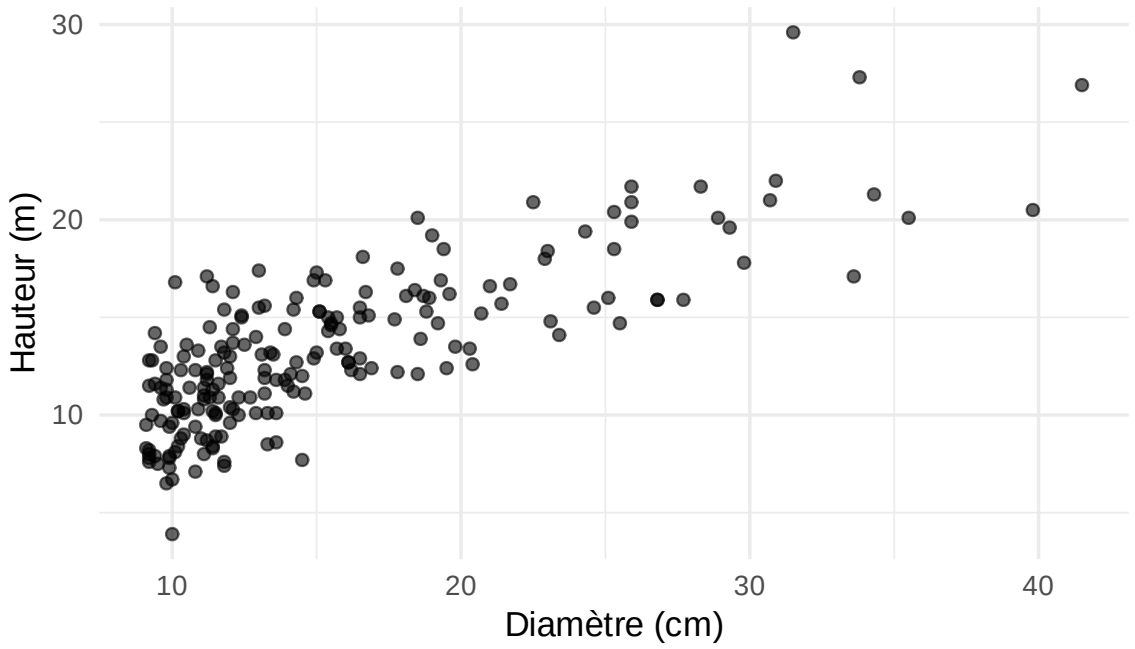


FIGURE 7.1 : Hauteur en fonction du diamètre pour les 200 arbres du jeu de données `arbres`.

7.1 Estimateur des k plus proches voisins (kNN)

7.1.1 Définition

On suppose que la relation entre X et Y s'écrit

$$Y_i = f(X_i) + \epsilon_i,$$

où f est une fonction inconnue à estimer, supposée assez lisse (deux fois dérivable), et ϵ_i vérifie $E[\epsilon_i \mid X_i] = 0$. Sous ces hypothèses,

$$f(x) = E[Y \mid X = x].$$

On suppose de plus X continue, de sorte que $P(X_i = X_j) = 0$ si $i \neq j$.

Posons $d_i(x) = |X_i - x|$ et ordonnons ces distances, $d_{(1)}(x) \leq \cdots \leq d_{(n)}(x)$. L'ensemble des k plus proches voisins de x est

$$N_k(x) = \{i : d_i(x) \leq d_{(k)}(x)\},$$

et l'estimateur des k plus proches voisins (kNN) est

$$\hat{f}(x) = \frac{1}{|N_k(x)|} \sum_{i \in N_k(x)} Y_i,$$

où $|N_k(x)|$ est le nombre d'éléments de $N_k(x)$. Si $d_{(k)}(x) < d_{(k+1)}(x)$, ce qui arrive avec probabilité 1 lorsque X est continue, $N_k(x)$ contient exactement k indices et $\hat{f}(x)$ est la moyenne des Y_i des k observations dont le X_i est le plus près de x. Si $d_{(k+1)}(x) = d_{(k)}(x)$, $N_k(x)$ contient aussi les observations ex æquo avec la k^{e}, donc plus de k indices.

En R, la fonction `knn.reg` de la librairie `FNN` calcule cet estimateur lorsqu'il n'y a pas d'ex æquo. Elle traite les ex æquo autrement (voir la remarque suivante), alors on programme plutôt l'estimateur directement dans les exemples.

Remarque — Valeurs ex æquo

Les diamètres d'`arbres` sont mesurés au dixième de centimètre près. On n'y trouve que 113 valeurs distinctes pour 200 arbres, contrairement à l'hypothèse $P(X_i = X_j) = 0$. Avec la définition ci-dessus, tous les ex æquo en $d_{(k)}(x)$ sont retenus, et $\hat{f}$ ne dépend pas de l'ordre des lignes du jeu de données. La fonction `knn.reg` retient plutôt exactement k voisins et départage les ex æquo de façon arbitraire. Plus grave, lorsqu'elle calcule le critère de validation croisée, elle peut retenir l'observation i elle-même parmi ses propres voisins quand un autre arbre a le même diamètre, ce qui sous-estime l'erreur de prédiction (voir l'exemple sur le choix de k). L'effet est d'autant plus grand que X est fortement discrétisée.

Remarque — Choix de la distance avec une seule variable explicative

L'estimateur kNN ne dépend de la distance que par l'ordre qu'elle induit sur les observations, c'est-à-dire que seule l'identité des k plus proches voisins compte, pas la valeur de $d_i(x)$. Avec une seule variable explicative, toute distance de la forme $d_i(x) = \phi(|X_i - x|)$, avec ϕ strictement croissante, donne le même ensemble $N_k(x)$, donc le même $\hat{f}$. Par exemple, $|X_i - x|$, $(X_i - x)^2$ et $e^{|X_i - x|}$ donnent les mêmes voisins. Le choix de la distance ne se pose qu'avec plusieurs variables explicatives, parce que la façon de combiner les écarts sur chaque variable change alors les voisins retenus.

7.1.2 Compromis biais-variance et choix de k

Le comportement de $\hat{f}$ dépend de k. Cette section examine comment son biais et sa variance varient avec k, puis en déduit comment choisir k. Pour simplifier, on suppose qu'il n'y a pas d'ex æquo, de sorte que $|N_k(x)| = k$. Conditionnellement aux X_i, et lorsque les X_i de $N_k(x)$ sont proches de x, un développement de Taylor de f autour de x donne

$$E[\hat{f}(x)] \approx f(x) + f'(x)\,\frac{1}{k}\sum_{i \in N_k(x)} (X_i - x) + \frac{1}{2k} f''(x) \sum_{i \in N_k(x)} (X_i - x)^2.$$

Le terme du premier ordre fait intervenir la moyenne des écarts $X_i - x$ des voisins, dont la taille dépend de la position des voisins autour de x.

> **Remarque — Hypothèse de symétrie du voisinage**
> Le terme du premier ordre est négligeable seulement si les X_i de $N_k(x)$ sont répartis à peu près symétriquement autour de x, c'est-à-dire si $\frac{1}{k}\sum_{i\in N_k(x)}(X_i - x) \approx 0$. C'est le cas loin des bords et lorsque la densité de X varie peu autour de x. Si l'une de ces deux conditions n'est pas remplie, le biais contient aussi le terme $f'(x)\,\frac{1}{k}\sum_{i\in N_k(x)}(X_i - x)$, qui peut dominer celui en $f''(x)$.

Sous cette hypothèse,

$$\text{biais}[\hat{f}(x)] \approx \frac{1}{2k} f''(x) \sum_{i\in N_k(x)} (X_i - x)^2,$$

quantité qui croît avec k (les voisins sont de plus en plus éloignés de x). Près des bords, au contraire, les voisins sont tous du même côté de x et c'est le terme du premier ordre $f'(x)\,\frac{1}{k}\sum_{i\in N_k(x)}(X_i - x)$ qui domine : c'est le biais de bord. Par exemple, les voisins des plus gros arbres ont tous un diamètre plus petit et, comme f est croissante ($f'(x) > 0$), $\hat{f}$ y sous-estime la hauteur. La Figure 7.2 oppose les deux situations.

```
k.ill <- 20
x.ill <- c(Centre = median(D), Bord = max(D))
vois <- do.call(rbind, lapply(names(x.ill), function(nom) {
  d <- abs(D - x.ill[[nom]])
  data.frame(panneau = nom, x = x.ill[[nom]], D = D, H = H,
             voisin = d <= sort(d)[k.ill] + 1e-8)
}))
vois <- mutate(vois, panneau = factor(panneau, levels = names(x.ill)))
# moyenne des diamètres et des hauteurs des voisins (hauteur moyenne =
 ↪  estimation kNN en x)
resume <- vois |>
  filter(voisin) |>
  group_by(panneau, x) |>
  summarise(D = mean(D), H = mean(H), .groups = "drop")

ggplot(vois, aes(D, H)) +
  geom_point(aes(colour = voisin), alpha = 0.6) +
  geom_vline(data = resume, aes(xintercept = x)) +
  geom_vline(data = resume, aes(xintercept = D), linetype = "dashed") +
  geom_point(data = resume, aes(x = x, y = H), shape = 17, size = 3) +
  facet_wrap(~ panneau) +
  scale_colour_manual(values = c(`FALSE` = "grey70", `TRUE` = "#3454d1"),
                      labels = c("Autres arbres", "Voisins"), name = NULL) +
  labs(x = "Diamètre (cm)", y = "Hauteur (m)") +
  theme_minimal(base_size = 13)
```

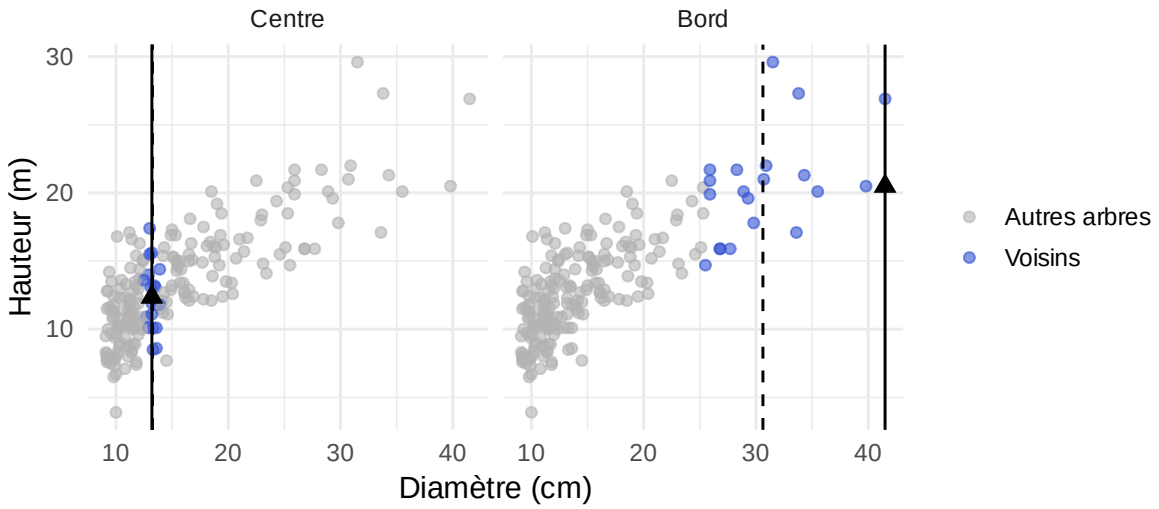


FIGURE 7.2 : Voisinages $N_k(x)$ pour $k = 20$ sur les données `arbres`, au centre (x = diamètre médian) et au bord (x = plus gros diamètre). Le trait plein marque x, le trait pointillé la moyenne des diamètres des voisins et le triangle l'estimation $\hat{f}(x)$. Au bord, les voisins sont tous à gauche de x : $\hat{f}(x)$ est la moyenne des hauteurs d'arbres plus petits.

Au centre, la moyenne des diamètres des voisins (13,2 cm) est presque égale à x (13,2 cm). Au bord, elle vaut 30,6 cm, loin sous x (41,5 cm), et le terme du premier ordre domine.

Quant à la variance, si les ϵ_i sont indépendants et $Var[\epsilon_i \mid X_i] = \sigma^2$, alors, conditionnellement aux X_i,

$$Var[\hat{f}(x)] = \frac{\sigma^2}{k},$$

qui décroît avec k. On sélectionne donc le k qui minimise l'erreur quadratique moyenne

$$\text{biais}^2[\hat{f}(x)] + Var[\hat{f}(x)].$$

En pratique, on ne connaît ni $f''(x)$ ni σ^2, et k est sélectionné par validation croisée. Le critère LOOCV est

$$VC_{(n)}(k) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{f}_{-i}(X_i)\right)^2,$$

où $\hat{f}_{-i}$ est l'estimateur kNN calculé sans l'observation i. Lorsqu'aucun jeu test ne lui est fourni, `knn.reg` calcule le critère `PRESS`, qui vaut $n\,VC_{(n)}(k)$ s'il n'y a pas d'ex æquo.

Exemple — Choix de k sur `arbres`

Le code suivant programme l'estimateur kNN avec la définition de $N_k(x)$ ci-dessus, puis

calcule $VC_{(n)}(k)$ pour $k = 1, \ldots, 60$.

```
# kNN : moyenne des Y_i tels que d_i(x) <= d_(k)(x), ex æquo compris
# (la tolérance 1e-8 évite qu'une erreur d'arrondi brise une égalité de
 ↪  distances)
knn.moy <- function(x, X, Y, k) {
  d <- abs(X - x)
  mean(Y[d <= sort(d)[k] + 1e-8])
}
# prédictions LOO : estimation de f(D[i]) sans l'observation i, i = 1,
 ↪  ..., n
knn.loo <- function(Y, k) sapply(1:n, function(i) knn.moy(D[i], D[-i],
 ↪  Y[-i], k))

ks <- 1:60
vc <- sapply(ks, function(k) mean((H - knn.loo(H, k))^2))
k.min <- ks[which.min(vc)]
k.min
```

```
[1] 18
```

```
# comparaison avec knn.reg pour k = k.min
press.fnn <- knn.reg(train = D, y = H, k = k.min)$PRESS / n
# nombre d'arbres i que knn.reg garde parmi leurs propres voisins
soi <- sum(rowSums(get.knn(matrix(D), k = k.min)$nn.index == 1:n) > 0)
c(VC = vc[k.min], PRESS.n = press.fnn, soi = soi)
```

```
       VC   PRESS.n       soi
 6.501744  6.199754 87.000000
```

```
ggplot(data.frame(k = ks, vc), aes(k, vc)) +
  geom_line() + geom_point() +
  geom_vline(xintercept = k.min, linetype = "dashed") +
  labs(x = "k", y = expression(VC[(n)](k))) +
  theme_minimal(base_size = 13)
```

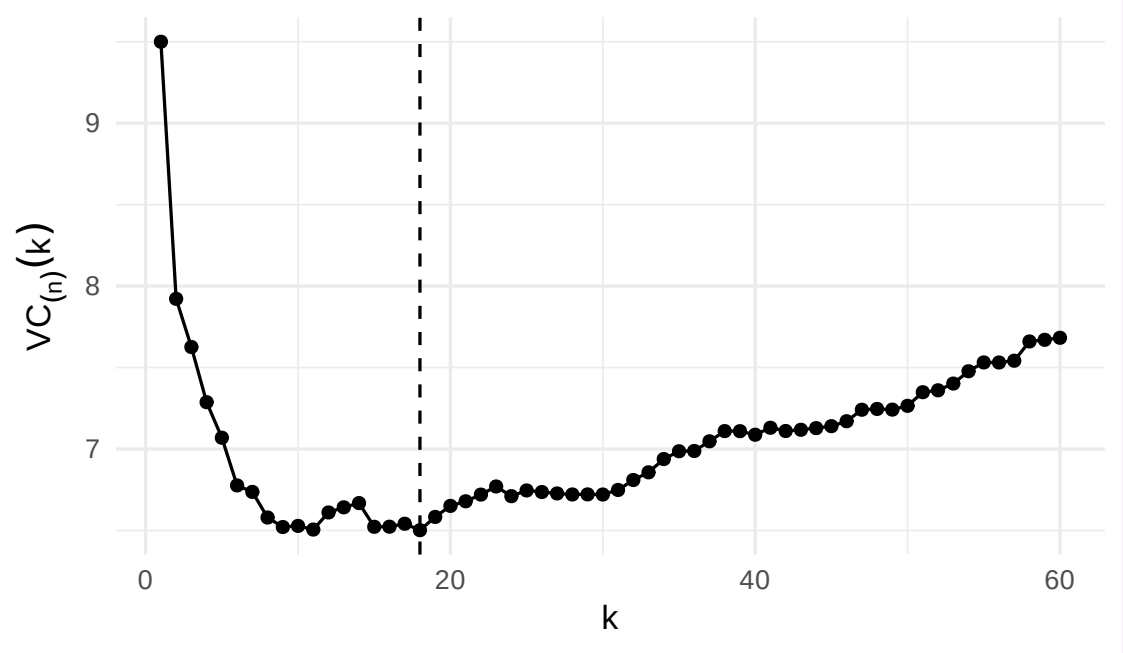


FIGURE 7.3 : Erreur quadratique moyenne LOOCV de l'estimateur kNN en fonction de k sur les données `arbres`.

Le critère est élevé pour $k = 1$ (variance maximale), décroît rapidement, puis forme un plateau dont le minimum est en $k_{\min} = 18$ ($VC_{(n)}(k_{\min}) = 6{,}50$), et remonte lentement lorsque le biais prend le dessus. Comme le plateau est assez plat, les valeurs de k voisines de $k_{\min}$ donnent une erreur semblable.
Pour $k = k_{\min}$, `knn.reg` donne plutôt PRESS$/n = 6{,}20$. Pour 87 des 200 arbres, l'arbre i figure parmi ses propres voisins : `knn.reg` retire un point à distance nulle de X_i, mais pas forcément l'arbre i lorsqu'un autre arbre a le même diamètre. Sa valeur de Y_i sert alors à se prédire elle-même, et le critère sous-estime l'erreur de prédiction.
La Figure 7.4 compare l'estimateur pour trois valeurs de k. Avec $k = 3$, la courbe suit le bruit des observations. Avec $k = 45$, elle est plus régulière, mais elle écrase la croissance rapide des petits diamètres et sous-estime la hauteur des plus gros arbres, qui ont peu de voisins de leur taille.

```
grille <- seq(min(D), max(D), length.out = 300)
k.fig <- c(3, k.min, 45)
courbes <- do.call(rbind, lapply(k.fig, function(k) {
  data.frame(k = paste("k =", k), x = grille,
             f = sapply(grille, function(x) knn.moy(x, D, H, k)))
}))
```

```r
courbes <- mutate(courbes, k = factor(k, levels = paste("k =", k.fig)))
 ↪  # ordre de la légende

ggplot(arbres, aes(diameter_cm, height_m)) +
  geom_point(alpha = 0.2) +
  geom_line(data = courbes, aes(x, f, colour = k), linewidth = 0.8) +
  labs(x = "Diamètre (cm)", y = "Hauteur (m)", colour = NULL) +
  theme_minimal(base_size = 13)
```

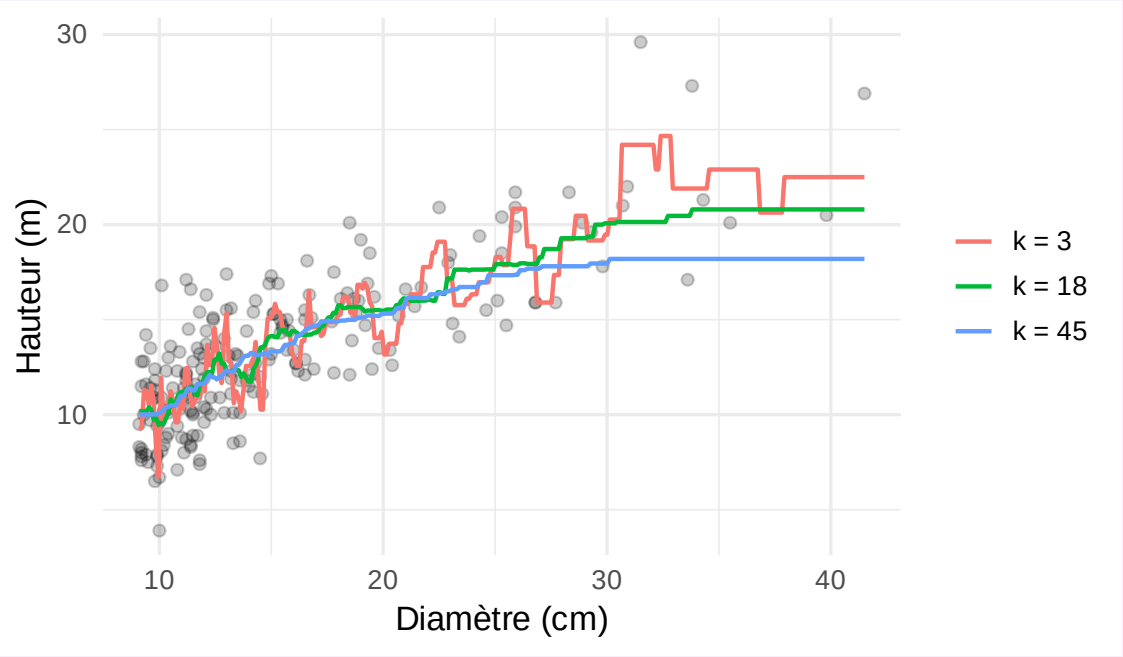


FIGURE 7.4 : Estimateur kNN de la hauteur en fonction du diamètre pour $k = 3$, $k = k_{\min}$ et $k = 45$.

7.1.3 Plusieurs variables explicatives et distance de Mahalanobis

Lorsqu'il y a plusieurs variables explicatives, l'estimateur kNN garde la même forme et seule la distance change. Avec deux variables, le modèle devient $Y_i = f(X_{i1}, X_{i2}) + \epsilon_i$, et la distance euclidienne

$$d_i(x) = \sqrt{(X_{i1} - x_1)^2 + (X_{i2} - x_2)^2}$$

pose problème si les ordres de grandeur des variables diffèrent. Supposons par exemple qu'il faille prédire la hauteur des arbres d'`arbres` à partir du diamètre et de l'âge (`age_years`, disponible pour 148 arbres). Parmi ces 148 arbres, le diamètre va de 9 à 31 cm, alors que l'âge

va de 15 à 153 ans. Comme la distance euclidienne traite une différence d'un an comme une différence d'un centimètre, l'âge domine le calcul des voisins. La correction consiste à pondérer par l'inverse de la matrice de variance-covariance empirique :

$$d_i(x) = \sqrt{\begin{pmatrix} X_{i1} - x_1 \\ X_{i2} - x_2 \end{pmatrix}^\top V^{-1} \begin{pmatrix} X_{i1} - x_1 \\ X_{i2} - x_2 \end{pmatrix}}, \qquad V = \begin{pmatrix} s_1^2 & s_{12} \\ s_{12} & s_2^2 \end{pmatrix}.$$

C'est la distance de Mahalanobis. Pour la suite, notons $z_i = (X_{i1}, \dots, X_{ip})^\top$ le vecteur des variables explicatives de l'observation i (sans constante), $z = (z_1, \dots, z_p)^\top$ le point où l'on veut estimer f, et Σ la matrice de variance-covariance des variables explicatives (estimée par V). En notation vectorielle, la distance s'écrit

$$d_i(z) = \sqrt{(z_i - z)^\top \Sigma^{-1} (z_i - z)},$$

ce qui la généralise à p variables continues. Lorsque $\Sigma = I$, elle se réduit à la distance euclidienne, et lorsque Σ est diagonale, à la distance euclidienne sur les variables standardisées. En pratique, on peut transformer les variables par $V^{-1/2}$ (ou, plus simplement, les standardiser) avant de calculer les voisins.

> **Remarque — Distances au carré**
> En pratique, on omet la racine carrée et on travaille avec les distances au carré, $(X_i - x)^2$ ou $(z_i - z)^\top \Sigma^{-1} (z_i - z)$. Comme la racine est strictement croissante, l'ordre des observations reste le même, donc $N_k(x)$ et $\hat{f}$ aussi. Les implémentations comme `knn.reg` procèdent ainsi pour s'épargner un calcul inutile.

7.2 Estimateur à noyau (Nadaraya-Watson)

7.2.1 Des voisins aux noyaux

On réécrit d'abord l'estimateur kNN comme une moyenne pondérée, puis on change les poids. L'estimateur kNN peut s'écrire

$$\hat{f}(x) = \frac{\sum_{i=1}^n \mathbf{1}_{\{i \in N_k(x)\}} Y_i}{\sum_{i=1}^n \mathbf{1}_{\{i \in N_k(x)\}}}.$$

Le voisinage peut aussi se définir autrement : x et X_i sont voisins si $|X_i - x| < h$, où le paramètre de fenêtre $h > 0$ joue le même rôle que k. On obtient alors

$$\hat{f}(x) = \frac{\sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) Y_i}{\sum_{i=1}^n K\left(\frac{X_i - x}{h}\right)} = \sum_{i=1}^n w_i(x)\, Y_i,$$

où $K(u) = \mathbf{1}_{\{|u|<1\}}$ est appelé noyau (ici, le noyau uniforme) et

$$w_i(x) = \frac{K\left(\frac{X_i - x}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{X_j - x}{h}\right)}$$

sont des poids non négatifs de somme 1 (définis lorsqu'au moins une observation est dans la fenêtre). Cet estimateur, avec un noyau K quelconque, est appelé estimateur de Nadaraya-Watson.

Avec le noyau uniforme, la contribution d'une observation passe brusquement d'une valeur positive, $1/n_h(x)$ (où $n_h(x)$ est le nombre d'observations dans la fenêtre), à 0 lorsque x sort de son voisinage. L'estimateur $\hat{f}$ est donc discontinu. Avec un noyau continu, les poids diminuent de façon continue à mesure que x s'éloigne de X_i. Par exemple, le noyau gaussien $K(u) = e^{-u^2/2}$ donne des poids continus, donc un estimateur $\hat{f}$ continu.

En R, la fonction `locpoly(x=X, y=Y, degree=0, bandwidth=h)` de la librairie `KernSmooth` calcule une approximation de cet estimateur avec le noyau gaussien, sur une grille de valeurs de x. Comme elle tronque le noyau à $4h$, elle ne donne pas de valeur (`NaN`) là où aucune observation n'est à moins de $4h$ de x. Par exemple, avec $h = 0{,}5$, elle n'estime rien entre 37,5 et 37,8 cm dans `arbres`, au milieu du trou de 35,5 à 39,8 cm où il n'y a aucun arbre. Dans les exemples, on calcule plutôt l'estimateur directement à partir de sa définition.

7.2.2 Choix du paramètre de fenêtre h

Comme pour kNN, le paramètre h est choisi par validation croisée, parce qu'un h trop petit produit une courbe oscillante (variance élevée) et qu'un h trop grand écrase la structure (biais élevé). Le critère LOOCV est $VC_{(n)}(h) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}_{-i}(X_i))^2$, où $\hat{f}_{-i}$ est l'estimateur calculé sans l'observation i. On montre qu'il se calcule sans réajuster l'estimateur n fois :

$$VC_{(n)}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{f}(X_i)}{1 - w_i(X_i)}\right)^2,$$

où $\hat{f}$ est l'estimateur calculé avec toutes les observations.

Exemple — Choix de h sur `arbres`

La formule ci-dessus se programme directement, puisque la matrice `W` contient les poids $w_j(X_i)$, donc `W %*% H` donne les valeurs ajustées et `diag(W)` donne les $w_i(X_i)$.

```
cv.noyau <- function(h) {
  W <- dnorm(outer(D, D, "-") / h)   # W[i, j] = K((D[j] - D[i]) / h)
  W <- W / rowSums(W)                 # poids w_j(D[i])
  f.hat <- as.vector(W %*% H)
```

```r
  mean(((H - f.hat) / (1 - diag(W)))^2)
}
hs <- seq(0.2, 10, by = 0.1)
cvs <- sapply(hs, cv.noyau)
h.min <- hs[which.min(cvs)]
h.min
```

```
[1] 2
```

```r
ggplot(data.frame(h = hs, cv = cvs), aes(h, cv)) +
  geom_line() +
  geom_vline(xintercept = h.min, linetype = "dashed") +
  labs(x = "h (cm)", y = expression(VC[(n)](h))) +
  theme_minimal(base_size = 13)
```

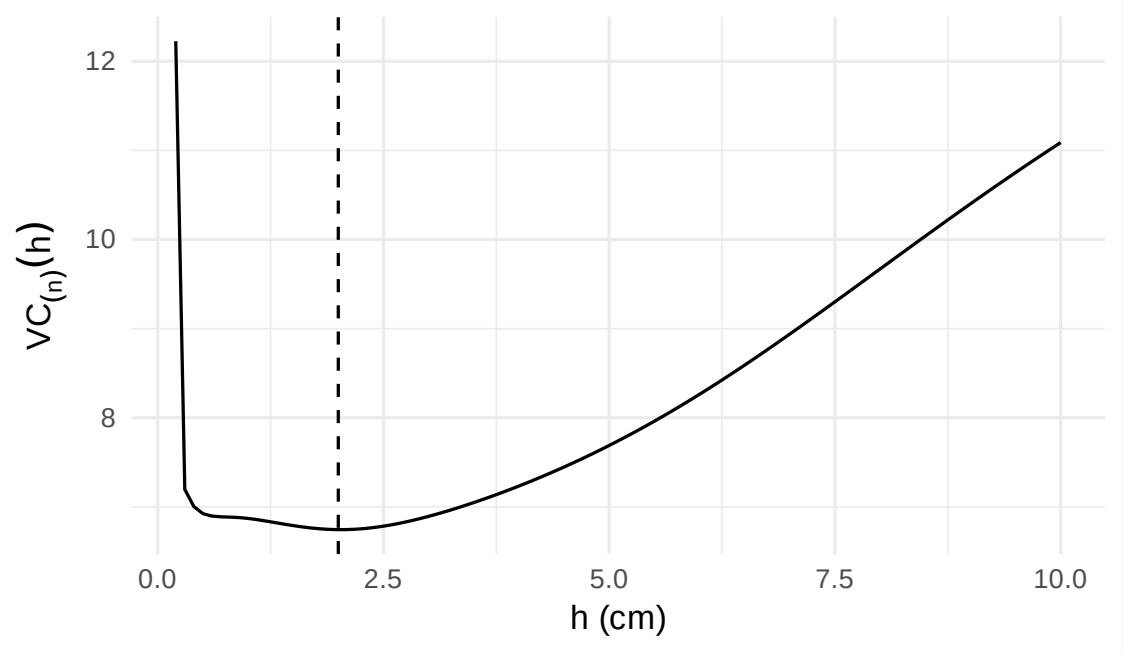


FIGURE 7.5 : Critère de validation croisée de l'estimateur de Nadaraya-Watson (noyau gaussien) en fonction de la fenêtre h sur les données `arbres`.

Avec le noyau gaussien, on obtient $h_{\min} = 2{,}0$ cm ($VC_{(n)}(h_{\min}) = 6{,}74$). Pour $h = h_{\min}$, l'estimateur de Nadaraya-Watson est proche de l'estimateur kNN avec $k = k_{\min}$, mais sans ses sauts (Figure 7.6), et les deux méthodes captent la même relation concave.

```r
# estimateur de Nadaraya-Watson (noyau gaussien) calculé exactement en x
nw.moy <- function(x, h) {
  w <- dnorm((D - x) / h)
  sum(w * H) / sum(w)
}
h.fig <- c(0.5, h.min, 8)
etiq.h <- paste("h =", sapply(h.fig, format, decimal.mark = ","))
nw <- do.call(rbind, lapply(seq_along(h.fig), function(j) {
  data.frame(h = etiq.h[j], x = grille, f = sapply(grille, nw.moy, h =
   ↪  h.fig[j]))
}))
nw <- mutate(nw, h = factor(h, levels = etiq.h))  # ordre de la légende
knn.min <- data.frame(x = grille, f = sapply(grille, function(x)
 ↪  knn.moy(x, D, H, k.min)))

ggplot(arbres, aes(diameter_cm, height_m)) +
  geom_point(alpha = 0.2) +
  geom_line(data = nw, aes(x, f, colour = h), linewidth = 0.9) +
  geom_line(data = knn.min, aes(x, f), linetype = "dashed") +
  labs(x = "Diamètre (cm)", y = "Hauteur (m)", colour = NULL) +
  theme_minimal(base_size = 13)
```

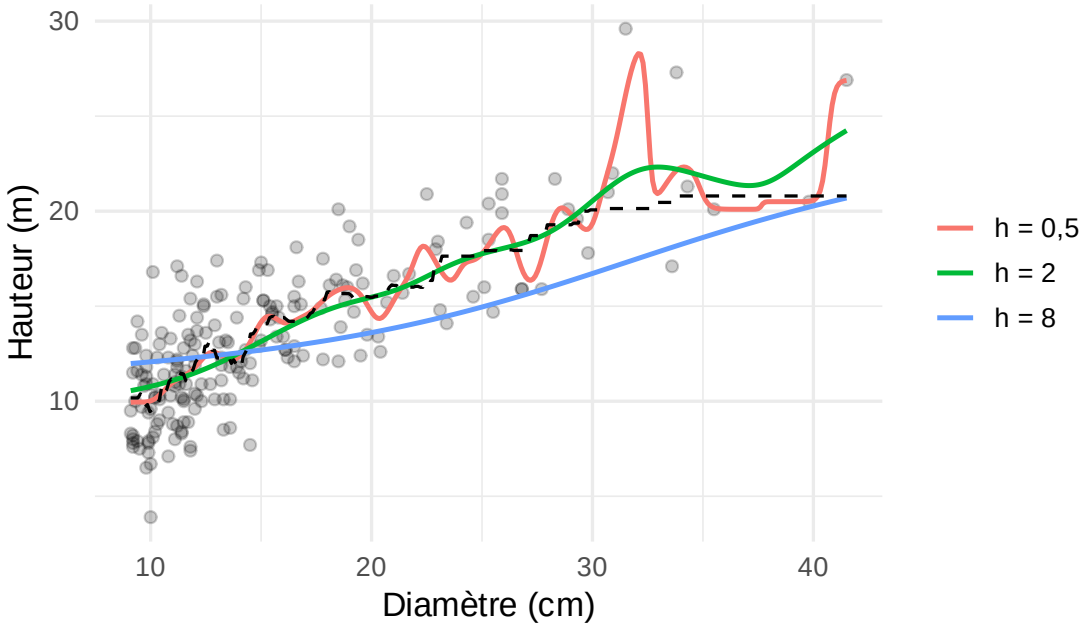


FIGURE 7.6 : Estimateur de Nadaraya-Watson pour $h = 0{,}5$, $h = h_{\min}$ et $h = 8$ cm, et estimateur kNN avec $k = k_{\min}$ (tirets).

Remarque — kNN et Nadaraya-Watson : quelle est la différence ?

Les deux estimateurs reposent sur la même idée : $f(x)$ est estimée par la moyenne des Y_i des observations dont le X_i est proche de x. Ils diffèrent par la façon de définir « proche » et de pondérer.

- Estimateur kNN : il fixe le nombre de voisins. Il prend toujours les k observations les plus proches de x (plus les ex æquo), quelle que soit leur distance, et donne à chacune le même poids $1/|N_k(x)|$, soit $1/k$ en l'absence d'ex æquo. La taille du voisinage s'adapte donc à la densité des données, petite là où les observations sont nombreuses et grande là où elles sont rares. Par contre, l'estimateur est une fonction en escalier, qui saute chaque fois qu'une observation entre dans le voisinage ou en sort. La variance, σ^2/k en l'absence d'ex æquo, est la même partout ;
- Estimateur de Nadaraya-Watson : il fixe la largeur du voisinage par h. Toutes les observations contribuent, avec un poids qui décroît continûment avec la distance $|X_i - x|/h$, de sorte que le nombre effectif de voisins varie d'un x à l'autre. Il est grand là où les données sont denses (variance faible) et petit là où elles sont rares (variance élevée, comme pour les gros arbres d'`arbres`). En contrepartie, l'estimateur est lisse, puisque les poids varient continûment avec x.

Autrement dit, kNN adapte la fenêtre et garde la variance constante ; Nadaraya-Watson

garde la fenêtre et laisse la variance varier. Les deux ont un paramètre de lissage (k ou h) qu'on choisit par validation croisée. Pour des valeurs comparables (un h tel que le voisinage contient environ k observations), ils donnent des courbes semblables, comme le montre la Figure 7.6.

7.3 Régression non paramétrique et classification

Appliqué à une réponse binaire, l'estimateur kNN devient un classifieur. Dans `arbres`, la variable `canopy_stratum` donne la position de l'arbre dans le couvert : un arbre est dominant ou codominant s'il reçoit la lumière directement, intermédiaire ou opprimé s'il pousse sous le couvert des autres. Posons $Y_i = 1$ si l'arbre i est dans la strate inférieure (intermédiaire ou opprimé), ce qui est le cas de 61 des 200 arbres, et $Y_i = 0$ sinon. On veut prédire la strate à partir du diamètre X_i. La fonction à estimer est $\pi(x) = P(Y = 1 \mid X = x)$, qu'on estime par kNN :

$$\hat{\pi}(x) = \frac{1}{|N_k(x)|} \sum_{i \in N_k(x)} Y_i \in [0, 1].$$

Deux critères de validation croisée permettent de choisir k :

1. Taux d'erreur test : on pose $\hat{Y}_i = \mathbf{1}_{\{\hat{\pi}_{-i}(X_i) > u\}}$ (avec $u = 1/2$ en pratique) et on minimise
$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{Y_i \neq \hat{Y}_i\}}.$$
2. Aire sous la courbe ROC : on choisit le k qui maximise l'AUC (voir le Section 2.5), un critère qui ne dépend d'aucun seuil.

Exemple — Prédire la strate d'un arbre à partir de son diamètre

On calcule, pour chaque k, les prédictions LOOCV $\hat{\pi}_{-i}(X_i)$ avec la fonction `knn.loo` de l'exemple sur le choix de k, puis on en tire le taux d'erreur pour trois seuils u et l'AUC. L'AUC se calcule à partir des rangs des $\hat{\pi}_{-i}(X_i)$ (statistique de Mann-Whitney), ce qui évite de tracer la courbe ROC pour chaque k.

```
Y <- as.integer(arbres$canopy_stratum %in% c("Intermédiaire", "Opprimé"))

# p.loo[i, k] : estimation kNN de pi(D[i]) sans l'observation i
ks.c <- 1:100
p.loo <- sapply(ks.c, function(k) knn.loo(Y, k))

taux.erreur <- function(u) apply(p.loo, 2, function(p) mean(Y != (p >
 ↪  u)))
```

```r
auc <- function(p) {
  r <- rank(p); n1 <- sum(Y == 1); n0 <- n - n1
  (sum(r[Y == 1]) - n1 * (n1 + 1) / 2) / (n1 * n0)
}

# k qui minimisent le taux d'erreur pour u = 1/4, 1/2 et 3/4 (ex æquo compris)
k.u <- lapply(c(0.25, 0.5, 0.75), function(u) {
  te <- taux.erreur(u); ks.c[te <= min(te) + 1e-12]
})
k.u
```

```
[[1]]
[1] 28

[[2]]
[1] 11 12

[[3]]
[1]  2  8  9 13
```

```r
aucs <- apply(p.loo, 2, auc)
k.auc <- ks.c[which.max(aucs)]
# AUC maximale du kNN et AUC obtenue en ordonnant les arbres par diamètre décroissant
c(k.auc = k.auc, auc.max = max(aucs), auc.diametre = auc(-D))
```

```
      k.auc      auc.max auc.diametre
 64.0000000    0.7296851    0.7713763
```

```r
res <- rbind(data.frame(critere = "Taux d'erreur (u = 1/2)", k = ks.c, valeur = taux.erreur(0.5)),
             data.frame(critere = "AUC", k = ks.c, valeur = aucs))
ggplot(res, aes(k, valeur)) +
  geom_line() +
  facet_wrap(~ critere, scales = "free_y") +
  labs(x = "k", y = NULL) +
  theme_minimal(base_size = 13)
```

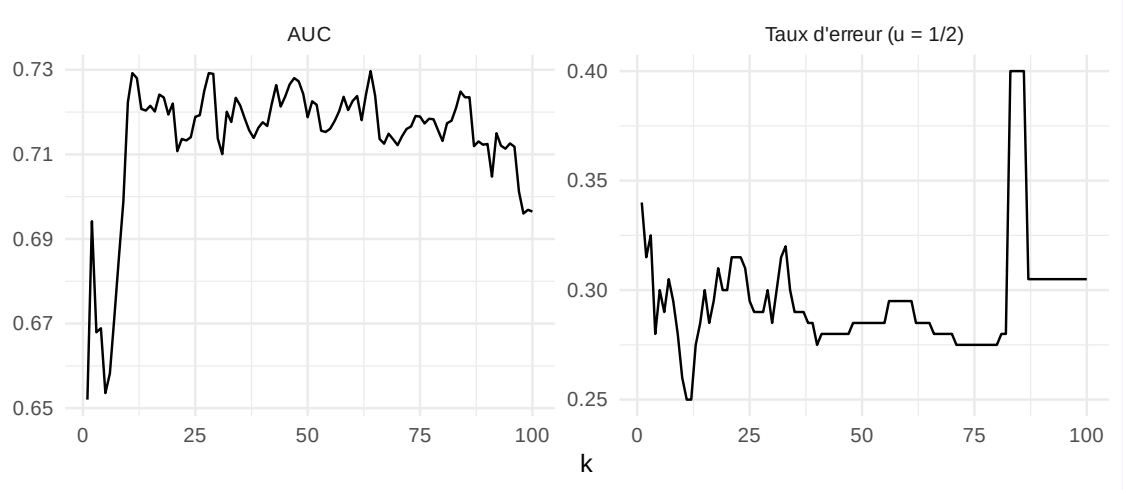


FIGURE 7.7 : Taux d'erreur LOOCV (seuil $u = 1/2$) et AUC de l'estimateur kNN de $P(\text{strate inférieure} \mid \text{diamètre})$ en fonction de k.

Avec $u = 1/2$, le taux d'erreur est minimal en $k = 11$ et $k = 12$ (à égalité), mais la courbe est irrégulière et le résultat dépend du seuil arbitraire u : avec $u = 1/4$, on obtient $k = 28$ et avec $u = 3/4$, le minimum est atteint en $k = 2$, 8, 9, 13. L'AUC varie plus régulièrement avec k : après une montée rapide pour les petites valeurs de k, elle forme un plateau autour de 0,72 sur presque toute la grille, avec un maximum de 0,73 en $k = 64$. L'estimateur kNN ne permet donc qu'une prédiction modeste de la strate (AUC LOOCV $\approx 0{,}73$). À titre de comparaison, ordonner simplement les arbres par diamètre décroissant donne une AUC de 0,77. Environ un arbre sur deux de moins de 12 cm est sous le couvert, contre moins d'un sur vingt au-delà de 20 cm. Entre les deux, la strate dépend surtout des arbres voisins, que le jeu de données ne décrit pas.

Remarque — Ce qu'on cherche à faire ici

Rien n'a changé dans la méthode. Y est une variable binaire, mais $\pi(x) = E[Y \mid X = x]$ reste une fonction de régression, que l'estimateur kNN estime comme avant par la moyenne des Y_i des k voisins de x. Lorsque Y vaut 0 ou 1, cette moyenne est simplement la proportion de voisins pour lesquels $Y_i = 1$, et $E[Y \mid X = x]$ est la probabilité $P(Y = 1 \mid X = x)$. On estime donc une probabilité, comme en régression logistique, mais sans supposer que son logit est linéaire en x : la courbe $\hat{\pi}$ prend la forme que les données lui donnent. C'est l'analogue non paramétrique du Chapitre 2.

Une fois $\hat{\pi}$ obtenue, on s'en sert de deux façons : classer un nouvel arbre dans la strate inférieure si $\hat{\pi}(x) > u$, ou ordonner les arbres selon leur probabilité estimée, sans fixer de seuil. Comme dans tout le chapitre, k joue le rôle du paramètre de lissage et il est encore choisi par validation croisée. Ce qui change, c'est le critère. L'erreur quadratique

n'est plus naturelle pour une réponse binaire, donc elle est remplacée par une mesure adaptée à l'usage prévu de $\hat{\pi}$. Si l'objectif est de classer, le critère est le taux de mauvaise classification. Si l'objectif est d'ordonner les individus du plus probable au moins probable, par exemple pour retenir les meilleurs candidats sans fixer de seuil à l'avance, le critère est l'AUC. L'exemple montre que le premier critère est instable et dépend d'un seuil arbitraire, alors que le second ne dépend d'aucun seuil et varie plus régulièrement avec k.

Remarque — À retenir

- kNN : $\hat{f}(x)$ est la moyenne des Y_i des k observations dont le X_i est le plus près de x, et k, qui règle le compromis entre le biais (qui croît avec k, loin des bords) et la variance σ^2/k, se choisit par validation croisée. Notez qu'avec des ex æquo, `knn.reg` peut compter l'observation i parmi ses propres voisins, ce qui sous-estime l'erreur de prédiction ;
- Biais de bord : près des bords du domaine, les voisins sont tous du même côté de x, et le terme du premier ordre en $f'(x)$ domine le biais (la hauteur des plus gros arbres est sous-estimée) ;
- Distance : avec une seule variable explicative, le choix de la distance ne change pas les voisins. Avec plusieurs, la distance euclidienne pose problème si les ordres de grandeur des variables diffèrent, d'où la distance de Mahalanobis ou, plus simplement, la standardisation des variables ;
- Nadaraya-Watson : $\hat{f}(x) = \sum_{i=1}^{n} w_i(x)\, Y_i$, avec des poids proportionnels à $K\left(\frac{X_i - x}{h}\right)$, et la fenêtre h se choisit par LOOCV sans réajuster l'estimateur n fois. Le kNN fixe le nombre de voisins (variance constante, estimateur en escalier), alors que Nadaraya-Watson fixe la largeur du voisinage (variance variable, estimateur lisse avec un noyau continu) ;
- Classification : pour une réponse binaire, $\hat{\pi}(x)$ est la proportion de voisins avec $Y_i = 1$, une estimation de $P(Y = 1 \mid X = x)$ qui ne suppose pas que son logit est linéaire en x. Pour choisir k, l'AUC, qui ne dépend d'aucun seuil, varie plus régulièrement que le taux de mauvaise classification, qui dépend du seuil arbitraire u.

7.4 Exercices

Les exercices sont classés par niveau de difficulté. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

7.4.1 Niveau facile

Exercice 1. On dispose des six observations suivantes :

X_i	1	2	3	5	8	9
Y_i	2	4	3	7	6	9

a) Calculez l'estimateur kNN en $x = 4$ pour $k = 2$ et $k = 3$.
b) Calculez l'estimateur de Nadaraya-Watson en $x = 4$ avec le noyau uniforme $K(u) = \mathbf{1}_{\{|u|<1\}}$, pour $h = 2{,}5$ puis pour $h = 1$. Que se passe-t-il dans le second cas ?
c) Calculez l'estimateur de Nadaraya-Watson en $x = 4$ avec le noyau gaussien $K(u) = e^{-u^2/2}$ et $h = 1$.

Voir la solution

Exercice 2. Supposez que les erreurs sont indépendantes, de variance σ^2, et raisonnez conditionnellement aux X_i.

a) Montrez que l'estimateur de Nadaraya-Watson a pour variance $Var[\hat{f}(x)] = \sigma^2 \sum_i w_i(x)^2$.
b) Avec le noyau uniforme, montrez que cette variance vaut $\sigma^2/n_h(x)$, où $n_h(x)$ est le nombre d'observations dans la fenêtre.
c) Montrez que, parmi toutes les moyennes pondérées de m observations données (poids positifs de somme 1), la variance est minimale lorsque les poids sont égaux. (Indice : inégalité de Cauchy-Schwarz.)

Voir la solution

Exercice 3. Supposez qu'il n'y a pas d'ex æquo parmi les X_i.

a) Pour $k = 1$, que vaut $\hat{f}(X_i)$ calculé avec toutes les observations ? Quelle est l'erreur d'entraînement ?
b) Pour $k = n$, que vaut $\hat{f}(x)$?
c) Expliquez pourquoi on ne peut pas choisir k en minimisant l'erreur d'entraînement, et pourquoi la validation croisée règle le problème.

Voir la solution

7.4.2 Niveau moyen

Exercice 1. Démontrez la formule de LOOCV de l'estimateur de Nadaraya-Watson énoncée au Chapitre 7 :

$$Y_i - \hat{f}_{-i}(X_i) = \frac{Y_i - \hat{f}(X_i)}{1 - w_i(X_i)},$$

où $\hat{f}_{-i}$ est l'estimateur calculé sans l'observation i et $w_i(X_i) = K(0)/\sum_j K\{(X_j - X_i)/h\}$. La même formule vaut-elle pour le kNN ?

Voir la solution

Exercice 2. Montrez que l'estimateur de Nadaraya-Watson en x est la solution d'un problème de moindres carrés pondérés :

$$\hat{f}(x) = \arg\min_{a \in \mathbb{R}} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)(Y_i - a)^2.$$

Expliquez pourquoi on l'appelle un estimateur localement constant, et décrivez comment on obtiendrait un estimateur localement linéaire.

Voir la solution

Exercice 3. Soit Σ une matrice de variance-covariance $p \times p$ définie positive et $d(z_i, z) = \{(z_i - z)^\top \Sigma^{-1}(z_i - z)\}^{1/2}$ la distance de Mahalanobis.

a) Montrez que si $\Sigma = \text{diag}(s_1^2, \ldots, s_p^2)$, la distance de Mahalanobis est la distance euclidienne entre les vecteurs standardisés $(z_{i1}/s_1, \ldots, z_{ip}/s_p)$ et $(z_1/s_1, \ldots, z_p/s_p)$.
b) On transforme les variables explicatives par $\tilde{z} = Bz + c$, avec B inversible, de sorte que leur matrice de variance-covariance devient $B\Sigma B^\top$. Montrez que la distance de Mahalanobis est inchangée. Qu'en déduisez-vous pour l'estimateur kNN ?

Voir la solution

7.4.3 Niveau difficile

Exercice 1. Supposez que les X_i sont équidistants, d'écart δ entre deux valeurs consécutives, que f est deux fois dérivable et que k est impair. On néglige les termes d'ordre supérieur à 2 dans le développement de Taylor de f.

a) Pour un point intérieur $x = X_j$ dont les k voisins sont X_j et les $(k-1)/2$ observations de chaque côté, montrez que le biais de l'estimateur kNN vaut approximativement

$$\text{biais}[\hat{f}(x)] \approx \frac{f''(x)\,\delta^2\,(k^2 - 1)}{24}.$$

(Indice : $\sum_{m=1}^{M} m^2 = M(M+1)(2M+1)/6$.)
b) Pour le point le plus à gauche, $x = X_{(1)}$, dont les voisins sont $X_{(1)}, \ldots, X_{(k)}$, montrez que le biais est approximativement $f'(x)\,\delta\,(k-1)/2$. Comparez les ordres de grandeur des deux biais lorsque δ est petit.

Voir la solution

Exercice 2. On poursuit l'exercice précédent pour un point intérieur, avec des X_i équidistants sur $[0,1]$, de sorte que $\delta = 1/n$. On suppose $Var[\hat{f}(x)] = \sigma^2/k$.

a) Écrivez l'erreur quadratique moyenne approximative $EQM(k)$ en traitant k comme une variable continue et en remplaçant $k^2 - 1$ par k^2.

b) Montrez que la valeur de k qui la minimise est $k^* = \left\{ \frac{144\,\sigma^2 n^4}{f''(x)^2} \right\}^{1/5}$, qui croît comme $n^{4/5}$.

c) Montrez que l'erreur quadratique moyenne optimale décroît comme $n^{-4/5}$. Comparez avec un modèle paramétrique bien spécifié, dont l'erreur quadratique moyenne décroît comme n^{-1}.

Voir la solution

Exercice 3. Soit (X, Y) un couple aléatoire, avec $Y \in \{0,1\}$ et $\pi(x) = P(Y = 1 \mid X = x)$. Un classifieur est une fonction $c(x) \in \{0,1\}$.

a) Montrez que, pour tout x, $P\{Y \neq c(X) \mid X = x\}$ vaut $1 - \pi(x)$ si $c(x) = 1$ et $\pi(x)$ si $c(x) = 0$.

b) Déduisez-en que le classifieur de Bayes, $c^*(x) = \mathbf{1}_{\{\pi(x) > 1/2\}}$, minimise le taux d'erreur $P\{Y \neq c(X)\}$ parmi tous les classifieurs.

c) Expliquez pourquoi le classifieur kNN du Chapitre 7, avec le seuil $u = 1/2$, est une approximation du classifieur de Bayes, et quand un autre seuil serait préférable.

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R*
— Faraway (2016). *Extending the Linear Model with R*
— James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R*
— Hastie, Tibshirani et Friedman (2009). *The Elements of Statistical Learning*
— Wasserman (2006). *All of Nonparametric Statistics*

8 Régression non linéaire : polynômes, fonctions par morceaux et splines

Ce chapitre utilise les librairies R suivantes.

```
library(ggplot2)
library(dplyr)
```

Le chapitre précédent a présenté deux estimateurs locaux de $f(x) = E[Y \mid X = x]$: l'estimateur kNN et l'estimateur à noyau (Chapitre 7). Ce chapitre présente une autre approche, fondée sur des bases de fonctions, où le modèle s'écrit

$$Y_i = \beta_1 \, b_1(X_i) + \beta_2 \, b_2(X_i) + \cdots + \beta_q \, b_q(X_i) + \epsilon_i,$$

où les $b_k(\cdot)$ sont des transformations connues de la variable explicative. Comme le modèle reste linéaire en β, tout ce qui découle des moindres carrés (estimation, intervalles de confiance, tests) s'applique tel quel.

8.1 Régression polynomiale

La régression polynomiale approxime $f(x)$ par un polynôme de degré p :

$$f(x) = \beta_0 + \beta_1 x + \cdots + \beta_p x^p,$$

ce qui donne le modèle linéaire $Y_i = \beta_0 + \beta_1 X_i + \cdots + \beta_p X_i^p + \epsilon_i$. On estime β par

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y, \qquad X = \begin{pmatrix} 1 & X_1 & \cdots & X_1^p \\ \vdots & \vdots & & \vdots \\ 1 & X_n & \cdots & X_n^p \end{pmatrix}.$$

Comme β contient $p+1$ éléments, le modèle a $p+1$ degrés de liberté.

Pour choisir p, on peut utiliser l'AIC, le BIC ou des tests F. On peut aussi procéder comme au Chapitre 5 et retenir la valeur qui minimise le taux d'erreur test estimé par validation croisée (la LOOCV admet une expression explicite).

Notons $x_* = (1,\ x,\ x^2,\ \dots,\ x^p)$ le vecteur des puissances de x. Une fois p choisi, on a $\hat{f}(x) = x_* \hat{\beta}$ et

$$Var[\hat{f}(x)] = x_*\, Var[\hat{\beta}]\, x_*^\top,$$

ce qui donne des bandes de confiance autour de $\hat{f}(x)$. En R : `lm(Y ~ poly(X, 2, raw = TRUE), data = data)`.

Exemple — Régression polynomiale sur `arbres`

Reprenons la hauteur en fonction du diamètre du Chapitre 7. On ajuste des polynômes de degré 1 à 6 et on choisit p par LOOCV. La formule explicite du Chapitre 5 s'applique : $VC_{(n)} = \frac{1}{n}\sum_i \left(\frac{e_i}{1-h_i}\right)^2$, où e_i est le résidu et h_i le levier de l'observation i.

```
arbres <- read.csv("Jeux de données/arbres.csv")

degres <- 1:6
modeles <- lapply(degres, function(p) {
  lm(height_m ~ poly(diameter_cm, p, raw = TRUE), data = arbres)
})
vc.n <- sapply(modeles, function(m) mean((residuals(m) / (1 -
 ↪  hatvalues(m)))^2))
rbind(degre = degres, vc.n = round(vc.n, 3),
      AIC = round(sapply(modeles, AIC), 1), BIC = round(sapply(modeles,
       ↪  BIC), 1))
```

```
         [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
degre   1.000   2.000   3.000   4.000   5.000   6.000
vc.n    6.241   6.217   6.306   6.641   6.949   6.367
AIC   934.400 932.300 933.400 934.700 936.500 933.700
BIC   944.300 945.500 949.900 954.500 959.600 960.100
```

La LOOCV et l'AIC retiennent le degré 2 et le BIC la droite, mais les écarts entre les degrés 1, 2 et 3 sont minces. Le polynôme de degré 2 a un coefficient négatif sur x^2, ce qui traduit la concavité de la relation. Au-delà du degré 3, l'erreur de validation croisée est plus élevée, parce que les termes supplémentaires suivent le bruit des observations. Le degré 2 donne $VC_{(n)} \approx 6{,}2$, du même ordre que l'estimateur à noyau du Chapitre 7 ($VC_{(n)}(h) \approx 6{,}7$ pour $h = 2$ cm).

```
grille <- data.frame(
  diameter_cm = seq(min(arbres$diameter_cm), max(arbres$diameter_cm),
   ↪  length.out = 300)
)
```

```r
ajust <- do.call(rbind, lapply(c(1, 2, 5), function(p) {
  data.frame(p = paste("p =", p), grille, f = predict(modeles[[p]],
   ↪  newdata = grille))
}))
bande <- cbind(grille, predict(modeles[[2]], newdata = grille, interval =
 ↪  "confidence"))

ggplot(arbres, aes(diameter_cm, height_m)) +
  geom_ribbon(data = bande, aes(x = diameter_cm, ymin = lwr, ymax = upr),
              inherit.aes = FALSE, alpha = 0.2) +
  geom_point(alpha = 0.2) +
  geom_line(data = ajust, aes(diameter_cm, f, colour = p), linewidth =
   ↪  0.8) +
  labs(x = "Diamètre (cm)", y = "Hauteur (m)", colour = NULL) +
  theme_minimal(base_size = 13)
```

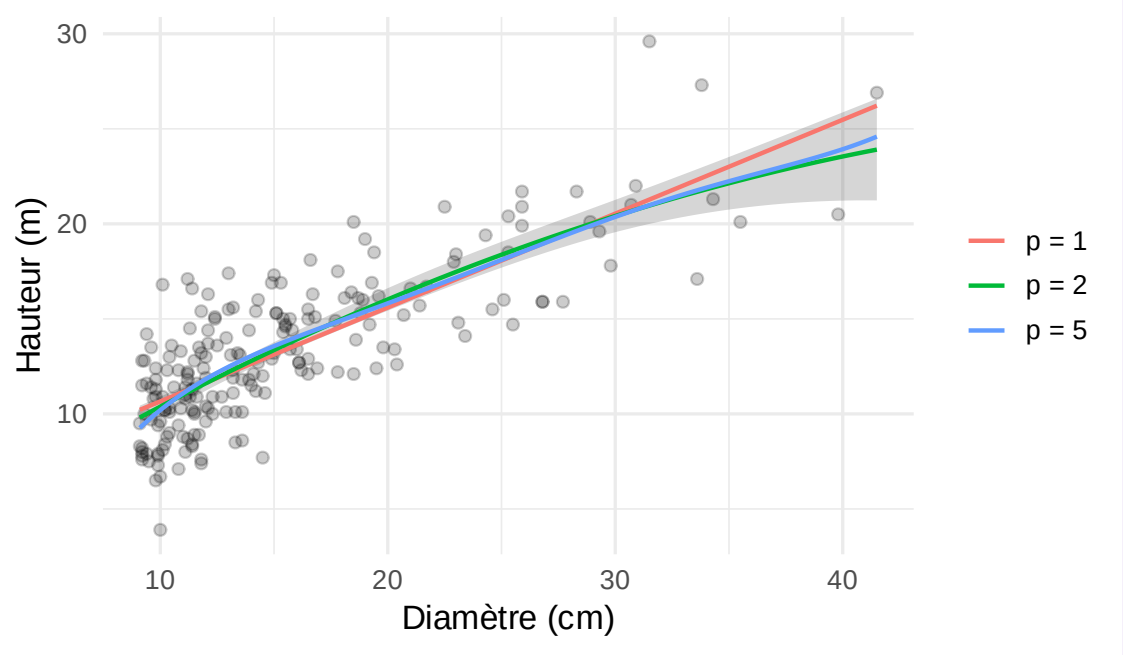


FIGURE 8.1 : Régression polynomiale de la hauteur sur le diamètre pour $p = 1$, 2 et 5. Pour $p = 2$, la bande grise est l'intervalle de confiance ponctuel à 95 % de la moyenne conditionnelle $f(x)$; ce n'est pas un intervalle de prédiction pour la hauteur d'un nouvel arbre.

La bande de confiance, obtenue avec `predict(..., interval = "confidence")`, s'élar-

git au-delà de 30 cm, là où les arbres sont rares, parce que $Var[\hat{f}(x)] = x_* Var[\hat{\beta}]\, x_*^\top$ croît avec les puissances de x. Le polynôme de degré 5 s'écarte légèrement des autres aux extrémités du domaine, là où les données sont rares. Avec un degré plus élevé, cet écart deviendrait erratique. Ce comportement des polynômes de degré élevé aux bords est un défaut, sur lequel nous reviendrons avec les splines naturelles (Section 8.3.4).
Il faut reconnaître que ce jeu de données n'est pas idéal pour la régression polynomiale. Comme la relation est monotone et que sa courbure est modérée, la droite fait déjà l'essentiel du travail, et les critères peinent d'ailleurs à départager les degrés 1 à 3. Les polynômes sont à leur avantage lorsque la courbe change de direction à l'intérieur du domaine, comme dans l'exemple suivant.

Exemple — Jeu de données qualite_air

Le jeu de données `qualite_air` (⬇) contient les mesures horaires de 2025 des 42 stations du Réseau de surveillance de la qualité de l'air du Québec (RSQAQ). Ces mesures sont diffusées par Données Québec et préparées par donneesbleues.ca. Chaque ligne correspond à une station (`Station`) et à une heure (`Date_Heure`, en heure normale de l'Est), l'heure indiquée marquant la fin de l'intervalle mesuré. Les colonnes donnent la concentration de différents contaminants, notamment l'ozone (`O3`, en ppb), le dioxyde d'azote (`NO2`, en ppb) et les particules fines (`PM2.5-T640`, en µg/m³). Le nombre de stations qui mesurent un contaminant varie de 2 à 40, d'où de nombreuses valeurs manquantes.

Exemple — Cycle journalier de l'ozone sur `qualite_air`

L'ozone troposphérique se forme par photochimie à partir des oxydes d'azote, sous l'effet du rayonnement solaire. Sa concentration suit donc un cycle journalier marqué, surtout en été. On retient les heures de juin, juillet et août à la station de Longueuil, soit $n = 2\,204$ mesures, et on modélise la concentration d'ozone Y en fonction de l'heure $X \in \{0, 1, \ldots, 23\}$.

```
air <- read.csv("Jeux de données/qualite_air.csv", fileEncoding =
 ↪  "UTF-8-BOM") |>
  mutate(heure = as.integer(substr(Date_Heure, 12, 13)),
         mois  = as.integer(substr(Date_Heure, 6, 7)))
ozone <- air |>
  filter(Station == "06600 - Longueuil", mois %in% 6:8, !is.na(O3))


ggplot(ozone, aes(heure, O3)) +
  geom_jitter(width = 0.25, height = 0, alpha = 0.08, size = 0.7) +
  labs(x = "Heure (HNE)", y = "Ozone (ppb)") +
  theme_minimal(base_size = 13)
```

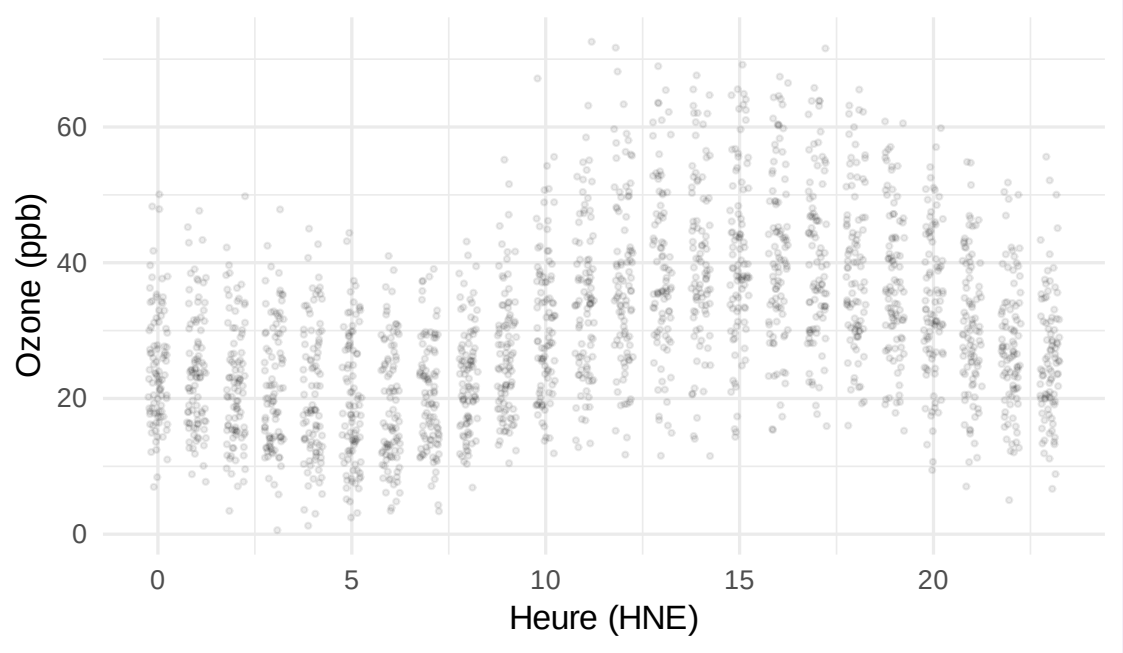


FIGURE 8.2 : Concentration horaire d'ozone à Longueuil, juin à août 2025, en fonction de l'heure. Les points sont légèrement décalés horizontalement pour la lisibilité.

Le nuage de points montre une bosse nette : la concentration est minimale au petit matin, monte rapidement en avant-midi, culmine en fin d'après-midi, puis redescend plus lentement en soirée. Comme pour les arbres, on ajuste des polynômes de degré 1 à 8 et on choisit p par LOOCV.

```
degres <- 1:8
modeles.o3 <- lapply(degres, function(p) {
  lm(O3 ~ poly(heure, p, raw = TRUE), data = ozone)
})
vc.o3 <- sapply(modeles.o3, function(m) mean((residuals(m) / (1 -
 ↪  hatvalues(m)))^2))
rbind(degre = degres, vc.n = round(vc.o3, 1), BIC =
 ↪  round(sapply(modeles.o3, BIC), 1))
```

```
          [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]
degre      1.0     2.0     3.0     4.0     5.0     6.0     7.0     8.0
vc.n     145.1   135.6   112.2   111.4   109.4   109.3   109.4   109.3
BIC    17244.5 17101.4 16688.6 16679.5 16645.9 16649.1 16656.8 16659.8
```

Le contraste avec `arbres` est net. La droite est inadéquate (la courbe a un minimum vers 5 h et un maximum vers 16 h). Le degré 2 ne suffit pas non plus, parce que la bosse est asymétrique (montée rapide le matin, descente plus lente en soirée). Entre les degrés 1 et 3, $VC_{(n)}$ chute d'environ 23 %, puis le critère devient pratiquement plat. Le BIC retient $p = 5$, mais les degrés 3 à 8 donnent des courbes presque identiques. Avec $n = 2\,204$, chaque paramètre supplémentaire coûte peu en variance : c'est le biais qui décide.

```
grille <- data.frame(heure = seq(0, 23, length.out = 300))
ajust <- do.call(rbind, lapply(c(1, 3, 5), function(p) {
  data.frame(p = paste("p =", p), grille, f = predict(modeles.o3[[p]],
   ↪  newdata = grille))
}))
bande <- cbind(grille, predict(modeles.o3[[5]], newdata = grille,
 ↪  interval = "confidence"))
pred  <- cbind(grille, predict(modeles.o3[[5]], newdata = grille,
 ↪  interval = "prediction"))

ggplot(ozone, aes(heure, O3)) +
  geom_jitter(width = 0.25, height = 0, alpha = 0.08, size = 0.7) +
  geom_ribbon(data = bande, aes(x = heure, ymin = lwr, ymax = upr),
              inherit.aes = FALSE, alpha = 0.3) +
  geom_line(data = ajust, aes(heure, f, colour = p), linewidth = 0.8) +
  geom_line(data = pred, aes(heure, lwr), linetype = "dotted") +
  geom_line(data = pred, aes(heure, upr), linetype = "dotted") +
  labs(x = "Heure (HNE)", y = "Ozone (ppb)", colour = NULL) +
  theme_minimal(base_size = 13)
```

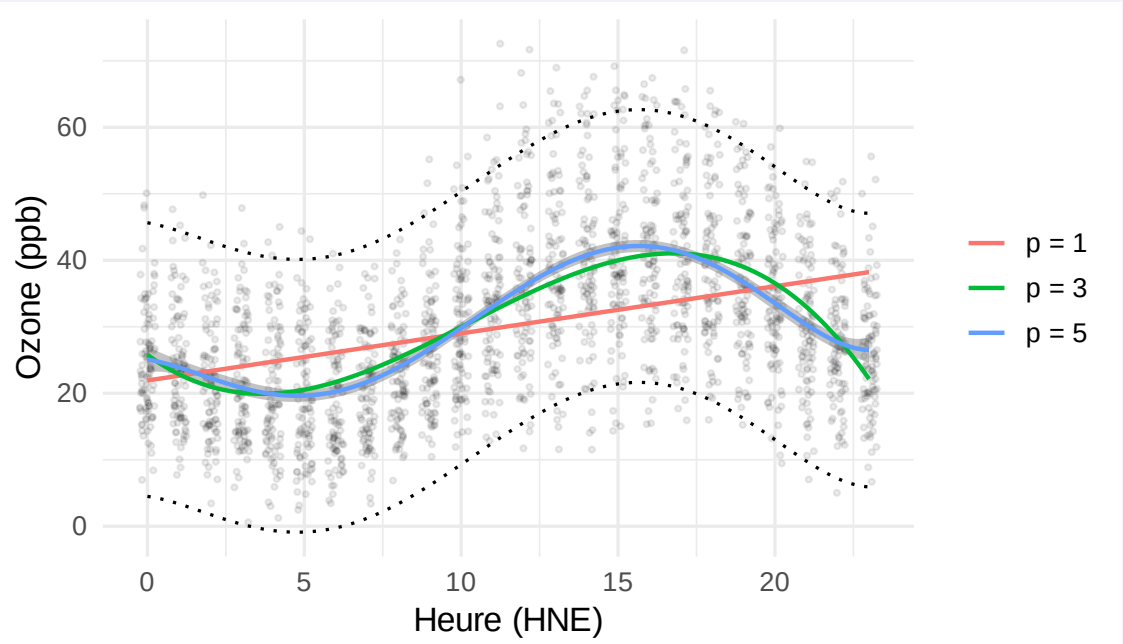


FIGURE 8.3 : Concentration horaire d'ozone à Longueuil (juin à août 2025) et régression polynomiale pour $p = 1$, 3 et 5. Pour $p = 5$, la bande grise est l'intervalle de confiance ponctuel à 95 % de la moyenne conditionnelle $f(x)$ et les pointillés délimitent l'intervalle de prédiction à 95 % d'une observation future. Les deux supposent des erreurs indépendantes, alors que les mesures horaires d'une même journée sont corrélées (voir la remarque plus bas). Les points sont légèrement décalés horizontalement pour la lisibilité.

Cet exemple est un cas favorable à la régression polynomiale, pour trois raisons. D'abord, la courbe est lisse et change deux fois de direction à l'intérieur du domaine, de sorte qu'un polynôme de degré modéré la décrit avec une poignée de paramètres. Ensuite, la variable explicative ne prend que 24 valeurs distinctes, chacune répétée une centaine de fois. L'estimateur kNN du Chapitre 7 s'en accommode mal (les voisins sont tous ex æquo), alors qu'un modèle linéaire en β n'y voit aucune difficulté. Enfin, la bande de confiance reste étroite sur tout le domaine (demi-largeur d'au plus 1,9 ppb), même si elle s'élargit un peu aux bords, parce que les observations y sont aussi nombreuses qu'ailleurs.

Remarque — Bande de confiance ou bande de prédiction ?

La bande grise de la Figure 8.3 peut sembler étroite par rapport à la dispersion des points. Il ne faut pas la voir comme une bande qui contient les observations. Elle porte sur la courbe $f(x) = E[Y \mid X = x]$, c'est-à-dire qu'elle répond à la question « où se trouve la

concentration moyenne à cette heure ? ». Sa demi-largeur est d'environ 0,8 ppb au centre du domaine. C'est moins que $2\hat{\sigma}/\sqrt{n_x} \approx 2$ ppb, où n_x est le nombre d'observations à l'heure x (avec $\hat{\sigma} \approx 10{,}4$ ppb et $n_x \approx 92$), parce que la courbe combine l'information de toutes les heures. Une nouvelle mesure Y_0 s'encadre plutôt par l'intervalle de prédiction, obtenu avec `predict(..., interval = "prediction")`. Sa demi-largeur est d'environ $2\hat{\sigma}$, soit une vingtaine de ppb, et il contient l'essentiel du nuage (pointillés). Le cycle journalier n'explique qu'une partie de la variabilité de l'ozone ; le reste tient à la météo et aux émissions de chaque jour.

Les deux intervalles supposent des erreurs indépendantes, alors que les mesures d'heures consécutives d'une même journée sont fortement corrélées. Par exemple, une journée chaude et ensoleillée reste au-dessus de la courbe du matin au soir. Le nombre effectif d'observations indépendantes est donc bien inférieur à $n = 2\,204$, plus proche du nombre de jours (92), et la bande de confiance sous-estime l'incertitude réelle sur f. Le Chapitre 10 présente des modèles qui tiennent compte de ce type de dépendance, et il reprend cet exemple avec des erreurs autorégressives à l'intérieur de chaque jour : la bande de confiance s'en trouve nettement élargie. D'ici là, gardez cette limite en tête lorsque vous appliquez une méthode de régression à une série chronologique.

Pour aller plus loin avec la théorie — Interprétation des coefficients β_1, β_2, ...

Dans la régression linéaire simple, β_1 s'interprète directement : une augmentation d'une unité de x fait varier $E[Y \mid X = x]$ de β_1. Dès que $p \geq 2$, cette lecture ne tient plus. On ne peut pas augmenter x d'une unité « en maintenant x^2 constant », parce que les termes $x, x^2, \ldots, x^p$ varient ensemble. L'effet d'une variation de x se lit plutôt sur la dérivée de la courbe,

$$f'(x) = \beta_1 + 2\beta_2\, x + 3\beta_3\, x^2 + \cdots + p\,\beta_p\, x^{p-1},$$

une pente *locale* qui dépend de l'endroit où l'on se trouve.

- β_1 est la pente de la courbe en $x = 0$, puisque $f'(0) = \beta_1$. Cette interprétation n'est utile que si $x = 0$ appartient au domaine des données ; sinon, on peut centrer la variable (remplacer x par $x - \bar{x}$) pour que β_1 devienne la pente au point moyen.
- β_2 gouverne la courbure : $f''(0) = 2\beta_2$. Pour $p = 2$, la courbure $2\beta_2$ est la même partout. Si $\beta_2 < 0$, la parabole est concave et culmine en $x^* = -\beta_1/(2\beta_2)$, comme pour la hauteur des arbres, dont la croissance ralentit avec le diamètre. Si $\beta_2 > 0$, la parabole est convexe et atteint son minimum au même point.
- En général, $\beta_k = f^{(k)}(0)/k!$: les coefficients sont ceux du développement de Taylor de f autour de 0. Au-delà de $k = 2$, ils n'ont pas d'interprétation pratique isolée, puisque leur valeur change du tout au tout si on recentre x ou si on passe du degré p au degré $p+1$. On interprète donc le modèle par la courbe ajustée dans son ensemble, plutôt que coefficient par coefficient, en regardant son tracé, l'emplacement de ses extremums ou sa pente $\widehat{f}'(x)$ en un point d'intérêt.

8.2 Régression avec des fonctions constantes par morceaux

Notons $[A, B]$ le domaine de X (en pratique $A = \min_i X_i - \delta$ et $B = \max_i X_i + \delta$ pour un petit $\delta > 0$). On choisit une partition $A = c_0 < c_1 < \cdots < c_K = B$ et on définit, pour $k = 1, \ldots, K$,

$$b_k(x) = \begin{cases} 1 & \text{si } x \in [c_{k-1}, c_k), \\ 0 & \text{sinon.} \end{cases}$$

On approxime alors f par une fonction en escalier, $f(x) = \beta_1 b_1(x) + \cdots + \beta_K b_K(x)$, ce qui revient à discrétiser X : on retrouve un modèle d'ANOVA à un facteur à K modalités. L'estimateur des moindres carrés est

$$\hat{\beta}_k = \frac{\sum_{i=1}^n b_k(X_i) Y_i}{n_k}, \qquad n_k = \sum_{i=1}^n b_k(X_i),$$

c'est-à-dire la moyenne des Y_i dans l'intervalle k, et $Var[\hat{\beta}_k] = \sigma^2 / n_k$. En R : `lm(O3 ~ cut(heure, breaks = c(0, 6, 12, 18, 24), right = FALSE), data = ozone)`, où `right = FALSE` donne les intervalles $[0, 6)$, $[6, 12)$, $[12, 18)$ et $[18, 24)$, fermés à gauche comme dans la définition.

Exemple — Fonctions constantes par morceaux sur `qualite_air`

Découpons les 24 heures en K intervalles de même largeur et comparons les valeurs de K par LOOCV. Avec $K = 24$, chaque intervalle contient une seule heure : $\hat{f}$ est simplement la moyenne horaire.

```
Ks <- c(2, 3, 4, 6, 8, 12, 24)
modeles.cut <- lapply(Ks, function(K) {
  # bornes fixes : predict() réutilise les mêmes intervalles sur une
   ↪  autre grille
  # right = FALSE : intervalles [c_{k-1}, c_k), comme dans la définition
  bornes <- seq(0, 24, length.out = K + 1)
  lm(O3 ~ cut(heure, bornes, right = FALSE), data = ozone)
})
vc.cut <- sapply(modeles.cut, function(m) mean((residuals(m) / (1 -
 ↪  hatvalues(m)))^2))
rbind(K = Ks, vc.n = round(vc.cut, 1))
```

```
      [,1] [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
K      2.0    3   4.0   6.0   8.0  12.0  24.0
vc.n 131.7  133 123.5 114.7 112.3 110.6 110.7
```

```r
escalier <- do.call(rbind, lapply(c(2, 5), function(j) {
  data.frame(K = paste("K =", Ks[j]), grille, f =
   ↪  predict(modeles.cut[[j]], newdata = grille))
}))

ggplot(ozone, aes(heure, O3)) +
  geom_jitter(width = 0.25, height = 0, alpha = 0.08, size = 0.7) +
  geom_step(data = escalier, aes(heure, f, colour = K), linewidth = 0.8)
   ↪  +
  labs(x = "Heure (HNE)", y = "Ozone (ppb)", colour = NULL) +
  theme_minimal(base_size = 13)
```

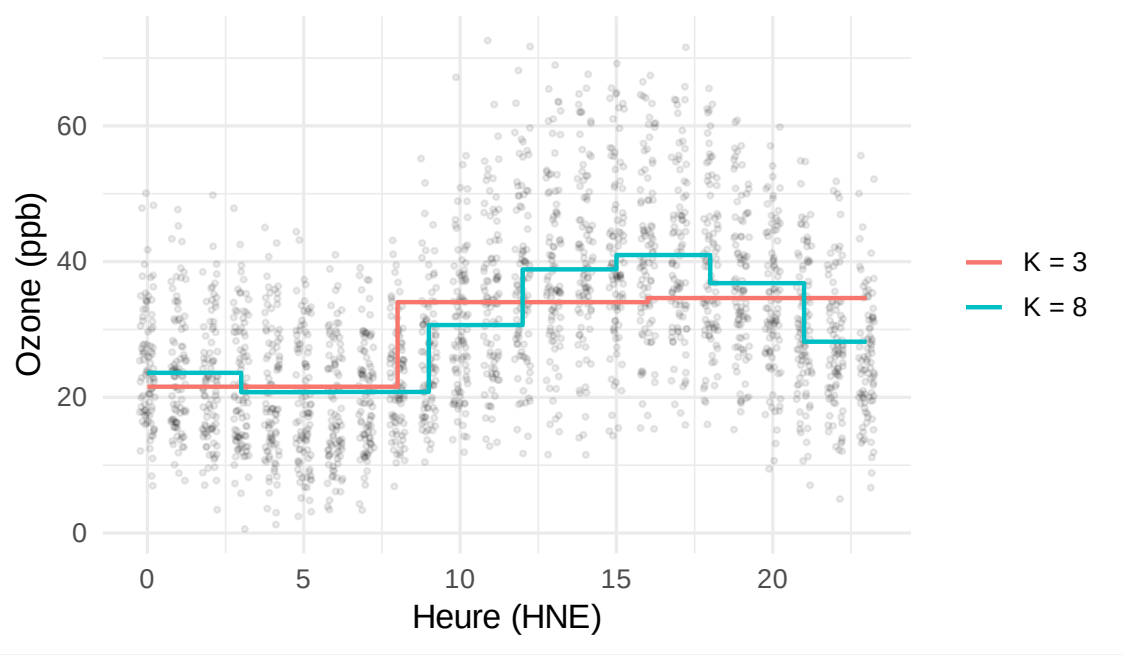


FIGURE 8.4 : Fonctions constantes par morceaux ajustées au cycle journalier de l'ozone pour $K = 3$ et $K = 8$ intervalles.

L'erreur de validation croisée diminue globalement jusqu'à $K = 12$, puis stagne : les moyennes horaires ($K = 24$) ne font pas mieux que $K = 12$ (110,7 contre 110,6), pour deux fois plus de paramètres. Même au mieux, l'approche reste en retrait du polynôme de degré 5 ($VC_{(n)} \approx 109{,}4$ avec 6 paramètres), parce qu'une fonction en escalier remplace la courbe lisse par une suite de plateaux. À l'intérieur de chaque intervalle, elle ignore la pente, et les sauts entre intervalles n'ont aucun sens physique. L'intérêt de la méthode est

ailleurs. Elle montre que discrétiser X revient à un modèle linéaire, ce qui ouvre la voie aux splines.

8.3 Régression avec des splines

Remarque — Modéliser plutôt qu'interpréter

Dans les chapitres précédents, les coefficients avaient un sens : une pente, un rapport de cotes, l'effet d'une classe. Ce n'est plus le cas ici. Les paramètres β_k d'une spline multiplient des fonctions de base qui n'ont pas d'interprétation propre, de sorte qu'il est inutile de les tester un à un ou de les commenter. L'objectif est de décrire le mieux possible la relation entre x et y, c'est-à-dire d'estimer la courbe $f(x) = E[Y \mid X = x]$ et de prédire avec précision. Les questions pertinentes portent sur la forme de $\hat{f}$, sur l'erreur de prédiction estimée par validation croisée et sur le nombre de degrés de liberté qu'on accorde à la courbe. La valeur de tel ou tel coefficient n'en fait pas partie.
Le même état d'esprit vaut pour la lecture de cette section. Je donne les détails techniques (récurrence de Cox-de Boor, conditionnement des bases, calcul de la matrice de pénalité) pour montrer comment les logiciels procèdent, mais il n'est pas nécessaire de les maîtriser pour utiliser ces méthodes. L'essentiel est de savoir ce que fait chaque méthode et ce qui la distingue des autres, quel paramètre contrôle sa flexibilité (le degré, le nombre et la position des nœuds, le paramètre λ) et comment comparer par validation croisée les ajustements obtenus sur un même jeu de données.

8.3.1 Splines de régression

Les splines sont des polynômes définis par intervalles, avec des contraintes de régularité aux points de jonction (les nœuds, *knots*).

Prenons d'abord le cas simple de $K = 2$ intervalles et du degré $p = 3$. Le domaine $[A, B]$ est divisé en $[c_0, c_1)$ et $[c_1, c_2)$, et f est approximée par

$$f(x) = \begin{cases} \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 & \text{si } x \in [c_0, c_1), \\ \gamma_0 + \gamma_1 x + \gamma_2 x^2 + \gamma_3 x^3 & \text{si } x \in [c_1, c_2). \end{cases}$$

Ce modèle a $4 + 4 = 8$ degrés de liberté. Si on impose que f, f' et f'' soient continues en c_1 (f deux fois continûment dérivable, 3 contraintes), il en reste $8 - 3 = 5$.

Dans le cas général, avec K intervalles et un degré p, il y a $K(p+1)$ paramètres et $p(K-1)$ contraintes (continuité de $f, f', \ldots, f^{(p-1)}$ aux $K-1$ nœuds intérieurs), d'où

$$K(p+1) - p(K-1) = K + p \text{ degrés de liberté.}$$

Exemple — Deux cubiques, avec ou sans contraintes, sur `qualite_air`

Coupons la journée à midi ($c_1 = 12$) et comparons deux ajustements. Le premier est formé de deux polynômes cubiques estimés séparément sur $[0, 12)$ et $[12, 23]$, soit 8 paramètres. Le second est la spline cubique à un nœud, qui impose la continuité de f, f' et f'' à midi et ne compte que 5 paramètres. Elle s'obtient avec `lm`, en ajoutant au polynôme cubique la seule fonction $(x-12)_+^3$ (voir la section suivante).

```
ozone  <- mutate(ozone,  apres = heure >= 12)
grille <- mutate(grille, apres = heure >= 12)
m.libre  <- lm(O3 ~ apres * poly(heure, 3, raw = TRUE), data = ozone)
m.spline <- lm(O3 ~ heure + I(heure^2) + I(heure^3) + I(pmax(heure - 12,
 ↪  0)^3), data = ozone)
c(libre = length(coef(m.libre)), spline = length(coef(m.spline)))
```

```
 libre spline
     8      5
```

```
vc.cub <- sapply(list(libre = m.libre, spline = m.spline),
                 function(m) mean((residuals(m) / (1 - hatvalues(m)))^2))
vc.cub
```

```
   libre   spline
109.4167 111.1680
```

```
morceaux <- rbind(
  data.frame(m = "Deux cubiques libres (8 paramètres)", grille,
             f = predict(m.libre, newdata = grille)),
  data.frame(m = "Spline cubique, nœud à 12 h (5 paramètres)", grille,
             f = predict(m.spline, newdata = grille))
)

ggplot(ozone, aes(heure, O3)) +
  geom_jitter(width = 0.25, height = 0, alpha = 0.08, size = 0.7) +
  geom_line(data = morceaux, aes(heure, f, colour = m, group =
   ↪  interaction(m, apres)),
            linewidth = 0.8) +
  geom_vline(xintercept = 12, linetype = "dashed", colour = "grey50") +
  labs(x = "Heure (HNE)", y = "Ozone (ppb)", colour = NULL) +
  theme_minimal(base_size = 13)
```

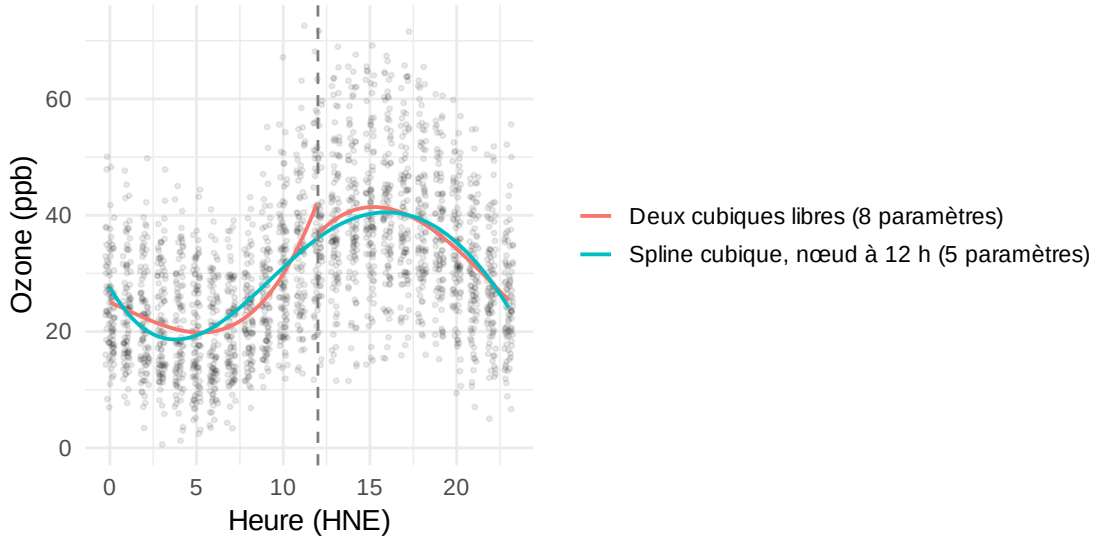


FIGURE 8.5 : Deux cubiques ajustées séparément de part et d'autre de midi (8 paramètres) et spline cubique à un nœud en 12 h (5 paramètres).

Les deux cubiques libres présentent un saut d'environ 5,7 ppb à midi, sans justification physique, puisque rien ne se passe à midi qui ne se passe pas à 11 h 59. Les trois contraintes de régularité suppriment ce défaut au prix d'une légère hausse de $VC_{(n)}$ (111 contre 109), et la courbe obtenue a l'allure attendue. C'est l'intérêt des splines : des polynômes par morceaux, mais raccordés.

8.3.2 Base des puissances tronquées

Pour p et K donnés, un choix commode de base de $K+p$ splines est

$$b_1(x) = 1, \quad b_2(x) = x, \ \ldots, \ b_{p+1}(x) = x^p, \quad b_{p+1+k}(x) = h(x, c_k), \ k = 1, \ldots, K-1,$$

où

$$h(x, c) = (x - c)_+^p = \begin{cases} (x-c)^p & \text{si } x \geq c, \\ 0 & \text{sinon.} \end{cases}$$

Il existe une infinité de bases équivalentes, puisque n'importe quelles $K+p$ combinaisons linéaires indépendantes de cette base forment encore une base de splines.

Le modèle s'écrit alors

$$Y_i = \beta_1 b_1(X_i) + \cdots + \beta_{K+p}\, b_{K+p}(X_i) + \epsilon_i,$$

et l'estimateur des moindres carrés est $\hat{\beta} = (X^\top X)^{-1} X^\top Y$, avec $X_{ik} = b_k(X_i)$, ce qui donne $\hat{f}(x) = \sum_k \hat{\beta}_k b_k(x)$.

Exemple — Base des puissances tronquées sur `qualite_air`

Prenons $p = 3$ et des nœuds intérieurs à 6 h, 12 h et 18 h ($K = 4$). La base compte alors $K + p = 7$ fonctions : 1, x, x^2, x^3, $(x-6)_+^3$, $(x-12)_+^3$ et $(x-18)_+^3$. On construit la matrice X à la main et on ajuste le modèle par `lm`.

```
h <- function(x, c) pmax(x - c, 0)^3
X.pt <- with(ozone, cbind(1, heure, heure^2, heure^3, h(heure, 6),
 ↪  h(heure, 12), h(heure, 18)))
m.pt <- lm(O3 ~ X.pt - 1, data = ozone)
round(coef(m.pt), 4)
```

```
     X.pt X.ptheure       X.pt       X.pt       X.pt       X.pt       X.pt
  24.4590    0.2275   -0.6356    0.0761   -0.1399    0.0662    0.0457
```

```
mean((residuals(m.pt) / (1 - hatvalues(m.pt)))^2)
```

```
[1] 109.1377
```

```
kappa.pt <- kappa(crossprod(X.pt), exact = TRUE)   # nombre de condition
 ↪  de X'X
kappa.pt
```

```
[1] 982328687
```

Les coefficients de $(x - c_k)_+^3$ se lisent comme des changements du coefficient cubique à chaque nœud, ce qui est peu parlant. Leurs ordres de grandeur sont aussi disparates : la colonne x^3 dépasse 10^4, alors que la première vaut 1. Le nombre de condition de $X^\top X$ (rapport de la plus grande à la plus petite valeur propre) est ici d'environ $9{,}8 \times 10^8$, soit de l'ordre de 10^9. La base est mathématiquement correcte, mais numériquement fragile, et elle le devient davantage à mesure qu'on ajoute des nœuds. C'est ce qui motive la base de la section suivante, qui engendre le même espace de fonctions, donc le même $\hat{f}$, avec une matrice bien conditionnée.

8.3.3 B-splines

La base des puissances tronquées est simple, mais numériquement instable ($X^\top X$ mal conditionnée, comme l'exemple précédent l'a montré). En pratique, les logiciels utilisent plutôt la base des B-splines, une base équivalente dont les fonctions sont à support compact. Chaque B-spline de degré p est non nulle sur au plus $p+1$ intervalles consécutifs.

Les B-splines se définissent par la récurrence de Cox-de Boor. Sur une suite de nœuds augmentée $\{t_j\}$, les B-splines de degré 0 sont

$$B_{j,0}(x) = \mathbf{1}_{\{t_j \le x < t_{j+1}\}},$$

et pour $d = 1, \dots, p$:

$$B_{j,d}(x) = \frac{x - t_j}{t_{j+d} - t_j} B_{j,d-1}(x) + \frac{t_{j+d+1} - x}{t_{j+d+1} - t_{j+1}} B_{j+1,d-1}(x),$$

avec la convention $0/0 = 0$, puisque les nœuds frontières sont répétés dans la suite augmentée. Ces fonctions ont les propriétés suivantes :

- positivité : chaque $B_{j,p}(x) \ge 0$;
- partition de l'unité : $\sum_j B_{j,p}(x) = 1$ pour tout $x \in [A, B]$;
- structure creuse : la matrice de design est creuse par bandes, ce qui rend les calculs stables et rapides.

En R, la fonction `bs` de la librairie `splines` construit la base de B-splines cubiques.

Exemple — B-splines cubiques sur `qualite_air`

Les trois nœuds intérieurs placés à 6 h, 12 h et 18 h donnent $K = 4$ intervalles et $K + p = 7$ degrés de liberté (ordonnée à l'origine comprise). La Figure 8.6 montre les six fonctions de base produites par `bs`, chacune non nulle sur au plus quatre intervalles consécutifs. Remarquez que `bs` omet par défaut la première B-spline, que l'ordonnée à l'origine du modèle remplace. La somme des sept B-splines complètes vaut 1 partout, alors que celle des six colonnes de `bs` vaut $1 - B_1(x)$, qui est inférieure à 1 sur le premier intervalle $[0, 6)$.

```
library(splines)
B <- bs(grille$heure, knots = c(6, 12, 18))
dim(B)
```

```
[1] 300   6
```

```
base <- data.frame(heure = grille$heure, k = factor(rep(1:ncol(B), each =
 ↪  nrow(B))),
```

```
                    b = as.vector(B))

ggplot(base, aes(heure, b, colour = k)) +
  geom_line(linewidth = 0.8) +
  geom_vline(xintercept = c(6, 12, 18), linetype = "dashed", colour =
   ↪  "grey50") +
  labs(x = "Heure (HNE)", y = expression(b[k](x)), colour = NULL) +
  theme_minimal(base_size = 13)
```

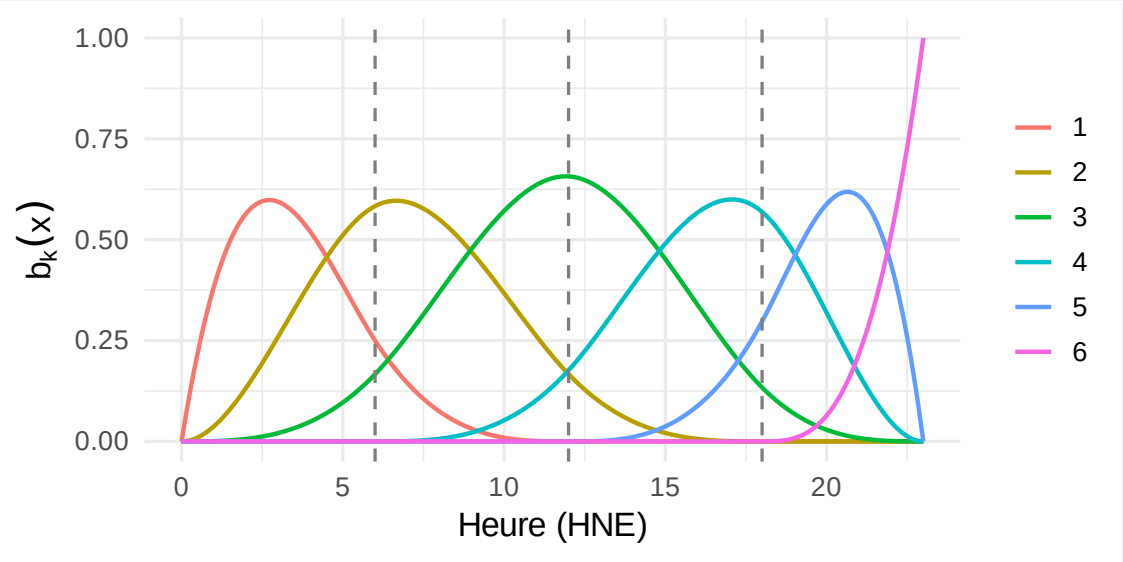


FIGURE 8.6 : Base de B-splines cubiques sur $[0, 23]$ avec nœuds intérieurs à 6, 12 et 18 h (tirets).

Le modèle s'ajuste avec `lm` comme n'importe quel modèle linéaire :

```
modele.bs <- lm(O3 ~ bs(heure, knots = c(6, 12, 18)), data = ozone)
summary(modele.bs)$coefficients
```

```
                                  Estimate Std. Error    t value
(Intercept)                     24.4589975  0.9926665 24.6396932
bs(heure, knots = c(6, 12, 18))1   0.4549517  2.0147284  0.2258129
bs(heure, knots = c(6, 12, 18))2 -13.8897033  1.5748454 -8.8197250
bs(heure, knots = c(6, 12, 18))3  17.4597645  1.7217356 10.1407932
bs(heure, knots = c(6, 12, 18))4  18.8803791  1.6895631 11.1747109
```

```
bs(heure, knots = c(6, 12, 18))5   4.8291070  1.7470414  2.7641629
bs(heure, knots = c(6, 12, 18))6   1.3632836  1.4136373  0.9643801
                                      Pr(>|t|)
(Intercept)                       1.442856e-118
bs(heure, knots = c(6, 12, 18))1  8.213679e-01
bs(heure, knots = c(6, 12, 18))2  2.287619e-18
bs(heure, knots = c(6, 12, 18))3  1.196750e-23
bs(heure, knots = c(6, 12, 18))4  3.081624e-28
bs(heure, knots = c(6, 12, 18))5  5.754533e-03
bs(heure, knots = c(6, 12, 18))6  3.349615e-01
```

```
vc.bs7 <- mean((residuals(modele.bs) / (1 - hatvalues(modele.bs)))^2)
vc.bs7
```

```
[1] 109.1377
```

```
kappa.bs <- kappa(crossprod(model.matrix(modele.bs)), exact = TRUE)   #
 ↪  nombre de condition de X'X
kappa.bs
```

```
[1] 211.7128
```

Les coefficients individuels ne s'interprètent pas, parce que chaque b_k n'agit que localement. Les valeurs ajustées et $VC_{(n)}$ sont exactement celles de la base des puissances tronquées avec les mêmes nœuds, ce qui est normal puisque les deux bases engendrent le même espace. En revanche, le nombre de condition de $X^\top X$ est passé d'environ 10^9 à environ 210. La courbe, tracée plus loin à la Figure 8.8, est pratiquement confondue avec le polynôme de degré 5, et la LOOCV est du même ordre ($VC_{(n)} \approx 109{,}1$). Sur une courbe aussi régulière, les deux bases décrivent la même fonction. La spline a quand même deux avantages : elle reste sage aux extrémités, et elle permet d'ajouter des nœuds là où la courbe change vite sans augmenter le degré partout.

8.3.4 Splines naturelles

L'estimateur par splines peut être instable aux extrémités du domaine de X, à cause du comportement erratique des polynômes au-delà des nœuds extrêmes. Les splines naturelles corrigent ce défaut. Elles sont définies exactement comme les splines cubiques, avec la contrainte

supplémentaire que la fonction soit linéaire au-delà des nœuds frontières c_0 et c_K. Par continuité de f'', cela revient à imposer $f''(c_0) = f''(c_K) = 0$, soit 2 contraintes, et une spline naturelle cubique compte donc $K + 3 - 2 = K + 1$ degrés de liberté. Ces 2 degrés de liberté libérés peuvent être réinvestis en nœuds intérieurs. En R : `lm(O3 ~ ns(heure, df = 4), data = ozone)`. L'argument `df` fixe le nombre de degrés de liberté, sans compter l'ordonnée à l'origine, et `ns` place les nœuds intérieurs aux quantiles de X. Les nœuds frontières de `ns` sont par défaut le minimum et le maximum de X. Sur l'étendue des données, la fonction ajustée reste donc cubique dans les intervalles extrêmes, mais avec une dérivée seconde nulle aux deux bords.

Exemple — Splines de régression et splines naturelles sur `qualite_air`

Comparons par LOOCV la spline cubique (`bs`) et la spline naturelle (`ns`), pour un même nombre de degrés de liberté.

```
dfs <- 3:10
vc.bs <- sapply(dfs, function(d) {
  m <- lm(O3 ~ bs(heure, df = d), data = ozone)
  mean((residuals(m) / (1 - hatvalues(m)))^2)
})
vc.ns <- sapply(dfs, function(d) {
  m <- lm(O3 ~ ns(heure, df = d), data = ozone)
  mean((residuals(m) / (1 - hatvalues(m)))^2)
})
rbind(df = dfs, bs = round(vc.bs, 1), ns = round(vc.ns, 1))
```

```
    [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]
df   3.0   4.0   5.0   6.0   7.0   8.0   9.0  10.0
bs 112.2 111.4 109.6 109.2 109.5 109.2 109.3 109.4
ns 110.0 109.8 109.5 109.1 109.2 109.2 109.3 109.4
```

Comme pour `ns`, l'argument `df` de `bs` ne compte pas l'ordonnée à l'origine : `df = 6` correspond à 7 paramètres. À degrés de liberté égaux, la spline naturelle fait un peu mieux pour les petites valeurs de `df`. En fait, les deux degrés de liberté que `bs` consacre à la courbure aux deux bords ($f''(c_0)$ et $f''(c_K)$ libres), `ns` les investit en nœuds intérieurs. À partir de six degrés de liberté, les deux se valent et rejoignent le polynôme de degré 5 et les B-splines à nœuds fixes. Avec $n = 2\,204$ et une courbe sans comportement erratique aux bords, l'avantage des splines naturelles est modeste. Il devient important lorsque les observations sont rares aux extrémités, comme pour les gros arbres d'`arbres`.

8.3.5 Splines de lissage (splines pénalisées)

Un autre défaut des splines de régression est qu'elles peuvent avoir tendance à « aller chercher » le maximum de points (surajustement). Pour y remédier, il suffit d'imposer une pénalité sur la rugosité de f, c'est-à-dire d'estimer β en minimisant

$$\sum_{i=1}^{n} \left\{ Y_i - \sum_k \beta_k b_k(X_i) \right\}^2 + \lambda \int_A^B f''(t)^2 \, dt.$$

Comme $f = \sum_k \beta_k b_k$, la pénalité s'écrit $\lambda \beta^\top M \beta$, avec $M_{lq} = \int_A^B b_l''(t)\, b_q''(t)\, dt$. L'estimateur qui en résulte est explicite :

$$\boxed{\hat{\beta}_\lambda = (X^\top X + \lambda M)^{-1} X^\top Y}$$

Le paramètre λ contrôle la régularité de $\hat{f}$:

- $\lambda = 0$: aucune pénalité, risque de surajustement ;
- $\lambda \to \infty$: $f'' \equiv 0$, ce qui redonne la droite des moindres carrés.

L'analogie avec la régression ridge du Chapitre 6 est directe : l'estimateur a la même forme et le paramètre de pénalité joue le même rôle. En pratique, on choisit λ par validation croisée. En R, la fonction `smooth.spline(x, y)` place un nœud à chaque valeur distincte de x. Elle choisit λ par validation croisée généralisée (GCV), une approximation de la LOOCV ; l'option `cv = TRUE` demande la LOOCV exacte.

Exemple — Spline de lissage sur `qualite_air`

```
modele.sp <- smooth.spline(ozone$heure, ozone$O3)
modele.sp
```

```
Call:
smooth.spline(x = ozone$heure, y = ozone$O3)

Smoothing Parameter  spar= 0.4916719  lambda= 0.01890238 (14 iterations)
Equivalent Degrees of Freedom (Df): 7.685062
Penalized Criterion (RSS): 405.0395
GCV: 109.2475
```

Les 24 heures distinctes donnent 24 nœuds, donc 24 degrés de liberté avant pénalisation, soit un paramètre par heure, comme pour les moyennes horaires de la Figure 8.4. La pénalité les ramène à 7,7 degrés de liberté effectifs, et le critère GCV (109,2) rejoint la LOOCV des meilleurs modèles précédents. La LOOCV exacte (`cv = TRUE`) n'est pas recommandée ici. Avec des valeurs de x répétées, exclure une observation à la fois ne change presque rien à l'ajustement en ce point. D'ailleurs, `smooth.spline` émet un avertissement.

```r
lissage <- do.call(rbind, lapply(list(c("df = 3", 3), c("GCV", NA), c("df
 ↪  = 20", 20)), function(z) {
  sp <- if (is.na(z[2])) modele.sp else smooth.spline(ozone$heure,
 ↪  ozone$O3, df = as.numeric(z[2]))
  data.frame(m = sprintf("%s (df = %.1f, lambda = %.2g)", z[1], sp$df,
   ↪  sp$lambda),
             heure = grille$heure, f = predict(sp, grille$heure)$y)
}))

ggplot(ozone, aes(heure, O3)) +
  geom_jitter(width = 0.25, height = 0, alpha = 0.08, size = 0.7) +
  geom_line(data = lissage, aes(heure, f, colour = m), linewidth = 0.8) +
  labs(x = "Heure (HNE)", y = "Ozone (ppb)", colour = NULL) +
  theme_minimal(base_size = 13)
```

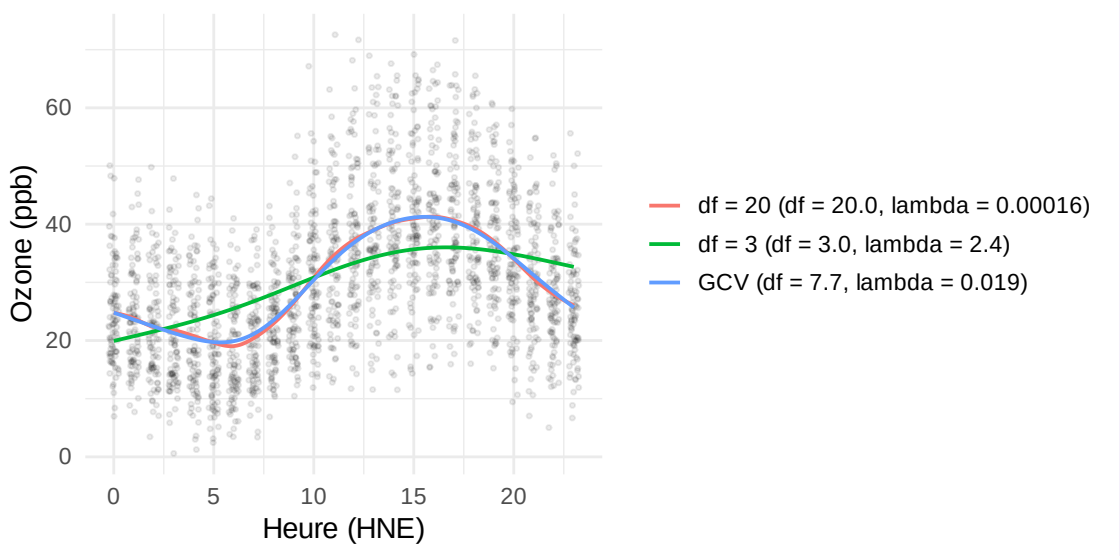


FIGURE 8.7 : Splines de lissage pour trois valeurs de λ : forte pénalisation (3 degrés de liberté effectifs), valeur choisie par GCV et faible pénalisation (20 degrés de liberté effectifs).

Avec trois degrés de liberté, la pénalité écrase la bosse. Avec vingt, la courbe est libre de suivre les moyennes horaires, mais comme celles-ci sont régulières, elle reste lisse et presque confondue avec la solution GCV. C'est l'avantage de la pénalisation : on part d'une base riche (un nœud par valeur de x) et on laisse λ décider de la complexité. Il n'est pas nécessaire de choisir le nombre ni la position des nœuds.

```r
modele.ns <- lm(O3 ~ ns(heure, df = 4), data = ozone)
courbes <- rbind(
  data.frame(m = "Polynôme, p = 5", heure = grille$heure,
             f = predict(modeles.o3[[5]], newdata = grille)),
  data.frame(m = "B-splines, nœuds 6, 12, 18", heure = grille$heure,
             f = predict(modele.bs, newdata = grille)),
  data.frame(m = "Spline naturelle, df = 4", heure = grille$heure,
             f = predict(modele.ns, newdata = grille)),
  data.frame(m = "Spline de lissage (GCV)", heure = grille$heure,
             f = predict(modele.sp, grille$heure)$y)
)

ggplot(ozone, aes(heure, O3)) +
  geom_jitter(width = 0.25, height = 0, alpha = 0.08, size = 0.7) +
  geom_line(data = courbes, aes(heure, f, colour = m), linewidth = 0.8) +
  labs(x = "Heure (HNE)", y = "Ozone (ppb)", colour = NULL) +
  theme_minimal(base_size = 13)
```

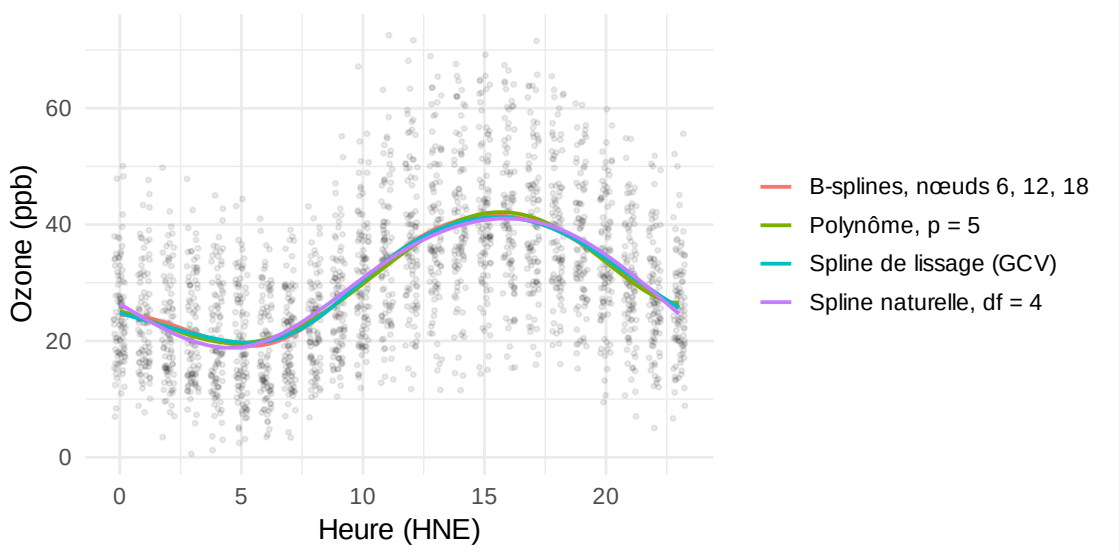


FIGURE 8.8 : Cycle journalier de l'ozone estimé par le polynôme de degré 5, les B-splines cubiques à nœuds 6, 12 et 18 h, la spline naturelle à 4 degrés de liberté et la spline de lissage (GCV).

Les quatre estimateurs sont pratiquement indiscernables. Lorsque n est grand et que f est lisse, le choix de la base importe peu : ce qui compte, c'est le nombre de degrés de

liberté effectifs. Ici, tous les critères de validation croisée s'accordent sur une complexité de 6 à 8 degrés de liberté, ordonnée à l'origine comprise.

Remarque — Degrés de liberté effectifs

Pour un estimateur linéaire $\hat{Y} = A_\lambda Y$, avec $A_\lambda = X(X^\top X + \lambda M)^{-1} X^\top$, les degrés de liberté effectifs sont $\mathrm{Tr}(A_\lambda)$. Ils décroissent de $K + p$ (pour $\lambda = 0$) vers 2 (pour $\lambda \to \infty$) et donnent ainsi une échelle interprétable pour calibrer le lissage.

Remarque — À retenir

- Bases de fonctions : polynômes, fonctions constantes par morceaux et splines de régression s'écrivent $\sum_k \beta_k b_k(X_i)$ et s'ajustent comme des modèles linéaires, avec `lm`. Les coefficients des splines ne s'interprètent pas un à un, et ce sont la courbe $\hat{f}$, son erreur de validation croisée et ses degrés de liberté qui comptent ;
- Splines de régression : polynômes de degré p par morceaux, raccordés aux nœuds par la continuité de $f, f', \ldots, f^{(p-1)}$, soit $K + p$ degrés de liberté pour K intervalles. La base des B-splines (`bs`) donne le même $\hat{f}$ que la base des puissances tronquées avec les mêmes nœuds, mais avec une matrice $X^\top X$ bien conditionnée ;
- Bords du domaine : les polynômes de degré élevé et les splines peuvent être instables aux extrémités, et les splines naturelles, linéaires au-delà des nœuds frontières, corrigent ce défaut ($K + 1$ degrés de liberté pour une spline naturelle cubique). Avec `ns(heure, df = 4)`, les nœuds intérieurs sont placés aux quantiles de X, et `df` (comme pour `bs`) ne compte pas l'ordonnée à l'origine ;
- Splines de lissage : la pénalité $\lambda \int_A^B f''(t)^2\, dt$ donne $\hat{\beta}_\lambda = (X^\top X + \lambda M)^{-1} X^\top Y$, de la même forme que ridge. `smooth.spline` place un nœud à chaque valeur distincte de x et choisit λ par GCV, et la complexité se lit sur les degrés de liberté effectifs $\mathrm{Tr}(A_\lambda)$;
- Confiance ou prédiction : `predict(..., interval = "confidence")` donne une bande pour la moyenne conditionnelle $f(x)$, et `interval = "prediction"` un intervalle, plus large, pour une nouvelle observation. Les deux supposent des erreurs indépendantes, et des mesures corrélées (les heures d'une même journée) rendent la bande de confiance trop étroite.

8.4 Exercices

Les exercices sont classés par niveau de difficulté. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

Les exercices de ce chapitre sont surtout appliqués : ils demandent d'ajuster et de comparer

des modèles en R. Ils utilisent trois jeux de données :

— les concentrations horaires de dioxyde d'azote (`NO2`, en ppb) mesurées à la station de Longueuil, du lundi au vendredi, tirées de `qualite_air` ;
— le jeu de données `arbres` ;
— le jeu de données `mcycle` de la librairie `MASS`, qui donne l'accélération de la tête d'un mannequin (`accel`, en multiples de g) en fonction du temps écoulé depuis l'impact (`times`, en millisecondes) lors de 133 essais simulés d'accidents de moto.

Le code suivant charge les données et définit la fonction `vc.n`, qui calcule $VC_{(n)}$ par la formule explicite du Chapitre 5.

```
library(splines)
library(MASS)   # jeu de données mcycle

air <- read.csv("Jeux de données/qualite_air.csv", fileEncoding =
 ↪  "UTF-8-BOM") |>
  mutate(heure = as.integer(substr(Date_Heure, 12, 13)),
         date  = as.Date(substr(Date_Heure, 1, 10)))
# jours de semaine seulement (%u : 1 = lundi, ..., 7 = dimanche)
no2 <- air |>
  filter(Station == "06600 - Longueuil", !is.na(NO2), as.integer(format(date,
   ↪  "%u")) <= 5)

arbres <- read.csv("Jeux de données/arbres.csv")

vc.n <- function(m) mean((residuals(m) / (1 - hatvalues(m)))^2)
```

8.4.1 Exercices appliqués

Exercice 1 (facile). Cycle journalier du dioxyde d'azote.

a) Calculez la concentration moyenne de NO2 à chaque heure et tracez-la. Décrivez le cycle journalier et proposez une explication.
b) Ajustez des régressions polynomiales de degré 1 à 12 de `NO2` sur `heure` et comparez-les par $VC_{(n)}$ et par le BIC. Quel degré retenez-vous ?
c) Pourquoi faut-il ici un degré nettement plus élevé que pour l'ozone au Chapitre 8 ? Pourquoi est-il préférable d'utiliser `poly(heure, p)` plutôt que `poly(heure, p, raw = TRUE)` pour les degrés élevés ?

Voir la solution

Exercice 2 (facile). Fonctions constantes par morceaux pour le NO2.

a) Découpez la journée en $K = 2, 3, 4, 6, 8, 12$ et 24 intervalles de même largeur et comparez les ajustements par $VC_{(n)}$.
b) Proposez plutôt un découpage en cinq intervalles qui suit le cycle observé à l'exercice précédent, par exemple la nuit, la pointe du matin, la journée, la pointe du soir et la soirée. Comparez-le aux découpages réguliers.
c) Ajustez votre modèle sans ordonnée à l'origine (`- 1` dans la formule) et vérifiez numériquement que les coefficients sont les moyennes de `NO2` dans chaque intervalle, comme l'annonce le Chapitre 8.

Voir la solution

Exercice 3 (moyen). Extrapolation de la hauteur des arbres.

Le plus gros arbre du jeu de données `arbres` a un diamètre de 41,5 cm.

a) Ajustez la hauteur en fonction du diamètre par des polynômes de degré 1 à 5, des splines cubiques `bs(diameter_cm, df = d)` pour $d = 3, \ldots, 6$ et des splines naturelles `ns(diameter_cm, df = d)` pour $d = 1, \ldots, 6$. Comparez-les par $VC_{(n)}$.
b) Pour chaque modèle, prédisez la hauteur d'arbres de 40, 45, 50 et 60 cm de diamètre. Commentez.
c) Pourquoi les splines naturelles se comportent-elles mieux au-delà des données ? Leur prédiction à 60 cm est-elle pour autant fiable ?

Voir la solution

Exercice 4 (moyen). Accélération lors d'un impact (`mcycle`).

a) Tracez `accel` en fonction de `times`. Pourquoi la régression polynomiale est-elle mal adaptée à ces données ? Vérifiez-le en calculant $VC_{(n)}$ pour les degrés 1 à 12.
b) Comparez par $VC_{(n)}$ les splines cubiques `bs(times, df = d)` et les splines naturelles `ns(times, df = d)` pour $d = 3, \ldots, 15$. Quel modèle retenez-vous ?
c) Superposez au nuage de points le meilleur polynôme et la spline retenue.
d) Prédisez l'accélération à 60 et 65 ms avec le polynôme de degré 8, la spline cubique et la spline naturelle à 8 degrés de liberté. Laquelle est la plus plausible physiquement ?

Voir la solution

Exercice 5 (moyen). Spline de lissage pour `mcycle`.

a) Ajustez une spline de lissage avec `smooth.spline`, en laissant la GCV choisir λ. Combien de degrés de liberté effectifs obtenez-vous ? Comparez avec la spline naturelle retenue à l'exercice précédent.
b) Tracez sur un même graphique les splines de lissage à 5 degrés de liberté, à la valeur choisie par la GCV et à 30 degrés de liberté. Décrivez le compromis biais-variance.

c) Calculez l'écart-type des résidus de la spline choisie par la GCV avant 14 ms, entre 14 et 40 ms, et après 40 ms. Qu'en concluez-vous sur les hypothèses du modèle ? Quelle conséquence cela a-t-il sur le choix de λ et sur les intervalles de confiance ?

Voir la solution

Exercice 6 (difficile). Simulation : splines cubiques et splines naturelles aux bords.

On veut vérifier par simulation (Chapitre 1) l'affirmation du Chapitre 8 selon laquelle les splines naturelles sont plus stables aux extrémités du domaine. Considérez le modèle $Y_i = \sin(2\pi X_i) + \epsilon_i$, avec $X_i \sim \mathcal{U}(0,1)$, $\epsilon_i \sim \mathcal{N}(0, 0{,}3^2)$ et $n = 50$.

a) Pour $B = 500$ jeux de données, ajustez `bs(x, df = 6)` et `ns(x, df = 6)` et conservez les prédictions $\hat{f}(x_0)$ en $x_0 = 0{,}02$, $0{,}5$ et $0{,}98$. Pour que les deux bases aient les mêmes nœuds frontières d'un jeu de données à l'autre, fixez-les à 0 et 1 (argument `Boundary.knots`).
b) Estimez le biais, la variance et l'erreur quadratique moyenne de chaque estimateur en chaque point. Commentez.
c) Les deux modèles ont-ils le même nombre de paramètres ? Comment `ns` utilise-t-elle les deux degrés de liberté que `bs` consacre aux bords ?
d) La fonction $\sin(2\pi x)$ vérifie $f''(0) = f''(1) = 0$, exactement la contrainte qu'impose une spline naturelle. Reprenez la simulation avec $f(x) = \cos(2\pi x)$, dont la dérivée seconde ne s'annule pas aux bords. Vos conclusions changent-elles ?

Voir la solution

Exercice 7 (difficile). Base des puissances tronquées et B-splines.

a) Écrivez une fonction qui construit la base des puissances tronquées cubique pour un vecteur de nœuds donné.
b) Pour les données `mcycle`, placez $K = 3$, 6, 10 et 15 nœuds intérieurs aux quantiles de `times`. Pour chaque K, ajustez le modèle avec la base des puissances tronquées et avec `bs(times, knots = ...)`, puis comparez les valeurs ajustées et le nombre de condition de $X^\top X$ (fonction `kappa(..., exact = TRUE)`).
c) Expliquez les résultats.

Voir la solution

8.4.2 Exercices théoriques

Exercice 1 (facile). Soit le modèle constant par morceaux $Y_i = \sum_{k=1}^{K} \beta_k b_k(X_i) + \epsilon_i$, où $b_k(x) = \mathbf{1}\{x \in [c_{k-1}, c_k)\}$ et où chaque intervalle contient $n_k \geq 1$ observations.

a) Montrez que $X^\top X = \text{diag}(n_1, \ldots, n_K)$ et déduisez-en que $\hat{\beta}_k$ est la moyenne des Y_i de l'intervalle k.
b) Montrez que $Var[\hat{\beta}_k] = \sigma^2/n_k$ et que les $\hat{\beta}_k$ sont non corrélés. Pourquoi un intervalle qui contient peu d'observations pose-t-il problème ?

Voir la solution

Exercice 2 (facile). Soit $h(x, c) = (x - c)_+^3$.

a) Calculez $h'(x, c)$, $h''(x, c)$ et $h'''(x, c)$ (dérivées par rapport à x) pour $x < c$ et pour $x > c$. Montrez que h, h' et h'' sont continues en $x = c$, mais que h''' ne l'est pas.
b) Déduisez-en que toute fonction $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K-1} \theta_k (x - c_k)_+^3$ est une spline cubique de nœuds $c_1, \ldots, c_{K-1}$, et vérifiez qu'on retrouve $K + 3$ degrés de liberté.

Voir la solution

Exercice 3 (moyen). On représente une spline cubique de nœuds $\xi_1 < \cdots < \xi_m$ dans la base des puissances tronquées :

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{m} \theta_k (x - \xi_k)_+^3.$$

a) Montrez que f est linéaire sur $(-\infty, \xi_1]$ si et seulement si $\beta_2 = \beta_3 = 0$.
b) Montrez que f est alors linéaire sur $[\xi_m, \infty)$ si et seulement si $\sum_k \theta_k = 0$ et $\sum_k \theta_k \xi_k = 0$.
c) Déduisez-en qu'une spline cubique naturelle à m nœuds, linéaire au-delà de ξ_1 et de ξ_m, a m degrés de liberté. Faites le lien avec le compte $K + 1$ du Chapitre 8, où les nœuds sont $c_0 < c_1 < \cdots < c_K$.

Voir la solution

Exercice 4 (moyen). Pour une régression quadratique $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2$ avec $\beta_2 \neq 0$, l'extremum de la courbe est atteint en $x^* = -\beta_1/(2\beta_2)$ (Chapitre 8).

a) Montrez, par la méthode du delta, que la variance approximative de $\hat{x}^* = -\hat{\beta}_1/(2\hat{\beta}_2)$ est

$$Var[\hat{x}^*] \approx \frac{1}{4\beta_2^2} \left\{ Var[\hat{\beta}_1] + 4x^* \, Cov[\hat{\beta}_1, \hat{\beta}_2] + 4x^{*2} \, Var[\hat{\beta}_2] \right\}.$$

b) Pourquoi cette approximation est-elle mauvaise lorsque $\hat{\beta}_2$ est proche de 0 ? Que représente alors x^* ?

Voir la solution

Exercice 5 (difficile). Soit l'estimateur pénalisé $\hat{\beta}_\lambda = (X^\top X + \lambda M)^{-1} X^\top Y$ et la matrice d'influence $A_\lambda = X(X^\top X + \lambda M)^{-1} X^\top$, avec $X^\top X + \lambda M$ définie positive. Notons $\hat{\beta}_{\lambda,(-i)}$ l'estimateur calculé sans l'observation i, avec la même pénalité et le même λ.

a) À l'aide de la formule de Sherman-Morrison (exercice 1 du niveau difficile du chapitre 5), montrez que

$$Y_i - X_i^\top \hat{\beta}_{\lambda,(-i)} = \frac{Y_i - \hat{Y}_i}{1 - A_{ii}}.$$

b) Concluez que la formule de $VC_{(n)}$ utilisée pour les splines de lissage et les modèles additifs (Chapitre 9) est exacte, et non une approximation.

Voir la solution

Exercice 6 (difficile). On reprend l'estimateur de l'exercice précédent, où M est la matrice de pénalité d'une spline de lissage cubique ($M_{lq} = \int b_l'' b_q''$) et où $X^\top X$ est inversible.

a) Soit $X^\top X = R^\top R$ une factorisation (de Cholesky, par exemple) et $R^{-\top} M R^{-1} = U D U^\top$ la décomposition spectrale, avec $D = \text{diag}(d_1, \dots, d_q)$ et $d_k \geq 0$. Montrez que

$$\text{Tr}(A_\lambda) = \sum_{k=1}^{q} \frac{1}{1 + \lambda d_k}.$$

b) Montrez que le nombre de valeurs propres d_k nulles est la dimension de l'ensemble des fonctions de la base qui ne sont pas pénalisées, c'est-à-dire telles que $\int f''(t)^2 dt = 0$, et que cette dimension vaut 2.

c) Déduisez-en que les degrés de liberté effectifs décroissent de q (pour $\lambda = 0$) vers 2 (quand $\lambda \to \infty$), comme l'affirme le Chapitre 8.

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R*
— Faraway (2016). *Extending the Linear Model with R*
— James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R*

— Wood (2017). *Generalized Additive Models : An Introduction with R*
— Hastie, Tibshirani et Friedman (2009). *The Elements of Statistical Learning*
— Wasserman (2006). *All of Nonparametric Statistics*
— Ruppert, Wand et Carroll (2003). *Semiparametric Regression*

9 Modèles additifs et modèles additifs généralisés (GAM)

Les modèles additifs, que présente ce chapitre, gardent la structure d'un modèle linéaire, mais on y remplace chaque terme $\beta_j x_{ij}$ par une fonction lisse $f_j(x_{ij})$, estimée par splines pénalisées. Le cas gaussien vient d'abord, puis les GAM (réponse de la famille exponentielle) et leur ajustement avec `mgcv`.

9.1 Motivation

Les méthodes non paramétriques des Chapitre 7 et Chapitre 8 fonctionnent bien avec une ou deux variables explicatives, mais elles se heurtent au fléau de la dimension quand p est grand. Les voisinages se vident alors et le nombre de fonctions de base explose. Les modèles additifs sont un compromis entre la flexibilité des méthodes non paramétriques et l'interprétabilité des modèles linéaires : on remplace la partie linéaire par une somme de fonctions univariées,

$$Y_i = \alpha + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i,$$

où chaque f_j est une fonction lisse inconnue.

Laissons les données montrer la forme fonctionnelle appropriée, plutôt que de la fixer d'avance.

Chaque f_j est représentée par une base de splines (Chapitre 8) : $f_j(x) = \sum_{k=1}^{K_j} b_{jk}(x)\,\beta_{jk}$. Tel quel, le modèle n'est pas identifiable, puisqu'une constante peut passer librement de α aux f_j. On impose donc la contrainte de centrage

$$\sum_{i=1}^{n} f_j(x_{ij}) = 0, \qquad j = 1, \dots, p.$$

9.2 Estimation par moindres carrés pénalisés

Les f_j ne sont pas estimées en choisissant avec soin le nombre de nœuds de chaque base. On prend plutôt des bases « suffisamment riches » et on pénalise la rugosité de chaque fonction, ce qui revient à minimiser

$$\|Y - X\beta\|^2 + \sum_{j=1}^{p} \lambda_j\, \beta^\top S_j\, \beta,$$

où X regroupe toutes les fonctions de base évaluées aux observations et où S_j est la matrice de pénalité de la j^{e} fonction ($\beta^\top S_j \beta = \int f_j''(t)^2 dt$). La solution est explicite :

$$\widehat{\beta} = \Big(X^\top X + \sum_{j=1}^{p} \lambda_j S_j \Big)^{-1} X^\top Y.$$

Chaque paramètre de lissage λ_j règle le compromis entre régularité et fidélité aux données pour la fonction f_j. Si $\lambda_j \to \infty$, on force f_j à être linéaire, alors que $\lambda_j = 0$ laisse f_j libre, avec un risque de surajustement.

9.2.1 Matrice d'influence et degrés de liberté effectifs

Les valeurs ajustées s'écrivent $\widehat{Y} = AY$ avec $A = X(X^\top X + \sum_j \lambda_j S_j)^{-1} X^\top$. Les degrés de liberté effectifs du modèle sont $\mathrm{Tr}(A)$, et leur décomposition par fonction f_j donne une mesure interprétable de la complexité de chaque effet. Notez que la pénalisation induit un biais, $E[\widehat{\beta}] \neq \beta$, exactement comme pour ridge.

9.2.2 Choix des paramètres de lissage

Les paramètres de lissage $\lambda = (\lambda_1, \dots, \lambda_p)$ sont choisis en minimisant un estimateur de l'erreur de prédiction. Dans les deux critères suivants, $\widehat{y}_i$ est la i^{e} valeur ajustée et A dépend de λ :

— validation croisée par exclusion d'une observation (LOOCV), dite validation croisée ordinaire (OCV) :

$$VC_{(n)}(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \widehat{y}_i}{1 - A_{ii}} \right)^2 ;$$

— validation croisée généralisée (GCV), qui remplace les A_{ii} par leur moyenne :

$$GCV(\lambda) = \frac{n \sum_i (y_i - \widehat{y}_i)^2}{\big(n - \mathrm{Tr}(A)\big)^2}.$$

9.2.3 Inférence

L'inférence repose sur la forme ridge de l'estimateur. Si les erreurs sont normales et en traitant les λ_j comme fixés, elle donne $\widehat{\beta} \sim \mathcal{N}(E[\widehat{\beta}],\, V_\beta)$ avec $V_\beta = \sigma^2 (X^\top X + \sum_j \lambda_j S_j)^{-1} X^\top X (X^\top X + \sum_j \lambda_j S_j)^{-1}$. Comme les λ_j sont en fait estimés, ce résultat n'est qu'approximatif. L'approche bayésienne, où la pénalité devient une loi a priori gaussienne sur β, mène à la forme plus simple $V_\beta = \sigma^2 (X^\top X + \sum_j \lambda_j S_j)^{-1}$. Les intervalles de crédibilité de niveau 95 % qui en découlent,

$$\widehat{f}_j(x) \pm 1{,}96 \sqrt{\widehat{\mathrm{V}}(\widehat{f}_j(x))},$$

ont une bonne couverture fréquentiste en moyenne sur l'ensemble des valeurs de x, mais pas nécessairement en chaque point.

Pour tester $H_0 : f_j = 0$, on utilise une statistique de type Wald,

$$T_j = \widehat{\beta}_j^\top V_{\beta_j}^{r-} \widehat{\beta}_j,$$

où $\widehat{\beta}_j$ regroupe les coefficients de f_j, V_{β_j} est le bloc correspondant de V_β et $V_{\beta_j}^{r-}$ en est un pseudo-inverse de rang r, proche des degrés de liberté effectifs de f_j. La pénalisation rend V_{β_j} presque singulière, d'où ce pseudo-inverse plutôt que l'inverse. On compare T_j à une loi χ^2 si σ^2 est connue, ou à une loi F si σ^2 est estimée, avec un nombre de degrés de liberté proche de r (colonne `Ref.df` de `summary`). Pour un GAM, la même statistique s'utilise avec le paramètre de dispersion ϕ défini plus bas à la place de σ^2.

> **Remarque — Portée de l'inférence d'un modèle additif**
> Les erreurs-types, les intervalles et les valeurs-p que donne `summary` traitent les paramètres de lissage λ_j comme fixés. Or ils sont choisis à partir des mêmes données. L'incertitude liée à ce choix n'est donc pas prise en compte, et les intervalles tendent à être trop étroits. Le test de $H_0 : f_j = 0$ ne s'interprète pas non plus exactement comme le test d'un coefficient en régression linéaire. Ses degrés de liberté (`Ref.df`) sont eux-mêmes approximatifs et dépendent du lissage retenu. Je vous conseille de lire ces valeurs-p comme des indications : une valeur-p de 0,04 et une de 0,06 ne racontent pas des histoires différentes.
> Avec `method = "REML"`, `mgcv` peut corriger en partie le premier problème : `vcov(modele, unconditional = TRUE)` et `plot(modele, unconditional = TRUE)` ajoutent l'incertitude sur les λ_j.

9.3 Modèles additifs généralisés

Les GAM combinent la structure additive avec les lois de la famille exponentielle, comme les GLM. Le modèle suppose que Y_i suit une loi de la famille exponentielle avec $E[Y_i] = \mu_i$ et

$$g(\mu_i) = \alpha + f_1(x_{i1}) + \cdots + f_p(x_{ip}),$$

où g est une fonction de lien. Deux cas particuliers sont le GAM logistique (réponse binaire) et le GAM Poisson (dénombrements, avec un *offset* au besoin).

Les paramètres sont estimés en minimisant la déviance pénalisée :

$$\widehat{\beta} = \arg\min_{\beta} \left\{ D(y, \mu) + \sum_{j=1}^{p} \lambda_j \, \beta^\top S_j \, \beta \right\},$$

où $D(y, \mu)$ est la déviance du Chapitre 3.

9.3.1 Algorithme d'ajustement (moindres carrés pondérés itératifs pénalisés)

L'ajustement combine l'algorithme des moindres carrés pondérés itératifs des GLM et le lissage pénalisé. À l'itération m :

1. construire la réponse de travail (*working response*)
$$z_i^{(m)} = \eta_i^{(m)} + (y_i - \mu_i^{(m)}) \left(\frac{\partial \eta_i}{\partial \mu_i}\right)^{(m)} ;$$
2. construire les poids
$$w_i^{(m)} = \frac{1}{V(\mu_i^{(m)})} \left[\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^{(m)}\right]^2 ;$$
3. ajuster un modèle additif pénalisé pondéré à $z^{(m)}$;
4. répéter jusqu'à convergence.

9.3.2 Choix des paramètres de lissage et comparaison de modèles

Le critère de choix des paramètres de lissage dépend de ce qu'on sait du paramètre de dispersion ϕ :

— si ϕ est connu (logistique, Poisson) : critère UBRE (*unbiased risk estimator*, équivalent au C_p de Mallows)
$$UBRE(\lambda) = \frac{D(y, \widehat{\mu})}{n} - \phi + \frac{2\phi \operatorname{Tr}(A)}{n} ;$$
le terme $-\phi$ ne change pas le λ choisi, mais c'est cette valeur que rapporte `summary` ;
— si ϕ est inconnu : GCV
$$GCV(\lambda) = \frac{n\, D(y, \widehat{\mu})}{(n - \operatorname{Tr}(A))^2}.$$

Pour comparer des modèles, on utilise l'$AIC = -2\,\ell(\widehat{\beta}; y) + 2\operatorname{Tr}(A)$, où les degrés de liberté effectifs remplacent le nombre de paramètres. La GCV peut aussi servir, et on préfère le modèle qui a le plus petit AIC ou la plus petite GCV. La proportion de déviance expliquée $(D_{null} - D)/D_{null}$ décrit la qualité de l'ajustement, mais elle ne pénalise pas la complexité (elle augmente toujours avec un modèle plus riche) et ne permet donc pas, seule, de choisir un modèle.

9.3.3 En pratique avec R

La librairie `mgcv` ajuste les GAM et choisit les paramètres de lissage automatiquement. Dans la formule, `s(x)` déclare une fonction lisse de `x`. Par défaut, c'est une spline de régression en plaques minces (*thin plate regression spline*, `bs = "tp"`) à `k = 10` fonctions de base (Wood 2003). Contrairement aux B-splines et aux splines cubiques de régression (`bs = "cr"`) du Chapitre 8, cette base ne demande pas de choisir le nombre ni la position des nœuds : elle est construite à partir de toutes les valeurs observées de `x`, puis réduite aux `k` directions les plus utiles. Les termes sans `s()` restent paramétriques, et `family` joue le même rôle que dans `glm`.

Exemple — Modèle additif gaussien : cycles journalier et saisonnier de l'ozone sur `qualite_air`

Au Chapitre 8, la concentration d'ozone à Longueuil a été modélisée en fonction de l'heure, pour les seuls mois d'été. Cet exemple reprend l'année entière ($n = 8\,727$ heures) avec deux variables explicatives, l'heure et le jour de l'année. Le modèle additif

$$O_3 = \alpha + f_1(\text{heure}) + f_2(\text{jour}) + \epsilon$$

suppose que le cycle journalier a la même forme toute l'année et que le cycle saisonnier est le même à toute heure.

```
library(mgcv)
longueuil <- read.csv("Jeux de données/qualite_air.csv", fileEncoding =
 ↪  "UTF-8-BOM") |>
  dplyr::mutate(heure = as.integer(substr(Date_Heure, 12, 13)),
                jour  = as.integer(format(as.Date(substr(Date_Heure, 1,
                 ↪  10)), "%j"))) |>
  dplyr::filter(Station == "06600 - Longueuil", !is.na(O3))

mod1 <- gam(O3 ~ s(heure) + s(jour), data = longueuil)
summary(mod1)
```

```
Family: gaussian
Link function: identity

Formula:
O3 ~ s(heure) + s(jour)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 27.83688    0.09997   278.4   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(heure) 7.813  8.635 269.2  <2e-16 ***
s(jour)  8.952  8.999 257.1  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.347   Deviance explained = 34.8%
GCV = 87.402  Scale est. = 87.224    n = 8727
```

```
par(mfrow = c(1, 2))
plot(mod1, shade = TRUE)
```

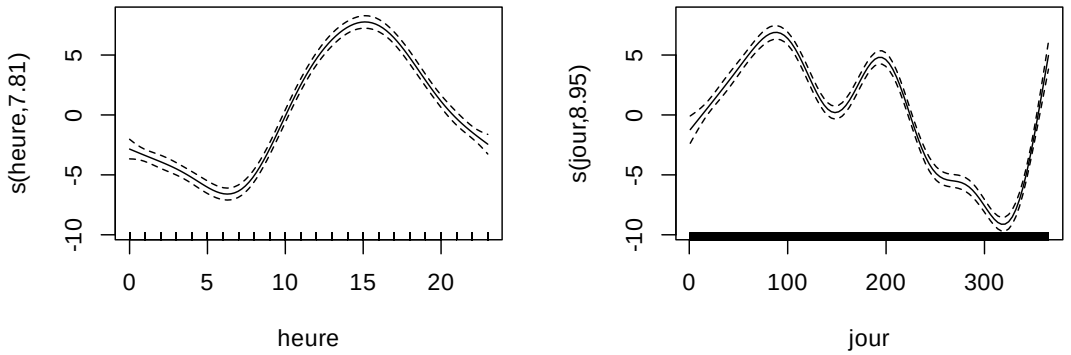


FIGURE 9.1 : Fonctions $\hat{f}_1$ (heure) et $\hat{f}_2$ (jour de l'année) du modèle additif pour l'ozone à Longueuil, avec bandes de crédibilité ponctuelles à 95 % pour ces fonctions (et non pour une observation future). Chaque fonction est centrée : elle représente l'écart à la moyenne $\hat{\alpha}$. Comme les mesures horaires d'une même journée sont corrélées, ces bandes sont trop étroites (voir le Chapitre 10).

La sortie de `summary` donne, pour chaque fonction lisse, ses degrés de liberté effectifs (`edf`) et un test approximatif de $H_0 : f_j = 0$. Les deux fonctions sont non nulles et non linéaires.

La fonction $\hat{f}_1$ décrit ici le cycle journalier sur l'année entière. Elle a la même forme que la bosse estivale du Chapitre 8 (minimum vers 6 h, maximum vers 15 h). La fonction $\hat{f}_2$ montre le maximum printanier de l'ozone (vers le jour 88, à la fin de mars), suivi d'un second sommet en juillet, d'un creux en novembre (vers le jour 319) et d'une remontée en décembre. Le modèle explique 35 % de la variance, contre 14 % pour le modèle linéaire en `heure` et `jour`.
Ce modèle appelle trois remarques. D'abord, `s(jour)` utilise presque tous ses 9 degrés de liberté disponibles, et si on augmente `k`, la courbe se met à suivre les épisodes météorologiques de quelques jours. En fait, les heures consécutives sont corrélées, ce qui fait sous-lisser la GCV, puisqu'elle suppose des erreurs indépendantes. Pour la même raison, les bandes de la Figure 9.1 et les valeurs-p de `summary` sont trop optimistes (voir la remarque « Bande de confiance ou bande de prédiction ? » du Chapitre 8, qui discute la même corrélation, et le Chapitre 10, qui présente des modèles qui en tiennent compte). Ensuite, `s(jour)` n'est pas cyclique : rien n'oblige $\hat{f}_2$ à se raccorder entre le 31 décembre et le 1er janvier, et la figure montre en effet un saut entre les deux extrémités. Une spline cyclique, `s(jour, bs = "cc")`, impose ce raccord. Enfin, l'hypothèse d'additivité est discutable, puisque l'amplitude du cycle journalier est plus grande en été qu'en hiver. Dans `mgcv`, on peut ajuster une surface $f(\text{heure}, \text{jour})$ avec `te(heure, jour)`, mais on perd alors l'interprétation séparée des deux effets.

Remarque — Concurvité

Dans un modèle additif, deux fonctions lisses peuvent se substituer l'une à l'autre si leurs variables sont liées de façon non linéaire, par exemple si $f_2(x_2)$ est presque une fonction de x_1. C'est l'analogue non linéaire de la colinéarité, appelé concurvité. Les fonctions estimées deviennent alors instables, et leurs bandes de crédibilité trompeuses. La fonction `concurvity` de `mgcv` mesure, pour chaque terme lisse, la part qui peut être reproduite par les autres termes : 0 signifie aucun problème, et une valeur proche de 1 signale un terme mal identifié.

```
round(concurvity(mod1), 3)
```

```
         para s(heure) s(jour)
worst       0        0       0
observed    0        0       0
estimate    0        0       0
```

Ici, les valeurs sont nulles à trois décimales près, parce que l'heure et le jour de l'année varient indépendamment l'une de l'autre dans les données.

Exemple — GAM logistique : strate du couvert sur `arbres`

Au Chapitre 7, on a estimé par kNN la probabilité qu'un arbre soit dans la strate inférieure du couvert en fonction de son diamètre, sans forme paramétrique. Le GAM logistique fait de même et ajoute l'espèce comme terme paramétrique. Notons $Y_i = 1$ si l'arbre i est dans la strate inférieure (intermédiaire ou opprimé) et $Y_i = 0$ sinon, d_i son diamètre (en cm) et $e(i) \in \{\text{bouleau}, \text{épinette}, \text{érable}, \text{sapin}\}$ son espèce. On suppose $Y_i \sim \text{Bernoulli}(\pi_i)$, indépendants, avec

$$\operatorname{logit} \pi_i = \log \frac{\pi_i}{1-\pi_i} = \alpha + f(d_i) + \gamma_{e(i)},$$

où

- f est une fonction lisse inconnue du diamètre, représentée par une spline de régression pénalisée à 10 fonctions de base (`s(diameter_cm)`) et centrée, $\sum_i f(d_i) = 0$, pour être identifiable séparément de α ;
- γ_e est l'effet de l'espèce, un terme paramétrique ordinaire. Comme le bouleau à papier est le niveau de référence ($\gamma_{\text{bouleau}} = 0$), e^{γ_e} est le rapport de cotes de la strate inférieure entre l'espèce e et le bouleau, à diamètre égal ;
- α est le logit de la probabilité pour un bouleau de diamètre « moyen », au sens de la contrainte de centrage.

C'est la régression logistique du Chapitre 2, à ceci près que l'effet du diamètre n'est pas contraint à être linéaire sur l'échelle logit. Le paramètre de lissage de f, choisi par UBRE, décide au vu des données du degré de courbure nécessaire.

```
arbres <- read.csv("Jeux de données/arbres.csv") |>
  dplyr::mutate(inferieure = as.integer(canopy_stratum %in%
↪  c("Intermédiaire", "Opprimé")),
                espece = factor(species_fr))

mod2 <- gam(inferieure ~ s(diameter_cm) + espece, family = binomial, data
↪  = arbres)
summary(mod2)
```

```
Family: binomial
Link function: logit

Formula:
inferieure ~ s(diameter_cm) + espece

Parametric coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          -1.7527     0.4148  -4.225 2.39e-05 ***
```

```
especeÉpinette noire   0.5016     0.4954   1.012   0.3113
especeÉrable rouge     0.1757     0.5265   0.334   0.7386
especeSapin baumier    0.7950     0.4830   1.646   0.0998 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
               edf Ref.df Chi.sq  p-value
s(diameter_cm)   1      1  21.01 5.11e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.188   Deviance explained = 18.5%
UBRE = 0.052107  Scale est. = 1         n = 200
```

```r
plot(mod2, shade = TRUE)
```

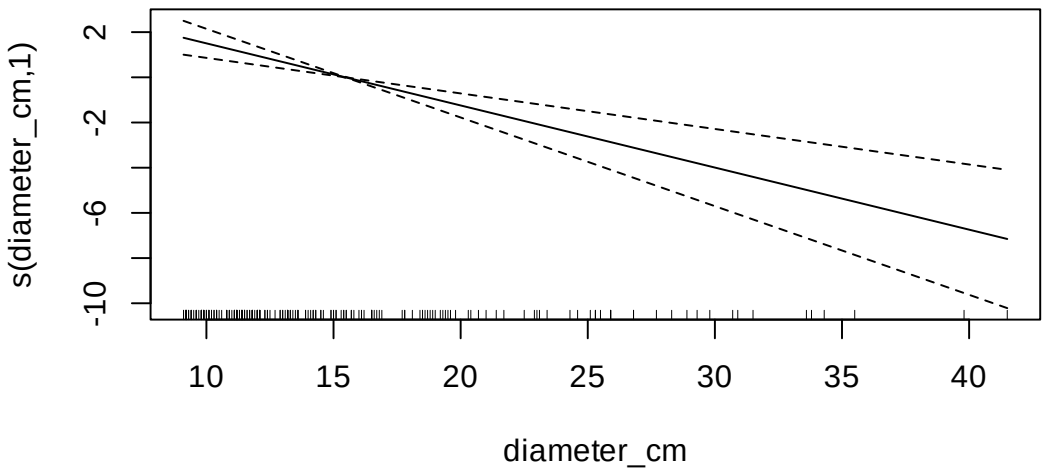


FIGURE 9.2 : Fonction lisse $\hat{f}$ du diamètre sur l'échelle logit, centrée, avec bande de crédibilité ponctuelle à 95 % pour f (et non une bande de prédiction). Les traits au bas de la figure marquent les diamètres observés.

La fonction lisse du diamètre a un degré de liberté effectif (1,0001), parce que la pénalisation a ramené $\hat{f}$ à une droite. Le GAM se réduit donc à la régression logistique avec le diamètre comme variable linéaire sur l'échelle logit (même AIC, à 10^{-4} près). La Figure 9.2 le montre : une droite décroissante, dont la bande de crédibilité s'évase là où les gros arbres se font rares. Cet exemple illustre l'usage diagnostique du GAM : quand on hésite sur la forme fonctionnelle, on laisse les données trancher. Si les degrés de liberté effectifs sont proches de 1, un GLM paramétrique suffit.
Sur l'échelle des probabilités, le modèle donne une courbe par espèce. Ces courbes ont toutes la même forme (effet additif sur le logit) et sont décalées verticalement par les $\hat{\gamma}_e$. À diamètre égal, le sapin baumier est l'espèce la plus souvent sous le couvert et le bouleau la moins souvent, mais aucun écart avec le bouleau n'est significatif au seuil de 5 %. La Figure 9.3 superpose l'estimateur kNN du Chapitre 7 avec $k = 64$, la valeur choisie par l'AUC.

```
library(ggplot2)
grille <- expand.grid(diameter_cm = seq(9, 42, length.out = 200), espece
 ↪  = levels(arbres$espece))
grille <- dplyr::mutate(grille, pi = predict(mod2, newdata = grille, type
 ↪  = "response"))
# kNN du chapitre kNN : moyenne des Y_i tels que d_i(x) <= d_(k)(x), ex
 ↪  æquo compris
knn.moy <- function(x, X, Y, k) {
  d <- abs(X - x)
  mean(Y[d <= sort(d)[k] + 1e-8])
}
knn <- data.frame(diameter_cm = seq(9, 42, length.out = 200)) |>
  dplyr::mutate(pi = sapply(diameter_cm, function(x)
    knn.moy(x, arbres$diameter_cm, arbres$inferieure, k.auc)))

set.seed(1)   # pour le décalage aléatoire des points
ggplot(arbres, aes(diameter_cm, inferieure)) +
  geom_jitter(aes(colour = espece), width = 0, height = 0.03, alpha =
   ↪  0.5, size = 1) +
  geom_line(data = grille, aes(diameter_cm, pi, colour = espece),
   ↪  linewidth = 0.8) +
  geom_line(data = knn, aes(diameter_cm, pi), linetype = "dashed") +
  labs(x = "Diamètre (cm)", y = "P(strate inférieure)", colour = NULL) +
  theme_minimal(base_size = 13)
```

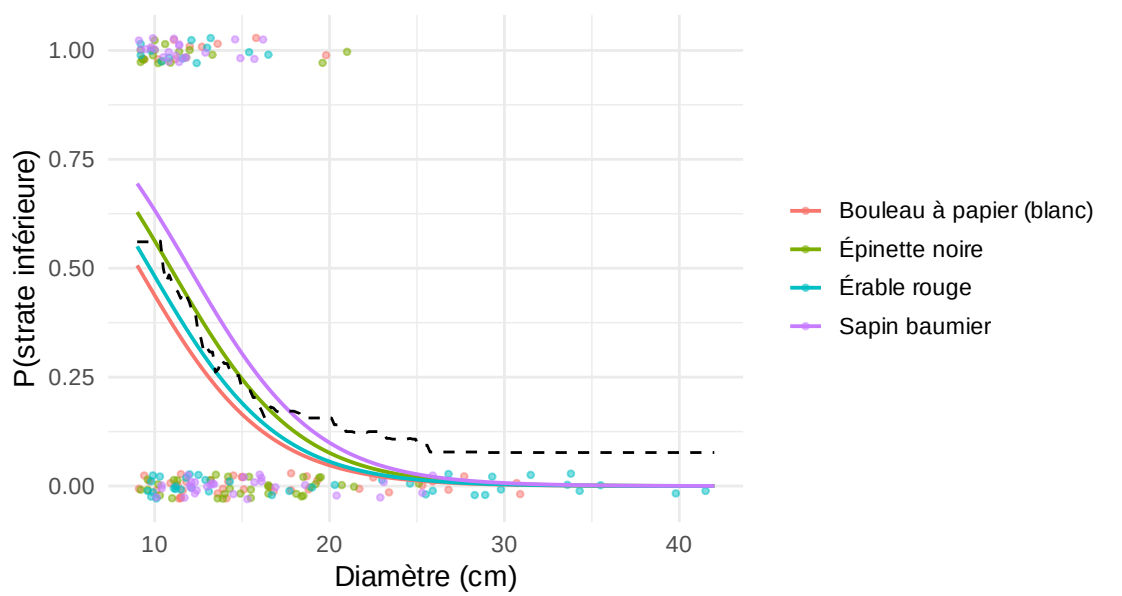


FIGURE 9.3 : Probabilité estimée d'être dans la strate inférieure selon le diamètre et l'espèce (GAM logistique), observations (0 ou 1, légèrement décalées verticalement) et estimateur kNN avec le k choisi par l'AUC au Chapitre 7 (tirets), qui ignore l'espèce.

Les deux approches donnent le même portrait : pour les plus petits arbres, la probabilité est d'environ 0,5 à 0,7 (selon l'espèce), puis elle tombe à presque 0 au-delà de 25 cm. Le kNN, en escalier, ne descend pas sous 0,08 pour les gros diamètres. Ses 64 voisins (65 avec les ex æquo) sont alors les plus gros arbres de l'échantillon, et 5 d'entre eux sont dans la strate inférieure, d'où le plancher $5/65 \approx 0{,}08$. Le GAM, lui, prolonge la décroissance. Avec $n = 200$, l'estimateur non paramétrique ne contenait pas d'information que la droite sur l'échelle logit ne capte déjà, et le GAM ajoute en plus l'espèce sans effort.

```
# réajustement par glm, puisque f est linéaire
mod2.glm <- glm(inferieure ~ diameter_cm + espece, family = binomial,
 ↪  data = arbres)
coef(mod2.glm)
```

```
        (Intercept)          diameter_cm especeÉpinette noire
          2.4994863           -0.2748570            0.5015752
 especeÉrable rouge  especeSapin baumier
          0.1756944            0.7949903
```

```r
# rapports de cotes et IC de Wald à 95 %
exp(cbind(RC = coef(mod2.glm), confint.default(mod2.glm)))
```

```
                             RC     2.5 %     97.5 %
(Intercept)          12.1762374 2.4016684 61.7324009
diameter_cm           0.7596807 0.6754517  0.8544131
especeÉpinette noire  1.6513204 0.6254029  4.3601635
especeÉrable rouge    1.1920737 0.4247947  3.3452389
especeSapin baumier   2.2144196 0.8592494  5.7069046
```

Contrairement aux splines du Chapitre 8, cet exemple comporte des paramètres qui s'interprètent. La partie paramétrique du modèle, les γ_e, se lit exactement comme en régression logistique. Par exemple, $e^{\hat{\gamma}_{\text{sapin}}} = e^{0,795} \approx 2,2$ est le rapport de cotes estimé de la strate inférieure entre un sapin baumier et un bouleau à papier de même diamètre. Son intervalle de confiance de Wald à 95 %, d'environ [0,9; 5,7], contient 1. C'est le propre des modèles additifs : les effets dont on veut une interprétation restent paramétriques, et seuls ceux dont la forme est incertaine passent par une fonction lisse. Ici, comme $\hat{f}$ s'est révélée linéaire, on peut aller plus loin et réajuster le modèle avec `glm`. La pente du diamètre vaut alors $-0,275$ par cm. Il s'agit d'un rapport de cotes de $e^{-0,275} \approx 0,76$ par centimètre de diamètre supplémentaire (IC de Wald à 95 % : [0,68; 0,85]), c'est-à-dire que les cotes sont divisées par environ 4 pour 5 cm de plus. Notez que cette interprétation n'aurait pas été disponible si les degrés de liberté effectifs de $\hat{f}$ avaient été nettement supérieurs à 1.

Dans les deux exemples, `summary` rapporte, pour chaque fonction lisse, ses degrés de liberté effectifs et un test approximatif de nullité. La fonction `plot` trace chaque $\hat{f}_j$ avec ses bandes de crédibilité. Un GAM Poisson s'ajuste avec la même syntaxe et `family = poisson`. Au besoin, on ajoute un terme `offset(log(exposition))` dans la formule, comme pour `glm`.

Remarque — À retenir

- Modèle additif : chaque terme $\beta_j x_{ij}$ est remplacé par une fonction lisse $f_j(x_{ij})$, centrée pour être identifiable, et un GAM relie $g(\mu_i)$ à $\alpha + f_1(x_{i1}) + \cdots + f_p(x_{ip})$. C'est un compromis entre la flexibilité des méthodes non paramétriques et l'interprétabilité des modèles linéaires ;
- Lissage : chaque f_j est représentée dans une base riche, pénalisée par $\lambda_j \beta^\top S_j \beta$, et les λ_j sont choisis par UBRE si ϕ est connu (logistique, Poisson) ou par GCV sinon. Des degrés de liberté effectifs (`edf`) proches de 1 indiquent une fonction linéaire, et un GLM paramétrique suffit alors ;
- En R : `gam` de `mgcv`, où `s(x)` déclare une fonction lisse (par défaut une spline en plaques minces à `k = 10` fonctions de base), les termes sans `s()` restent paramétriques et `family` joue le même rôle que dans `glm`. `summary` donne les `edf` et un test

approximatif de $H_0 : f_j = 0$, et `plot` trace chaque $\hat{f}_j$ avec sa bande de crédibilité ;
— Inférence approximative : les erreurs-types, intervalles et valeurs-p de `summary` traitent les λ_j comme fixés, alors qu'ils sont choisis à partir des mêmes données, de sorte que les intervalles tendent à être trop étroits et que les valeurs-p sont des indications. Avec `method = "REML"`, l'argument `unconditional = TRUE` corrige en partie ce problème ;
— Comparaison et limites : les modèles se comparent par l'AIC (calculé avec les degrés de liberté effectifs) ou la GCV, et non par la seule proportion de déviance expliquée, qui augmente toujours avec la complexité. Des erreurs corrélées font sous-lisser la GCV, et `te(heure, jour)` relâche l'additivité au prix de l'interprétation séparée des effets.

9.4 Exercices

Les exercices sont classés par niveau de difficulté. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

Comme ceux du chapitre précédent, ces exercices sont surtout appliqués. Ils reprennent les jeux de données `arbres` et `qualite_air`, chargés comme suit.

```
library(mgcv)

arbres <- read.csv("Jeux de données/arbres.csv") |>
  dplyr::mutate(espece = factor(species_fr))

longueuil <- read.csv("Jeux de données/qualite_air.csv", fileEncoding =
 ↪  "UTF-8-BOM") |>
  dplyr::mutate(heure = as.integer(substr(Date_Heure, 12, 13)),
                jour  = as.integer(format(as.Date(substr(Date_Heure, 1, 10)),
                 ↪  "%j"))) |>
  dplyr::filter(Station == "06600 - Longueuil", !is.na(O3))
```

9.4.1 Exercices appliqués

Exercice 1 (facile). Hauteur des arbres.

a) Ajustez le modèle additif `height_m ~ s(diameter_cm) + espece`. Combien de degrés de liberté effectifs la fonction du diamètre utilise-t-elle ? Tracez $\hat{f}$ et décrivez-la.
b) Comparez ce modèle, par l'AIC, au modèle linéaire `lm(height_m ~ diameter_cm + espece)`. Quelle conclusion en tirez-vous sur la forme de la relation ?

c) Au Chapitre 5, la hauteur était modélisée sur l'échelle log-log. Expliquez pourquoi on ne peut pas comparer directement l'AIC du modèle `lm(log(height_m) ~ log(diameter_cm) + espece)` à celui du modèle additif. Montrez qu'il faut lui ajouter $2\sum_i \log H_i$ pour l'exprimer sur l'échelle de H, puis faites la comparaison.

Voir la solution

Exercice 2 (moyen). Une courbe par espèce ?

a) Ajustez le modèle `height_m ~ s(diameter_cm, by = espece) + espece`, qui estime une fonction lisse distincte pour chaque espèce. Quels sont les degrés de liberté effectifs de chaque fonction ? Tracez-les.
b) Comparez ce modèle à celui de l'exercice précédent par l'AIC et par le test approximatif de `anova(..., test = "F")`.
c) Comparez maintenant les deux modèles, ainsi que le modèle linéaire, par validation croisée à 10 blocs (mêmes blocs pour les trois modèles). Les conclusions concordent-elles ? Comment l'expliquez-vous ?

Voir la solution

Exercice 3 (moyen). Splines cycliques pour l'ozone.

Le Chapitre 9 a ajusté le modèle `O3 ~ s(heure) + s(jour)` aux données de Longueuil et noté que $\hat{f}_2$ ne se raccorde pas entre le 31 décembre et le 1er janvier.

a) Ajustez ce modèle et prédisez la concentration à midi le 1er janvier (`jour = 1`) et le 31 décembre (`jour = 365`).
b) Ajustez le même modèle avec des splines cycliques, `s(heure, bs = "cc", k = 12)` et `s(jour, bs = "cc", k = 20)`, en précisant les bornes du cycle avec l'argument `knots = list(heure = c(-0.5, 23.5), jour = c(0.5, 365.5))`. Refaites les prédictions de a) et comparez les AIC.
c) Utilisez `gam.check` pour vérifier si la dimension de base `k` est suffisante. Que se passe-t-il si vous prenez `k = 40` pour le jour ? Interprétez, en lien avec la corrélation entre les heures consécutives.

Voir la solution

Exercice 4 (difficile). Interaction entre l'heure et la saison.

Le Chapitre 9 note que l'hypothèse d'additivité est discutable, puisque l'amplitude du cycle journalier de l'ozone semble plus grande en été qu'en hiver.

a) Ajoutez au modèle cyclique de l'exercice précédent un terme d'interaction `ti(heure, jour, bs = c("cc", "cc"), k = c(10, 12))`. Comparez les AIC et commentez les degrés de liberté effectifs de l'interaction.

b) Tracez le cycle journalier prédit pour quatre jours de l'année (15 janvier, 15 avril, 15 juillet et 15 octobre). Calculez l'amplitude du cycle (maximum moins minimum) en janvier et en juillet.

c) Que vaut la valeur-p du test de l'interaction ? Pourquoi faut-il l'interpréter avec prudence ?

Voir la solution

Exercice 5 (difficile). GAM binomial pour les dépassements d'ozone.

Pour chaque jour de l'année, comptez le nombre d'heures où la concentration d'ozone à Longueuil dépasse 40 ppb, ainsi que le nombre d'heures mesurées.

a) Ajustez un GAM binomial du nombre d'heures de dépassement, avec une spline cyclique du jour (`k = 20`). Décrivez la courbe sur l'échelle des probabilités.

b) Estimez le paramètre de dispersion par la statistique de Pearson divisée par les degrés de liberté résiduels. Qu'en concluez-vous ? D'où vient le problème ?

c) Réajustez le modèle avec `family = quasibinomial`. Comment changent les degrés de liberté effectifs et la largeur des bandes ? Pourquoi ?

Voir la solution

Exercice 6 (difficile). Ce que calcule `gam`.

On reprend le modèle `g.arbres` de l'exercice 1. Les objets `gam` contiennent la matrice de design (`model.matrix(g.arbres)`), les paramètres de lissage (`g.arbres$sp`) et, pour chaque fonction lisse, la matrice de pénalité (`g.arbres$smooth[[1]]$S[[1]]`), dont les colonnes correspondent aux coefficients `g.arbres$smooth[[1]]$first.para` à `last.para`.

a) Construisez la matrice de pénalité complète λS, de la taille de β, et vérifiez numériquement que $\widehat{\beta} = (X^\top X + \lambda S)^{-1} X^\top Y$ redonne les coefficients de `coef(g.arbres)`.

b) Calculez la matrice d'influence A et vérifiez que $\mathrm{Tr}(A)$ est la somme des degrés de liberté effectifs (`sum(g.arbres$edf)`).

c) Calculez le critère GCV à partir de sa définition et comparez-le à `g.arbres$gcv.ubre`. Calculez aussi le critère OCV.

Voir la solution

9.4.2 Exercices théoriques

Exercice 1 (facile). Considérez le modèle additif $Y_i = \alpha + f_1(x_{i1}) + f_2(x_{i2}) + \epsilon_i$.

a) Montrez que, sans contrainte, le modèle n'est pas identifiable : pour toute constante c, $(\alpha, f_1 + c, f_2 - c)$ et $(\alpha - c, f_1 + c, f_2)$ donnent les mêmes valeurs de $E[Y_i]$.

b) Montrez que, sous la contrainte de centrage $\sum_i f_j(x_{ij}) = 0$ pour $j = 1, 2$, l'ordonnée à l'origine est déterminée par $\alpha = \frac{1}{n}\sum_i E[Y_i]$, et que $\hat{\alpha} = \bar{Y}$ lorsque les valeurs ajustées somment à $\sum_i Y_i$.

c) Comment interprète-t-on alors le graphique de $\hat{f}_j$ produit par `plot` pour un objet `gam` ?

Voir la solution

Exercice 2 (moyen). L'algorithme des moindres carrés pondérés itératifs pénalisés du Chapitre 9 utilise une réponse de travail z_i et des poids w_i.

a) Calculez z_i et w_i pour un GAM logistique ($Y_i \sim \text{Bernoulli}(\pi_i)$, lien logit).

b) Calculez z_i et w_i pour un GAM Poisson (lien log).

c) Montrez que, pour un modèle additif gaussien avec lien identité et des λ_j fixés, l'algorithme converge en une seule itération, quelle que soit la valeur de départ.

Voir la solution

Exercice 3 (moyen). Soit un lisseur linéaire $\hat{Y} = AY$, de critères

$$VC_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - A_{ii}}\right)^2 \qquad \text{et} \qquad GCV = \frac{n\sum_i (y_i - \hat{y}_i)^2}{\{n - \operatorname{Tr}(A)\}^2}.$$

a) Montrez que la GCV s'obtient en remplaçant chaque A_{ii} de $VC_{(n)}$ par leur moyenne $\operatorname{Tr}(A)/n$.

b) Montrez que, lorsque $\operatorname{Tr}(A)/n$ est petit,

$$GCV \approx \frac{RSS}{n} + \frac{2\operatorname{Tr}(A)}{n} \cdot \frac{RSS}{n},$$

où $RSS = \sum_i (y_i - \hat{y}_i)^2$. Comparez avec le critère UBRE du Chapitre 9 pour $\phi = \sigma^2$ connu.

c) Donnez une raison de préférer la GCV à $VC_{(n)}$ en pratique.

Voir la solution

Exercice 4 (difficile). Interprétation bayésienne de la pénalité.

Considérez le modèle $Y \mid \beta \sim \mathcal{N}(X\beta, \sigma^2 I)$ et la loi a priori $\beta \sim \mathcal{N}(0, \sigma^2 P^{-1})$, où $P = \sum_j \lambda_j S_j$ est supposée définie positive (en pratique, P est singulière et la loi a priori est impropre dans les directions non pénalisées, mais le calcul est le même).

a) Montrez que la densité a posteriori de β est proportionnelle à

$$\exp\left\{-\frac{1}{2\sigma^2}\left(\|Y - X\beta\|^2 + \beta^\top P\beta\right)\right\}.$$

b) En complétant le carré, montrez que la loi a posteriori est normale, de moyenne $\widehat{\beta} = (X^\top X + P)^{-1}X^\top Y$ et de matrice de variance-covariance $\sigma^2(X^\top X + P)^{-1}$.

c) Faites le lien avec les intervalles de crédibilité du Chapitre 9 et avec l'exercice 3 du niveau moyen du chapitre 6.

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Faraway (2016). *Extending the Linear Model with R*
— James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R*
— Wood (2017). *Generalized Additive Models : An Introduction with R*
— Hastie, Tibshirani et Friedman (2009). *The Elements of Statistical Learning*
— Ruppert, Wand et Carroll (2003). *Semiparametric Regression*

10 Données corrélées : modèles mixtes et GEE

Ce chapitre utilise les librairies R suivantes.

```
library(ggplot2)
library(lme4)
library(geepack)
library(mgcv)
library(dplyr)
```

Tous les modèles des chapitres précédents supposent des observations indépendantes. Cette hypothèse est souvent fausse. Les mesures répétées sur un même patient se ressemblent, les arbres d'une même région écologique poussent dans des conditions semblables, et les heures consécutives d'une même journée partagent la même météo, comme on l'a vu avec l'ozone au Chapitre 8 et au Chapitre 9. Ce chapitre présente deux façons d'en tenir compte lorsque les observations sont regroupées. La première modélise explicitement la source de dépendance par des effets aléatoires (modèles mixtes). La seconde se contente de spécifier la moyenne et corrige l'inférence (équations d'estimation généralisées, ou GEE).

Dans tout le chapitre, Y_{ij} désigne la j^{e} observation du groupe i, avec $i = 1, \dots, m$ et $j = 1, \dots, n_i$, et $N = \sum_i n_i$ est le nombre total d'observations. Les observations de groupes différents sont indépendantes, mais celles d'un même groupe peuvent être corrélées.

10.1 Conséquences de la dépendance

Ignorer la corrélation ne biaise généralement pas l'estimation des coefficients de la moyenne, mais fausse leurs erreurs-types. Le cas le plus simple le montre.

Propriété — Effet du plan

Supposons m groupes de même taille c, des observations de même variance σ_Y^2 et une corrélation ρ entre deux observations d'un même groupe. La moyenne globale $\bar{Y}$ des $N = mc$ observations a pour variance

$$Var[\bar{Y}] = \frac{\sigma_Y^2}{N}\left\{1 + (c-1)\rho\right\}.$$

Le facteur $1 + (c-1)\rho$ est l'effet du plan (*design effect*), et $N/\{1 + (c-1)\rho\}$ est la taille d'échantillon effective.

Par exemple, avec des groupes de $c = 24$ observations (les heures d'une journée) et $\rho = 0{,}5$, l'effet du plan vaut 12,5 : un an de mesures horaires ne contient pas plus d'information sur la moyenne que 700 observations indépendantes environ. La formule qui suppose l'indépendance sous-estime alors l'erreur-type d'un facteur $\sqrt{12{,}5} \approx 3{,}5$. L'effet n'est pas toujours dans ce sens. Pour une variable explicative qui varie à l'intérieur des groupes (la dose administrée à un même patient à des moments différents, par exemple), comparer chaque groupe à lui-même élimine la variabilité entre groupes, et l'erreur-type naïve peut être trop grande. Dans les deux cas, l'inférence qui suppose l'indépendance est fausse.

10.2 Modèle linéaire mixte

10.2.1 Modèle à ordonnée à l'origine aléatoire

Définition — Modèle linéaire mixte à ordonnée à l'origine aléatoire

$$Y_{ij} = x_{ij}^\top \beta + b_i + \epsilon_{ij}, \qquad b_i \sim \mathcal{N}(0, \sigma_b^2), \qquad \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

où les b_i et les ϵ_{ij} sont mutuellement indépendants, et indépendants des variables explicatives. Les coefficients β sont les effets fixes, communs à tous les groupes. Les b_i sont les effets aléatoires : chaque groupe a sa propre ordonnée à l'origine, tirée d'une loi normale centrée.

Plutôt que d'estimer un paramètre par groupe, comme le ferait un effet fixe du groupe, le modèle estime la seule variance σ_b^2 de ces ordonnées. C'est utile quand les groupes sont nombreux et vus comme un échantillon d'une population de groupes (des régions, des patients, des écoles), et quand on veut généraliser au-delà des groupes observés.

Propriété — Corrélation intraclasse

Sous ce modèle, $Var[Y_{ij}] = \sigma_b^2 + \sigma^2$ et, pour $j \neq k$, $Cov[Y_{ij}, Y_{ik}] = Var[b_i] = \sigma_b^2$. La corrélation entre deux observations d'un même groupe est donc

$$\rho = \frac{\sigma_b^2}{\sigma_b^2 + \sigma^2},$$

appelée coefficient de corrélation intraclasse. C'est l'effet aléatoire partagé qui crée la dépendance.

10.2.2 Forme générale

Définition — Modèle linéaire mixte

Pour le groupe i, on écrit

$$y_i = X_i\beta + Z_i b_i + \epsilon_i, \qquad b_i \sim \mathcal{N}(0, D), \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2 I_{n_i}),$$

où X_i est la matrice de design $n_i \times (p+1)$ des effets fixes et Z_i celle des effets aléatoires, de dimension $n_i \times q$. Une colonne de 1 dans Z_i donne une ordonnée à l'origine aléatoire, et les colonnes $(1, t_{ij})$ donnent une ordonnée et une pente aléatoires.

En intégrant les effets aléatoires, on obtient la loi marginale

$$y_i \sim \mathcal{N}(X_i\beta,\ V_i), \qquad V_i = Z_i D Z_i^\top + \sigma^2 I_{n_i}.$$

Le modèle mixte est donc un modèle linéaire dont la matrice de variance-covariance n'est plus $\sigma^2 I$, mais diagonale par blocs, un bloc V_i par groupe. Les effets aléatoires sont une façon économique de paramétrer ces blocs : avec une ordonnée à l'origine aléatoire, deux paramètres (σ_b^2 et σ^2) suffisent, quelle que soit la taille des groupes.

Remarque — Effets emboîtés, effets croisés et erreurs corrélées

La forme générale couvre plusieurs structures courantes.

- Effets emboîtés : des élèves dans des classes, elles-mêmes dans des écoles. Chaque classe appartient à une seule école, et on pose un effet aléatoire par école et un effet par classe à l'intérieur de l'école. En R : `(1 | ecole/classe)`, qui équivaut à `(1 | ecole) + (1 | ecole:classe)`.
- Effets croisés : des évaluateurs qui notent tous les mêmes copies. Une note dépend de la copie et de l'évaluateur, sans que l'un soit emboîté dans l'autre. En R : `(1 | copie) + (1 | evaluateur)`. Les observations ne se découpent plus en groupes indépendants, mais la loi marginale reste normale, de matrice $V = ZGZ^\top + \sigma^2 I$, et l'estimation se fait de la même façon. La fonction `lmer` traite ces deux cas.
- Erreurs corrélées : on peut aussi remplacer $\sigma^2 I_{n_i}$ par une matrice $\sigma^2 R_i$ structurée, par exemple autorégressive pour des mesures rapprochées dans le temps. La fonction `lmer` ne le permet pas, mais la fonction `lme` de la librairie `nlme` et la fonction `gamm` de `mgcv` le font (argument `correlation`). L'exemple sur l'ozone, plus loin dans ce chapitre, en montre l'utilité.

10.2.3 Estimation

Propriété — Moindres carrés généralisés

Si les V_i sont connues, l'estimateur du maximum de vraisemblance de β est l'estimateur des moindres carrés généralisés

$$\hat{\beta} = \left(\sum_{i=1}^{m} X_i^\top V_i^{-1} X_i\right)^{-1} \sum_{i=1}^{m} X_i^\top V_i^{-1} y_i, \qquad Var[\hat{\beta}] = \left(\sum_{i=1}^{m} X_i^\top V_i^{-1} X_i\right)^{-1}.$$

En pratique, D et σ^2 sont inconnus. Notons θ le vecteur de ces paramètres de variance, de sorte que $V_i = V_i(\theta)$. Ils s'estiment en maximisant la log-vraisemblance de la loi marginale.

Définition — Vraisemblances ML et REML

La log-vraisemblance marginale est

$$\ell(\beta, \theta) = -\frac{1}{2} \sum_{i=1}^{m} \left\{ n_i \log(2\pi) + \log |V_i| + (y_i - X_i\beta)^\top V_i^{-1} (y_i - X_i\beta) \right\}.$$

Pour θ fixé, elle est maximale en $\hat{\beta}(\theta)$, l'estimateur des moindres carrés généralisés. L'estimateur du maximum de vraisemblance (ML) de θ maximise donc la vraisemblance profilée $\ell\{\hat{\beta}(\theta), \theta\}$. Le maximum de vraisemblance restreint (REML) maximise plutôt

$$\ell_R(\theta) = \ell\{\hat{\beta}(\theta), \theta\} - \frac{1}{2} \log \left| \sum_{i=1}^{m} X_i^\top V_i^{-1} X_i \right| + \text{constante}.$$

Le terme supplémentaire de ℓ_R corrige un biais connu. Le maximum de vraisemblance sous-estime les composantes de variance, pour la même raison que $\hat{\sigma}^2_{MV} = SSE/n$ sous-estime σ^2 en régression linéaire : il ne tient pas compte des degrés de liberté utilisés pour estimer β. La vraisemblance REML est celle de $N - (p+1)$ combinaisons linéaires des observations dont la loi ne dépend pas de β (des contrastes d'erreurs). En régression linéaire ordinaire, REML redonne exactement l'estimateur sans biais $SSE/\{N - (p+1)\}$. Une fois $\hat{\theta}$ obtenu, on calcule $\hat{\beta} = \hat{\beta}(\hat{\theta})$.

Remarque — ML ou REML ?

REML est le choix par défaut pour estimer les composantes de variance. Par contre, la vraisemblance REML dépend de la matrice X, de sorte que deux modèles qui n'ont pas les mêmes effets fixes ne se comparent pas par leurs vraisemblances REML. Pour un test du rapport de vraisemblance sur les effets fixes, ou pour comparer leurs AIC, on ajuste les deux modèles par ML. En R, la fonction `lmer` de la librairie `lme4` utilise REML par

défaut (`REML = TRUE`), et `anova` réajuste automatiquement les modèles par ML avant de les comparer. Pour comparer deux modèles qui ont les mêmes effets fixes mais des effets aléatoires différents, les vraisemblances REML conviennent.

10.2.4 Inférence sur les effets fixes

L'erreur-type de $\hat{\beta}_j$ s'obtient en remplaçant θ par $\hat{\theta}$ dans $Var[\hat{\beta}]$. Cette formule ignore l'incertitude sur $\hat{\theta}$, et elle sous-estime donc un peu la variance réelle de $\hat{\beta}$. L'écart est négligeable quand les groupes sont nombreux, mais pas avec une dizaine de groupes. Trois approches sont courantes.

— Le test de Wald compare $\hat{\beta}_j / \mathrm{ET}(\hat{\beta}_j)$ à la loi normale. C'est l'approche la plus simple, valide quand m est grand.
— Les tests t et F avec degrés de liberté approchés comparent la même statistique à une loi de Student (ou de Fisher pour plusieurs coefficients), dont les degrés de liberté sont estimés à partir des données. Les méthodes de Satterthwaite et de Kenward-Roger sont les plus utilisées, et la seconde corrige aussi l'erreur-type. La librairie `lmerTest` les ajoute aux sorties de `lmer`.
— Le test du rapport de vraisemblance compare deux modèles emboîtés ajustés par ML. Il repose lui aussi sur une approximation asymptotique, et pour les effets fixes, il rejette trop souvent quand les groupes sont peu nombreux. Dans ce cas, on préfère les tests t et F de Kenward-Roger, ou on calibre le test par un bootstrap paramétrique : simuler des données sous le modèle réduit, ajuster les deux modèles à chaque jeu simulé et comparer la statistique observée à sa loi simulée (Chapitre 1).

Les degrés de liberté approchés reflètent la quantité d'information disponible sur chaque coefficient. Pour un coefficient qui varie à l'intérieur des groupes, ils sont proches du nombre d'observations. Pour un coefficient qui ne varie qu'entre les groupes, ou pour un effet dont la variabilité entre groupes est modélisée par une pente aléatoire, ils sont proches du nombre de groupes.

10.2.5 Prédiction des effets aléatoires

Les b_i ne sont pas des paramètres, mais on peut les prédire à partir des données du groupe.

Propriété — BLUP et rétrécissement
Le vecteur (b_i, y_i) suit une loi normale, avec $Cov[b_i, y_i] = DZ_i^\top$. Par les propriétés de la loi normale multivariée,

$$E[b_i \mid y_i] = DZ_i^\top V_i^{-1}(y_i - X_i\beta), \qquad Var[b_i \mid y_i] = D - DZ_i^\top V_i^{-1} Z_i D.$$

Le prédicteur $\hat{b}_i = DZ_i^\top V_i^{-1}(y_i - X_i\hat{\beta})$ est le meilleur prédicteur linéaire sans biais (BLUP, *best linear unbiased predictor*). Pour le modèle à ordonnée à l'origine aléatoire, il se simplifie en

$$\hat{b}_i = \frac{n_i\sigma_b^2}{n_i\sigma_b^2 + \sigma^2}\left(\bar{y}_i - \bar{x}_i^\top\hat{\beta}\right),$$

où $\bar{y}_i$ et $\bar{x}_i$ sont les moyennes du groupe i.

Au sens strict, le nom de BLUP suppose D et σ^2 connus. Avec les composantes de variance estimées, on parle parfois de BLUP empirique (EBLUP), mais l'usage garde souvent le terme BLUP.

Le facteur $n_i\sigma_b^2/(n_i\sigma_b^2 + \sigma^2)$ est compris entre 0 et 1 : le BLUP rétrécit le résidu moyen du groupe vers 0, d'autant plus que le groupe est petit. Un groupe d'une seule observation est fortement ramené vers la moyenne générale, alors qu'un grand groupe garde presque son écart observé. C'est le même rétrécissement que celui de ridge (Chapitre 6), et la Section 10.3 montre que ce n'est pas une coïncidence.

Remarque — Prédire pour un groupe observé ou pour un nouveau groupe

Un modèle mixte donne deux prédictions différentes de la réponse en x_0. Pour un groupe i déjà observé, la prédiction conditionnelle est $x_0^\top\hat{\beta} + \hat{b}_i$, et sa variance d'erreur contient σ^2 et la variance résiduelle de b_i sachant les données, qui diminue avec n_i. Pour un nouveau groupe, rien ne renseigne sur son effet, et la prédiction est $x_0^\top\hat{\beta}$, avec une variance d'erreur d'au moins $\sigma_b^2 + \sigma^2$. En R, `predict(modele, newdata)` donne la première, et `predict(modele, newdata, re.form = NA)` donne la seconde. Ce choix compte aussi en validation croisée (Chapitre 5) : retirer des observations à l'intérieur des groupes évalue la prédiction pour des groupes connus, alors que retirer des groupes entiers évalue la prédiction pour de nouveaux groupes. C'est presque toujours la seconde qui intéresse quand le modèle sert à extrapoler.

10.2.6 Tester une composante de variance

Pour tester $H_0 : \sigma_b^2 = 0$ (pas d'effet de groupe), on compare le modèle mixte au modèle linéaire ordinaire par un test du rapport de vraisemblance. Comme pour le test de surdispersion, la valeur $\sigma_b^2 = 0$ est à la frontière de l'espace des paramètres. La statistique ne suit donc pas une loi χ^2_1 sous H_0.

Propriété — Loi du rapport de vraisemblance à la frontière

Sous $H_0 : \sigma_b^2 = 0$, et avec des groupes indépendants nombreux, la statistique du rapport de vraisemblance suit approximativement le mélange

$$\tfrac{1}{2}\chi^2_0 + \tfrac{1}{2}\chi^2_1,$$

où χ^2_0 désigne une masse en 0. La valeur-p est donc la moitié de celle que donne une χ^2_1 (pour une statistique strictement positive). Pour tester une pente aléatoire, on compare le modèle à ordonnée et pente aléatoires au modèle à ordonnée aléatoire seule. L'hypothèse nulle fixe alors à 0 une variance (à la frontière) et une covariance (à l'intérieur de l'espace), et la loi de référence devient $\tfrac{1}{2}\chi^2_1 + \tfrac{1}{2}\chi^2_2$.

Dans les deux cas, utiliser la loi χ^2 usuelle donne un test conservateur : il rejette moins souvent que prévu. Les vraisemblances REML conviennent ici, puisque les deux modèles ont les mêmes effets fixes.

Exemple — Relation hauteur-diamètre par région écologique sur `arbres`

Les 200 arbres du jeu de données viennent de 30 régions écologiques, qui comptent de 1 à 23 arbres. Deux arbres d'une même région partagent le climat, les sols et l'histoire des perturbations, de sorte qu'à diamètre égal, leurs hauteurs se ressemblent sans doute plus que celles d'arbres de régions différentes. On reprend le modèle log-log du Chapitre 5, avec une ordonnée à l'origine propre à chaque espèce, et on ajoute une ordonnée à l'origine aléatoire par région :

$$\log H_{ij} = \beta_{0,e(ij)} + \beta_1 \log D_{ij} + b_i + \epsilon_{ij}.$$

```
arbres <- read.csv("Jeux de données/arbres.csv") |>
  mutate(espece = factor(species_fr), region = factor(ecological_region))

mod.lm  <- lm(log(height_m) ~ log(diameter_cm) + espece, data = arbres)
mod.lmm <- lmer(log(height_m) ~ log(diameter_cm) + espece + (1 | region),
 ↪  data = arbres)
print(summary(mod.lmm), correlation = FALSE)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: log(height_m) ~ log(diameter_cm) + espece + (1 | region)
   Data: arbres

REML criterion at convergence: -89.9

Scaled residuals:
```

```
    Min      1Q  Median      3Q     Max
-4.3822 -0.5955  0.0753  0.6228  2.2010

Random effects:
 Groups   Name        Variance Std.Dev.
 region   (Intercept) 0.004109 0.0641
 Residual             0.030957 0.1759
Number of obs: 200, groups:  region, 30

Fixed effects:
                     Estimate Std. Error t value
(Intercept)          1.047360   0.102607  10.208
log(diameter_cm)     0.590805   0.036419  16.222
especeÉpinette noire -0.145287   0.036199  -4.014
especeÉrable rouge   -0.008069   0.036670  -0.220
especeSapin baumier  -0.142863   0.036665  -3.896
```

La corrélation intraclasse estimée vaut $\hat{\rho} = 0{,}064^2/(0{,}064^2 + 0{,}176^2) \approx 0{,}12$: environ 12 % de la variance résiduelle de $\log H$ est attribuable à la région. Le test de $H_0 : \sigma_b^2 = 0$ compare les deux modèles ajustés par ML :

```
anova(mod.lmm, mod.lm)
```

```
refitting model(s) with ML (instead of REML)

Data: arbres
Models:
mod.lm: log(height_m) ~ log(diameter_cm) + espece
mod.lmm: log(height_m) ~ log(diameter_cm) + espece + (1 | region)
        npar      AIC     BIC logLik -2*log(L)  Chisq Df Pr(>Chisq)
mod.lm     6  -96.598 -76.808 54.299   -108.60
mod.lmm    7 -101.936 -78.848 57.968   -115.94 7.3382  1   0.006751 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La statistique vaut 7,34. La valeur-p affichée, 0,0068, suppose une loi χ^2_1. Avec la correction pour la frontière, elle vaut la moitié, 0,0034. L'effet de la région est donc significatif. Les coefficients du diamètre et leurs erreurs-types changent peu :

```
rbind(lm = coef(summary(mod.lm))[2, 1:2], lmm = coef(summary(mod.lmm))[2,
 ↪  1:2])
```

```
     Estimate Std. Error
lm  0.5968473 0.03785142
lmm 0.5908048 0.03641922
```

C'est attendu : le diamètre varie surtout à l'intérieur des régions, et la corrélation intraclasse est modérée. Les tests t avec degrés de liberté de Satterthwaite le confirment. Le diamètre a environ 190 degrés de liberté, presque autant que d'arbres, de sorte que la loi de Student et la loi normale donnent ici les mêmes conclusions.

```
round(summary(lmerTest::lmer(log(height_m) ~ log(diameter_cm) + espece +
 ↪  (1 | region),
                             data = arbres))$coefficients, 4)
```

```
                     Estimate Std. Error       df t value Pr(>|t|)
(Intercept)            1.0474     0.1026 193.9505 10.2075   0.0000
log(diameter_cm)       0.5908     0.0364 187.2411 16.2223   0.0000
especeÉpinette noire  -0.1453     0.0362 187.4860 -4.0135   0.0001
especeÉrable rouge    -0.0081     0.0367 193.1805 -0.2200   0.8261
especeSapin baumier   -0.1429     0.0367 191.4136 -3.8965   0.0001
```

L'effet de la région se voit plutôt dans les prédictions. La Figure 10.1 compare, pour chaque région, le résidu moyen du modèle sans effet de région au BLUP $\hat{b}_i$.

```
blup <- arbres |>
  mutate(residu = residuals(mod.lm)) |>
  group_by(region) |>
  summarise(n = n(), moyenne = mean(residu)) |>
  mutate(region = as.character(region),
         blup = ranef(mod.lmm)$region[region, 1])
ggplot(blup, aes(moyenne, blup, size = n)) +
  geom_abline(linetype = "dotted") +
  geom_hline(yintercept = 0, colour = "grey60") +
  geom_point(alpha = 0.6) +
  coord_equal() +
  labs(x = "Résidu moyen de la région", y = expression(hat(b)[i]), size =
   ↪  "Arbres") +
  theme_minimal(base_size = 13)
```

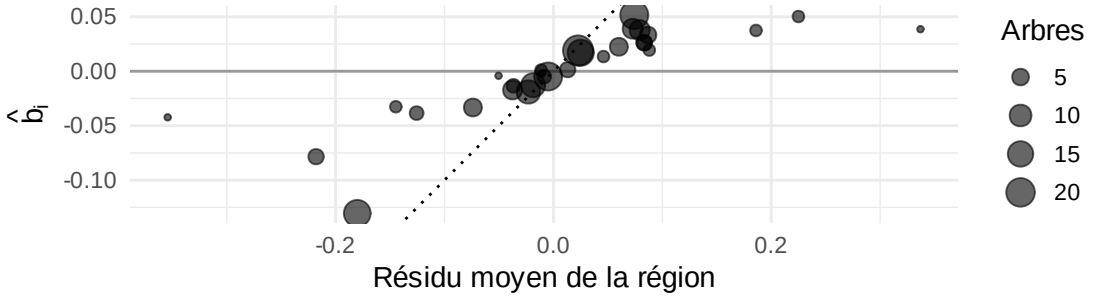


FIGURE 10.1 : Résidu moyen de chaque région (modèle linéaire sans effet de région) et BLUP $\widehat{b}_i$ du modèle mixte, sur des axes à la même échelle. La taille des points est proportionnelle au nombre d'arbres de la région ; la droite pointillée est la diagonale.

Les BLUP sont presque tous plus près de 0 que les résidus moyens. Les rares exceptions viennent de ce que la formule de rétrécissement s'applique aux résidus calculés avec l'estimateur $\widehat{\beta}$ du modèle mixte, légèrement différent de celui de `lm`. Les petites régions, dont le résidu moyen repose sur un ou deux arbres, sont presque entièrement ramenées vers 0, alors que les régions qui comptent une vingtaine d'arbres gardent l'essentiel de leur écart. Le modèle emprunte de l'information aux autres régions pour les régions peu échantillonnées.

10.2.7 Effets intra-groupes et inter-groupes

La définition du modèle mixte suppose que les effets aléatoires sont indépendants des variables explicatives. Cette hypothèse est plus forte qu'elle n'en a l'air. Une variable x_{ij} qui varie à l'intérieur des groupes se décompose en

$$x_{ij} = \bar{x}_i + (x_{ij} - \bar{x}_i),$$

une partie inter-groupes $\bar{x}_i$ et une partie intra-groupe $x_{ij} - \bar{x}_i$. Le modèle mixte usuel donne le même coefficient aux deux parties. Or elles ne mesurent pas forcément la même chose. Dans l'exemple des arbres, l'effet intra-région compare des arbres de diamètres différents dans un même milieu. L'effet inter-régions compare des régions où les arbres sont en moyenne plus gros

à des régions où ils sont plus petits, et ces régions diffèrent aussi par le climat ou les sols, qui influencent la hauteur. Si b_i est corrélé avec $\bar{x}_i$, l'estimateur du modèle mixte est biaisé.

Propriété — Modèle intra-inter

Le modèle

$$Y_{ij} = \beta_0 + \beta_W(x_{ij} - \bar{x}_i) + \beta_B\,\bar{x}_i + b_i + \epsilon_{ij}$$

sépare l'effet intra-groupe β_W et l'effet inter-groupes β_B. L'estimateur de β_W ne dépend que des écarts à l'intérieur des groupes. Il reste donc valide même si b_i est corrélé avec $\bar{x}_i$, comme l'estimateur du modèle à effets fixes de groupe (un coefficient par groupe, sans effet aléatoire). Lorsque toutes les variables qui varient à l'intérieur des groupes sont ainsi décomposées, cet estimateur coïncide exactement avec celui du modèle à effets fixes. Le modèle mixte usuel correspond à $\beta_W = \beta_B$, une hypothèse qu'on peut tester par un test du rapport de vraisemblance ou un test de Wald.

Ce test est une version du test de Hausman utilisé en économétrie pour choisir entre effets fixes et effets aléatoires. Quand l'hypothèse $\beta_W = \beta_B$ est rejetée, on garde le modèle intra-inter, qui conserve les avantages du modèle mixte (effets des variables constantes dans les groupes, prédiction pour de nouveaux groupes) tout en protégeant l'estimation de l'effet intra-groupe.

Exemple — Effet intra et inter-régions du diamètre

```
arbres <- arbres |>
  group_by(region) |>
  mutate(ld = log(diameter_cm),
         ld.inter = mean(ld),          # moyenne de la région
         ld.intra = ld - ld.inter) |>
  ungroup()

mod.ii <- lmer(log(height_m) ~ ld.intra + ld.inter + espece + (1 |
 ↪  region), data = arbres)
mod.fe <- lm(log(height_m) ~ ld + espece + region, data = arbres)   #
 ↪  effets fixes de région
rbind(intra = coef(summary(mod.ii))["ld.intra", 1:2],
      inter = coef(summary(mod.ii))["ld.inter", 1:2],
      effets.fixes = coef(summary(mod.fe))["ld", 1:2])
```

```
              Estimate Std. Error
intra        0.5731660 0.03800694
inter        0.7642171 0.11662212
effets.fixes 0.5729230 0.03788639
```

```
anova(mod.lmm, mod.ii)
```

```
refitting model(s) with ML (instead of REML)

Data: arbres
Models:
mod.lmm: log(height_m) ~ log(diameter_cm) + espece + (1 | region)
mod.ii: log(height_m) ~ ld.intra + ld.inter + espece + (1 | region)
        npar     AIC     BIC logLik -2*log(L)  Chisq Df Pr(>Chisq)
mod.lmm    7 -101.94 -78.848 57.968   -115.94
mod.ii     8 -102.47 -76.079 59.233   -118.47 2.5293  1     0.1118
```

L'effet intra-région, 0,573, est pratiquement égal à celui du modèle à effets fixes de région. Les deux ne coïncident pas exactement parce que l'espèce, qui varie aussi à l'intérieur des régions, n'a pas été décomposée. L'effet inter-régions, 0,764, est plus grand, mais il est beaucoup moins précis, puisqu'il ne repose que sur une trentaine de moyennes régionales. Le test de $\beta_W = \beta_B$ donne une valeur-p de 0,11, de sorte que les données ne contredisent pas le modèle mixte usuel. Son coefficient, 0,591, est une moyenne pondérée des deux effets, dominée par l'effet intra-région, le plus précis.

10.2.8 Pentes aléatoires

Avec des mesures répétées dans le temps, l'effet du temps varie souvent d'un individu à l'autre. On ajoute alors une pente aléatoire,

$$Y_{ij} = \beta_0 + \beta_1 t_{ij} + b_{0i} + b_{1i} t_{ij} + \epsilon_{ij}, \qquad \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix} \sim \mathcal{N}\left\{0,\ D = \begin{pmatrix} d_{00} & d_{01} \\ d_{01} & d_{11} \end{pmatrix}\right\}.$$

En R, on écrit `(1 + t | groupe)`, ou simplement `(t | groupe)`. La pente aléatoire change aussi la structure de variance : $Var[Y_{ij}] = d_{00} + 2t_{ij}d_{01} + t_{ij}^2 d_{11} + \sigma^2$ varie avec le temps, et deux mesures d'un même individu ne sont pas toutes corrélées de la même façon (voir les exercices). La covariance d_{01} dépend du choix de l'origine du temps, de sorte qu'il est souvent utile de centrer t à une valeur qui a du sens.

Exemple — Privation de sommeil

Le jeu de données `sleepstudy` de la librairie `lme4` donne le temps de réaction moyen (`Reaction`, en ms) de 18 sujets privés de sommeil, mesuré chaque jour pendant 10 jours (`Days`, de 0 à 9).

```
mod.sleep <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy)
VarCorr(mod.sleep)
```

```
 Groups   Name        Std.Dev. Corr
 Subject  (Intercept) 24.7407
          Days         5.9221  0.066
 Residual             25.5918
```

```
rbind(lm = coef(summary(lm(Reaction ~ Days, data = sleepstudy)))[2, 1:2],
      lmm = coef(summary(mod.sleep))[2, 1:2])
```

```
    Estimate Std. Error
lm  10.46729   1.238195
lmm 10.46729   1.545790
```

Chaque jour de privation allonge le temps de réaction de 10,5 ms en moyenne. L'écart-type des pentes individuelles est de 5,9 ms par jour, de sorte que la pente varie beaucoup d'un sujet à l'autre : à peu près entre -1 et 22 ms par jour pour 95 % des sujets. L'estimation de la pente moyenne est la même qu'avec `lm`, parce que le plan est équilibré, mais son erreur-type est plus grande. La régression ordinaire traite les 180 mesures comme indépendantes, alors que l'information sur la pente moyenne vient en fait de 18 sujets dont les pentes diffèrent. Les degrés de liberté de Satterthwaite le disent aussi : ils valent 17 pour la pente moyenne, et non 178.

```
summary(lmerTest::lmer(Reaction ~ Days + (Days | Subject), data =
 ↪  sleepstudy))$coefficients
```

```
             Estimate Std. Error       df   t value     Pr(>|t|)
(Intercept) 251.40510   6.824597 16.99973 36.838090 1.171558e-17
Days         10.46729   1.545790 16.99998  6.771481 3.263824e-06
```

La pente aléatoire est-elle nécessaire ? On compare le modèle au modèle à ordonnée à l'origine aléatoire seule. Les deux modèles ont les mêmes effets fixes, et on garde les vraisemblances REML (`refit = FALSE`).

```
mod.sleep0 <- lmer(Reaction ~ Days + (1 | Subject), data = sleepstudy)
test.pente <- anova(mod.sleep, mod.sleep0, refit = FALSE)
q.pente <- test.pente$Chisq[2]
c(statistique = q.pente,
  p.chi2.2 = pchisq(q.pente, 2, lower.tail = FALSE),
  p.melange = 0.5 * pchisq(q.pente, 1, lower.tail = FALSE) +
              0.5 * pchisq(q.pente, 2, lower.tail = FALSE))
```

```
 statistique     p.chi2.2    p.melange
4.283681e+01 4.990041e-10 2.792530e-10
```

La statistique vaut 42,8, et la valeur-p est négligeable avec ou sans correction pour la frontière : les sujets ne réagissent pas tous de la même façon à la privation de sommeil. La Figure 10.2 montre le rétrécissement en deux dimensions. Pour chaque sujet, elle compare la droite ajustée par `lm` sur ses seules 10 mesures à la droite prédite par le modèle mixte, $(\hat{\beta}_0 + \hat{b}_{0i}) + (\hat{\beta}_1 + \hat{b}_{1i})t$.

```
sujets <- c("309", "330", "335", "337", "350", "370")
indiv <- t(sapply(split(sleepstudy, sleepstudy$Subject),
                  function(d) coef(lm(Reaction ~ Days, data = d))))
droites <- rbind(
  data.frame(Subject = rownames(indiv), a = indiv[, 1], b = indiv[, 2],
   ↪ ajustement = "Séparé"),
  data.frame(Subject = rownames(coef(mod.sleep)$Subject), a =
   ↪ coef(mod.sleep)$Subject[, 1],
             b = coef(mod.sleep)$Subject[, 2], ajustement = "Modèle
              ↪ mixte")
)
droites <- filter(droites, Subject %in% sujets)
ggplot(filter(sleepstudy, Subject %in% sujets), aes(Days, Reaction)) +
  geom_abline(intercept = fixef(mod.sleep)[1], slope =
   ↪ fixef(mod.sleep)[2], colour = "grey70") +
  geom_point(size = 1, alpha = 0.6) +
  geom_abline(data = droites, aes(intercept = a, slope = b, linetype =
   ↪ ajustement)) +
  facet_wrap(~ Subject, nrow = 2) +
  scale_linetype_manual(values = c("Séparé" = "dotted", "Modèle mixte" =
   ↪ "solid")) +
  labs(x = "Jours de privation", y = "Temps de réaction (ms)", linetype =
   ↪ NULL) +
  theme_minimal(base_size = 13) + theme(legend.position = "bottom")
```

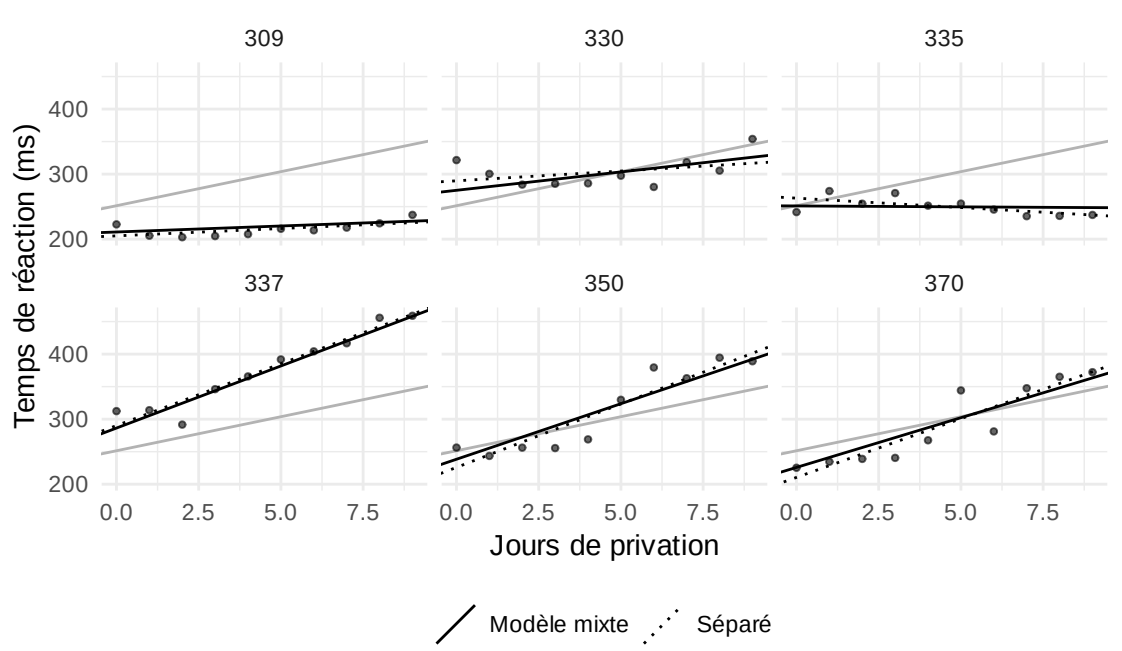


FIGURE 10.2 : Droites ajustées séparément pour chaque sujet (pointillés) et droites prédites par le modèle mixte (traits pleins), pour six sujets de `sleepstudy`. La droite grise est la droite moyenne $\hat{\beta}_0 + \hat{\beta}_1 t$.

Les droites du modèle mixte sont tirées vers la droite moyenne, en ordonnée à l'origine comme en pente. Le rétrécissement est faible pour les sujets 309 et 337, dont les mesures suivent de près une droite. Il est plus marqué pour les sujets dont les mesures sont dispersées autour de leur droite ou dont la droite est atypique. La pente du sujet 335, par exemple, est négative quand elle est estimée seule, et le modèle mixte la ramène près de 0. Celle du sujet 370, très forte et appuyée sur des mesures dispersées, est réduite de près de 3 ms par jour.

Remarque — Ajustement singulier

La fonction `lmer` affiche parfois le message `boundary (singular) fit: see help('isSingular')`. Il signale qu'une composante de variance est estimée à 0, ou qu'une corrélation entre effets aléatoires est estimée à ± 1. Ce n'est pas une erreur de calcul : la structure des effets aléatoires est trop riche pour ce que les données permettent d'estimer. Les effets fixes restent souvent fiables, mais les écarts-types des effets aléatoires et les BLUP ne le sont plus. Il vaut mieux simplifier la structure, une étape à la fois :

— retirer la corrélation entre l'ordonnée à l'origine et la pente aléatoires, avec `(1 | Subject) + (0 + Days | Subject)` (ou `(Days || Subject)`) au lieu de `(Days`

`| Subject)` ;
— retirer la pente aléatoire si sa variance est estimée à 0 ;
— vérifier le résultat avec `isSingular(modele)` et `VarCorr(modele)`.

10.2.9 Diagnostics

Les outils de validation du Chapitre 5 et des chapitres précédents s'adaptent, en distinguant deux niveaux.

— Les résidus conditionnels $y_{ij} - x_{ij}^\top \hat{\beta} - z_{ij}^\top \hat{b}_i$ estiment les ϵ_{ij}. On les trace en fonction des valeurs prédites et des variables explicatives, comme en régression linéaire, pour vérifier la forme de la moyenne et l'homoscédasticité. En R, `plot(modele)` et `residuals(modele)`.
— Les BLUP estiment les effets aléatoires. Un graphique quantile-quantile des $\hat{b}_i$ vérifie grossièrement leur normalité, et un BLUP très éloigné des autres signale un groupe atypique. En R, `qqnorm(ranef(modele)$groupe[, 1])`.

Ces graphiques demandent un peu de prudence : les BLUP sont rétrécis et n'ont pas tous la même variance, de sorte que leur distribution ne reproduit pas exactement celle des b_i. Un graphique quantile-quantile des BLUP n'est donc pas un test direct de la normalité des effets aléatoires. Des BLUP d'allure normale peuvent cacher des b_i qui ne le sont pas, en particulier avec de petits groupes, où le rétrécissement est fort. Il vaut mieux croiser plusieurs sources : les résidus conditionnels, la qualité des prédictions (par validation croisée sur des groupes entiers) et, au besoin, des données simulées à partir du modèle ajusté (`simulate(modele)`), dont on compare les graphiques à ceux des données observées. En pratique, l'estimation de β est assez robuste à la non-normalité des effets aléatoires, alors que les prédictions pour un groupe et les intervalles de prédiction le sont moins.

Pour un GLMM, les résidus d'un modèle binomial ou de Poisson sont encore plus difficiles à lire. Les résidus simulés de la librairie `DHARMa`, présentés au Chapitre 3 (remarque « Résidus simulés avec `DHARMa` »), s'appliquent directement à un modèle ajusté avec `glmer` : `DHARMa::simulateResiduals(modele)`.

10.3 Effets aléatoires et pénalisation

Le rétrécissement des BLUP rappelle celui de ridge, et ce n'est pas un hasard.

Propriété — Équations de Henderson

Pour D et σ^2 fixés, l'estimateur $\hat{\beta}$ des moindres carrés généralisés et les BLUP $\hat{b}$ sont la solution du problème pénalisé

$$\min_{\beta, b} \; \|y - X\beta - Zb\|^2 + \sigma^2 \, b^\top G^{-1} b,$$

où y, X et Z empilent les données de tous les groupes et $G = \mathrm{Var}[b]$ est diagonale par blocs, avec les blocs D. Pour une ordonnée à l'origine aléatoire, la pénalité vaut $\lambda \sum_i b_i^2$, avec $\lambda = \sigma^2 / \sigma_b^2$.

Les effets aléatoires sont donc des coefficients estimés avec une pénalité ridge, dont le paramètre λ est déterminé par le rapport des variances plutôt que par validation croisée. Le critère pénalisé est d'ailleurs, à une constante près, $-2\sigma^2$ fois la log-densité jointe de (y, b) : les BLUP sont le mode de la loi a posteriori de b, comme l'estimateur ridge est le mode a posteriori de β sous une loi a priori normale. La réciproque est vraie aussi : une fonction lisse pénalisée d'un GAM (Chapitre 9) peut s'écrire comme un effet aléatoire, dont la variance est inversement proportionnelle à λ. C'est ce qui permet à `mgcv` de choisir les paramètres de lissage par REML (`method = "REML"`), une méthode souvent plus stable que la GCV. Dans `mgcv`, le terme `s(groupe, bs = "re")` est d'ailleurs un effet aléatoire.

Exemple — Le même modèle avec `lmer` et avec `gam`

```
mod.gam.re <- gam(log(height_m) ~ log(diameter_cm) + espece + s(region,
 ↪  bs = "re"),
                  data = arbres, method = "REML")
gam.vcomp(mod.gam.re)
```

```
Standard deviations and 0.95 confidence intervals:

             std.dev      lower     upper
s(region) 0.06410488 0.03456763 0.1188810
scale     0.17594524 0.15830660 0.1955492

Rank: 2/2

Linear smooth components:
[1] 0.06410488
```

```
max(abs(coef(mod.gam.re)[1:5] - fixef(mod.lmm)))
```

```
[1] 5.350831e-11
```

Les écarts-types estimés de l'effet de région et de l'erreur sont exactement ceux de `lmer`, et les effets fixes coïncident à l'erreur d'arrondi près. On peut ainsi combiner dans un même modèle des fonctions lisses et des effets aléatoires, ou ajuster un GAM dont les erreurs sont corrélées avec la fonction `gamm`.

Exemple — Cycle journalier de l'ozone avec erreurs corrélées sur `qualite_air`

Au Chapitre 8, la bande de confiance du cycle journalier de l'ozone à Longueuil (juin à août) supposait des erreurs indépendantes, alors que les heures d'une même journée sont fortement corrélées. On ajuste ici le même cycle avec une spline cyclique, d'abord par `gam` avec des erreurs indépendantes, puis par `gamm` avec deux structures de dépendance : un effet aléatoire par jour (toutes les heures d'une journée ont la même corrélation), puis, en plus, des erreurs autorégressives d'ordre 1 à l'intérieur de chaque jour ($Cor[\epsilon_{jh}, \epsilon_{jh'}] = \phi^{|h-h'|}$).

```
ozone <- read.csv("Jeux de données/qualite_air.csv", fileEncoding =
 ↪  "UTF-8-BOM") |>
  mutate(heure = as.integer(substr(Date_Heure, 12, 13)),
         mois  = as.integer(substr(Date_Heure, 6, 7))) |>
  filter(Station == "06600 - Longueuil", mois %in% 6:8, !is.na(O3)) |>
  mutate(jour = factor(substr(Date_Heure, 1, 10)))
noeuds <- list(heure = c(0, 24))   # raccord de la spline cyclique entre
 ↪  24 h et 0 h

o3.indep <- gam(O3 ~ s(heure, bs = "cc", k = 10), data = ozone, knots =
 ↪  noeuds,
                method = "REML")
o3.jour  <- gamm(O3 ~ s(heure, bs = "cc", k = 10), random = list(jour = ~
 ↪  1),
                 data = ozone, knots = noeuds)
o3.ar1   <- gamm(O3 ~ s(heure, bs = "cc", k = 10), random = list(jour = ~
 ↪  1),
                 correlation = nlme::corAR1(form = ~ heure | jour),
                 data = ozone, knots = noeuds)
anova(o3.jour$lme, o3.ar1$lme)
```

```
           Model df      AIC      BIC    logLik   Test  L.Ratio p-value
o3.jour$lme     1  4 15428.84 15451.64 -7710.422
o3.ar1$lme      2  5 11636.54 11665.03 -5813.269 1 vs 2 3794.307  <.0001
```

```
o3.ar1$lme$modelStruct$corStruct
```

```
Correlation structure of class corARMA representing
     Phi1
0.9457333
```

Le modèle autorégressif est de loin le meilleur, et la corrélation entre deux heures consécutives est estimée à $\hat{\phi} \approx 0{,}95$. Avec une corrélation aussi forte, la structure autorégressive rend aussi compte du niveau propre à chaque journée : l'écart-type de l'effet de jour passe de 7,3 ppb dans le modèle à effet de jour seul à 0,002 ppb, c'est-à-dire pratiquement 0. La Figure 10.3 compare les bandes de confiance du cycle journalier.

```
bandes <- rbind(
  data.frame(grille.h, f = predict(o3.indep, grille.h), et = et.indep,
             modele = "Erreurs indépendantes"),
  data.frame(grille.h, f = predict(o3.ar1$gam, grille.h), et = et.ar1,
             modele = "Erreurs AR(1) par jour")
)
ggplot(bandes, aes(heure, f, fill = modele, colour = modele)) +
  geom_ribbon(aes(ymin = f - 1.96 * et, ymax = f + 1.96 * et), alpha =
   ↪  0.25, colour = NA) +
  geom_line() +
  labs(x = "Heure", y = "Ozone (ppb)", fill = NULL, colour = NULL) +
  theme_minimal(base_size = 13) + theme(legend.position = "bottom")
```

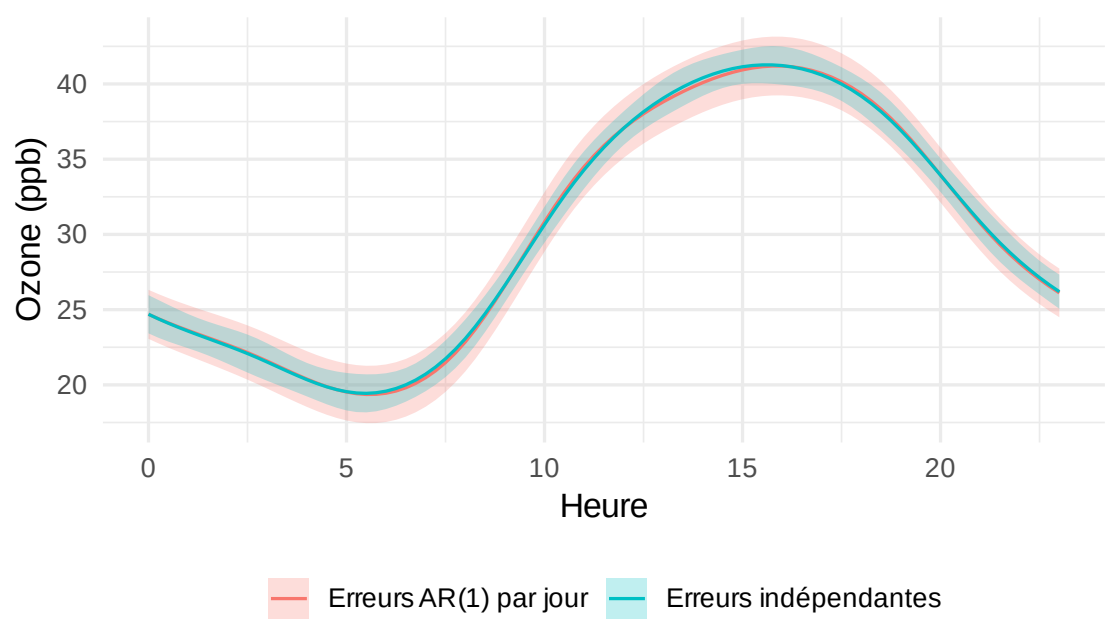


FIGURE 10.3 : Cycle journalier estimé de l'ozone à Longueuil (juin à août 2025) et bandes de confiance ponctuelles à 95 % de la moyenne conditionnelle, en supposant des erreurs indépendantes (`gam`) ou des erreurs autorégressives à l'intérieur de chaque jour (`gamm`).

Les deux courbes sont presque identiques, mais la bande qui tient compte de la dépendance est environ 1,6 fois plus large en moyenne. La bande du Chapitre 8 sous-estimait donc nettement l'incertitude sur le cycle. L'effet de jour seul élargit lui aussi la bande (d'un facteur 1,5 environ), mais il suppose la même corrélation entre 1 h et 2 h qu'entre 1 h et 23 h, ce que les données rejettent.

10.4 Modèle linéaire généralisé mixte

Définition — GLMM

Le modèle linéaire généralisé mixte (GLMM) étend le GLM comme le modèle mixte étend la régression linéaire. Conditionnellement à b_i, les Y_{ij} sont indépendantes, de loi dans la famille exponentielle, avec

$$g\left(E[Y_{ij} \mid b_i]\right) = x_{ij}^\top \beta + z_{ij}^\top b_i, \qquad b_i \sim \mathcal{N}(0, D).$$

La vraisemblance du groupe i s'obtient en intégrant l'effet aléatoire :

$$L_i(\beta, D) = \int \prod_{j=1}^{n_i} f(y_{ij} \mid b)\, \phi(b; 0, D)\, db,$$

où $\phi(\cdot; 0, D)$ est la densité normale. Contrairement au cas linéaire, cette intégrale n'a pas de forme fermée, et il faut l'approcher.

Propriété — Approximation de Laplace

Écrivons $L_i = \int e^{h_i(b)}\, db$, avec $h_i(b) = \sum_j \log f(y_{ij} \mid b) + \log \phi(b; 0, D)$, et notons $\tilde{b}_i$ le maximum de h_i. Un développement de Taylor d'ordre 2 de h_i autour de $\tilde{b}_i$, où le gradient est nul, donne

$$L_i \approx \int \exp\left\{ h_i(\tilde{b}_i) - \frac{1}{2}(b - \tilde{b}_i)^\top H_i (b - \tilde{b}_i) \right\} db = e^{h_i(\tilde{b}_i)}\, (2\pi)^{q/2}\, |H_i|^{-1/2},$$

où $H_i = -\partial^2 h_i / \partial b\, \partial b^\top$ est évaluée en $\tilde{b}_i$. L'intégrande est remplacée par une densité normale centrée en son mode.

L'approximation est exacte quand h_i est quadratique, ce qui est le cas du modèle linéaire mixte (voir les exercices). Elle est d'autant meilleure que h_i ressemble à une parabole, c'est-à-dire que les groupes sont grands et que la loi conditionnelle est proche de la normale (comptes élevés, proportions binomiales avec beaucoup d'essais). Elle est la moins bonne pour des réponses binaires avec peu d'observations par groupe et une grande variance des effets aléatoires, où elle peut biaiser nettement l'estimation de σ_b et des coefficients. La quadrature de Gauss-Hermite adaptative généralise l'approximation de Laplace (qui correspond à un seul point de quadrature) en évaluant l'intégrande en plusieurs points autour du mode. Elle est plus précise, mais son coût croît exponentiellement avec la dimension q des effets aléatoires. Dans `glmer`, l'approximation de Laplace est utilisée par défaut, et la quadrature (argument `nAGQ`) n'est offerte que pour un seul effet aléatoire scalaire.

Remarque — Interprétation conditionnelle

Les coefficients β d'un GLMM s'interprètent conditionnellement à l'effet aléatoire, c'est-à-dire pour un groupe donné (interprétation dite spécifique au sujet). La moyenne marginale, obtenue en intégrant sur b_i,

$$E[Y_{ij}] = \int g^{-1}(x_{ij}^\top \beta + z_{ij}^\top b)\, \phi(b; 0, D)\, db,$$

n'a en général pas la même forme. Pour une ordonnée à l'origine aléatoire $b_i \sim \mathcal{N}(0, \sigma_b^2)$ avec le lien probit, les coefficients marginaux valent exactement $\beta / \sqrt{1 + \sigma_b^2}$ (voir les exercices). Avec le lien logit, ils valent approximativement $\beta / \sqrt{1 + c^2 \sigma_b^2}$, où $c = 16\sqrt{3}/(15\pi) \approx$

0,588. Avec le lien log, $E[Y_{ij}] = \exp(x_{ij}^\top \beta + \sigma_b^2/2)$: seule l'ordonnée à l'origine change. Pour les liens logit et probit, l'effet d'une variable sur la moyenne de la population est donc atténué par rapport à son effet dans un groupe donné.

Remarque — Surdispersion et effet aléatoire par observation

Un GLMM de Poisson ou binomial hérite de la relation entre la moyenne et la variance de sa loi conditionnelle. Si les données sont surdispersées même à l'intérieur des groupes, on peut ajouter un effet aléatoire propre à chaque observation, `(1 | obs)`, où `obs` numérote les lignes du jeu de données. Pour un modèle de Poisson, cet effet transforme la loi conditionnelle en un mélange de Poisson et de log-normale, proche de la loi binomiale négative du Chapitre 3.

Exemple — Péripneumonie contagieuse bovine

Le jeu de données `cbpp` de la librairie `lme4` provient d'une étude de la péripneumonie contagieuse bovine dans 15 troupeaux de zébus en Éthiopie. Chaque troupeau a été suivi pendant quatre périodes (`period`). Pour chaque troupeau et chaque période, on connaît le nombre de bêtes (`size`) et le nombre de nouveaux cas (`incidence`). Les quatre observations d'un même troupeau ne sont pas indépendantes : certains troupeaux sont plus touchés que d'autres pendant toute l'étude.

```
data(cbpp, package = "lme4")
mod.glm  <- glm(cbind(incidence, size - incidence) ~ period, family =
 ↪  binomial, data = cbpp)
mod.glmm <- glmer(cbind(incidence, size - incidence) ~ period + (1 |
 ↪  herd),
                  family = binomial, data = cbpp)
mod.glmm.q <- update(mod.glmm, nAGQ = 20)   # quadrature adaptative à 20
 ↪  points
summary(mod.glmm)$varcor
```

```
 Groups Name        Std.Dev.
 herd   (Intercept) 0.64227
```

```
round(rbind(Laplace = fixef(mod.glmm), Quadrature = fixef(mod.glmm.q)),
 ↪  3)
```

```
           (Intercept) period2 period3 period4
Laplace         -1.399  -0.992  -1.129  -1.580
Quadrature      -1.399  -0.991  -1.128  -1.579
```

L'écart-type estimé de l'effet aléatoire des troupeaux est de 0,64 sur l'échelle logit, et l'approximation de Laplace donne pratiquement les mêmes estimations que la quadrature. C'est attendu, puisque chaque observation est une proportion binomiale sur plusieurs bêtes. L'incidence diminue nettement après la première période. Pour un troupeau donné, la cote d'un nouveau cas à la deuxième période est environ $e^{-0,99} \approx 0,37$ fois celle de la première.

Exemple — Respiration sifflante chez les enfants de Steubenville

Le jeu de données `ohio` de la librairie `geepack` provient d'une étude sur les effets de la pollution de l'air sur la santé respiratoire. Pour 537 enfants de Steubenville, en Ohio, on a noté chaque année, de 7 à 10 ans, la présence d'une respiration sifflante (`resp`, 0 ou 1). La variable `age` vaut l'âge moins 9 ans (de -2 à 1), et `smoke` indique si la mère fumait au début de l'étude. Chaque enfant compte 4 observations binaires, la situation la plus défavorable pour l'approximation de Laplace.

```
data(ohio, package = "geepack")
ohio.laplace <- glmer(resp ~ age + smoke + (1 | id), family = binomial,
 ↪  data = ohio)
ohio.glmm    <- update(ohio.laplace, nAGQ = 25)
sd.b <- function(m) attr(VarCorr(m)$id, "stddev")[[1]]
round(rbind(Laplace    = c(fixef(ohio.laplace), sigma.b =
 ↪  sd.b(ohio.laplace)),
            Quadrature = c(fixef(ohio.glmm),    sigma.b =
             ↪  sd.b(ohio.glmm))), 3)
```

```
           (Intercept)    age smoke sigma.b
Laplace         -3.374 -0.177 0.415   2.343
Quadrature      -3.102 -0.176 0.399   2.165
```

Les deux méthodes donnent ici des estimations différentes de l'ordonnée à l'origine et de σ_b, que l'approximation de Laplace surestime d'environ 8 %. Les pentes, elles, sont peu touchées. On retient les estimations par quadrature. Comme prévu, l'écart-type des effets aléatoires est grand : 2,16 sur l'échelle logit selon la quadrature. Les enfants diffèrent beaucoup par leur propension à la respiration sifflante, et celle-ci persiste d'une année à l'autre. Pour un enfant donné, la cote d'une respiration sifflante est multipliée par $e^{-0,176} \approx 0,84$ chaque année, et elle est multipliée par environ 1,49 si la mère fume, un effet qui n'est pas significatif.
La Figure 10.4 illustre l'écart entre l'interprétation conditionnelle et l'interprétation marginale. Les courbes grises sont les probabilités conditionnelles $\text{logit}^{-1}(x^\top \hat{\beta} + b)$ pour des enfants dont l'effet aléatoire b est à différents quantiles de sa loi. La courbe pointillée

est celle de l'enfant « typique » ($b = 0$), et la courbe pleine est la probabilité marginale, c'est-à-dire la moyenne des courbes conditionnelles sur la loi de b, calculée par intégration numérique.

```
ages <- seq(-2, 1, length.out = 61)
eta <- b.ohio[["(Intercept)"]] + b.ohio[["age"]] * ages
marginale <- sapply(eta, function(e)
  integrate(function(u) plogis(e + u) * dnorm(u, 0, sb.ohio), -Inf,
   ↪  Inf)$value)
quantiles <- qnorm(c(0.10, 0.25, 0.75, 0.90), 0, sb.ohio)
courbes <- data.frame(age = rep(ages, length(quantiles)) + 9,
                      p = as.vector(plogis(outer(eta, quantiles, "+"))),
                      q = rep(seq_along(quantiles), each = length(ages)))
observe <- ohio |>
  filter(smoke == 0) |>
  group_by(age) |>
  summarise(resp = mean(resp))
ggplot() +
  geom_line(data = courbes, aes(age, p, group = q), colour = "grey70") +
  geom_line(aes(ages + 9, plogis(eta)), linetype = "dashed") +
  geom_line(aes(ages + 9, marginale)) +
  geom_point(data = observe, aes(age + 9, resp)) +
  scale_y_continuous(limits = c(0, 1)) +
  labs(x = "Âge (ans)", y = "Probabilité de respiration sifflante") +
  theme_minimal(base_size = 13)
```

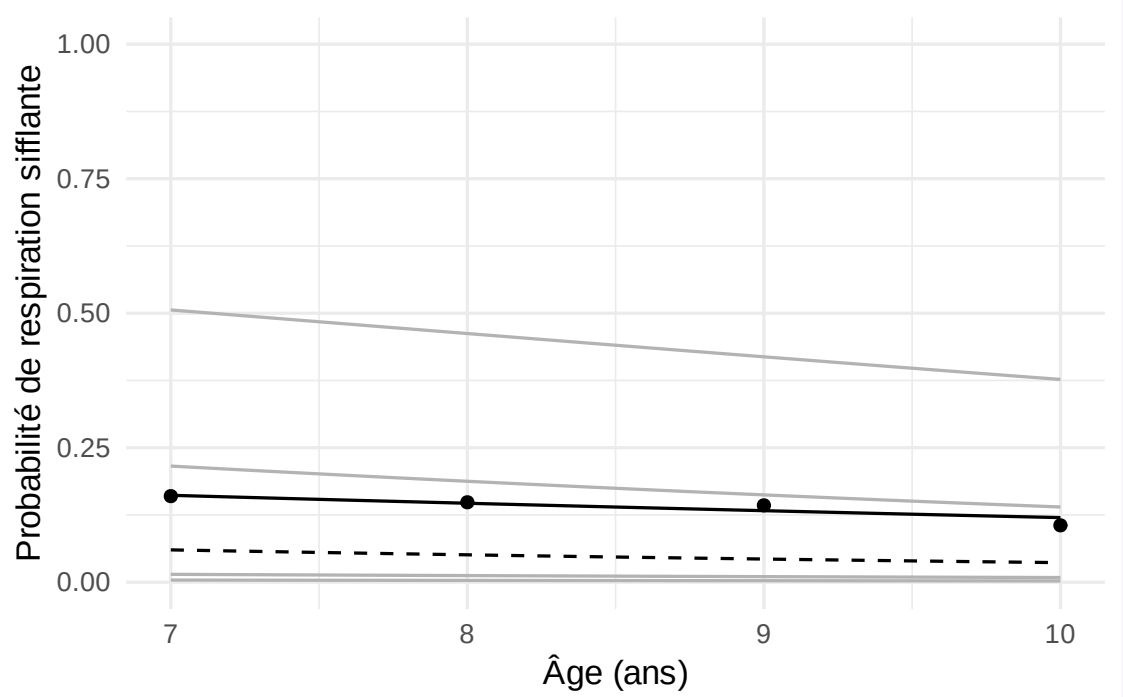


FIGURE 10.4 : Probabilité de respiration sifflante selon l'âge (mère non fumeuse) dans le GLMM de `ohio`. Courbes grises : enfants aux quantiles 10 %, 25 %, 75 % et 90 % de l'effet aléatoire. Pointillés : enfant typique ($b = 0$). Trait plein : probabilité marginale obtenue en intégrant sur b. Points : proportions observées.

L'enfant typique a une probabilité de respiration sifflante d'environ 4 % à 9 ans, alors que la probabilité marginale est d'environ 13 %, proche de la proportion observée. Ce n'est pas une contradiction. La plupart des enfants ont une faible probabilité de respiration sifflante, mais une minorité a une probabilité élevée, et la moyenne de la population est tirée vers le haut par ce petit groupe. La courbe marginale est aussi plus plate que les courbes conditionnelles : sur l'échelle logit, sa pente moyenne entre 7 et 10 ans est environ 0,65 fois celle d'un enfant donné, proche du facteur d'atténuation approximatif $1/\sqrt{1 + c^2\sigma_b^2} \approx 0{,}62$.

10.5 Équations d'estimation généralisées (GEE)

L'approche GEE ne spécifie pas la loi jointe des observations d'un groupe. Elle modélise seulement la moyenne marginale et la variance, comme en quasi-vraisemblance (Section 3.6.5),

et remplace la structure de corrélation inconnue par une structure de travail.

Définition — GEE

On suppose

$$g(\mu_{ij}) = x_{ij}^\top \beta, \qquad Var[Y_{ij}] = \phi\, V(\mu_{ij}),$$

et on postule une matrice de corrélation de travail $R_i(\alpha)$ pour les observations du groupe i : indépendance ($R_i = I$), échangeable (même corrélation α pour toutes les paires), autorégressive d'ordre 1 ($\alpha^{|j-k|}$) ou non structurée (une corrélation par paire de temps). L'estimateur $\hat{\beta}$ résout

$$U(\beta) = \sum_{i=1}^{m} \Delta_i^\top V_i^{-1}(y_i - \mu_i) = 0, \qquad \Delta_i = \frac{\partial \mu_i}{\partial \beta^\top}, \qquad V_i = \phi\, A_i^{1/2} R_i(\alpha) A_i^{1/2},$$

où $A_i = \mathrm{diag}\{V(\mu_{ij})\}$. Avec $R_i = I$, on retrouve exactement le score d'un GLM.

Les paramètres de travail ϕ et α s'estiment par la méthode des moments, à partir des résidus de Pearson $r_{ij} = (y_{ij} - \hat{\mu}_{ij})/\sqrt{V(\hat{\mu}_{ij})}$. Pour la structure échangeable, par exemple,

$$\hat{\phi} = \frac{1}{N - (p+1)} \sum_{i=1}^{m} \sum_{j=1}^{n_i} r_{ij}^2, \qquad \hat{\alpha} = \frac{1}{\hat{\phi}} \cdot \frac{\sum_{i=1}^{m} \sum_{j<k} r_{ij} r_{ik}}{\sum_{i=1}^{m} n_i(n_i-1)/2 - (p+1)}.$$

Ici, p compte seulement les variables explicatives, de sorte que $\hat{\beta}$ contient $p+1$ paramètres avec l'ordonnée à l'origine.

Pour aller plus loin avec la théorie — D'où vient l'estimateur de α sous la structure échangeable

L'estimateur $\hat{\alpha}$ est un estimateur des moments : on calcule l'espérance d'un produit de résidus, puis on la remplace par une moyenne empirique.
Espérance d'un produit de résidus. Travaillons d'abord avec les résidus calculés aux vraies moyennes,

$$e_{ij} = \frac{Y_{ij} - \mu_{ij}}{\sqrt{V(\mu_{ij})}}.$$

Le modèle suppose $E[Y_{ij}] = \mu_{ij}$ et $Var[Y_{ij}] = \phi\, V(\mu_{ij})$, de sorte que

$$E[e_{ij}] = 0, \qquad Var[e_{ij}] = \frac{\phi\, V(\mu_{ij})}{V(\mu_{ij})} = \phi.$$

Comme μ_{ij} et μ_{ik} sont des constantes, chaque résidu est de la forme $aY + c$, avec $a = 1/\sqrt{V(\mu_{ij})}$ et $c = -\mu_{ij}/\sqrt{V(\mu_{ij})}$. Par la bilinéarité de la covariance, $Cov[aX +$

$c,\, bY + d] = ab\, Cov[X, Y]$, les translations disparaissent et

$$Cov[e_{ij}, e_{ik}] = \frac{Cov[Y_{ij}, Y_{ik}]}{\sqrt{V(\mu_{ij})}\sqrt{V(\mu_{ik})}}.$$

Sous la structure échangeable, $Corr[Y_{ij}, Y_{ik}] = \alpha$ pour tout $j \neq k$, d'où $Cov[Y_{ij}, Y_{ik}] = \alpha\,\phi\sqrt{V(\mu_{ij})V(\mu_{ik})}$. Les racines se simplifient, et comme les e_{ij} sont centrés,

$$E[e_{ij}e_{ik}] = Cov[e_{ij}, e_{ik}] = \phi\,\alpha.$$

Cette égalité est exacte : la standardisation de Pearson ramène toutes les observations à la même variance ϕ, et leur covariance vaut alors ϕ fois leur corrélation.
Passage aux résidus observés. En pratique, μ_{ij} est inconnu, et on utilise r_{ij}, calculé avec $\hat{\mu}_{ij} = g^{-1}(x_{ij}^{\top}\hat{\beta})$. Les constantes de standardisation deviennent alors aléatoires, et l'égalité n'est plus qu'approximative. Comme $\hat{\beta}$ est convergent, $r_{ij} \approx e_{ij}$ quand m est grand, et $E[r_{ij}r_{ik}] \approx \phi\,\alpha$.
Le dénominateur. On estime $\phi\,\alpha$ par la moyenne des produits $r_{ij}r_{ik}$ sur toutes les paires disponibles, puis on divise par $\hat{\phi}$. Un groupe de taille n_i contient $\binom{n_i}{2} = n_i(n_i-1)/2$ paires distinctes $j < k$, de sorte que le nombre total de paires est

$$N^* = \sum_{i=1}^{m} \frac{n_i(n_i-1)}{2}.$$

Diviser par N^* donnerait la moyenne empirique simple des produits. On soustrait $p+1$ par analogie avec la correction des degrés de liberté de $\hat{\phi}$, ou de $\hat{\sigma}^2 = SSE/\{n-(p+1)\}$ en régression linéaire : les résidus sont calculés avec $\hat{\beta}$, ajusté aux données, et sont donc en moyenne un peu plus petits que les vraies erreurs.
Cette dernière correction est heuristique. Pour les produits croisés, on ne perd pas exactement $p+1$ degrés de liberté comme dans le cas linéaire, et la correction n'a aucun effet asymptotique, puisque N^* croît avec m. Les conventions varient d'ailleurs d'une source et d'un logiciel à l'autre : Liang et Zeger (1986) écrivent $N^* - p$ parce que leur p inclut l'ordonnée à l'origine, ce qui revient au même, et certaines implémentations n'appliquent aucune correction.

L'algorithme alterne entre la résolution de $U(\beta) = 0$ par un pas de Fisher scoring, à ϕ et α fixés, et la mise à jour de ϕ et de α à partir des nouveaux résidus, jusqu'à convergence.

Propriété — Convergence et estimateur sandwich

Si le modèle de la moyenne est correct, $E[y_{_i} - \mu_{_i}] = 0$, de sorte que $E[U(\beta)] = 0$ au vrai β, quelle que soit la matrice $V_{_i}$ utilisée. Les équations sont sans biais, et $\hat{\beta}$ est convergent

même si $R_i(\alpha)$ est mal spécifiée. Un développement de Taylor de U autour du vrai β donne $\hat{\beta} - \beta \approx B^{-1}U(\beta)$, avec $B = \sum_i \Delta_i^\top V_i^{-1}\Delta_i$, et donc

$$Var[\hat{\beta}] \approx B^{-1}\left\{\sum_{i=1}^{m}\Delta_i^\top V_i^{-1}\,Var[y_i]\,V_i^{-1}\Delta_i\right\}B^{-1}.$$

En remplaçant la vraie variance $Var[y_i]$, inconnue, par le produit des résidus observés, on obtient l'estimateur sandwich

$$\hat{\mathrm{V}}(\hat{\beta}) = B^{-1}\left\{\sum_{i=1}^{m}\Delta_i^\top V_i^{-1}(y_i-\hat{\mu}_i)(y_i-\hat{\mu}_i)^\top V_i^{-1}\Delta_i\right\}B^{-1},$$

qui reste valide même si la corrélation de travail est fausse. Si V_i est la vraie variance de y_i (corrélation, fonction de variance et ϕ), le terme central estime B, et la variance se réduit à B^{-1}, l'estimateur « modèle ».

Le terme central de l'estimateur sandwich remplace la variance modélisée par le produit des résidus observés de chaque groupe. Chaque produit $(y_i-\hat{\mu}_i)(y_i-\hat{\mu}_i)^\top$ est un très mauvais estimateur de $Var[y_i]$ pris seul, mais leur somme pondérée estime bien la variance du score, parce que les erreurs se compensent d'un groupe à l'autre. L'estimateur ne repose donc que sur m groupes indépendants.

Remarque — L'unité indépendante est le groupe

Pour un GEE, l'argument asymptotique porte sur le nombre m de groupes indépendants, pas sur le nombre total N d'observations. Mille mesures réparties dans dix groupes restent dix unités indépendantes. Avec peu de groupes (une quarantaine ou moins), l'estimateur sandwich est instable et tend à sous-estimer la variance, de sorte que les erreurs-types sont trop petites et que les tests de Wald rejettent trop souvent. Dans ce cas, il vaut mieux :

- utiliser une correction pour petits échantillons (Mancl et DeRouen, Kauermann et Carroll), ou l'approximation par jackknife de `geeglm` (argument `std.err = "jack"`), qui retire un groupe entier à la fois ;
- ou recourir à un bootstrap adapté au plan, qui rééchantillonne des groupes entiers plutôt que des observations ;
- et, dans tous les cas, rapporter m en plus de N, puisque c'est m qui détermine la fiabilité de l'inférence.

10.5.1 Choisir la structure de travail

Puisque $\hat{\beta}$ est convergent et que l'estimateur sandwich est valide pour toute structure de travail, le choix de R_i n'influence en général que l'efficacité (avec une exception, dans la première règle

ci-dessous). Une structure proche de la vraie donne des erreurs-types plus petites, surtout pour les variables qui varient à l'intérieur des groupes. Quelques règles pratiques :

— l'indépendance est un choix sûr, parfois peu efficace, et c'est le seul qui reste valide quand des variables explicatives qui changent dans le temps sont liées aux réponses passées ;
— la structure échangeable convient à des groupes sans ordre (des élèves dans des classes) ;
— la structure autorégressive convient à des mesures répétées régulièrement espacées dans le temps ;
— la structure non structurée demande beaucoup de groupes, puisqu'elle estime $n(n-1)/2$ corrélations.

Comme le GEE n'a pas de vraisemblance, l'AIC n'est pas disponible. Pan (2001) a proposé un critère analogue, le QIC, qui remplace la log-vraisemblance par la quasi-vraisemblance calculée sous l'indépendance, et la pénalité $2(p+1)$ par un terme qui compare la variance sandwich à la variance modèle. Le QIC sert surtout à choisir les variables de la moyenne. Il distingue mal les structures de corrélation entre elles, et on ne doit pas s'y fier seul pour ce choix. La fonction `QIC` de `geepack` le calcule.

Exemple — Péripneumonie contagieuse bovine, approche GEE

La fonction `geeglm` de la librairie `geepack` prend la réponse sous forme de proportion, avec les tailles des troupeaux comme poids, et l'argument `id` indique les groupes, qui doivent être contigus dans le jeu de données.

```
cbpp <- mutate(cbpp, prop = incidence / size)
mod.gee <- geeglm(prop ~ period, id = herd, weights = size, family =
 ↪  binomial,
                  corstr = "exchangeable", data = cbpp)
comparaison <- cbind(
  GLM = coef(mod.glm), ET.GLM = sqrt(diag(vcov(mod.glm))),
  GEE = coef(mod.gee), ET.GEE = summary(mod.gee)$coefficients[,
   ↪  "Std.err"],
  GLMM = fixef(mod.glmm), ET.GLMM = sqrt(diag(vcov(mod.glmm)))
)
round(comparaison, 3)
```

```
               GLM ET.GLM    GEE ET.GEE   GLMM ET.GLMM
(Intercept) -1.269  0.145 -1.270  0.276 -1.399   0.232
period2     -1.171  0.291 -1.156  0.436 -0.992   0.307
period3     -1.301  0.313 -1.284  0.491 -1.129   0.327
period4     -1.782  0.413 -1.754  0.375 -1.580   0.427
```

Les estimations du GEE sont presque identiques à celles du GLM : les deux décrivent la même moyenne marginale, et la corrélation de travail estimée est faible. Les erreurs-types du GEE, elles, sont nettement plus grandes pour l'ordonnée à l'origine et les périodes 2 et 3 : le GLM ignore que les quatre périodes d'un même troupeau sont liées. Les coefficients du GLMM ont une interprétation conditionnelle et diffèrent un peu de ceux du GEE. Avec $\hat{\sigma}_b \approx 0{,}64$, le facteur d'atténuation de la remarque précédente vaut environ 0,94, soit un écart théorique d'à peine 6 %. Les effets de période du GLMM sont même un peu plus petits en valeur absolue que ceux du GEE, à l'inverse de ce que prévoit l'atténuation : avec si peu de troupeaux, le fait que les deux méthodes ne pondèrent pas les troupeaux de la même façon l'emporte sur cet écart théorique. Enfin, les 56 observations ne viennent que de $m = 15$ troupeaux. Avec si peu de groupes indépendants, l'estimateur sandwich est imprécis, ce qui peut expliquer que l'erreur-type GEE de la quatrième période soit plus petite que celle du GLM. Les erreurs-types du GEE de cet exemple sont donc à prendre avec prudence.

Exemple — Respiration sifflante, approche GEE

Le jeu de données `ohio` compte $N = 2148$ observations, mais ce sont les $m = 537$ enfants qui sont les unités indépendantes. Ils sont assez nombreux pour que l'estimateur sandwich soit fiable. On compare les quatre structures de travail.

```
structures <- c("independence", "exchangeable", "ar1", "unstructured")
ohio.gee <- lapply(structures, function(s)
  geeglm(resp ~ age + smoke, id = id, family = binomial, corstr = s, data
   ↪  = ohio))
names(ohio.gee) <- structures
t(sapply(ohio.gee, function(m) {
  et <- summary(m)$coefficients[, "Std.err"]
  names(et) <- paste0("ET.", names(coef(m)))
  round(c(coef(m), et, QIC = QIC(m)[["QIC"]]), 3)
}))
```

```
             (Intercept)    age smoke ET.(Intercept) ET.age ET.smoke      QIC
independence      -1.884 -0.113 0.272          0.114  0.044    0.178 1829.493
exchangeable      -1.880 -0.113 0.265          0.114  0.044    0.178 1829.483
ar1               -1.902 -0.115 0.234          0.115  0.045    0.181 1830.255
unstructured      -1.889 -0.115 0.253          0.114  0.044    0.178 1829.597
```

```
summary(ohio.gee$exchangeable)$corr
```

```
       Estimate    Std.err
alpha 0.3543049 0.06244265
```

Les quatre structures donnent pratiquement les mêmes coefficients et les mêmes erreurs-types, et leurs QIC sont presque égaux. La corrélation échangeable estimée entre deux années d'un même enfant vaut 0,35 sur l'échelle des résidus de Pearson. Cette robustesse au choix de la structure est typique quand les groupes sont nombreux et que les variables explicatives sont équilibrées entre les groupes, comme ici, où chaque enfant est observé aux quatre âges.
La comparaison avec le GLMM illustre l'atténuation de façon nette :

```
round(rbind(GEE = gee.ex, GLMM = b.ohio,
            GLMM.attenue = b.ohio / sqrt(1 + 0.588^2 * sb.ohio^2)), 3)
```

```
              (Intercept)    age smoke
GEE                -1.880 -0.113 0.265
GLMM               -3.102 -0.176 0.399
GLMM.attenue       -1.916 -0.108 0.246
```

Les coefficients du GLMM sont environ 1,6 fois plus grands que ceux du GEE. Une fois multipliés par le facteur d'atténuation, ils sont très proches des coefficients du GEE : les deux modèles décrivent la même moyenne marginale, l'un directement et l'autre en intégrant sur les effets aléatoires (Figure 10.4). Pour la question « de combien la fréquence de la respiration sifflante dans la population diminue-t-elle avec l'âge ? », le coefficient pertinent est celui du GEE. Pour la question « de combien le risque d'un enfant donné diminue-t-il avec l'âge ? », c'est celui du GLMM.

Remarque — Données manquantes

Dans les études longitudinales, des observations manquent souvent : un enfant absent une année, un patient qui abandonne l'étude. Les deux approches ne les traitent pas de la même façon. Le GEE n'est valide que si les données manquent complètement au hasard (*MCAR*), c'est-à-dire si le fait qu'une observation manque ne dépend pas des réponses, observées ou non. Il peut dépendre des variables explicatives du modèle. Les modèles mixtes, ajustés par maximum de vraisemblance, restent valides sous l'hypothèse plus faible que les données manquent au hasard (*MAR*) : le fait qu'une observation manque peut dépendre des réponses observées du même groupe, par exemple un patient qui abandonne parce que ses mesures précédentes se détérioraient. Cette garantie suppose toutefois que le modèle complet, structure des effets aléatoires comprise, est bien spécifié. Des versions pondérées du GEE (par l'inverse de la probabilité d'être observé) corrigent ce défaut, au prix d'un modèle supplémentaire pour cette probabilité.

Exemple — Abandon qui dépend des réponses passées

On simule une étude longitudinale de $m = 500$ sujets mesurés aux temps $t = 0, 1, 2, 3$, selon un modèle à ordonnée à l'origine aléatoire, $Y_{it} = 10 + t + b_i + \epsilon_{it}$, avec $b_i \sim \mathcal{N}(0, 2^2)$ et $\epsilon_{it} \sim \mathcal{N}(0, 1)$. La vraie pente vaut donc 1. Après chaque mesure, un sujet abandonne l'étude avec une probabilité qui augmente avec sa dernière réponse observée : les données manquent au hasard (MAR), mais pas complètement au hasard.

```
set.seed(4300)
m_sim <- 500
complet <- expand.grid(temps = 0:3, id = 1:m_sim) |>
  as_tibble() |>
  group_by(id) |>
  mutate(b = rnorm(1, 0, 2)) |>
  ungroup() |>
  mutate(y = 10 + temps + b + rnorm(n(), 0, 1))

# Abandon après une réponse élevée ; une fois parti, le sujet ne revient
 ↪  pas
observe <- complet |>
  group_by(id) |>
  mutate(p_abandon = plogis(-1 + 1.5 * (lag(y) - 12)),
                abandon = if_else(temps == 0, 0L,
                                  rbinom(n(), 1,
                                   ↪  coalesce(p_abandon, 0))),
                present = cumprod(1 - abandon)) |>
  ungroup() |>
  filter(present == 1) |>
  arrange(id, temps)

table(observe$temps)   # sujets encore présents à chaque temps
```

```
  0   1   2   3
500 422 348 274
```

```
lmm_mar  <- lmer(y ~ temps + (1 | id), data = observe, REML = FALSE)
gee_ind  <- geeglm(y ~ temps, id = id, data = observe, corstr =
 ↪  "independence")
gee_ech  <- geeglm(y ~ temps, id = id, data = observe, corstr =
 ↪  "exchangeable")
```

```r
pentes_mar <- c(vraie = 1, GEE.independance = coef(gee_ind)[["temps"]],
                GEE.echangeable = coef(gee_ech)[["temps"]], LMM =
                 ↪  fixef(lmm_mar)[["temps"]])
round(pentes_mar, 3)
```

```
           vraie GEE.independance  GEE.echangeable             LMM
           1.000            0.565            0.963           0.992
```

Les sujets qui restent jusqu'au bout sont ceux dont les réponses étaient plus basses, de sorte que la moyenne observée aux derniers temps sous-estime la vraie moyenne. Le GEE avec la structure d'indépendance suit ces moyennes observées et sous-estime nettement la pente (environ 0,57 au lieu de 1). Le modèle mixte, ajusté par maximum de vraisemblance, reste près de la vraie valeur (0,99), parce que la vraisemblance tient compte des réponses observées avant l'abandon. Le GEE avec la structure échangeable s'en approche aussi (0,96) : dans ce modèle linéaire normal, la corrélation de travail échangeable est la vraie structure, et le GEE se rapproche alors de l'estimateur du maximum de vraisemblance. En général, avec une corrélation de travail mal spécifiée ou un GLM non linéaire, le GEE reste biaisé sous MAR.

```r
moyennes_obs <- observe |>
  group_by(temps) |>
  summarise(y = mean(y))
droites <- data.frame(
  modele = c("GEE (indépendance)", "Modèle mixte"),
  ordonnee = c(coef(gee_ind)[["(Intercept)"]],
   ↪  fixef(lmm_mar)[["(Intercept)"]]),
  pente = c(coef(gee_ind)[["temps"]], fixef(lmm_mar)[["temps"]])
)
ggplot(moyennes_obs, aes(temps, y)) +
  geom_abline(intercept = 10, slope = 1, linetype = "dashed") +
  geom_abline(data = droites, aes(intercept = ordonnee, slope = pente,
   ↪  colour = modele)) +
  geom_point(size = 2.5) +
  labs(x = "Temps", y = "Réponse moyenne", colour = NULL) +
  theme_minimal(base_size = 13) + theme(legend.position = "bottom")
```

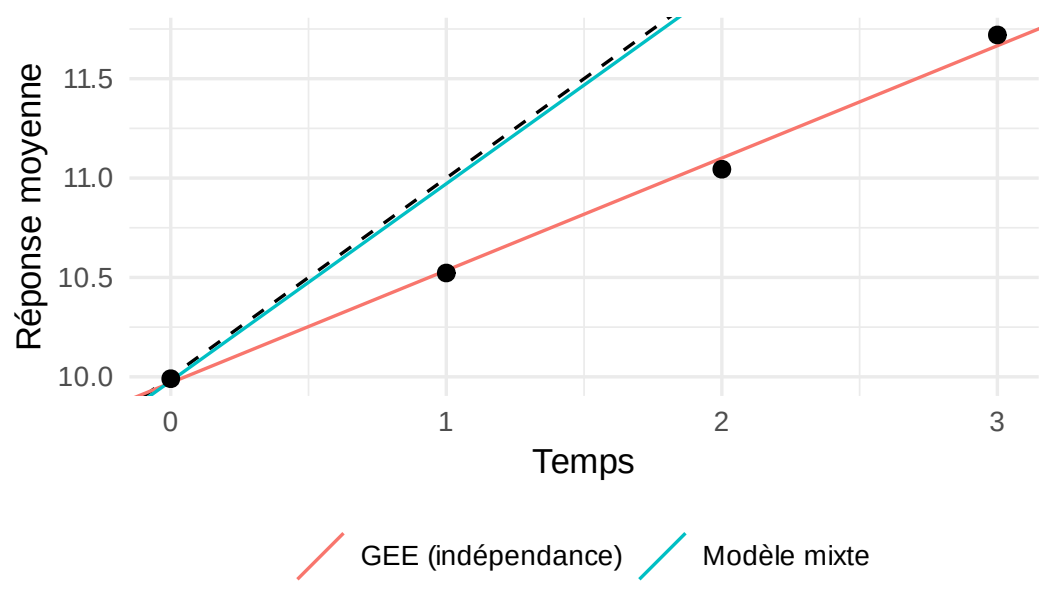


FIGURE 10.5 : Moyennes observées à chaque temps (points) et droites estimées par le GEE avec structure d'indépendance et par le modèle mixte. Le trait pointillé est la vraie moyenne, $10 + t$.

Remarque — GEE ou GLMM ?

Les coefficients d'un GEE ont une interprétation marginale (l'effet moyen dans la population), alors que ceux d'un GLMM ont une interprétation conditionnelle (l'effet pour un groupe donné). Le GEE convient quand on s'intéresse à l'effet moyen d'une intervention sur une population et qu'on dispose de nombreux groupes. Il ne demande aucune hypothèse sur la loi des effets aléatoires et protège l'inférence contre une mauvaise spécification de la dépendance. Le GLMM convient quand on veut décrire l'hétérogénéité entre groupes, prédire pour un groupe précis, quand les groupes sont peu nombreux ou quand des données manquent au hasard, puisqu'il s'appuie sur un modèle complet. Comme le GEE ne repose pas sur une vraisemblance, l'AIC et les tests du rapport de vraisemblance n'y sont pas disponibles ; on utilise des tests de Wald fondés sur l'estimateur sandwich. Pour un modèle linéaire normal avec ordonnée à l'origine aléatoire, les deux interprétations coïncident.

Remarque — Résumé des approches

	Modèle linéaire mixte	GLMM	GEE
Réponse	normale	famille exponentielle	moyenne et variance seulement
Dépendance	effets aléatoires (et erreurs corrélées)	effets aléatoires	corrélation de travail
Interprétation de β	conditionnelle = marginale	conditionnelle	marginale
Estimation	ML ou REML	ML (Laplace, quadrature)	équations d'estimation
Erreurs-types	modèle (avec df approchés)	modèle	sandwich
Comparaison de modèles	AIC, rapport de vraisemblance	AIC, rapport de vraisemblance	Wald, QIC
Prédiction pour un groupe	oui (BLUP)	oui	non
Données manquantes	valide sous MAR	valide sous MAR	valide sous MCAR
En R	`lmer`, `lme`, `gamm`	`glmer`, `gamm`	`geeglm`

Remarque — À retenir

- Dépendance : ignorer la corrélation ne biaise généralement pas les coefficients de la moyenne, mais fausse leurs erreurs-types, dans un sens ou dans l'autre. Avec m groupes de taille c et une corrélation ρ, l'effet du plan vaut $1+(c-1)\rho$ et la taille d'échantillon effective $N/\{1+(c-1)\rho\}$;
- Modèle mixte : une ordonnée à l'origine aléatoire $b_i \sim \mathcal{N}(0, \sigma_b^2)$ crée la corrélation intraclasse $\rho = \sigma_b^2/(\sigma_b^2+\sigma^2)$, et les BLUP $\hat{b}_i$ sont rétrécis vers 0, d'autant plus que le groupe est petit (le même rétrécissement que ridge). En R, `lmer` de `lme4` avec `(1 | groupe)`, ou `(t | groupe)` pour une pente aléatoire ;
- ML ou REML : `lmer` utilise REML par défaut, mais deux modèles qui n'ont pas les mêmes effets fixes se comparent par ML (`anova` réajuste automatiquement). Pour $H_0 : \sigma_b^2 = 0$, la valeur testée est à la frontière, et la valeur-p est la moitié de celle d'une χ^2_1 ;
- GLMM ou GEE : les coefficients d'un GLMM (`glmer`) s'interprètent pour un groupe donné, ceux d'un GEE (`geeglm` de `geepack`) pour la moyenne de la population, et avec les liens logit et probit, l'effet marginal est atténué. L'approximation de Laplace de `glmer` est la moins fiable pour des réponses binaires avec peu d'observations par groupe et une grande variance des effets aléatoires (argument `nAGQ` pour la quadrature) ;

— GEE : si la moyenne est bien spécifiée, $\widehat{\beta}$ est convergent et l'estimateur sandwich reste valide même si la corrélation de travail est fausse, mais l'unité indépendante est le groupe. Avec une quarantaine de groupes ou moins, les erreurs-types sandwich tendent à être trop petites, et faute de vraisemblance, l'AIC et le test du rapport de vraisemblance ne sont pas disponibles (tests de Wald et QIC).

10.6 Exercices

Les exercices sont classés par niveau de difficulté. Leurs solutions sont regroupées dans l'annexe Solutions des exercices ; le lien « Voir la solution » sous chaque énoncé y mène directement.

10.6.1 Niveau facile

Exercice 1. On dispose de m groupes de c observations chacun, de même variance σ^2_Y. Deux observations d'un même groupe ont une corrélation ρ, et deux observations de groupes différents sont indépendantes. Montrez que la moyenne $\bar{Y}$ des $N = mc$ observations a pour variance

$$Var[\bar{Y}] = \frac{\sigma^2_Y}{N}\left\{1 + (c-1)\rho\right\}.$$

Que vaut la taille d'échantillon effective si $m = 20$, $c = 10$ et $\rho = 0{,}3$?

Voir la solution

Exercice 2. Dans le modèle à ordonnée à l'origine aléatoire $Y_{ij} = x_{ij}^\top\beta + b_i + \epsilon_{ij}$, avec $\sigma_b = 2$ et $\sigma = 4$:

a) calculez $Var[Y_{ij}]$, $Cov[Y_{ij}, Y_{ik}]$ pour $j \neq k$ et la corrélation intraclasse ;
b) que vaut $Cov[Y_{ij}, Y_{lk}]$ pour deux groupes $i \neq l$?

Voir la solution

Exercice 3. Dans un modèle à ordonnée à l'origine aléatoire, $\sigma^2_b = 1$ et $\sigma^2 = 4$. Deux groupes ont le même résidu moyen $\bar{y}_i - \bar{x}_i^\top\widehat{\beta} = 2$, mais le premier compte 1 observation et le second 16. Calculez leurs BLUP et interprétez.

Voir la solution

Exercice 4. Dans le modèle à ordonnée et pente aléatoires ajusté aux données `sleepstudy` (Chapitre 10), les estimations sont $d_{00} = 612$, $d_{01} = 9{,}6$, $d_{11} = 35{,}1$ et $\sigma^2 = 655$, avec t en jours.

a) Calculez $Var[Y_{ij}]$ au jour 0 et au jour 9.

b) Calculez la corrélation entre deux mesures d'un même sujet prises aux jours 0 et 9, puis aux jours 8 et 9. Comparez-les à la corrélation intraclasse du modèle à ordonnée à l'origine aléatoire seule, ajusté aux mêmes données, pour lequel $\sigma_b^2 \approx 1\,378$ et $\sigma^2 \approx 960$.

Voir la solution

Exercice 5. On teste $H_0 : \sigma_b^2 = 0$ dans un modèle à ordonnée à l'origine aléatoire, et la statistique du rapport de vraisemblance vaut 3,2.

a) Calculez la valeur-p avec la loi χ^2_1, puis avec le mélange $\frac{1}{2}\chi^2_0 + \frac{1}{2}\chi^2_1$. Que conclut-on au seuil de 5 % dans chaque cas ?

b) Quelle est la valeur critique du test au seuil de 5 % sous le mélange ? Expliquez pourquoi elle correspond au quantile d'ordre 0,90 de la loi χ^2_1.

Voir la solution

10.6.2 Niveau moyen

Exercice 1. Pour le modèle à ordonnée à l'origine aléatoire, la matrice de variance-covariance du groupe i est $V_i = \sigma^2 I_{n_i} + \sigma_b^2 1 1^\top$, où 1 est le vecteur de n_i uns.

a) Vérifiez que
$$V_i^{-1} = \frac{1}{\sigma^2}\left(I_{n_i} - \frac{\sigma_b^2}{\sigma^2 + n_i\sigma_b^2} 1 1^\top\right).$$

b) À partir de la formule générale $\hat{b}_i = \sigma_b^2 1^\top V_i^{-1}(y_i - X_i\hat{\beta})$, retrouvez la formule de rétrécissement du Chapitre 10.

Voir la solution

Exercice 2. Pour un seul groupe et β fixé, notons $r_j = y_{ij} - x_{ij}^\top\beta$. Montrez que la valeur de b qui minimise
$$\sum_{j=1}^{n_i}(r_j - b)^2 + \lambda b^2$$
est $\hat{b} = n_i\bar{r}/(n_i + \lambda)$. Déduisez-en que le BLUP est la solution de ce problème ridge avec $\lambda = \sigma^2/\sigma_b^2$, comme l'affirment les équations de Henderson. Que se passe-t-il quand $\sigma_b^2 \to \infty$ et quand $\sigma_b^2 \to 0$?

Voir la solution

Exercice 3. Considérez le GLMM probit à ordonnée à l'origine aléatoire, $P(Y_{ij} = 1 \mid b_i) = \Phi(\eta_{ij} + b_i)$, avec $\eta_{ij} = x_{ij}^\top\beta$ et $b_i \sim \mathcal{N}(0, \sigma_b^2)$.

a) En écrivant $\Phi(\eta + b) = P(Z \leq \eta + b \mid b)$ avec $Z \sim \mathcal{N}(0,1)$ indépendante de b, montrez que la probabilité marginale vaut $P(Y_{ij} = 1) = \Phi\left(\eta_{ij}/\sqrt{1+\sigma_b^2}\right)$.

b) Déduisez-en que les coefficients marginaux valent $\beta/\sqrt{1+\sigma_b^2}$. Calculez le facteur d'atténuation pour $\sigma_b = 1$.

Voir la solution

Exercice 4. Considérez le modèle linéaire mixte à ordonnée à l'origine aléatoire, $Y_{ij} = x_{ij}^\top \beta + b_i + \epsilon_{ij}$, et notons $r_j = y_{ij} - x_{ij}^\top \beta$ pour un groupe i de taille n_i, avec β, σ_b^2 et σ^2 fixés.

a) Écrivez $h_i(b) = \sum_j \log f(y_{ij} \mid b) + \log \phi(b; 0, \sigma_b^2)$ et montrez que c'est un polynôme de degré 2 en b. Déduisez-en que l'approximation de Laplace de la vraisemblance du groupe est exacte.

b) Montrez que le mode $\tilde{b}_i$ de h_i est le BLUP du Chapitre 10, et que $1/\{-h_i''(\tilde{b}_i)\}$ est égal à $Var[b_i \mid y_i]$.

Voir la solution

10.6.3 Niveau difficile

Exercice 1. Considérez un GEE avec le lien identité, $V(\mu) = 1$ et la structure de travail d'indépendance ($R_i = I$).

a) Montrez que l'estimateur GEE est l'estimateur des moindres carrés ordinaires, $\hat{\beta} = (X^\top X)^{-1} X^\top y$, où X et y empilent tous les groupes.

b) Montrez que l'estimateur sandwich devient

$$\hat{\mathrm{V}}(\hat{\beta}) = (X^\top X)^{-1} \left(\sum_{i=1}^{m} X_i^\top e_i e_i^\top X_i \right) (X^\top X)^{-1},$$

où $e_i = y_i - X_i \hat{\beta}$, et comparez-le à l'estimateur usuel $\hat{\sigma}^2 (X^\top X)^{-1}$.

c) Que devient cet estimateur si chaque groupe ne contient qu'une observation ? Quelle hypothèse du modèle linéaire permet-il alors de relâcher ?

Voir la solution

Exercice 2 (appliqué). On étudie par simulation (Chapitre 1) la couverture des intervalles de confiance lorsqu'on ignore la corrélation intra-groupe. Générez $B = 300$ jeux de données de $m = 20$ groupes de $c = 10$ observations selon

$$Y_{ij} = x_i + b_i + \epsilon_{ij}, \qquad x_i \sim \mathcal{N}(0,1), \quad b_i \sim \mathcal{N}(0;\ 0{,}3), \quad \epsilon_{ij} \sim \mathcal{N}(0;\ 0{,}7),$$

où la variable explicative x_i est constante à l'intérieur d'un groupe (la vraie pente vaut 1).

a) Pour chaque jeu de données, construisez l'intervalle de confiance à 95 % de la pente avec `lm`, puis avec `lmer` et le quantile de la loi normale, puis avec `lmer` et le quantile de la loi de Student à $m-2$ degrés de liberté. Estimez la couverture de chaque intervalle.
b) Expliquez les résultats à l'aide de l'effet du plan.

Voir la solution

Exercice 3 (appliqué). On reprend les données `ohio` et on compare les moyennes marginales décrites par le GLMM et par le GEE.

a) Ajustez le GLMM `glmer(resp ~ age + smoke + (1 | id), family = binomial, nAGQ = 25)` et le GEE `geeglm(resp ~ age + smoke, id = id, family = binomial, corstr = "exchangeable")`.
b) Pour chacune des huit combinaisons de `age` et de `smoke`, calculez la probabilité marginale prédite par le GLMM en intégrant numériquement sur l'effet aléatoire, la probabilité prédite par le GEE et la proportion observée. Commentez.
c) Calculez aussi, pour chaque combinaison, la probabilité conditionnelle de l'enfant typique ($b=0$). Pourquoi est-elle beaucoup plus petite que la proportion observée ?

Voir la solution

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Faraway (2016). *Extending the Linear Model with R*
— Agresti (2013). *Categorical Data Analysis*
— Wood (2017). *Generalized Additive Models : An Introduction with R*
— Heritier, Cantoni, Copt et Victoria-Feser (2009). *Robust Methods in Biostatistics*
— Agresti (2010). *Analysis of Ordinal Categorical Data*
— Ruppert, Wand et Carroll (2003). *Semiparametric Regression*
— Fitzmaurice, Laird et Ware (2011). *Applied Longitudinal Analysis*
— Pinheiro et Bates (2000). *Mixed-Effects Models in S and S-PLUS*

11 Au-delà des modèles de ce livre

Les chapitres précédents ont présenté une famille cohérente de modèles de régression : le modèle linéaire, les GLM (Chapitre 3) et leurs extensions aux réponses catégorielles et aux comptes avec excès de zéros (Chapitre 4), leurs versions pénalisées (Chapitre 6), non paramétriques (Chapitre 7, Chapitre 8) ou additives (Chapitre 9), et enfin les modèles pour données corrélées (Chapitre 10). Dans tous ces modèles, seule l'espérance de la réponse (ou une fonction de celle-ci) dépend des variables explicatives. Ce chapitre présente brièvement quelques modèles classiques qui relâchent cette hypothèse, ainsi qu'un type de réponse que le livre n'a pas couvert, les durées de vie. Je donne seulement l'idée générale de chacun et sa formulation mathématique la plus simple, même si chacun mériterait un cours à lui seul.

11.1 Au-delà de l'espérance : GAMLSS et régression quantile

11.1.1 GAMLSS

Définition — GAMLSS

Dans un GLM ou un GAM, seule l'espérance dépend des variables explicatives : la dispersion ϕ est constante. Les modèles additifs généralisés pour la position, l'échelle et la forme (GAMLSS) laissent chaque paramètre de la loi de Y_i dépendre de ses propres covariables. Pour une loi $\mathcal{D}(\mu, \sigma, \nu, \kappa)$ à quatre paramètres (position, échelle, asymétrie, aplatissement), on pose

$$Y_i \sim \mathcal{D}(\mu_i, \sigma_i, \nu_i, \kappa_i), \qquad g_k(\theta_{ki}) = x_{ki}^\top \beta_k + \sum_j f_{kj}(x_{kji}), \quad k = 1, \dots, 4,$$

où $\theta_{1i} = \mu_i$, $\theta_{2i} = \sigma_i$, $\theta_{3i} = \nu_i$, $\theta_{4i} = \kappa_i$, chaque g_k est une fonction de lien et les f_{kj} sont des fonctions lisses comme dans un GAM.
Le cas le plus simple est le modèle normal hétéroscédastique

$$Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2), \qquad \mu_i = x_i^\top \beta, \qquad \log \sigma_i = z_i^\top \gamma.$$

Remarque — Une famille très large

La loi $\mathcal{D}$ n'est plus restreinte à la famille exponentielle, et la librairie `gamlss` propose plus d'une centaine de lois (continues, discrètes, mixtes, tronquées, avec excès de zéros). Le GLM et le GAM sont des cas particuliers où seul μ est modélisé. Ce cadre est utile pour construire des courbes de référence, comme les courbes de centiles de croissance, où la variabilité et l'asymétrie changent avec l'âge.

11.1.2 Régression quantile

Définition — Régression quantile

Plutôt que l'espérance, on modélise le quantile d'ordre $\tau \in (0,1)$ de la réponse :

$$Q_\tau(Y_i \mid x_i) = x_i^\top \beta(\tau).$$

L'estimateur $\widehat{\beta}(\tau)$ minimise

$$\sum_{i=1}^{n} \rho_\tau(y_i - x_i^\top \beta)\,, \qquad \rho_\tau(u) = u\,\{\tau - \mathbf{1}(u<0)\},$$

où ρ_τ est la fonction de perte quantile (*check function*, aussi appelée perte *pinball*). Le cas $\tau = 1/2$ correspond à la régression médiane, qui minimise la somme des valeurs absolues des résidus.

Remarque — Pourquoi les quantiles ?

La régression quantile ne fait aucune hypothèse de loi ni d'homoscédasticité. Sa perte est linéaire dans le résidu plutôt que quadratique, ce qui limite l'influence des grands résidus en Y par rapport aux moindres carrés. Elle n'est pas robuste pour autant au sens du Chapitre 12 : les points à fort levier (atypiques en x) peuvent toujours être très influents, et sa fonction d'influence n'est pas bornée. Ajuster le modèle pour plusieurs valeurs de τ donne une description beaucoup plus complète de la loi conditionnelle que la seule espérance, ce qui permet de détecter des effets qui diffèrent selon la queue de la distribution. Par exemple, une variable peut influencer les hauts salaires sans toucher les bas salaires. Le problème de minimisation est un programme linéaire, résolu en R par `rq` de la librairie `quantreg`.

11.2 Durées de vie : le modèle de Cox

Définition — Modèle à risques proportionnels de Cox

Soit T_i le temps jusqu'à un événement (décès, panne, rechute). Sa fonction de risque $h(t \mid x_i)$ est le taux instantané d'occurrence au temps t, sachant que l'événement n'a pas encore eu lieu. Le modèle de Cox pose

$$h(t \mid x_i) = h_0(t) \exp(x_i^\top \beta)\,,$$

où $h_0(t)$ est un risque de base commun, laissé entièrement non spécifié. Le rapport des risques de deux individus, $\exp\{(x_i - x_j)^\top \beta\}$, ne dépend pas de t, d'où le nom de risques proportionnels.

Remarque — Censure et vraisemblance partielle

La particularité des données de survie est la censure : pour certains individus, on sait seulement que T_i dépasse la durée observée. Le modèle de Cox traite la censure naturellement et estime β sans jamais estimer h_0, en maximisant la vraisemblance partielle

$$L_P(\beta) = \prod_{i\,:\,\delta_i = 1} \frac{\exp(x_i^\top \beta)}{\sum_{j \in \mathcal{R}(t_i)} \exp(x_j^\top \beta)},$$

où $t_i = \min(T_i, C_i)$ est la durée observée (C_i : temps de censure), $\delta_i = 1$ si l'événement a été observé et $\mathcal{R}(t_i)$ est l'ensemble des individus encore à risque juste avant t_i. En R, la fonction est `coxph` de la librairie `survival`. Il existe aussi des modèles paramétriques où la loi de T_i est entièrement spécifiée (Weibull, log-normale), ajustés par `survreg` de `survival`.

11.3 Pour aller plus loin

Chacun des modèles de ce chapitre a son ouvrage de référence : Stasinopoulos et al. (2017) pour les GAMLSS, Koenker (2005) pour la régression quantile et Therneau et Grambsch (2000) pour le modèle de Cox. Les références complémentaires donnent des lectures pour les chapitres du livre.

11.4 Vue d'ensemble des modèles

La Figure 11.1 donne un premier repère pour choisir un modèle selon la nature de la réponse et la structure des observations. Le tableau qui suit la complète.

```
Warning: The `label.size` argument of `geom_label()` is deprecated as of ggplot2 3.5.0.
i Please use the `linewidth` argument instead.
```

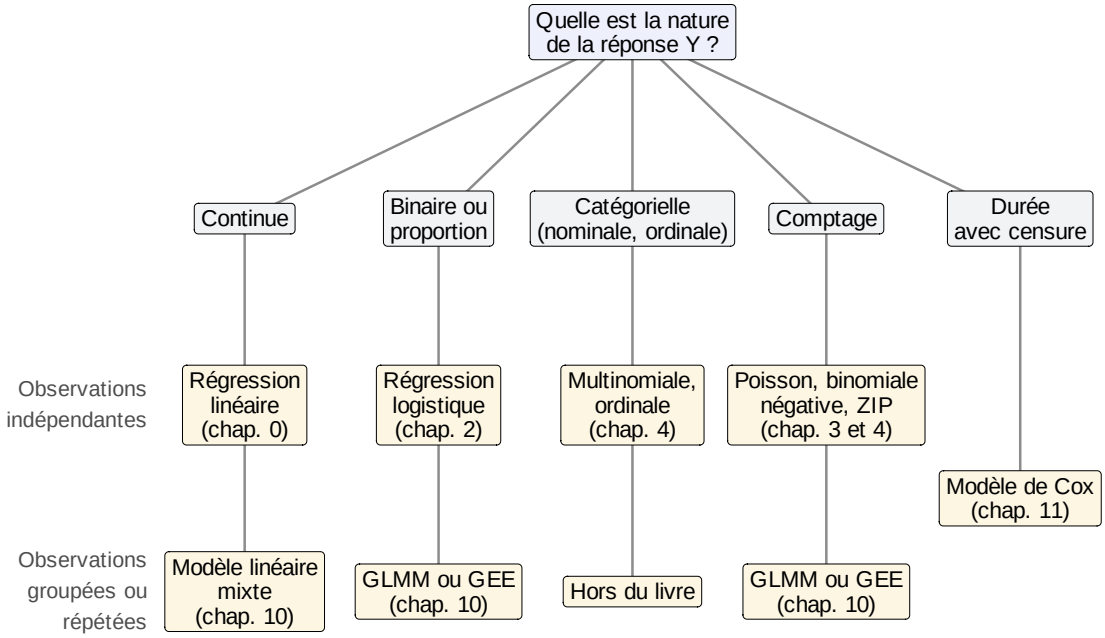


FIGURE 11.1 : Quel modèle pour mes données ? Premier choix selon la nature de la réponse et la dépendance entre les observations.

Quelle que soit la branche, quatre questions s'ajoutent :

- effets non linéaires : si la forme de l'effet d'une variable continue est inconnue, des splines ou un GAM (chapitres 8 et 9) remplacent le terme linéaire, aussi dans un GLM ou un modèle mixte ;
- nombreuses variables explicatives : la régularisation (ridge, lasso, elastic net, Chapitre 6) s'applique à toute réponse d'un GLM ;
- observations atypiques : les méthodes robustes (Chapitre 12) limitent leur influence ;
- quantité visée : si ce sont les quantiles de la réponse plutôt que sa moyenne, la régression quantile s'impose.

Le tableau suivant récapitule les modèles vus dans ce livre et ceux présentés dans ce chapitre, en donnant pour chacun le type de réponse traité et les applications typiques. Les trois derniers ne sont présentés que dans ce chapitre.

Modèle	Réponse	Idée clé	Applications typiques	Fonction R
Régression linéaire	Continue	$E[Y] = x^\top \beta$, erreurs normales homoscédastiques	Relations approximativement linéaires : prix, rendements, mesures physiques	`lm`
Régression logistique	Binaire	$\text{logit}\, P(Y = 1) = x^\top \beta$	Diagnostic, défaut de paiement, succès/échec, classification à deux classes	`glm(family = binomial)`
Régression de Poisson	Comptage	$\log E[Y] = x^\top \beta$, avec offset pour l'exposition	Nombre d'accidents, de réclamations, de cas d'une maladie ; taux d'incidence	`glm(family = poisson)`
Quasi-Poisson, binomiale négative	Comptage surdispersé	Variance supérieure à la moyenne (Section 3.6.5)	Mêmes applications que Poisson quand la variabilité est trop grande	`glm(family = quasipoisson)`, `MASS::glm.nb`
GLM gamma, inverse gaussienne	Continue positive asymétrique	Variance croissant avec la moyenne	Montants de sinistres, durées, coûts, concentrations	`glm(family = Gamma)`, `glm(family = inverse.gaussian)`
Régression multinomiale	Catégorielle nominale	Logit de chaque catégorie contre une référence	Choix de produit, de mode de transport, classification à plusieurs classes	`nnet::multinom`
Régression ordinale	Catégorielle ordonnée	Logits cumulatifs à cotes proportionnelles	Échelles de Likert, stades d'une maladie, niveaux de satisfaction	`MASS::polr`, `ordinal::clm`

Modèle	Réponse	Idée clé	Applications typiques	Fonction R
ZIP, modèle à barrière	Comptage avec excès de zéros	Mélange d'un zéro structurel et d'un modèle de comptage	Nombre de visites, de consommations, d'espèces observées	`pscl::zeroinfl`, `pscl::hurdle`
Ridge, lasso	Toute réponse d'un GLM	Pénalisation de β pour réduire la variance	Nombreuses variables explicatives, colinéarité, sélection automatique de variables	`glmnet::glmnet`
k plus proches voisins	Continue ou catégorielle	Moyenne locale des k voisins	Prédiction sans forme fonctionnelle, peu de variables explicatives	`class::knn`, `FNN::knn.reg`
Estimateur à noyau	Continue	Moyenne pondérée localement par un noyau	Lissage d'une relation univariée ou bivariée	`KernSmooth::locpoly`, `ksmooth`
Splines	Continue ou GLM	Polynômes par morceaux raccordés aux nœuds	Effet non linéaire d'une variable (âge, dose, temps) dans un modèle paramétrique	`splines::bs`, `splines::ns`, `smooth.spline`
GAM	Toute réponse d'un GLM	$g(E[Y]) = \alpha + \sum_j f_j(x_j)$	Plusieurs effets non linéaires additifs, exploration de la forme des effets	`mgcv::gam`
Modèle linéaire mixte	Continue, données groupées	Effets aléatoires par groupe	Mesures répétées, données longitudinales, élèves dans des écoles	`lme4::lmer`

Modèle	Réponse	Idée clé	Applications typiques	Fonction R
GLMM	GLM, données groupées	Effets aléatoires dans le prédicteur linéaire	Réponses binaires ou de comptage répétées par sujet ; interprétation sujet-spécifique	`lme4::glmer`
GEE	GLM, données groupées	Moyenne marginale et corrélation de travail	Essais cliniques, enquêtes longitudinales ; interprétation populationnelle	`geepack::geeglm`
GAMLSS	Toute loi, continue ou discrète	Covariables dans chaque paramètre de la loi	Hétéroscédasticité, courbes de centiles, lois hors famille exponentielle	`gamlss::gamlss`
Régression quantile	Continue	$Q_\tau(Y) = x^\top \beta(\tau)$	Salaires, données hétéroscédastiques ou avec valeurs aberrantes, queues de distribution	`quantreg::rq`
Modèle de Cox	Durée, avec censure	Risques proportionnels, risque de base non spécifié	Survie de patients, fiabilité, attrition de clients	`survival::coxph`

Remarque — À retenir

— Au-delà de l'espérance : dans les modèles des chapitres précédents, seule l'espérance de la réponse (ou une fonction de celle-ci) dépend des variables explicatives, et les GAMLSS et la régression quantile relâchent cette hypothèse ;

— GAMLSS : chaque paramètre de la loi de Y_i (position, échelle, asymétrie, aplatissement) dépend de ses propres covariables, et la loi n'est plus restreinte à la famille exponentielle (librairie `gamlss`). Le cas le plus simple est le modèle normal hétéroscédastique, avec $\log \sigma_i = z_i^\top \gamma$;

— Régression quantile : $Q_\tau(Y_i \mid x_i) = x_i^\top \beta(\tau)$, estimé avec la perte ρ_τ par `rq` de `quantreg` ($\tau = 1/2$ donne la régression médiane). Elle ne suppose ni loi ni homoscédasticité et limite l'influence des grands résidus en Y, mais elle n'est pas robuste aux points à fort levier ;
— Modèle de Cox : $h(t \mid x_i) = h_0(t)\exp(x_i^\top \beta)$, avec un risque de base $h_0(t)$ non spécifié et un rapport des risques qui ne dépend pas de t. La vraisemblance partielle tient compte de la censure et estime β sans estimer h_0 (`coxph` de `survival`).

12 Ouverture sur la statistique robuste

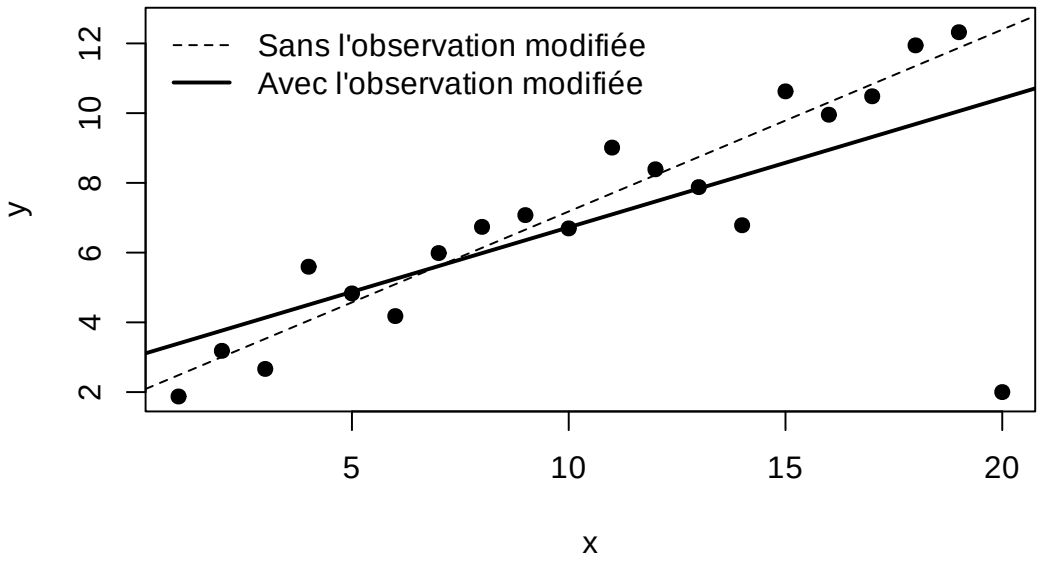
Ce chapitre donne l'idée générale de la statistique robuste, sans en faire la théorie. Tous les modèles de ce livre supposent que les données sont engendrées par le modèle postulé. Les diagnostics de la régression linéaire et de la régression logistique (leviers, distance de Cook, DFBETAS) servent à repérer, après coup, les observations qui s'en écartent et qui tirent l'ajustement vers elles. La statistique robuste prend l'autre chemin. Au lieu de détecter puis de retirer ces observations à la main, elle construit des estimateurs robustes, qui, par conception, ne se laissent pas entraîner par une petite fraction de données atypiques.

12.1 Pourquoi les moindres carrés sont fragiles

Exemple — Une seule observation suffit

Prenez la moyenne $\bar{y}$ d'un échantillon $y_1, \dots, y_n$. Remplacer une seule valeur y_n par $y_n + d$ déplace la moyenne de d/n, de sorte qu'il suffit de déplacer une seule observation pour rendre la moyenne aussi grande qu'on veut. La médiane, elle, reste bornée tant que moins de la moitié des observations sont contaminées : elle peut se déplacer, mais jamais au-delà des valeurs non contaminées.

L'estimateur des moindres carrés $\hat{\beta} = (X^\top X)^{-1} X^\top y$ a le même défaut, parce qu'il est linéaire en y. Une seule observation aberrante peut déplacer toute la droite ajustée, surtout si son levier h_i est élevé.



Un seul point déplacé (à droite) fait pivoter la droite des moindres carrés. Notez que rien dans la sortie de `lm` ne le signale : il faut regarder les diagnostics.

Le problème est structurel, puisque la perte quadratique $(y_i - x_i^\top \beta)^2$ croît sans borne avec le résidu. Les observations les plus éloignées du modèle sont donc précisément celles qui pèsent le plus dans l'estimation. L'estimateur du maximum de vraisemblance d'un GLM a le même défaut, pour la même raison.

12.2 Deux mesures de robustesse

Définition — Point de rupture

Le point de rupture (*breakdown point*) d'un estimateur est la plus petite fraction d'observations qu'il faut remplacer par des valeurs arbitraires pour que l'estimateur puisse prendre une valeur arbitraire. La moyenne et les moindres carrés ont un point de rupture de $1/n$, soit 0 asymptotiquement, alors que la médiane atteint le maximum possible, $1/2$.

Définition — Fonction d'influence

La fonction d'influence mesure l'effet d'une contamination infinitésimale au point z sur un estimateur T, vu comme une fonctionnelle de la loi F :

$$IF(z; T, F) = \lim_{\varepsilon \to 0} \frac{T\{(1-\varepsilon)F + \varepsilon\, \delta_z\} - T(F)}{\varepsilon},$$

où δ_z est la masse ponctuelle en z. Un estimateur est dit B-robuste si sa fonction d'influence est bornée, auquel cas aucune observation isolée, aussi extrême soit-elle, ne peut le déplacer de façon illimitée. Par exemple, la fonction d'influence de la moyenne, $IF(z) = z - \mu$, n'est pas bornée, alors que celle de la médiane l'est.

Remarque — Deux aspects complémentaires

Le point de rupture décrit la résistance à une contamination massive, alors que la fonction d'influence décrit la sensibilité à une contamination infinitésimale. Un bon estimateur robuste a un point de rupture élevé et une fonction d'influence bornée. Il garde aussi une efficacité raisonnable sous le modèle (une variance pas trop supérieure à celle de l'estimateur classique). On accepte de perdre un peu d'efficacité sous le modèle pour gagner beaucoup de stabilité en dehors.

12.3 L'idée centrale : borner la contribution de chaque observation

Il faut distinguer dès maintenant deux sortes d'observations atypiques. Une observation peut avoir un grand résidu (elle est atypique en y, compte tenu de x), ou un fort levier (elle est atypique en x). Les deux ne demandent pas la même protection, et les estimateurs de cette section ne protègent pas tous contre les deux (voir le tableau à la fin de la section sur les M-estimateurs).

12.3.1 M-estimateurs

Définition — M-estimateur de régression

Au lieu de minimiser une somme de carrés, on minimise

$$\sum_{i=1}^{n} \rho\left(\frac{y_i - x_i^\top \beta}{\sigma}\right),$$

où ρ est une fonction de perte symétrique qui croît moins vite que le carré pour les grands résidus. L'échelle σ est estimée de façon robuste (par exemple, la déviation absolue médiane des résidus). En dérivant, on obtient les équations d'estimation

$$\sum_{i=1}^{n} \psi\left(\frac{y_i - x_i^\top \beta}{\sigma}\right) x_i = 0, \qquad \psi = \rho'.$$

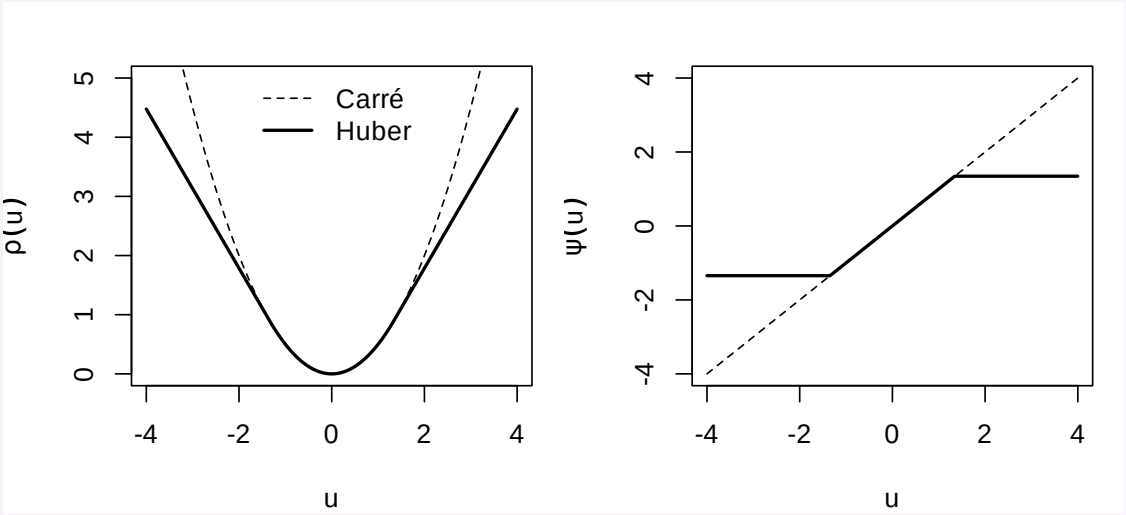
Les moindres carrés correspondent à $\rho(u) = u^2/2$ et $\psi(u) = u$, non bornée.

Exemple — Fonction de Huber

Le choix classique est

$$\rho_c(u) = \begin{cases} u^2/2 & \text{si } |u| \le c, \\ c|u| - c^2/2 & \text{si } |u| > c, \end{cases} \qquad \psi_c(u) = \max\{-c,\ \min(u, c)\}.$$

La perte est quadratique pour les petits résidus (comme les moindres carrés) et linéaire au-delà de c. L'effet du résidu sur le score, $\psi_c(u)$, est donc borné par c (mais pas celui de x_i, voir plus bas). Le choix $c \approx 1{,}345$ conserve 95 % de l'efficacité des moindres carrés quand les erreurs sont normales.



Remarque — Poids et interprétation

Posons $w_i = \psi(r_i)/r_i$, où $r_i = (y_i - x_i^\top \beta)/\sigma$ est le résidu standardisé ; les équations d'estimation s'écrivent alors $\sum_i w_i\,(y_i - x_i^\top \beta)\,x_i = 0$. Un M-estimateur est donc une régression par moindres carrés pondérés dont les poids décroissent avec la taille du résidu. En fait, il se calcule par IRLS, comme un GLM. Les poids $\hat{w}_i$ en sortie sont un diagnostic utile, puisqu'une observation de poids proche de 0 a été reconnue comme atypique par l'estimateur lui-même. Notez qu'un poids faible signale une observation que l'estimateur a sous-pondérée (*downweighted*), pas nécessairement une erreur de saisie (voir la remarque

« Deux lectures d'une observation atypique » plus bas).
Un M-estimateur de Huber protège contre les valeurs aberrantes en y, mais pas contre les points à fort levier (aberrants en x), parce que x_i reste non borné dans le score. Les estimateurs à point de rupture élevé (S-estimateurs, MM-estimateurs, moindres carrés tronqués) corrigent ce défaut et sont aujourd'hui le choix par défaut. En R, ils sont offerts par `MASS::rlm` (M- et MM-estimateurs) et `robustbase::lmrob` (MM-estimateur).

Le tableau suivant résume la protection qu'offre chaque estimateur. Les deux colonnes du milieu décrivent la fonction d'influence, dans la direction des résidus et dans celle des leviers. La colonne du point de rupture décrit la résistance à une contamination massive.

Estimateur	Grand résidu en y	Point à fort levier	Point de rupture	Calcul
Moindres carrés	influence non bornée ($\psi(u) = u$)	influence non bornée	$1/n$	explicite
Régression médiane (L_1)	influence bornée	influence non bornée	$1/n$	programme linéaire
M-estimateur de Huber	influence bornée par c	influence non bornée	$1/n$	IRLS
Estimateurs à point de rupture élevé (S, MM, moindres carrés tronqués)	influence bornée	point rejeté s'il est mal ajusté	jusqu'à $1/2$	plus complexe (optimisation non convexe, ré-échantillonnage)

Un point à fort levier n'est pas toujours nuisible : s'il suit la tendance des autres, il améliore la précision de l'estimation. C'est le point à fort levier mal ajusté qui fait pivoter la droite, comme dans l'exemple « Une seule observation suffit ».

12.3.2 Extension aux GLM

Remarque — GLM robustes

L'idée s'étend aux GLM en bornant le score. On remplace les équations du maximum de vraisemblance, $\sum_i (y_i - \mu_i)\, x_i\, \mu_i' / V(\mu_i) = 0$, où $\mu_i' = \partial \mu_i / \partial \eta_i$, par

$$\sum_{i=1}^{n} \left[\psi_c(r_i)\, w(x_i)\, \frac{\mu_i'}{\sqrt{\phi\, V(\mu_i)}}\, x_i - a(\beta) \right] = 0, \qquad r_i = \frac{y_i - \mu_i}{\sqrt{\phi\, V(\mu_i)}},$$

où ψ_c borne l'effet des résidus de Pearson standardisés r_i et où $w(x_i)$, un poids sur les

x différent des w_i de la remarque précédente, réduit le poids des observations à fort levier. Le paramètre de dispersion vaut $\phi = 1$ pour les lois binomiale et de Poisson. Pour la loi gamma, ϕ est inconnu et estimé de façon robuste. Le terme $a(\beta)$ est une correction qui assure la convergence de l'estimateur vers β sous le modèle. L'approche s'étend à la régression logistique polytomique (Miron et al. 2022). En R, la fonction `robustbase::glmrob` traite les lois binomiale, de Poisson et gamma.

12.4 Robustesse et modélisation

Remarque — Deux lectures d'une observation atypique

Une observation atypique est soit une erreur (saisie, mesure, individu hors population cible), soit un signal que le modèle est trop simple. La statistique robuste ne tranche pas cette question : elle fournit un ajustement qui décrit la majorité des données, et des poids qui désignent la minorité à examiner. C'est ensuite à vous de décider quoi faire : corriger les données, enrichir le modèle (une variable manquante, un lien différent, une loi à queues plus lourdes) ou simplement rapporter les deux ajustements.

La robustesse ne remplace donc pas les diagnostics : elle les rend fiables. Avec les moindres carrés, une observation influente attire la droite vers elle, de sorte que son propre résidu paraît petit. Plusieurs observations aberrantes peuvent aussi se masquer les unes les autres (effet de masquage) : en retirer une seule ne change presque rien, et les diagnostics calculés une observation à la fois ne détectent aucune d'elles. Un ajustement robuste évite ce piège, puisque ses résidus sont calculés par rapport à une droite peu déformée par ces observations.

Exemple — En R

```r
library(MASS)
# x et y_cont : données simulées de l'exemple « Une seule observation
 ↪  suffit »
# (set.seed(1), y = 2 + 0.5 x + erreur N(0, 1), puis y_cont[20] <- 2)
mod.ls <- lm(y_cont ~ x)
mod.rob <- rlm(y_cont ~ x, method = "MM")
rbind(moindres.carres = coef(mod.ls), robuste = coef(mod.rob))
```

```
                (Intercept)         x
moindres.carres    3.023297 0.3702411
robuste            1.928542 0.5292928
```

```r
round(mod.rob$w, 2)
```

```
 [1] 0.97 1.00 0.93 0.78 0.99 0.92 0.99 0.97 0.99 0.97 0.85 1.00 0.92 0.48 0.95
[16] 0.98 0.98 0.98 0.99 0.00
```

L'estimateur robuste retrouve la pente d'origine, et l'observation modifiée reçoit un poids nul ou presque.

Remarque — Pour aller plus loin

Pour aller plus loin, je vous suggère d'abord Heritier et al. (2009). Ce livre applique les idées de ce chapitre aux modèles de ce livre, soit la régression linéaire, les GLM, les modèles mixtes, les données longitudinales marginales (GEE) et le modèle de Cox, avec des exemples réels et des indications pour R. Pour la théorie, les références classiques sont Huber et Ronchetti (2009) et Maronna et al. (2019), et l'estimateur robuste des GLM de la section précédente vient de Cantoni et Ronchetti (2001).

Remarque — À retenir

- Fragilité des moindres carrés : comme la perte quadratique croît sans borne avec le résidu, une seule observation peut déplacer toute la droite ajustée, surtout si son levier est élevé. L'estimateur du maximum de vraisemblance d'un GLM a le même défaut ;
- Deux mesures : le point de rupture décrit la résistance à une contamination massive ($1/n$ pour les moindres carrés, $1/2$ pour la médiane), et la fonction d'influence la sensibilité à une contamination infinitésimale (un estimateur est B-robuste si elle est bornée) ;
- M-estimateurs : le carré est remplacé par une perte ρ qui croît moins vite, comme celle de Huber ($c \approx 1{,}345$ conserve 95 % de l'efficacité des moindres carrés sous normalité), et l'estimateur se calcule par IRLS. Un poids $\hat{w}_i$ proche de 0 désigne une observation sous-pondérée, pas nécessairement une erreur ;
- Fort levier : le M-estimateur de Huber ne protège pas contre les points aberrants en x, contrairement aux estimateurs à point de rupture élevé (S, MM, moindres carrés tronqués), offerts par `MASS::rlm` (`method = "MM"`) et `robustbase::lmrob`. Pour les GLM, l'estimateur robuste borne l'effet des résidus de Pearson et réduit le poids des points à fort levier (`robustbase::glmrob`) ;
- Robustesse et diagnostics : un ajustement robuste ne remplace pas les diagnostics, il les rend fiables, parce que ses résidus sont calculés par rapport à un ajustement peu déformé, ce qui évite l'effet de masquage. C'est ensuite à vous de décider si une observation atypique est une erreur ou le signe d'un modèle trop simple.

Références complémentaires

La page Références complémentaires décrit chacun de ces ouvrages.

— Heritier, Cantoni, Copt et Victoria-Feser (2009). *Robust Methods in Biostatistics*

Solutions des exercices

Cette annexe regroupe les solutions des exercices proposés à la fin de chaque chapitre. Les exercices sont classés en trois niveaux de difficulté (facile, moyen et difficile), sauf ceux des chapitres 8 et 9, qui sont surtout appliqués. Les chapitres 1, 2, 3, 5 et 6 se terminent aussi par des exercices appliqués. Ces exercices demandent d'ajuster des modèles en R, et leurs solutions contiennent le code et ses résultats. Le lien « Retour à l'énoncé » sous chaque numéro ramène à l'énoncé, dans son chapitre.

Les solutions qui contiennent du code R utilisent la librairie `dplyr` pour la préparation des données, en plus des librairies chargées dans chaque solution.

```
library(dplyr)
```

Chapitre 0 : Rappels de régression linéaire

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

Chaque ligne de la matrice de design s'écrit $X_i^\top = (1, x_i)$, de sorte que

$$X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \qquad X^\top X = \begin{pmatrix} n & n\bar{x} \\ n\bar{x} & \sum x_i^2 \end{pmatrix}, \qquad X^\top Y = \begin{pmatrix} n\bar{Y} \\ \sum x_i Y_i \end{pmatrix}.$$

Le déterminant de $X^\top X$ est $n\sum x_i^2 - (n\bar{x})^2 = n\sum_i (x_i - \bar{x})^2$, de sorte que

$$(X^\top X)^{-1} = \frac{1}{n\sum_i (x_i - \bar{x})^2} \begin{pmatrix} \sum x_i^2 & -n\bar{x} \\ -n\bar{x} & n \end{pmatrix}.$$

En multipliant par $X^\top Y$, la seconde composante donne

$$\hat{\beta}_1 = \frac{n\sum x_i Y_i - \sum x_i \sum Y_i}{n\sum_i (x_i - \bar{x})^2} = \frac{\sum_i (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_i (x_i - \bar{x})^2},$$

en utilisant l'identité $n\sum x_i Y_i - \sum x_i \sum Y_i = n\sum_i (x_i - \bar{x})(Y_i - \bar{Y})$. La première composante donne, après simplification, $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$.

Exercice 2. Retour à l'énoncé

En notant $e = Y - X\hat{\beta}$, les équations normales s'écrivent $X^\top e = 0$. Comme la première colonne de X ne contient que des 1, la première ligne de ce système est exactement $\sum_{i=1}^n e_i = 0$, alors que chaque autre ligne, associée à la colonne des x_{ij}, donne $\sum_{i=1}^n x_{ij} e_i = 0$.

Comme $\sum_i e_i = \sum_i (Y_i - \hat{Y}_i) = 0$, on a $\sum_i Y_i = \sum_i \hat{Y}_i$, de sorte que la moyenne des valeurs prédites est égale à la moyenne observée : $\bar{\hat{Y}} = \bar{Y}$.

Exercice 3. Retour à l'énoncé

Le modèle à p variables est un cas particulier du modèle à $p+1$ variables, puisqu'il suffit de fixer à 0 le coefficient de la variable ajoutée. Les moindres carrés minimisent SSE sur l'ensemble de tous les coefficients possibles, et cet ensemble est plus grand pour le modèle à $p+1$ variables (il contient en particulier la solution du modèle à p variables). On a donc nécessairement $SSE_{p+1} \leq SSE_p$. Comme $R^2 = 1 - SSE/SST$ et que SST ne dépend que de Y (donc reste inchangé), le R^2 ne peut qu'augmenter ou rester constant.

Le R^2 ajusté,

$$R^2_{adj} = 1 - \frac{SSE/(n-(p+1))}{SST/(n-1)},$$

divise SSE par les degrés de liberté résiduels $n-(p+1)$, qui diminuent quand on ajoute une variable. Si la baisse de SSE est trop faible pour compenser celle des degrés de liberté, le ratio $SSE/(n-(p+1))$ augmente et R^2_{adj} diminue, ce qui pénalise l'ajout d'une variable peu informative.

Niveau moyen

Exercice 1. Retour à l'énoncé

Symétrie : $H^\top = X\{(X^\top X)^{-1}\}^\top X^\top = X(X^\top X)^{-1}X^\top = H$, car $(X^\top X)^{-1}$ est symétrique.

Idempotence : $H^2 = X(X^\top X)^{-1}X^\top X(X^\top X)^{-1}X^\top = X(X^\top X)^{-1}X^\top = H$.

La symétrie de $I - H$ est immédiate, et son idempotence découle du calcul $(I-H)^2 = I - 2H + H^2 = I - 2H + H = I - H$.

Comme $e = (I-H)Y$ et $Var[Y] = \sigma^2 I$, la variance des résidus vaut

$$Var[e] = (I-H)\,\sigma^2 I\,(I-H)^\top = \sigma^2 (I-H)^2 = \sigma^2 (I-H).$$

L'élément diagonal i de cette matrice donne $Var[e_i] = \sigma^2(1-h_i)$.

Exercice 2. Retour à l'énoncé

La log-vraisemblance est

$$\ell(\beta, \sigma^2; Y) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(Y - X\beta)^\top (Y - X\beta).$$

À σ^2 fixé, maximiser ℓ en β revient à minimiser $SSE(\beta) = (Y-X\beta)^\top(Y-X\beta)$, de sorte que $\widehat{\beta}_{MV} = \widehat{\beta}$, c'est-à-dire l'estimateur des moindres carrés.

Pour σ^2, on annule la dérivée par rapport à σ^2 :

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{SSE}{2\sigma^4} = 0 \quad \Longrightarrow \quad \hat{\sigma}^2_{MV} = \frac{SSE}{n}.$$

Comme $\hat{\sigma}^2 = SSE/(n-(p+1))$, les deux estimateurs sont liés par

$$\hat{\sigma}^2_{MV} = \frac{n-(p+1)}{n}\,\hat{\sigma}^2.$$

Puisque $E[\hat{\sigma}^2] = \sigma^2$ (exercice suivant), on a $E[\hat{\sigma}^2_{MV}] = \frac{n-(p+1)}{n}\sigma^2 < \sigma^2$: l'estimateur du maximum de vraisemblance sous-estime donc systématiquement σ^2.

Exercice 3. Retour à l'énoncé

Comme $I-H$ est symétrique et idempotente, la somme des carrés des résidus s'écrit $SSE = e^\top e = Y^\top (I-H)^\top (I-H) Y = Y^\top (I-H) Y$. Le calcul de son espérance repose sur la formule d'une forme quadratique $Y^\top A Y$: si $E[Y] = \mu$ et $Var[Y] = \sigma^2 I$, on a $E[Y^\top A Y] = \sigma^2 \operatorname{Tr}(A) + \mu^\top A \mu$.

Ici, $\mu = X\beta$ et $(I-H)X = X - X(X^\top X)^{-1}X^\top X = X - X = 0$, de sorte que le second terme s'annule et qu'il reste

$$E[SSE] = \sigma^2 \operatorname{Tr}(I-H) = \sigma^2\{n - \operatorname{Tr}(H)\}.$$

Or $\operatorname{Tr}(H) = \operatorname{Tr}\{X(X^\top X)^{-1}X^\top\} = \operatorname{Tr}\{(X^\top X)^{-1}X^\top X\} = \operatorname{Tr}(I_{p+1}) = p+1$ (invariance cyclique de la trace), d'où $E[SSE] = \sigma^2(n-(p+1))$: l'estimateur $\hat{\sigma}^2$ est sans biais.

Niveau difficile

Exercice 1. Retour à l'énoncé

Soit $\tilde{\beta} = CY$ un estimateur linéaire quelconque, dont le biais est $E[\tilde{\beta}] - \beta = CX\beta - \beta = (CX - I)\beta$. Pour que $\tilde{\beta}$ soit sans biais quel que soit β, il faut donc $CX = I_{p+1}$.

En écrivant $C = (X^\top X)^{-1}X^\top + D$, la condition $CX = I + DX = I$ implique $DX = 0$.

La variance de $\tilde{\beta}$ est $Var[\tilde{\beta}] = \sigma^2 CC^\top$, où le produit $CC^\top$ se développe, après substitution de C, en

$$CC^\top = (X^\top X)^{-1} + (X^\top X)^{-1}X^\top D^\top + DX(X^\top X)^{-1} + DD^\top.$$

Comme $DX = 0$ (et donc $X^\top D^\top = 0$), les deux termes croisés s'annulent. Il reste

$$Var[\tilde{\beta}] = \sigma^2(X^\top X)^{-1} + \sigma^2 DD^\top = Var[\hat{\beta}] + \sigma^2 DD^\top.$$

Comme $DD^\top$ est semi-définie positive, $Var[\tilde{\beta}] \succeq Var[\hat{\beta}]$ (au sens des matrices), avec égalité si et seulement si $D = 0$. L'estimateur des moindres carrés est donc BLUE.

Exercice 2. Retour à l'énoncé

Loi du khi-deux. Avec $z = \epsilon/\sigma \sim \mathcal{N}(0, I)$ et $(I-H)X = 0$, on a $SSE = Y^\top(I-H)Y = \epsilon^\top(I-H)\epsilon = \sigma^2 z^\top(I-H)z$. La matrice $I-H$, symétrique et idempotente, est celle d'un projecteur orthogonal de rang $\mathrm{Tr}(I-H) = n-(p+1)$. Par décomposition spectrale, $I-H = P\Lambda P^\top$, avec $n-(p+1)$ valeurs propres égales à 1 et les autres à 0, et $P^\top z \sim \mathcal{N}(0, I)$. La forme $z^\top(I-H)z$ est donc une somme de $n-(p+1)$ carrés de variables $\mathcal{N}(0,1)$ indépendantes, soit une $\chi^2_{n-(p+1)}$, d'où $(n-(p+1))\hat{\sigma}^2/\sigma^2 = SSE/\sigma^2 \sim \chi^2_{n-(p+1)}$.

Indépendance. Les vecteurs $\hat{\beta} = (X^\top X)^{-1}X^\top Y$ et $e = (I-H)Y$ sont deux transformations linéaires du même vecteur gaussien Y, donc conjointement gaussiens. Il suffit alors de montrer qu'ils sont non corrélés pour conclure à l'indépendance. On calcule

$$Cov[\hat{\beta}, e] = (X^\top X)^{-1}X^\top\, \sigma^2 I\, (I-H)^\top = \sigma^2(X^\top X)^{-1}(X^\top - X^\top H).$$

Or $X^\top H = X^\top X(X^\top X)^{-1}X^\top = X^\top$, d'où $Cov[\hat{\beta}, e] = 0$. Comme $\hat{\sigma}^2$ est fonction de e (via $SSE = e^\top e$), on en conclut que $\hat{\sigma}^2$ est indépendant de $\hat{\beta}$.

Exercice 3. Retour à l'énoncé

Rapport de vraisemblance. Pour un modèle linéaire gaussien dont la somme des carrés résiduels est SSE, la log-vraisemblance maximisée en β et en σ^2 s'obtient en remplaçant σ^2 par $\hat{\sigma}^2_{MV} = SSE/n$ (exercice 2 du niveau moyen) :

$$\ell_{\max} = -\frac{n}{2}\log\left(\frac{2\pi\, SSE}{n}\right) - \frac{n}{2}.$$

Le rapport de vraisemblance entre le modèle réduit et le modèle complet vaut donc

$$\Lambda = \frac{L_{\text{réduit}}}{L_{\text{complet}}} = \left(\frac{SSE_{\text{complet}}}{SSE_{\text{réduit}}}\right)^{n/2}, \qquad \text{d'où} \qquad \Lambda^{-2/n} = \frac{SSE_{\text{réduit}}}{SSE_{\text{complet}}} = 1 + \frac{p-q}{n-(p+1)}\, F_{obs}.$$

Rejeter H_0 pour de petites valeurs de Λ revient donc à la rejeter pour de grandes valeurs de F_{obs}, puisque $-2\log\Lambda = n\log\{1 + (p-q)F_{obs}/(n-(p+1))\}$ est une fonction strictement croissante de F_{obs}. Il reste à trouver la loi exacte de F_{obs} sous H_0.

Loi sous H_0. Notons H_c et H_r les matrices chapeaux des modèles complet et réduit, de rangs respectifs $p+1$ et $q+1$. Comme le modèle réduit est emboîté dans le modèle complet, on a $H_r H_c = H_c H_r = H_r$, ce qui rend la matrice $H_c - H_r$ elle-même symétrique et idempotente, de rang $(p+1) - (q+1) = p - q$. Elle est aussi orthogonale à $I - H_c$, c'est-à-dire que $(H_c - H_r)(I - H_c) = 0$.

Sous H_0 (le modèle réduit est adéquat, c'est-à-dire $E[Y] = X_r\beta_r$), on montre comme à l'exercice précédent que

$$\frac{SSE_{\text{réduit}} - SSE_{\text{complet}}}{\sigma^2} = \frac{Y^\top(H_c - H_r)Y}{\sigma^2} \sim \chi^2_{p-q}, \qquad \frac{SSE_{\text{complet}}}{\sigma^2} \sim \chi^2_{n-(p+1)},$$

et que ces deux statistiques sont indépendantes, puisque $H_c - H_r$ et $I - H_c$ sont des projecteurs orthogonaux l'un à l'autre, appliqués au même vecteur gaussien. Par définition, le rapport de deux variables χ^2 indépendantes, chacune divisée par son nombre de degrés de liberté, suit une loi F :

$$F_{obs} = \frac{\{Y^\top(H_c - H_r)Y/\sigma^2\}/(p-q)}{\{SSE_{\text{complet}}/\sigma^2\}/(n-(p+1))} = \frac{(SSE_{\text{réduit}} - SSE_{\text{complet}})/(p-q)}{SSE_{\text{complet}}/(n-(p+1))} \sim F_{p-q,\, n-(p+1)}.$$

Les hypothèses nécessaires sont les suivantes :

— erreurs : indépendantes et identiquement distribuées selon $\mathcal{N}(0, \sigma^2)$, avec la même variance dans les deux modèles ;
— modèles emboîtés : l'espace des moyennes du modèle réduit est inclus dans celui du modèle complet.

Exercice 4. Retour à l'énoncé

Comme $\hat{Y}_k - \hat{Y}_{k(-i)} = X_k^\top(\hat{\beta} - \hat{\beta}_{(-i)})$, la définition se réécrit

$$D_i = \frac{(\hat{\beta} - \hat{\beta}_{(-i)})^\top X^\top X (\hat{\beta} - \hat{\beta}_{(-i)})}{(p+1)\hat{\sigma}^2}.$$

En substituant la formule de mise à jour $\hat{\beta} - \hat{\beta}_{(-i)} = \dfrac{(X^\top X)^{-1} X_i e_i}{1 - h_i}$, on obtient

$$(\hat{\beta}-\hat{\beta}_{(-i)})^\top X^\top X(\hat{\beta}-\hat{\beta}_{(-i)}) = \frac{e_i^2}{(1-h_i)^2}\, X_i^\top (X^\top X)^{-1} X^\top X (X^\top X)^{-1} X_i = \frac{e_i^2}{(1-h_i)^2}\, X_i^\top (X^\top X)^{-1} X_i.$$

Or $X_i^\top (X^\top X)^{-1} X_i = h_i$ est l'élément diagonal i de H, de sorte que l'expression ci-dessus vaut $\dfrac{e_i^2 h_i}{(1-h_i)^2}$, et on obtient

$$D_i = \frac{e_i^2 h_i}{(p+1)\hat{\sigma}^2 (1-h_i)^2} = \frac{e_i^2}{\hat{\sigma}^2(1-h_i)} \cdot \frac{h_i}{(p+1)(1-h_i)} = \frac{r_i^2}{p+1} \cdot \frac{h_i}{1-h_i},$$

en reconnaissant le résidu standardisé $r_i = e_i / \{\hat{\sigma}\sqrt{1-h_i}\}$, dont le carré est $r_i^2 = e_i^2 / \{\hat{\sigma}^2(1-h_i)\}$.

Chapitre 1 : Simulations de Monte Carlo

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

a) Par linéarité de l'espérance, $E[\hat{\theta}_B] = \frac{1}{B}\sum_{b=1}^{B} E[g(Y^{(b)})] = \frac{1}{B} \cdot B\theta = \theta$. Comme les $g(Y^{(b)})$ sont indépendants et de même variance τ^2, la variance de leur somme est la somme des variances :

$$Var[\hat{\theta}_B] = \frac{1}{B^2}\sum_{b=1}^{B} Var[g(Y^{(b)})] = \frac{B\tau^2}{B^2} = \frac{\tau^2}{B}.$$

b) Si g ne prend que les valeurs 0 et 1, alors $g(Y)$ suit une loi de Bernoulli de paramètre $P\{g(Y)=1\} = E[g(Y)] = \theta$, d'où $\tau^2 = \theta(1-\theta)$. Cette fonction de θ est maximale en $\theta = 1/2$ (sa dérivée $1-2\theta$ s'y annule et sa dérivée seconde vaut -2), où elle vaut $1/4$.

Exercice 2. Retour à l'énoncé

a) L'erreur-type vaut $\sqrt{\theta(1-\theta)/B}$. La condition $\sqrt{0{,}95 \times 0{,}05/B} \leq 0{,}005$ équivaut à

$$B \geq \frac{0{,}95 \times 0{,}05}{0{,}005^2} = \frac{0{,}0475}{0{,}000025} = 1\,900.$$

b) Sans information sur θ, on se place dans le pire cas, $\theta(1-\theta) \leq 1/4$ (exercice précédent), ce qui donne $B \geq 0{,}25/0{,}005^2 = 10\,000$. Il faut donc environ cinq fois plus de répétitions, parce qu'une proportion proche de $1/2$ est plus variable qu'une proportion proche de 0 ou de 1.

Exercice 3. Retour à l'énoncé

Sous l'hypothèse $\theta = 0{,}95$, l'erreur-type de Monte Carlo vaut $\sqrt{0{,}95 \times 0{,}05/1\,000} \approx 0{,}0069$, et l'écart observé correspond à

$$z = \frac{0{,}93 - 0{,}95}{0{,}0069} \approx -2{,}90.$$

Comme $|z| > 1{,}96$, on rejette l'hypothèse que la couverture réelle est de 0,95 (la valeur-p bilatérale est d'environ 0,004). L'intervalle est donc trop étroit.

Avec l'erreur-type estimée $\sqrt{0{,}93 \times 0{,}07/1\,000} \approx 0{,}0081$, l'intervalle de confiance approximatif à 95 % pour la couverture réelle est $0{,}93 \pm 1{,}96 \times 0{,}0081 = [0{,}914;\ 0{,}946]$, qui ne contient pas 0,95.

Niveau moyen

Exercice 1. Retour à l'énoncé

a) On écrit $\hat{\beta}^{(b)} - \beta = (\hat{\beta}^{(b)} - \bar{\beta}) + (\bar{\beta} - \beta)$ et on développe le carré :

$$\frac{1}{B}\sum_b \left(\hat{\beta}^{(b)} - \beta\right)^2 = \frac{1}{B}\sum_b \left(\hat{\beta}^{(b)} - \bar{\beta}\right)^2 + \frac{2}{B}\left(\bar{\beta} - \beta\right)\sum_b \left(\hat{\beta}^{(b)} - \bar{\beta}\right) + \left(\bar{\beta} - \beta\right)^2.$$

Le terme croisé est nul, puisque $\sum_b (\hat{\beta}^{(b)} - \bar{\beta}) = B\bar{\beta} - B\bar{\beta} = 0$, d'où le résultat. C'est la version empirique de la décomposition $E[(\hat{\beta} - \beta)^2] = \text{biais}^2 + Var[\hat{\beta}]$.

b) Le biais estimé est $\bar{\beta} - \beta$, où β est une constante connue. Sa variance est donc celle de $\bar{\beta}$, soit $Var[\hat{\beta}]/B$ par l'exercice 1 du niveau facile. On l'estime en remplaçant $Var[\hat{\beta}]$ par la variance empirique des $\hat{\beta}^{(b)}$, ce qui donne l'erreur-type

$$\frac{s_{\hat{\beta}}}{\sqrt{B}}, \qquad s^2_{\hat{\beta}} = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\beta}^{(b)} - \bar{\beta}\right)^2.$$

C'est le calcul fait dans l'exemple du Chapitre 1, où l'écart-type théorique de $\hat{\beta}_1$ était connu.

Exercice 2. Retour à l'énoncé

a) Pour $u \in [0,1]$, comme F_0 est continue et strictement croissante,

$$P(p \le u) = P\{F_0(T) \ge 1-u\} = P\{T \ge F_0^{-1}(1-u)\} = 1 - F_0\{F_0^{-1}(1-u)\} = u,$$

ce qui est la fonction de répartition de la loi uniforme sur $[0,1]$.

b) Le test rejette H_0 au seuil α lorsque $p \le \alpha$, ce qui arrive avec probabilité α sous H_0 d'après a). Le taux de rejet sur B jeux de données simulés sous H_0 est la moyenne de B indicatrices $\mathbf{1}\{p^{(b)} \le \alpha\}$: il estime sans biais $P(p \le \alpha)$, c'est-à-dire le niveau réel. Si le test a le bon niveau, l'erreur-type est $\sqrt{0{,}05 \times 0{,}95 / 2\,000} \approx 0{,}0049$.

c) Si F_0 n'est que la loi asymptotique de T, la loi exacte de T sous H_0 pour le n considéré est une autre fonction F_n, et $P(p \le \alpha) = 1 - F_n\{F_0^{-1}(1-\alpha)\}$, qui diffère de α en général. Le taux de rejet simulé estime alors ce niveau réel, et la simulation sert justement à mesurer l'écart entre le niveau réel et le niveau nominal. C'est la raison d'être de la simulation, comme le rappelle l'introduction du Chapitre 1.

Exercice 3. Retour à l'énoncé

a) On a $\hat{\theta}_A - \hat{\theta}_B = \frac{1}{B}\sum_b (G_A^{(b)} - G_B^{(b)})$, une moyenne de B différences indépendantes et de même loi. Par l'exercice 1 du niveau facile, sa variance vaut $Var[G_A - G_B]/B$, et

$$Var[G_A - G_B] = Var[G_A] + Var[G_B] - 2\,Cov[G_A, G_B].$$

b) Avec des jeux de données indépendants pour chaque méthode, la covariance est nulle. Avec les mêmes jeux de données, les deux intervalles réagissent aux mêmes échantillons. Par exemple, un échantillon dont $\hat{\beta}$ tombe loin de β fait souvent manquer la cible aux deux intervalles, de sorte que $Cov[G_A, G_B] > 0$ et que la variance de la différence diminue. La comparaison est alors plus précise pour le même B : c'est la technique des nombres aléatoires communs. La réinitialisation de la graine avant chaque appel de `cv.glm` applique la même idée à la validation croisée, puisque tous les modèles sont alors évalués sur la même partition en blocs.

Niveau difficile

Exercice 1. Retour à l'énoncé

a) La variable Y ne prend que $n+1$ valeurs, et pour chacune l'intervalle est entièrement déterminé. En notant $IC(y)$ l'intervalle obtenu lorsque $Y = y$, la couverture réelle est

$$C(\pi) = P\{\pi \in IC(Y)\} = \sum_{y=0}^{n} \mathbf{1}\{\pi \in IC(y)\} \binom{n}{y} \pi^y (1-\pi)^{n-y}.$$

C'est une somme finie, qu'on calcule exactement. La simulation n'est utile que lorsque l'espace des données est trop grand pour être parcouru, ce qui est le cas dès qu'une variable est continue ou que les variables explicatives sont nombreuses.

b) Avec $n = 10$, on vérifie pour chaque y si 0,1 est dans l'intervalle :

— $y = 0$: $\hat{\pi} = 0$ et l'intervalle se réduit à $\{0\}$, qui ne contient pas 0,1 ;
— $y = 1$: $0{,}1 \pm 1{,}96\sqrt{0{,}009} = [-0{,}086;\ 0{,}286]$, qui contient 0,1 ;
— $y = 2, 3, 4$: les intervalles $[-0{,}048;\ 0{,}448]$, $[0{,}016;\ 0{,}584]$ et $[0{,}096;\ 0{,}704]$ contiennent 0,1 ;
— $y = 5$: $0{,}5 \pm 0{,}310 = [0{,}190;\ 0{,}810]$ ne contient pas 0,1, et les intervalles des valeurs suivantes, centrés plus haut et plus étroits, ne le contiennent pas non plus.

La couverture vaut donc $P(1 \le Y \le 4) = 1 - P(Y = 0) - P(Y \ge 5)$. Comme $P(Y = 0) = 0{,}9^{10} \approx 0{,}349$ et $P(Y \ge 5) \approx 0{,}002$, on obtient $C(0{,}1) \approx 0{,}650$, très loin des 95 % annoncés. L'essentiel de l'écart vient du cas $y = 0$, qui survient une fois sur trois.

c) La simulation estimerait cette même quantité : $\hat{\theta}_B$ est sans biais pour $C(0{,}1)$, avec une erreur-type d'environ $\sqrt{0{,}650 \times 0{,}350/10\,000} \approx 0{,}0048$. Elle donnerait donc une valeur à moins d'un point de pourcentage de 0,650 dans la plupart des cas, mais rien qu'on ne puisse obtenir exactement ici.

Exercice 2. Retour à l'énoncé

a) Comme $1 - U \sim \mathcal{U}(0, 1)$, on a $E[g(1-U)] = E[g(U)] = \theta$, et les deux estimateurs sont des moyennes de termes d'espérance θ, donc sans biais. Pour l'estimateur usuel, l'exercice 1 du niveau facile donne $Var[g(U)]/(2B)$. Pour l'estimateur antithétique, les B paires sont indépendantes, et

$$Var\left[\frac{g(U) + g(1-U)}{2}\right] = \frac{Var[g(U)] + Var[g(1-U)] + 2\,Cov[g(U), g(1-U)]}{4} = \frac{Var[g(U)] + Cov[g(U)}{2}$$

puisque $Var[g(1-U)] = Var[g(U)]$. En divisant par B, on obtient le résultat.

b) Avec V indépendante de U et de même loi,

$$\frac{1}{2}E[\{a(U)-a(V)\}\{c(U)-c(V)\}] = \frac{1}{2}\{2E[a(U)c(U)] - 2E[a(U)]E[c(V)]\} = Cov[a(U), c(U)],$$

puisque $E[a(U)c(V)] = E[a(U)]E[c(V)]$ par indépendance et que $E[a(V)c(V)] = E[a(U)c(U)]$. Supposons g croissante et posons $a(u) = g(u)$ et $c(u) = g(1-u)$, qui est décroissante. Si $U > V$, alors $a(U) - a(V) \ge 0$ et $c(U) - c(V) \le 0$, et inversement si $U < V$. Le produit est donc toujours négatif ou nul, de même que son espérance. Le cas où g est décroissante est symétrique.

c) Pour $g(u) = e^u$, on a $E[e^U] = e - 1$ et $E[e^{2U}] = (e^2 - 1)/2$, d'où

$$Var[e^U] = \frac{e^2 - 1}{2} - (e-1)^2 \approx 0{,}2420, \qquad Cov[e^U, e^{1-U}] = E[e^U e^{1-U}] - (e-1)^2 = e - (e-1)^2 \approx -0{,}2342.$$

Le facteur de réduction vaut donc

$$\frac{Var[\hat{\theta}_{\text{ind}}]}{Var[\hat{\theta}_{\text{anti}}]} = \frac{0{,}2420}{0{,}2420 - 0{,}2342} \approx 31.$$

À précision égale, l'estimateur antithétique demande donc environ 31 fois moins d'évaluations de g. Le gain vient de ce que e^U est presque linéaire sur $[0,1]$, de sorte que $g(U)$ et $g(1-U)$ sont presque parfaitement corrélés négativement.

Exercices appliqués

Exercice appliqué 1. Retour à l'énoncé

a) On ajuste la droite d'étalonnage :

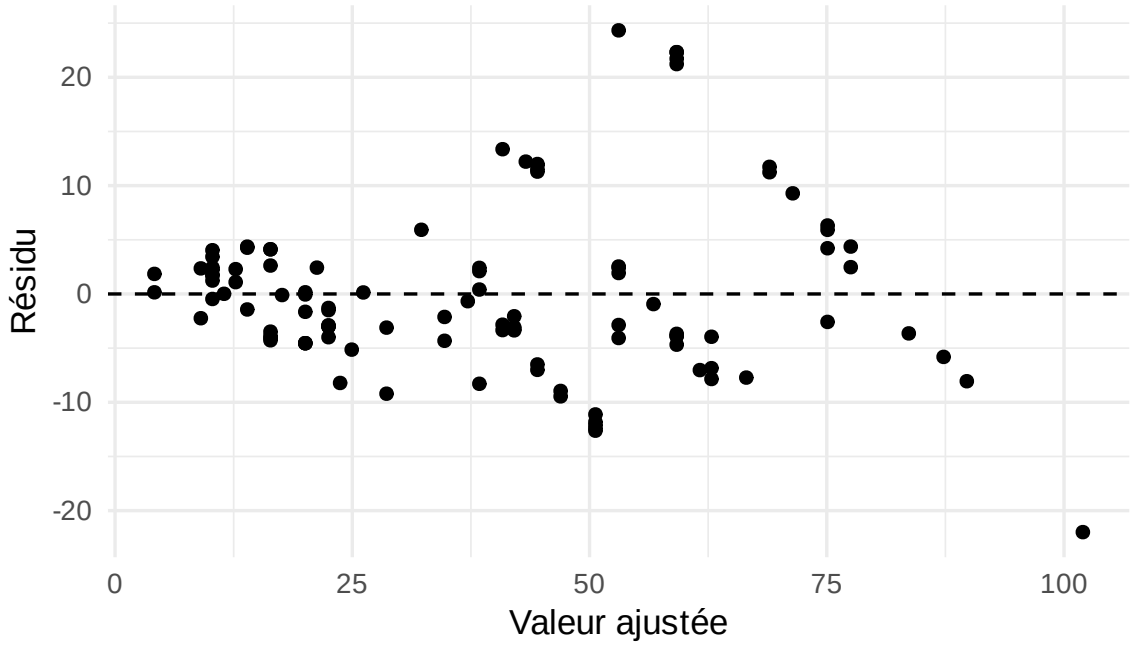
```
library(dplyr)
library(ggplot2)
data(pipeline, package = "faraway")

m.pipe <- lm(Lab ~ Field, data = pipeline)
summary(m.pipe)$coefficients
```

```
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) -1.967500 1.57478705 -1.249375 2.143059e-01
Field        1.222968 0.04106933 29.778123 5.197117e-53
```

La pente estimée vaut 1,223. Comme elle dépasse 1, la mesure sur le terrain sous-estime en moyenne la profondeur mesurée en laboratoire dès que `Field` dépasse environ 8,8, c'est-à-dire pour presque tous les défauts.

On regarde ensuite les résidus, puis leur écart-type dans chaque quart des valeurs de `Field` :

```
diag.pipe <- pipeline |>
  mutate(ajuste = fitted(m.pipe), residu = residuals(m.pipe))

ggplot(diag.pipe, aes(ajuste, residu)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Valeur ajustée", y = "Résidu") +
  theme_minimal(base_size = 13)
```




```
resume.pipe <- diag.pipe |>
  mutate(classe = cut(Field, quantile(Field, 0:4 / 4), include.lowest =
   ↪  TRUE)) |>
  group_by(classe) |>
  summarise(n = n(), Field.moyen = mean(Field), ecart.type = sd(residu),
            rapport = ecart.type / Field.moyen)
resume.pipe
```

```
# A tibble: 4 x 5
  classe        n Field.moyen ecart.type rapport
  <fct>     <int>       <dbl>      <dbl>   <dbl>
1 [5,18]       32        12.8       2.96   0.231
2 (18,35]      24        26.3       4.76   0.181
3 (35,46.5]    24        40.6      10.1    0.248
4 (46.5,85]    27        58.4      11.1    0.190
```

```
gamma.chapeau <- sqrt(mean((diag.pipe$residu / diag.pipe$Field)^2))
gamma.chapeau
```

```
[1] 0.2155968
```

Le nuage des résidus s'ouvre en entonnoir. L'écart-type des résidus passe d'environ 3,0 dans le premier quart des profondeurs à environ 11,1 dans le dernier, de sorte que l'hypothèse d'homoscédasticité n'est pas plausible. Le rapport entre l'écart-type et la profondeur moyenne reste, lui, à peu près stable (de 0,18 à 0,25). C'est ce qu'on attend d'un appareil dont l'erreur est proportionnelle à la grandeur mesurée. Le modèle $\sigma_i = \gamma x_i$ est donc raisonnable, avec $\hat{\gamma} \approx 0{,}216$: l'écart-type de l'erreur vaut environ 22 % de la profondeur mesurée sur le terrain.

b) On suit la recette du Chapitre 1. Le vrai modèle utilise les estimations de a), et les valeurs de `Field` restent celles des données. La fonction suivante prend en argument la fonction qui donne l'écart-type de l'erreur en chaque x, ce qui servira aussi en c) et en d).

```
b.pipe <- coef(m.pipe)   # vraies valeurs de beta_0 et beta_1 dans la
 ↪  simulation
x.pipe <- pipeline$Field
x0 <- c(10, 70)   # mesures sur le terrain pour les deux prédictions

simuler.couverture <- function(x, B, ecart.type) {
  res <- replicate(B, {
    y <- b.pipe[[1]] + b.pipe[[2]] * x + rnorm(length(x), sd = ecart.type(x))
    ajust <- lm(y ~ x)
    ic <- confint(ajust)["x", ]
    ip <- predict(ajust, data.frame(x = x0), interval = "prediction")
    y0 <- b.pipe[[1]] + b.pipe[[2]] * x0 + rnorm(length(x0), sd =
 ↪  ecart.type(x0))
    c(pente   = ic[[1]] <= b.pipe[[2]] & b.pipe[[2]] <= ic[[2]],
      pred.10 = ip[1, "lwr"] <= y0[1] & y0[1] <= ip[1, "upr"],
      pred.70 = ip[2, "lwr"] <= y0[2] & y0[2] <= ip[2, "upr"])
  })
  couv <- rowMeans(res)
  rbind(couverture = couv, erreur.type.MC = sqrt(couv * (1 - couv) / B))
}

B <- 2000
set.seed(4300)
couv.constante <- simuler.couverture(x.pipe, B, function(x)
 ↪  rep(sigma(m.pipe), length(x)))
round(couv.constante, 3)
```

```
               pente pred.10 pred.70
couverture     0.949   0.949   0.943
erreur.type.MC 0.005   0.005   0.005
```

Avec une variance constante et des erreurs normales, l'intervalle de Student pour β_1 et l'intervalle de prédiction ont une couverture exactement égale à 0,95. Les trois couvertures estimées (de 0,943 à 0,949) sont à moins de 2 erreurs-types de Monte Carlo de cette valeur, l'erreur-type étant d'environ 0,005. On commence par ce cas parce que la réponse y est connue : il sert à vérifier le code avant de l'appliquer à une situation où on ne connaît pas la réponse (voir les précautions du Chapitre 1).

c) On remplace l'écart-type constant par $\hat{\gamma} x_i$:

```
set.seed(4300)
couv.hetero <- simuler.couverture(x.pipe, B, function(x) gamma.chapeau * x)
round(couv.hetero, 3)
```

```
                pente pred.10 pred.70
couverture      0.871       1   0.717
erreur.type.MC  0.008       0   0.010
```

La couverture de l'intervalle de β_1 tombe à 0,871, nettement sous 0,95 compte tenu de l'erreur de Monte Carlo (environ 0,008). Les intervalles de prédiction se trompent dans les deux sens. À `Field` $= 10$, la couverture est pratiquement égale à 1 (1,000) : l'intervalle est beaucoup trop large. À `Field` $= 70$, elle n'est que de 0,717. L'intervalle de prédiction utilise le même $\hat{\sigma}$ partout, une sorte de moyenne des variances, trop grande pour les petits défauts et trop petite pour les gros.

Pour l'erreur-type de la pente, on écrit $\hat{\beta}_1 = \sum_i c_i Y_i$ avec $c_i = (x_i - \bar{x})/S_{xx}$ et $S_{xx} = \sum_i (x_i - \bar{x})^2$. Comme les Y_i sont indépendants de variances σ_i^2,

$$Var[\hat{\beta}_1] = \sum_{i=1}^{n} c_i^2 \sigma_i^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sigma_i^2}{S_{xx}^2}.$$

La fonction `lm` estime plutôt $\hat{\sigma}^2 / S_{xx}$. Comme $SSE = Y^\top (I - H) Y$, on a $E[SSE] = \mathrm{tr}\{(I - H)\Sigma\} = \sum_i \sigma_i^2 (1 - h_i)$, où $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$. En remplaçant $\hat{\sigma}^2$ par son espérance $E[SSE]/(n-2)$, on obtient la valeur typique de l'erreur-type calculée par `lm` :

```
sxx <- sum((x.pipe - mean(x.pipe))^2)
sigma.i <- gamma.chapeau * x.pipe
et.vraie <- sqrt(sum((x.pipe - mean(x.pipe))^2 * sigma.i^2)) / sxx
h <- hatvalues(m.pipe)
et.lm <- sqrt(sum(sigma.i^2 * (1 - h)) / (length(x.pipe) - 2) / sxx)
couv.approx <- 2 * pnorm(qt(0.975, length(x.pipe) - 2) * et.lm / et.vraie) -
 ↪  1
c(et.vraie = et.vraie, et.lm = et.lm, couverture.approx = couv.approx)
```

```
  et.vraie       et.lm couverture.approx
0.05560830 0.04303393        0.87508179
```

L'erreur-type calculée par `lm` est trop petite d'un facteur d'environ 1,29. Les observations dont x_i est loin de $\bar{x}$ pèsent le plus dans $\hat{\beta}_1$, et ici ce sont surtout les gros défauts, dont la variance est la plus grande. En traitant l'erreur-type estimée comme fixe, la couverture approximative vaut $2\Phi(t_{0{,}025,\,n-2}\, se_{\text{lm}}/se_{\text{vraie}}) - 1 \approx 0{,}875$, proche de la valeur simulée. Ce calcul confirme le résultat de la simulation par un autre chemin.

d) On répète quatre fois chaque valeur de `Field` :

```
set.seed(4300)
couv.hetero.4n <- simuler.couverture(rep(x.pipe, 4), B, function(x)
 ↪  gamma.chapeau * x)
round(couv.hetero.4n, 3)
```

```
               pente pred.10 pred.70
couverture     0.883       1   0.714
erreur.type.MC 0.007       0   0.010
```

Les couvertures ne s'améliorent pas : elles restent au même niveau, aux erreurs de Monte Carlo près (0,883 pour la pente et 0,714 pour la prédiction à `Field` $= 70$). Le problème n'est pas une approximation asymptotique qui serait mauvaise pour $n = 107$. Le rapport $se_{\text{vraie}}/se_{\text{lm}}$ ne dépend pas de n quand on répète le même plan, et $\hat{\sigma}^2$ converge vers une variance moyenne qui ne correspond à la variance d'aucune observation en particulier.

En pratique, ce sont les défauts profonds qui comptent pour la sécurité du pipeline. Avec le modèle à variance constante, environ 28 % des nouvelles mesures en laboratoire tomberaient hors de l'intervalle de prédiction à 95 % quand la mesure sur le terrain vaut 70, et le risque de sous-estimer la profondeur réelle serait sous-évalué. Il faut modéliser la variance. On pourrait, par exemple, utiliser les moindres carrés pondérés avec des poids $1/x_i^2$ (argument `weights` de `lm`), ou modéliser $\log(\texttt{Lab})$, dont la variance est à peu près constante si l'erreur est proportionnelle. Une erreur-type robuste à l'hétéroscédasticité (estimateur sandwich) corrigerait l'intervalle de la pente, mais pas l'intervalle de prédiction, qui demande un modèle pour la variance d'une nouvelle observation.

Chapitre 2 : Modèles de régression pour données binaires

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

Logit. Si $g^{-1}(\eta) = \dfrac{e^\eta}{1+e^\eta} = \pi$, alors $1-\pi = \dfrac{1}{1+e^\eta}$ et

$$g(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \log(e^\eta) = \eta.$$

Probit. Comme Φ est une bijection strictement croissante de $\mathbb{R}$ vers $(0,1)$, les fonctions $g^{-1}(\eta) = \Phi(\eta)$ et $g(u) = \Phi^{-1}(u)$ sont réciproques l'une de l'autre par construction, ce qui donne directement $g\{g^{-1}(\eta)\} = \Phi^{-1}\{\Phi(\eta)\} = \eta$.

C-log-log. Si $g(u) = \log\{-\log(1-u)\} = \eta$, alors $-\log(1-u) = e^\eta$, donc $1-u = e^{-e^\eta}$ et $u = g^{-1}(\eta) = 1 - e^{-e^\eta} = \pi$. En substituant,

$$g(\pi) = \log\{-\log(1-\pi)\} = \log\{-\log(e^{-e^\eta})\} = \log(e^\eta) = \eta.$$

Exercice 2. Retour à l'énoncé

Comme $o_i = e^{\eta_i}$ et $\pi_i = o_i/(1+o_i)$, les trois cas donnent :

- $\eta_i = 0$: $o_i = e^0 = 1$, donc $\pi_i = 1/2$ (les deux issues sont équiprobables) ;
- $\eta_i = \log 2$: $o_i = e^{\log 2} = 2$, donc $\pi_i = 2/3$;
- $\eta_i = -\log 3$: $o_i = e^{-\log 3} = 1/3$, donc $\pi_i = (1/3)/(4/3) = 1/4$.

Exercice 3. Retour à l'énoncé

Comme $\mu_i = m_i\pi_i$, on a $m_i - \mu_i = m_i(1-\pi_i)$, d'où

$$\frac{\mu_i(m_i-\mu_i)}{m_i} = \frac{m_i\pi_i \cdot m_i(1-\pi_i)}{m_i} = m_i\pi_i(1-\pi_i) = Var[Y_i].$$

À m_i fixé, il s'agit donc de maximiser $\pi_i(1-\pi_i)$. Sa dérivée par rapport à π_i, $\dfrac{d}{d\pi_i}\{\pi_i(1-\pi_i)\} = 1-2\pi_i$, s'annule en $\pi_i = 1/2$, et comme la dérivée seconde vaut $-2 < 0$, ce point est bien un maximum, où $Var[Y_i] = m_i/4$.

Le poids $w_i = m_i\pi_i(1-\pi_i)$ est exactement $Var[Y_i]$: les observations dont la probabilité prédite est proche de $1/2$ (les plus incertaines) reçoivent donc le plus grand poids dans l'algorithme IRLS. À l'inverse, celles dont π_i est proche de 0 ou 1 (presque certaines) reçoivent un poids proche de 0. Cela explique en partie l'instabilité de l'estimation en présence de séparation, où de nombreux π_i s'approchent de 0 ou 1.

Niveau moyen

Exercice 1. Retour à l'énoncé

On dérive terme à terme, pour $j = 0, 1, \dots, p$, avec $\partial \eta_i / \partial \beta_j = x_{ij}$ (et $x_{i0} \equiv 1$). Le terme $\log \binom{m_i}{y_i}$ ne dépend pas de β et sa dérivée est donc nulle, alors que pour $y_i \eta_i$, on a $\partial(y_i \eta_i)/\partial \beta_j = y_i x_{ij}$. Le dernier terme se dérive en chaîne :

$$\frac{\partial}{\partial \beta_j} \log(1 + e^{\eta_i}) = \frac{e^{\eta_i}}{1 + e^{\eta_i}} \cdot x_{ij} = \pi_i x_{ij},$$

donc $\partial\{m_i \log(1 + e^{\eta_i})\}/\partial \beta_j = m_i \pi_i x_{ij}$. En combinant,

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} (y_i - m_i \pi_i) x_{ij} = \sum_{i=1}^{n} (y_i - \mu_i) x_{ij}.$$

Comme x_{ij} est l'élément (i, j) de X, il suffit d'empiler ces $p + 1$ dérivées partielles pour obtenir $U(\beta) = X^\top (y - \mu)$.

Exercice 2. Retour à l'énoncé

La composante j du score s'écrit $U_j(\beta) = \sum_i (y_i - \mu_i) x_{ij}$, où $\mu_i = m_i \pi_i = m_i \, e^{\eta_i}/(1 + e^{\eta_i})$. Comme y_i ne dépend pas de β, seule la dérivée de μ_i intervient :

$$\frac{\partial U_j}{\partial \beta_k} = -\sum_i \frac{\partial \mu_i}{\partial \beta_k} x_{ij} = -\sum_i m_i \frac{\partial \pi_i}{\partial \eta_i} \cdot \frac{\partial \eta_i}{\partial \beta_k} \, x_{ij}.$$

Comme $\pi_i = e^{\eta_i}/(1 + e^{\eta_i})$ est la fonction logistique, sa dérivée vérifie $\partial \pi_i / \partial \eta_i = \pi_i (1 - \pi_i)$ (propriété classique de la fonction logistique). Avec $\partial \eta_i / \partial \beta_k = x_{ik}$, on obtient

$$\frac{\partial U_j}{\partial \beta_k} = -\sum_i m_i \pi_i (1 - \pi_i) \, x_{ij} x_{ik} = -\sum_i w_i x_{ij} x_{ik}.$$

Comme $\partial U_j / \partial \beta_k$ ne dépend pas de y, les informations observée et espérée coïncident, et l'élément (j, k) de l'information de Fisher est $I_{jk} = -E[\partial U_j / \partial \beta_k] = -\partial U_j / \partial \beta_k = \sum_i w_i x_{ij} x_{ik}$, c'est-à-dire exactement l'élément (j, k) de $X^\top W X$, d'où $I(\beta) = X^\top W X$.

Exercice 3. Retour à l'énoncé

Posons $t = \hat{\beta}_0 + \hat{\beta}_1 x$ et $s(t) = e^t/(1 + e^t)$, de sorte que $\hat{\pi}(x) = s(t)$, et vérifions d'abord que $s'(t) = s(t)\{1 - s(t)\}$:

$$s'(t) = \frac{e^t (1 + e^t) - e^t \cdot e^t}{(1 + e^t)^2} = \frac{e^t}{(1 + e^t)^2} = \frac{e^t}{1 + e^t} \cdot \frac{1}{1 + e^t} = s(t)\{1 - s(t)\}.$$

Par dérivation en chaîne, $\hat{\pi}'(x) = s'(t) \cdot \dfrac{dt}{dx} = \hat{\beta}_1\, s(t)\{1 - s(t)\} = \hat{\beta}_1\, \hat{\pi}(x)\{1 - \hat{\pi}(x)\}$.

L'exercice 3 du niveau facile a montré que $u(1-u)$ (pour $u = \hat{\pi}(x)$) est maximal en $u = 1/2$, où il vaut $1/4$. Si $\hat{\beta}_1 > 0$, la pente $\hat{\pi}'(x)$ est donc maximale lorsque $\hat{\pi}(x) = 1/2$, où elle vaut $\hat{\beta}_1/4$. Si $\hat{\beta}_1 < 0$, la courbe est décroissante et la pente est au contraire minimale (la plus négative) en ce point, où elle vaut $\hat{\beta}_1/4$. Dans les deux cas, c'est la valeur absolue $|\hat{\pi}'(x)| = |\hat{\beta}_1|\, \hat{\pi}(x)\{1-\hat{\pi}(x)\}$ qui est maximale en $\hat{\pi}(x) = 1/2$, où elle vaut $|\hat{\beta}_1|/4$.

Niveau difficile

Exercice 1. Retour à l'énoncé

Posons $\delta = y_i - \hat{\mu}_i$, de sorte que $y_i = \hat{\mu}_i + \delta$ et $m_i - y_i = (m_i - \hat{\mu}_i) - \delta$, et utilisons $\log(1+t) \approx t - t^2/2$ pour t petit. Le premier terme de d_i devient :

$$y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) = (\hat{\mu}_i + \delta) \log\left(1 + \frac{\delta}{\hat{\mu}_i}\right) \approx (\hat{\mu}_i + \delta)\left(\frac{\delta}{\hat{\mu}_i} - \frac{\delta^2}{2\hat{\mu}_i^2}\right) \approx \delta + \frac{\delta^2}{2\hat{\mu}_i},$$

en ne conservant que les termes jusqu'à l'ordre δ^2. Le second terme s'approxime de la même façon :

$$(m_i - y_i) \log\left(\frac{m_i - y_i}{m_i - \hat{\mu}_i}\right) = \{(m_i - \hat{\mu}_i) - \delta\} \log\left(1 - \frac{\delta}{m_i - \hat{\mu}_i}\right) \approx -\delta + \frac{\delta^2}{2(m_i - \hat{\mu}_i)}.$$

En additionnant les deux approximations, les termes en δ s'annulent :

$$y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) + (m_i - y_i) \log\left(\frac{m_i - y_i}{m_i - \hat{\mu}_i}\right) \approx \frac{\delta^2}{2}\left(\frac{1}{\hat{\mu}_i} + \frac{1}{m_i - \hat{\mu}_i}\right) = \frac{\delta^2}{2} \cdot \frac{m_i}{\hat{\mu}_i(m_i - \hat{\mu}_i)}.$$

En multipliant par 2 (définition de d_i), on obtient

$$d_i \approx \frac{\delta^2\, m_i}{\hat{\mu}_i(m_i - \hat{\mu}_i)} = \frac{(y_i - \hat{\mu}_i)^2\, m_i}{\hat{\mu}_i(m_i - \hat{\mu}_i)} = r_{P,i}^2.$$

En sommant sur $i = 1, \ldots, n$, on obtient alors $D = \sum_i d_i \approx \sum_i r_{P,i}^2 = X_P^2$. L'approximation devient exacte lorsque $\delta/\hat{\mu}_i$ et $\delta/(m_i - \hat{\mu}_i)$ tendent vers 0, ce qui se produit quand $m_i \to \infty$ à π_i fixé, si le modèle est bien spécifié : $\delta = y_i - \hat{\mu}_i$ est alors d'ordre $\sqrt{m_i}$, alors que $\hat{\mu}_i$ et $m_i - \hat{\mu}_i$ sont d'ordre m_i.

Exercice 2. Retour à l'énoncé

Par définition, $D_{\text{réduit}} = 2\{\ell(y; y) - \ell(\hat{\mu}_r; y)\}$ et $D_{\text{complet}} = 2\{\ell(y; y) - \ell(\hat{\mu}_c; y)\}$, où $\ell(y; y)$ est la log-vraisemblance du modèle saturé. Il ne faut pas voir ce terme comme propre à chaque

modèle : il ne dépend que des données observées y, pas du modèle ajusté, et il est donc le même dans les deux expressions. En soustrayant, on a

$$D_{\text{réduit}} - D_{\text{complet}} = 2\{\ell(y;y) - \ell(\hat{\mu}_r;y)\} - 2\{\ell(y;y) - \ell(\hat{\mu}_c;y)\} = 2\{\ell(\hat{\mu}_c;y) - \ell(\hat{\mu}_r;y)\},$$

ce qui est exactement $-2\{\ell(\text{réduit}) - \ell(\text{complet})\} = G^2$. Le terme $\ell(y;y)$ du modèle saturé s'annule dans la différence parce qu'il est commun aux deux déviances. Il ne dépend en effet que des données, qui sont les mêmes pour deux modèles ajustés sur le même échantillon. C'est pour cette raison que comparer des déviances revient à comparer des vraisemblances, et on n'a donc jamais à évaluer $\ell(y;y)$ explicitement.

Exercice 3. Retour à l'énoncé

Fixons $\beta \in \mathbb{R}$ et regardons le terme $(y_i - \pi_i)x_i$ pour chaque observation i, où $\pi_i = e^{\beta x_i}/(1 + e^{\beta x_i}) \in (0,1)$ pour tout β fini.

- Si $x_i > 0$, alors $y_i = 1$ par hypothèse de séparation, donc $(y_i - \pi_i)x_i = (1 - \pi_i)x_i$. Comme $1 - \pi_i > 0$ et $x_i > 0$, ce terme est strictement positif ;
- Si $x_i < 0$, alors $y_i = 0$, donc $(y_i - \pi_i)x_i = -\pi_i x_i$. Comme $\pi_i > 0$ et $x_i < 0$, on a $-\pi_i x_i > 0$: ce terme est aussi strictement positif.

Chaque terme de la somme $U(\beta) = \sum_i (y_i - \pi_i)x_i$ est donc strictement positif, quel que soit $\beta \in \mathbb{R}$ fini, d'où $U(\beta) > 0$ pour tout β.

Comme $U(\beta) = \ell'(\beta)$, la log-vraisemblance ℓ est strictement croissante sur tout $\mathbb{R}$ et n'a donc aucun point stationnaire ($U(\beta) = 0$ n'a pas de solution finie). On a alors $\sup_{\beta\in\mathbb{R}} \ell(\beta) = \lim_{\beta\to\infty} \ell(\beta)$, et ce supremum n'est jamais atteint pour une valeur finie de β : l'estimateur du maximum de vraisemblance $\hat{\beta}$ n'existe pas au sens usuel. Ce résultat confirme formellement, dans ce cas simple, le phénomène décrit qualitativement à la Section 2.2.2.

Exercice 4. Retour à l'énoncé

Prenons d'abord un $x_i > 0$ fixé : comme $\pi_i = \dfrac{1}{1 + e^{-\hat{\beta}x_i}}$, on a $1 - \pi_i = \dfrac{e^{-\hat{\beta}x_i}}{1 + e^{-\hat{\beta}x_i}} \approx e^{-\hat{\beta}x_i}$ quand $\hat{\beta} \to \infty$, puisque le dénominateur tend vers 1. Comme $\pi_i \to 1$, le poids vérifie

$$w_i = \pi_i(1 - \pi_i) \approx e^{-\hat{\beta}x_i} \longrightarrow 0$$

à un taux exponentiel en $\hat{\beta}$. Un calcul symétrique donne $w_i \approx e^{-\hat{\beta}|x_i|}$ pour $x_i < 0$. Dans les deux cas, w_i décroît donc exponentiellement vite vers 0 quand $\hat{\beta} \to \infty$, c'est-à-dire plus vite que n'importe quelle décroissance polynomiale (en $1/\hat{\beta}$, $1/\hat{\beta}^2$, etc.).

Par conséquent,

$$I(\hat{\beta}) = \sum_{i=1}^{n} w_i x_i^2 \approx \sum_{i=1}^{n} x_i^2 \, e^{-\hat{\beta}|x_i|} \longrightarrow 0$$

également à un taux exponentiel. La statistique de Wald s'écrit

$$z_{obs} = \frac{\hat{\beta}}{\sqrt{\hat{\mathrm{V}}(\hat{\beta})}} = \hat{\beta}\sqrt{I(\hat{\beta})}.$$

Deux vitesses s'affrontent ici : $\hat{\beta}$ croît linéairement, alors que $\sqrt{I(\hat{\beta})}$ décroît exponentiellement, et c'est l'exponentielle qui l'emporte. Pour conclure proprement, notons que $w_i = e^{-\hat{\beta}|x_i|}/(1+e^{-\hat{\beta}|x_i|})^2 \leq e^{-\hat{\beta}|x_i|}$ exactement, pour $x_i > 0$ comme pour $x_i < 0$. Avec $c = \min_i |x_i| > 0$, on a donc $I(\hat{\beta}) \leq e^{-c\hat{\beta}} \sum_i x_i^2$, d'où

$$0 < z_{obs} = \hat{\beta}\sqrt{I(\hat{\beta})} \leq \Big(\sum_{i=1}^{n} x_i^2\Big)^{1/2} \hat{\beta}\, e^{-c\hat{\beta}/2} \longrightarrow 0 \quad \text{quand } \hat{\beta} \to \infty,$$

puisque $t\, e^{-at} \to 0$ pour tout $a > 0$. Par exemple, avec $x = (-2, -1, 1, 2)$, on obtient $z_{obs} \approx 0{,}58$, 0,095 et 0,0013 pour $\hat{\beta} = 5$, 10 et 20. C'est le phénomène de Hauck-Donner : plus l'algorithme pousse $\hat{\beta}$ loin avant de s'arrêter, plus la statistique de Wald z_{obs} s'effondre vers 0, ce qui donne l'illusion d'un effet non significatif.

Exercices appliqués

Exercice appliqué 1. Retour à l'énoncé

a) On construit la réponse et on compare les deux modèles :

```
library(dplyr)
library(GLMsData)
data(nminer)

nminer <- nminer |>
  mutate(Detecte = as.numeric(Minerab > 0))
all(nminer$Detecte == nminer$Miners)
```

```
[1] TRUE
```

```
mod1.nminer <- glm(Detecte ~ Area + Grazed + Shrubs + Timber,
                   data = nminer, family = binomial)
mod2.nminer <- glm(Detecte ~ Eucs + Grazed + Bulokes,
                   data = nminer, family = binomial)
c(modele.1 = AIC(mod1.nminer), modele.2 = AIC(mod2.nminer))
```

```
modele.1 modele.2
46.86053 24.94172
```

La variable construite coïncide avec `Miners`. Le modèle 2 a le plus petit AIC (24,9 contre 46,9), et on le préfère. Les deux modèles ne sont pas emboîtés, puisque chacun contient des variables que l'autre n'a pas (`Eucs` et `Bulokes` d'un côté, `Area`, `Shrubs` et `Timber` de l'autre). On ne peut donc pas les comparer par une différence de déviances, alors que l'AIC s'applique.

b) Les trois méthodes partent du modèle additif à six variables (ou du modèle vide, pour *forward*). L'argument `trace = 0` supprime l'affichage des étapes.

```r
# warning: false masque l'avertissement « fitted probabilities numerically 0
 ↪  or 1 »,
# répété à presque chaque étape (voir c)
complet.nminer <- glm(Detecte ~ Eucs + Area + Grazed + Shrubs + Bulokes +
 ↪  Timber,
                      data = nminer, family = binomial)
vide.nminer <- glm(Detecte ~ 1, data = nminer, family = binomial)

selection.nminer <- list(
  backward = step(complet.nminer, direction = "backward", trace = 0),
  forward  = step(vide.nminer, direction = "forward",
                  scope = formula(complet.nminer), trace = 0),
  stepwise = step(complet.nminer, direction = "both", trace = 0)
)
t(sapply(selection.nminer, function(m)
  c(termes = paste(sort(attr(terms(m), "term.labels")), collapse = " + "),
    AIC = format(round(AIC(m), 2)))))
```

```
         termes                              AIC
backward "Eucs + Grazed + Shrubs + Timber" "19.27"
forward  "Eucs + Grazed + Shrubs + Timber" "19.27"
stepwise "Eucs + Grazed + Shrubs + Timber" "19.27"
```

Les trois méthodes retiennent le même modèle, avec `Eucs`, `Grazed`, `Shrubs` et `Timber`, et un AIC de 19,27. Ce n'est pas garanti en général, puisque chaque méthode ne parcourt qu'un chemin parmi les $2^6 = 64$ sous-modèles possibles.

c) On réajuste le modèle retenu, cette fois en laissant les avertissements visibles :

```
retenu.nminer <- glm(Detecte ~ Eucs + Grazed + Shrubs + Timber,
                     data = nminer, family = binomial)
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(retenu.nminer)$coefficients
```

```
             Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -6.7085742  4.3062196 -1.557880 0.11926166
Eucs         1.4940621  0.8688497  1.719586 0.08550766
Grazed      12.7870136  8.1296038  1.572895 0.11574312
Shrubs      -5.9791510  4.5601279 -1.311181 0.18979678
Timber      -0.5258369  0.3294490 -1.596110 0.11046417
```

```
round(sort(fitted(retenu.nminer)), 4)
```

```
    23     25     24      1     27      9      2     19     22     12     26
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0041 0.0041 0.0757 0.1245
    21      7     14     18     29     28     15      8      3      5     11
0.2406 0.2556 0.2724 0.6192 0.7319 0.7875 0.9318 0.9654 0.9876 0.9998 0.9999
     4      6     31     16     20     13     10     17     30
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
```

R signale des probabilités ajustées numériquement égales à 0 ou à 1. La plus petite vaut environ $8{,}66 \times 10^{-12}$, et 18 des 31 transects ont une probabilité ajustée à moins de 0,001 de 0 ou de 1. Les coefficients sont grands pour des variables de cette échelle (celui de `Grazed` vaut environ 12,8, soit un rapport de cotes de l'ordre de e^{13}), et leurs erreurs-types sont du même ordre que les coefficients eux-mêmes. Ce sont les symptômes de la séparation (Section 2.2.2).

Pour distinguer une séparation d'une quasi-séparation, on resserre le critère de convergence. Sous séparation complète ou quasi complète, la vraisemblance augmente sans fin et les coefficients continuent de croître à chaque itération supplémentaire. Si l'estimateur du maximum de vraisemblance existe, ils ne bougent plus.

```
strict.nminer <- update(retenu.nminer,
                        control = glm.control(epsilon = 1e-14, maxit = 100))
rbind(defaut = coef(retenu.nminer), strict = coef(strict.nminer))
```

```
       (Intercept)     Eucs   Grazed    Shrubs     Timber
defaut   -6.708574 1.494062 12.78701 -5.979151 -0.5258369
strict   -6.708574 1.494062 12.78701 -5.979151 -0.5258369
```

```
c(iterations.defaut = retenu.nminer$iter, iterations.strict =
 ↪  strict.nminer$iter)
```

```
iterations.defaut iterations.strict
                9                10
```

Les coefficients sont identiques et l'algorithme converge en 10 itérations. L'estimateur du maximum de vraisemblance existe donc : il n'y a pas de séparation au sens strict, mais les données en sont proches. Le tableau suivant montre d'où vient ce problème :

```
nminer |>
  mutate(eucalyptus = if_else(Eucs >= 12, "12 ou plus", "moins de 12")) |>
  count(eucalyptus, Detecte)
```

```
   eucalyptus Detecte  n
1  12 ou plus       0  1
2  12 ou plus       1 15
3 moins de 12       0 13
4 moins de 12       1  2
```

Avec au moins 12 eucalyptus, le mineur bruyant est détecté dans presque tous les transects, et avec moins de 12, il est absent de presque tous. `Eucs` sépare donc presque les deux groupes à elle seule, et les autres variables classent les quelques transects restants. Quelques observations mal classées suffisent à garder les coefficients finis.

d) On compare les tests de Wald aux tests du rapport de vraisemblance :

```
rv.nminer <- drop1(retenu.nminer, test = "Chisq")
rv.nminer
```

```
Single term deletions

Model:
Detecte ~ Eucs + Grazed + Shrubs + Timber
       Df Deviance    AIC     LRT  Pr(>Chi)
```

```
<none>       9.272 19.272
Eucs    1   37.161 45.161 27.8892 1.285e-07 ***
Grazed  1   14.986 22.986  5.7137  0.016833 *
Shrubs  1   12.075 20.075  2.8026  0.094111 .
Timber  1   16.865 24.865  7.5926  0.005861 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Aucune valeur-p de Wald n'est sous 5 % (`Eucs` : 0,086 ; `Grazed` : 0,12 ; `Timber` : 0,11). Par contre, le test du rapport de vraisemblance conclut que `Eucs` (valeur-p $\approx 1{,}28 \times 10^{-7}$), `Timber` (valeur-p $\approx$ 0,0059) et `Grazed` (valeur-p $\approx$ 0,017) sont significatives. C'est le phénomène de Hauck-Donner : quand les données sont presque séparées, la log-vraisemblance est plate autour de $\hat{\beta}_j$, l'erreur-type de Wald gonfle et le test perd sa puissance. Le test du rapport de vraisemblance compare directement les vraisemblances avec et sans la variable et n'a pas ce défaut. Ici, il faut donc rapporter ce dernier.

Seule `Shrubs` n'est pas significative au seuil de 5 % (valeur-p $\approx$ 0,094). Elle reste dans le modèle parce que la sélection se fait par l'AIC, et non par un test au seuil de 5 %. Retirer un terme à un paramètre fait baisser l'AIC exactement quand $G^2 < 2$, ce qui équivaut à un test du rapport de vraisemblance au seuil $P(\chi^2_1 > 2) \approx 0{,}157$. Comme la valeur-p de `Shrubs` est sous ce seuil, son retrait ferait augmenter l'AIC (de 19,27 à 20,07).

e) Avec le BIC, la pénalité par paramètre passe de 2 à $\log(31) \approx 3{,}43$:

```r
bic.nminer <- step(complet.nminer, direction = "both", k = log(nrow(nminer)),
 ↪  trace = 0)
attr(terms(bic.nminer), "term.labels")
```

```
[1] "Eucs"   "Grazed" "Timber"
```

```r
rc.nminer <- exp(cbind(RC = coef(bic.nminer), confint(bic.nminer)))
signif(rc.nminer, 3)
```

```
                  RC    2.5 %   97.5 %
(Intercept) 1.76e-03 1.45e-08 2.68e-01
Eucs        2.13e+00 1.34e+00 7.47e+00
Grazed      8.66e+02 2.90e+00 8.03e+08
Timber      8.18e-01 5.65e-01 9.82e-01
```

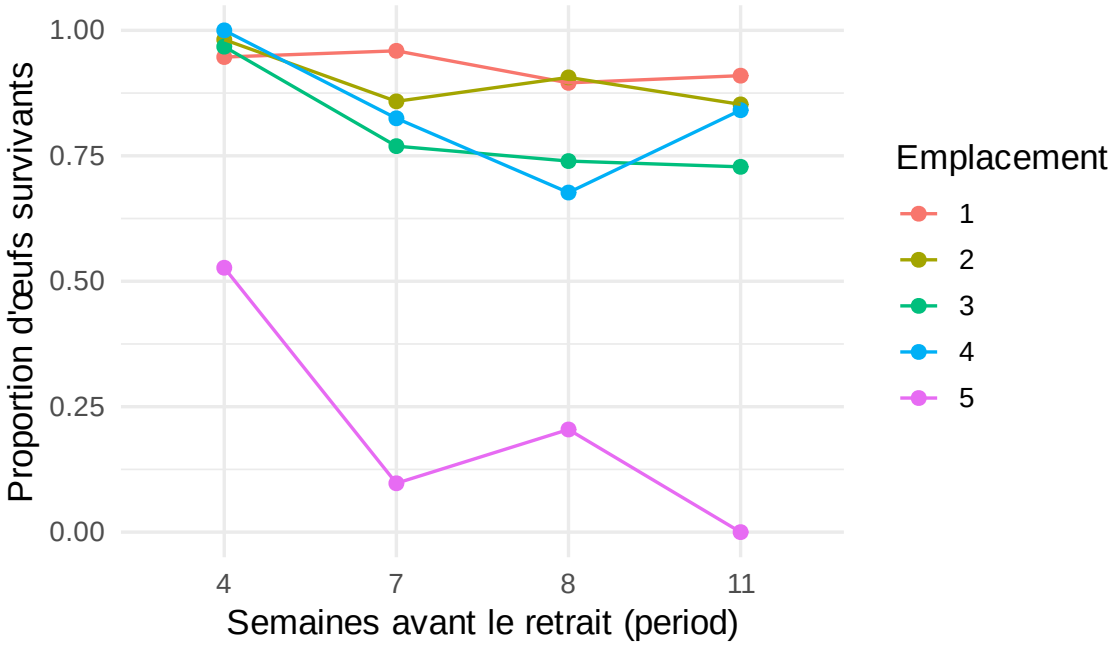
Le BIC écarte `Shrubs`, ce qui est cohérent avec d) : son seuil implicite est d'environ 0,064, et la valeur-p de `Shrubs` le dépasse. Dans ce modèle, chaque eucalyptus supplémentaire dans le transect multiplie la cote de détection du mineur bruyant par environ 2,1, les autres variables restant fixes. L'intervalle par vraisemblance profilée, de 1,3 à 7,5, est très large. Le sens de l'effet est clair, mais sa taille est mal déterminée. Le coefficient de `Grazed` est positif, et comme `Grazed` vaut 1 pour une parcelle non pâturée, la détection est plus probable dans les parcelles non pâturées. Son intervalle va de 2,9 à plus de 10^8 : les données ne disent presque rien de la taille de cet effet.

Ces conclusions reposent sur 31 transects et sur un modèle choisi parmi 64 à l'aide des mêmes données, de sorte que les valeurs-p et les intervalles sont optimistes. Le tableau de c) suggère d'ailleurs un effet de seuil du nombre d'eucalyptus plutôt qu'un effet linéaire sur l'échelle logit, ce qui est l'hypothèse de l'étude d'origine (Maron 2007). Avec des données aussi proches de la séparation, une méthode qui stabilise l'estimation (régression logistique pénalisée de Firth, ou régularisation, Chapitre 6) donnerait des rapports de cotes plus raisonnables.

Exercice appliqué 2. Retour à l'énoncé

a) On calcule les proportions observées et on les représente :

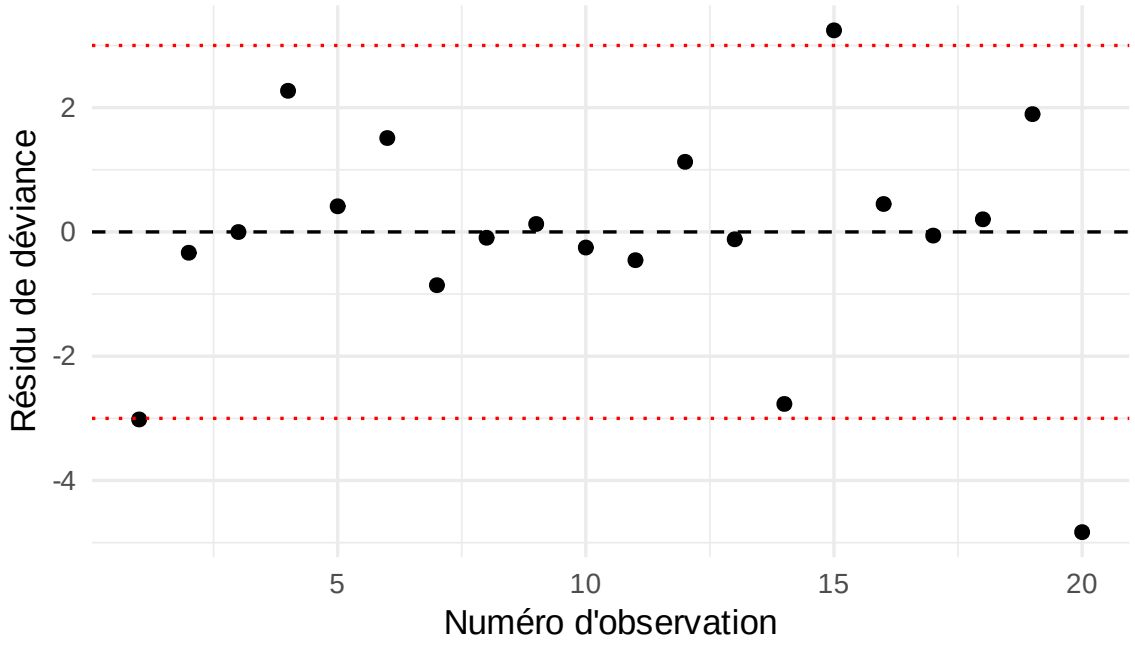
```
library(dplyr)
library(ggplot2)
data(troutegg, package = "faraway")

troutegg <- troutegg |>
  mutate(prop = survive / total)

ggplot(troutegg, aes(period, prop, colour = location, group = location)) +
  geom_point(size = 2) +
  geom_line() +
  labs(x = "Semaines avant le retrait (period)", y = "Proportion d'œufs
   ↪  survivants",
       colour = "Emplacement") +
  theme_minimal(base_size = 13)
```



La survie diminue avec la durée d'enfouissement à tous les emplacements, surtout entre 4 et 7 semaines. L'emplacement 5 se distingue nettement : la proportion de survivants y passe de 0,53 à 4 semaines à 0,00 à 11 semaines (aucun des 138 œufs de la dernière boîte n'a survécu). Les courbes ne sont pas parallèles, même sur l'échelle des proportions, ce qui laisse présager que le modèle additif aura de la difficulté à reproduire toutes les cellules.

b) On ajuste le modèle binomial et on trace les résidus de déviance :

```
mod.troutegg <- glm(cbind(survive, total - survive) ~ location + period,
                    family = binomial, data = troutegg)

res.troutegg <- troutegg |>
  mutate(numero = row_number(),
         residu = residuals(mod.troutegg, type = "deviance"))

ggplot(res.troutegg, aes(numero, residu)) +
  geom_point(size = 2) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_hline(yintercept = c(-3, 3), linetype = "dotted", colour = "red") +
  labs(x = "Numéro d'observation", y = "Résidu de déviance") +
  theme_minimal(base_size = 13)
```




```
res.troutegg |>
  filter(abs(residu) > 2) |>
  dplyr::select(numero, location, period, survive, total, residu)
```

```
   numero location period survive total    residu
1       1        1      4      89    94 -3.017430
4       4        4      4     104   104  2.270686
14     14        4      8      67    99 -2.767887
15     15        5      8      18    88  3.243402
20     20        5     11       0   138 -4.830463
```

```
X2.troutegg <- sum(residuals(mod.troutegg, type = "pearson")^2)
ddl.troutegg <- df.residual(mod.troutegg)
data.frame(
  Statistique = c("Déviance", "Khi-deux de Pearson"),
  Valeur = c(deviance(mod.troutegg), X2.troutegg),
  ddl = ddl.troutegg,
  valeur.p = pchisq(c(deviance(mod.troutegg), X2.troutegg), df =
   ↪  ddl.troutegg,
                    lower.tail = FALSE)
)
```

```
          Statistique   Valeur ddl     valeur.p
1            Déviance 64.49512  12 3.379416e-09
2 Khi-deux de Pearson 63.96386  12 4.235011e-09
```

3 résidus dépassent 3 en valeur absolue (observations 1, 15, 20), sur seulement 20 observations. Comme les m_i vont de 86 à 155 œufs et que les effectifs attendus $m_i\hat{\pi}_i$ et $m_i(1-\hat{\pi}_i)$ sont d'au moins 5 dans 16 des 20 boîtes, l'approximation χ^2 des tests d'adéquation est raisonnable (les exceptions sont des boîtes à 4 semaines, où presque tous les œufs survivent). Les deux tests rejettent nettement l'adéquation du modèle (valeurs-p $\approx 3{,}38 \times 10^{-9}$ pour la déviance et $4{,}24 \times 10^{-9}$ pour le khi-deux de Pearson). La déviance, 64,5, vaut environ 5 fois ses 12 degrés de liberté.

c) On estime ϕ et on ajuste le modèle quasibinomial :

```r
phi.troutegg <- X2.troutegg / ddl.troutegg
phi.troutegg
```

```
[1] 5.330322
```

```r
mod.quasi.troutegg <- glm(cbind(survive, total - survive) ~ location +
 ↪  period,
                          family = quasibinomial, data = troutegg)

coef.bin <- summary(mod.troutegg)$coefficients
coef.quasi <- summary(mod.quasi.troutegg)$coefficients
round(cbind(Estimate          = coef.bin[, "Estimate"],
            "SE binomial"     = coef.bin[, "Std. Error"],
            "SE quasi"        = coef.quasi[, "Std. Error"],
            "Pr(>|z|) binom." = coef.bin[, "Pr(>|z|)"],
            "Pr(>|t|) quasi"  = coef.quasi[, "Pr(>|t|)"]), 4)
```

```
            Estimate SE binomial SE quasi Pr(>|z|) binom. Pr(>|t|) quasi
(Intercept)   4.6358      0.2813   0.6495          0.0000          0.0000
location2    -0.4168      0.2461   0.5682          0.0903          0.4773
location3    -1.2421      0.2194   0.5066          0.0000          0.0305
location4    -0.9509      0.2288   0.5281          0.0000          0.0970
location5    -4.6138      0.2502   0.5777          0.0000          0.0000
period7      -2.1702      0.2384   0.5504          0.0000          0.0020
period8      -2.3256      0.2429   0.5609          0.0000          0.0014
period11     -2.4500      0.2341   0.5405          0.0000          0.0007
```

On obtient $\tilde{\phi}_P \approx 5{,}3$: la variance observée est environ 5,3 fois celle que prévoit le modèle binomial. Les estimations ne changent pas, mais les erreurs-types sont multipliées par $\sqrt{\tilde{\phi}_P} \approx 2{,}31$, et les valeurs-p viennent maintenant d'un test t à 12 degrés de liberté. Une seule conclusion change au seuil de 5 %. `location4` (emplacement 4 contre emplacement 1) passe d'une valeur-p d'environ $3{,}23 \times 10^{-5}$ à 0,097, et n'est plus significatif. `location3` le reste, mais de justesse (valeur-p $\approx 0{,}031$), et `location2`, qui ne l'était déjà pas, l'est encore moins (valeur-p $\approx 0{,}48$).

Pour juger des facteurs en entier, on compare les modèles emboîtés avec le test F du Chapitre 3, qui tient compte de l'estimation de ϕ :

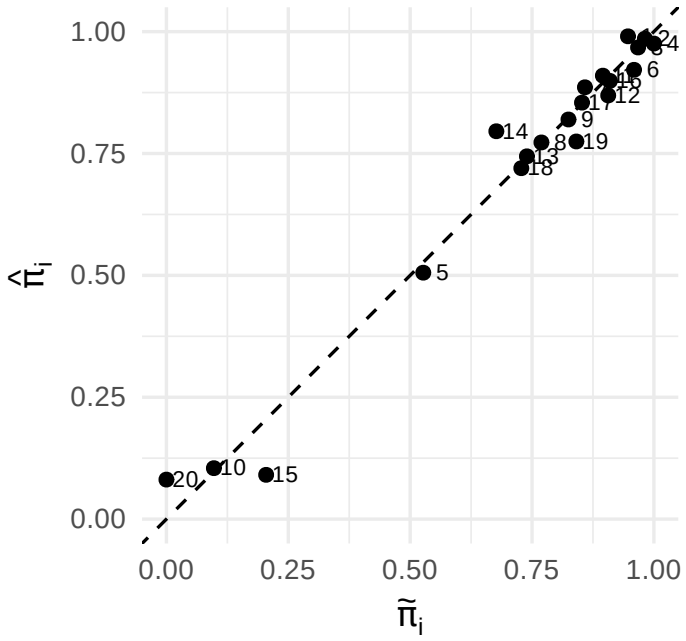
```
drop1(mod.quasi.troutegg, test = "F")
```

```
Single term deletions

Model:
cbind(survive, total - survive) ~ location + period
         Df Deviance F value    Pr(>F)
<none>         64.50
location  4   913.56  39.494 8.142e-07 ***
period    3   228.57  10.176  0.001288 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Les deux facteurs restent nettement significatifs (valeur-p $\approx 8{,}14 \times 10^{-7}$ pour l'emplacement et 0,0013 pour la période). Le modèle binomial exagérait la précision des estimations : la conclusion globale tient, mais certaines comparaisons deux à deux ne sont plus établies.

d) On reprend les diagnostics du chapitre :

```
diag.troutegg <- troutegg |>
  mutate(numero = row_number(),
         pi.chapeau = fitted(mod.quasi.troutegg),
         levier = hatvalues(mod.quasi.troutegg),
         cook = cooks.distance(mod.quasi.troutegg))

ggplot(diag.troutegg, aes(prop, pi.chapeau)) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  geom_point(size = 2) +
  geom_text(aes(label = numero), nudge_x = 0.04, size = 3) +
  coord_equal(xlim = c(0, 1), ylim = c(0, 1)) +
  labs(x = expression(tilde(pi)[i]), y = expression(hat(pi)[i])) +
  theme_minimal(base_size = 13)
```




```
p1.troutegg <- length(coef(mod.quasi.troutegg))
seuil.levier <- 2 * p1.troutegg / nrow(troutegg)
seuil.levier
```

```
[1] 0.8
```

```
diag.troutegg |>
  arrange(desc(cook)) |>
  dplyr::select(numero, location, period, prop, pi.chapeau, levier, cook) |>
  head(5)
```

```
   numero location period      prop pi.chapeau    levier       cook
14     14        4      8 0.6767677 0.79565154 0.4647627 0.32740019
20     20        5     11 0.0000000 0.08106621 0.3786318 0.27996745
19     19        4     11 0.8409091 0.77469241 0.5591774 0.22375827
15     15        5      8 0.2045455 0.09082420 0.2992036 0.19690345
5       5        5      4 0.5268817 0.50550147 0.7826887 0.06609963
```

Les points suivent la droite identité, sans courbure systématique, de sorte que la linéarité sur l'échelle logit n'est pas en cause. Aucun levier ne dépasse le seuil $2(p+1)/n = 0{,}8$ (le plus

grand vaut environ 0,78). Les distances de Cook les plus grandes sont celles des observations 14, 20, 19, 15, toutes à 8 ou à 11 semaines et aux emplacements 4 ou 5, là où l'écart entre les courbes de a) et un modèle additif est le plus visible. L'observation 20 (emplacement 5, 11 semaines) est la plus mal ajustée : aucun œuf n'a survécu, alors que le modèle prédit une probabilité de survie d'environ 0,081.

e) Le paramètre de dispersion représente ici une variabilité de la survie d'une boîte à l'autre qui n'est pas expliquée par l'emplacement et la durée. Le modèle binomial suppose que les œufs d'une même boîte survivent indépendamment les uns des autres, avec la même probabilité. Or les œufs d'une boîte partagent le même microhabitat (sédiments, oxygénation, température de l'eau, éventuelle infection fongique qui se propage dans la boîte). Leurs sorts sont donc positivement corrélés, ce qui gonfle la variance du nombre de survivants au-delà de $m_i\pi_i(1-\pi_i)$. Une variation de π_i d'une boîte à l'autre produit le même effet.

Ce paramètre ne corrige pas tout. Les résidus de d) montrent aussi un défaut de la moyenne : à l'emplacement 5, la survie chute beaucoup plus vite avec la durée qu'ailleurs, ce que le modèle additif sur l'échelle logit ne peut pas reproduire. Une partie de $\tilde{\phi}_P$ vient donc d'une interaction `location:period` absente du modèle. On ne peut cependant pas l'ajouter pour le vérifier, parce que le modèle avec interaction compte 20 paramètres pour 20 observations : il est saturé, sans degré de liberté pour estimer ϕ. Avec une seule boîte par combinaison, les données ne permettent pas de séparer la surdispersion d'une interaction. Seules des répétitions (plusieurs boîtes par emplacement et par période) le permettraient.

Chapitre 3 : Modèles linéaires généralisés (GLM)

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

Écrivons $f(y) = \exp\{-\lambda y + \log\lambda\}$. En posant $\theta = -\lambda$ (donc $\lambda = -\theta$), on a $-\lambda y = y\theta$ et $\log\lambda = \log(-\theta)$, ce qui donne

$$f(y) = \exp\{y\theta - b(\theta)\}, \qquad b(\theta) = -\log(-\theta), \qquad \phi = 1, \qquad a(y,\phi) = 1.$$

Les dérivées valent $b'(\theta) = -1/\theta$ et $b''(\theta) = 1/\theta^2$, et la substitution $\theta = -\lambda$ donne :

$$E[Y] = b'(\theta) = -\frac{1}{-\lambda} = \frac{1}{\lambda}, \qquad Var[Y] = \phi\, b''(\theta) = \frac{1}{(-\lambda)^2} = \frac{1}{\lambda^2},$$

ce qui correspond bien aux moments usuels de la loi exponentielle. Notez que $b(\theta) = -\log(-\theta)$ est la même fonction cumulante que pour la loi gamma, parce que la loi exponentielle est le cas particulier $\alpha = 1$ de la loi gamma.

Exercice 2. Retour à l'énoncé

La dérivée première vaut $b'(\theta) = m \cdot \dfrac{e^\theta}{1+e^\theta} = m\pi = \mu$, et une seconde dérivation (dérivée de la fonction logistique) donne :

$$b''(\theta) = m\pi(1-\pi).$$

En substituant $\pi = \mu/m$ et $1-\pi = (m-\mu)/m$:

$$V(\mu) = b''(\theta) = m \cdot \frac{\mu}{m} \cdot \frac{m-\mu}{m} = \frac{\mu(m-\mu)}{m}.$$

Exercice 3. Retour à l'énoncé

Ici, $b'(\theta) = e^\theta$. En posant $\mu = b'(u) = e^u$, la condition de lien canonique $g\{b'(u)\} = u$ devient $g(e^u) = u$. Comme $\mu = e^u$ équivaut à $u = \log(\mu)$, on obtient $g(\mu) = \log(\mu)$, ce qui est bien le lien canonique annoncé pour la loi de Poisson.

Niveau moyen

Exercice 1. Retour à l'énoncé

On calcule $\dfrac{\partial \log f}{\partial \theta} = \dfrac{y - b'(\theta)}{\phi}$, et la première identité donne

$$E\left[\frac{Y - b'(\theta)}{\phi}\right] = 0 \quad \Longrightarrow \quad E[Y] = b'(\theta).$$

En dérivant une seconde fois, $\dfrac{\partial^2 \log f}{\partial \theta^2} = -\dfrac{b''(\theta)}{\phi}$, donc $-E\left[\dfrac{\partial^2 \log f}{\partial \theta^2}\right] = \dfrac{b''(\theta)}{\phi}$, et la seconde identité donne

$$Var\left[\frac{Y - b'(\theta)}{\phi}\right] = \frac{Var[Y]}{\phi^2} = \frac{b''(\theta)}{\phi} \quad \Longrightarrow \quad Var[Y] = \phi\, b''(\theta).$$

Exercice 2. Retour à l'énoncé

Chaque facteur de la règle en chaîne se calcule comme suit :

- $\dfrac{\partial \ell_i}{\partial \theta_i} = \dfrac{y_i - b'(\theta_i)}{\phi} = \dfrac{y_i - \mu_i}{\phi}$;
- puisque $\mu_i = b'(\theta_i)$, on a $\dfrac{\partial \mu_i}{\partial \theta_i} = b''(\theta_i) = V(\mu_i)$, donc $\dfrac{\partial \theta_i}{\partial \mu_i} = \dfrac{1}{V(\mu_i)}$;

— puisque $g(\mu_i) = \eta_i$, on a $\dfrac{\partial \eta_i}{\partial \mu_i} = g'(\mu_i)$, donc $\dfrac{\partial \mu_i}{\partial \eta_i} = \dfrac{1}{g'(\mu_i)}$;

— $\dfrac{\partial \eta_i}{\partial \beta_j} = x_{ij}$.

En multipliant les quatre facteurs,

$$\frac{\partial \ell_i}{\partial \beta_j} = \frac{y_i - \mu_i}{\phi} \cdot \frac{1}{V(\mu_i)} \cdot \frac{1}{g'(\mu_i)} \cdot x_{ij},$$

et en sommant sur i, on obtient la formule annoncée.

Cas du lien canonique. Le lien canonique satisfait $g = (b')^{-1}$, c'est-à-dire $\theta = g(\mu)$. En dérivant l'identité $b'\{g(\mu)\} = \mu$ par rapport à μ, on obtient $1 = b''\{g(\mu)\}\, g'(\mu) = V(\mu)\, g'(\mu)$, donc $V(\mu)g'(\mu) = 1$. Dans le score général, le facteur $1/\{V(\mu_i)g'(\mu_i)\}$ vaut alors 1 pour chaque i, d'où

$$U_j(\beta) = \frac{1}{\phi}\sum_{i=1}^{n}(y_i - \mu_i)x_{ij} \quad \Longrightarrow \quad U(\beta) = \frac{1}{\phi}X^\top(y - \mu).$$

Ce résultat explique la simplification du score (à ϕ près) pour la régression logistique (Chapitre 2) et la régression de Poisson (Section 3.6), puisque les liens logit et log sont les liens canoniques de leurs lois respectives.

Exercice 3. Retour à l'énoncé

Comme $g(\mu) = -1/\mu = -\mu^{-1}$, la dérivée vaut $g'(\mu) = \mu^{-2} = 1/\mu^2$, et la multiplication par $V(\mu) = \mu^2$ donne :

$$V(\mu)\, g'(\mu) = \mu^2 \cdot \frac{1}{\mu^2} = 1,$$

ce qui est cohérent avec le fait que $g(\mu) = -1/\mu$ est le lien canonique de la loi gamma. La réciproque ne vaut qu'à une constante additive près : $V(\mu)g'(\mu) = 1$ donne $g'(\mu) = 1/V(\mu) = d\theta/d\mu$, donc $g(\mu) = \theta + c$. Le score associé a donc la forme simplifiée $U(\beta) = X^\top(y - \mu)/\phi$.

Niveau difficile

Exercice 1. Retour à l'énoncé

Posons $c_i = 1/\{V(\mu_i)g'(\mu_i)\}$, de sorte que $U_j(\beta) = \frac{1}{\phi}\sum_i (y_i - \mu_i)c_i x_{ij}$. On dérive ensuite par rapport à β_k : comme μ_i (et donc c_i) dépend de β, la règle du produit donne deux termes :

$$\frac{\partial U_j}{\partial \beta_k} = \frac{1}{\phi}\sum_i \left[-\frac{\partial \mu_i}{\partial \beta_k}c_i x_{ij} + (y_i - \mu_i)\frac{\partial c_i}{\partial \beta_k}x_{ij}\right].$$

En prenant l'espérance, le second terme s'annule car $E[y_i - \mu_i] = 0$ (et $\partial c_i / \partial \beta_k$ ne dépend pas de y_i) :

$$E\left[\frac{\partial U_j}{\partial \beta_k}\right] = -\frac{1}{\phi}\sum_i c_i x_{ij}\,\frac{\partial \mu_i}{\partial \beta_k}.$$

Par la règle en chaîne, $\dfrac{\partial \mu_i}{\partial \beta_k} = \dfrac{\partial \mu_i}{\partial \eta_i}\cdot\dfrac{\partial \eta_i}{\partial \beta_k} = \dfrac{x_{ik}}{g'(\mu_i)}$ (comme à l'exercice 2 du niveau moyen). En substituant,

$$E\left[\frac{\partial U_j}{\partial \beta_k}\right] = -\frac{1}{\phi}\sum_i \frac{1}{V(\mu_i)g'(\mu_i)}\cdot\frac{1}{g'(\mu_i)}\,x_{ij}x_{ik} = -\frac{1}{\phi}\sum_i \frac{x_{ij}x_{ik}}{V(\mu_i)[g'(\mu_i)]^2}.$$

En posant $w_i = 1/\{V(\mu_i)[g'(\mu_i)]^2\}$, l'élément (j,k) de l'information de Fisher est

$$I_{jk}(\beta) = -E\left[\frac{\partial U_j}{\partial \beta_k}\right] = \frac{1}{\phi}\sum_i w_i x_{ij} x_{ik},$$

ce qui correspond exactement à l'élément (j,k) de $\frac{1}{\phi}X^\top W X$, où $W = \mathrm{diag}(w_1, \ldots, w_n)$. Ainsi, $I(\beta) = \frac{1}{\phi}X^\top W X$ et la variance asymptotique de $\hat\beta$ est $I(\beta)^{-1} = \phi\,(X^\top W X)^{-1}$. En pratique, on l'estime en évaluant W en $\hat\beta$.

Exercice 2. Retour à l'énoncé

En substituant $\pi = \mu/m$:

$$\begin{aligned}\binom{m}{y}\pi^y(1-\pi)^{m-y} &= \frac{m(m-1)\cdots(m-y+1)}{y!}\left(\frac{\mu}{m}\right)^y\left(1-\frac{\mu}{m}\right)^{m-y}\\ &= \frac{\mu^y}{y!}\cdot\underbrace{\frac{m(m-1)\cdots(m-y+1)}{m^y}}_{(A)}\cdot\underbrace{\left(1-\frac{\mu}{m}\right)^{m-y}}_{(B)}.\end{aligned}$$

Terme (A). Pour y fixé,

$$\frac{m(m-1)\cdots(m-y+1)}{m^y} = \prod_{k=0}^{y-1}\left(1-\frac{k}{m}\right) \longrightarrow 1 \quad \text{quand } m\to\infty,$$

puisque chaque facteur tend vers 1 et qu'il y en a un nombre fini (y facteurs).

Terme (B). Écrivons $(1-\mu/m)^{m-y} = (1-\mu/m)^m\cdot(1-\mu/m)^{-y}$: le second facteur tend vers 1 puisque $\mu/m\to 0$, et le premier est la limite classique

$$\left(1-\frac{\mu}{m}\right)^m \longrightarrow e^{-\mu} \quad \text{quand } m\to\infty.$$

Ainsi, $(B)\to e^{-\mu}$.

En combinant (A) et (B),

$$\binom{m}{y}\pi^y(1-\pi)^{m-y} \longrightarrow \frac{\mu^y}{y!}\cdot 1\cdot e^{-\mu} = \frac{e^{-\mu}\mu^y}{y!},$$

ce qui est bien la fonction de masse de la loi $\mathcal{P}(\mu)$.

Exercice 3. Retour à l'énoncé

Sous H_0, $\hat{\alpha}_{MV} \dot{\sim} \mathcal{N}(0, v)$, et comme la loi normale est symétrique autour de 0, on a $P(\hat{\alpha}_{MV} > 0) = P(\hat{\alpha}_{MV} < 0) = 1/2$ (asymptotiquement).

Cas $\hat{\alpha}_{MV} \leq 0$ (probabilité $\to 1/2$). L'estimateur contraint est alors $\hat{\alpha}^+ = 0$, c'est-à-dire que le maximum de la log-vraisemblance sur $[0, \infty)$ est atteint exactement à la frontière. Dans ce cas, la statistique vaut $Q = 2\{\ell(\hat{\alpha}^+) - \ell(0)\} = 2\{\ell(0) - \ell(0)\} = 0$, de sorte que Q prend la valeur 0 avec probabilité $\to 1/2$: c'est la composante χ^2_0 (loi dégénérée en 0) du mélange.

Cas $\hat{\alpha}_{MV} > 0$ (probabilité $\to 1/2$). La contrainte $\alpha \geq 0$ n'est alors pas active : $\hat{\alpha}^+ = \hat{\alpha}_{MV}$, et $Q = 2\{\ell(\hat{\alpha}_{MV}) - \ell(0)\}$ coïncide avec la statistique non contrainte, qui suit une loi χ^2_1 par le résultat (iii). Il faut en fait cette loi conditionnellement à $\hat{\alpha}_{MV} > 0$, alors que (iii) ne donne que la loi marginale. Asymptotiquement, $2\{\ell(\hat{\alpha}_{MV}) - \ell(0)\} \approx \hat{\alpha}^2_{MV}/v = Z^2$, avec $Z = \hat{\alpha}_{MV}/\sqrt{v} \dot{\sim} \mathcal{N}(0,1)$. Par symétrie de la loi normale, Z^2 suit encore une loi χ^2_1 conditionnellement à $Z > 0$.

En combinant les deux cas, chacun de probabilité asymptotique 1/2, la loi de Q sous H_0 est bien le mélange

$$Q \;\dot{\sim}\; \tfrac{1}{2}\chi^2_0 + \tfrac{1}{2}\chi^2_1,$$

c'est-à-dire une masse ponctuelle de poids 1/2 en 0 et, pour le reste, une loi χ^2_1 de poids 1/2. La valeur-p se calcule donc par $P(\chi^2_1 > Q_{obs})/2$ plutôt que $P(\chi^2_1 > Q_{obs})$, puisque seule la moitié de la probabilité, celle associée au cas où la contrainte n'est pas active, est retenue.

Exercice 4. Retour à l'énoncé

a) L'intervalle pour η^* est $[3 - 1{,}96 \times 2;\ 3 + 1{,}96 \times 2] = [-0{,}92;\ 6{,}92]$, et l'application de $g^{-1}(\eta) = e^\eta/(1+e^\eta)$ à ses deux bornes donne :

$$g^{-1}(-0{,}92) \approx 0{,}285, \qquad g^{-1}(6{,}92) \approx 0{,}999.$$

L'intervalle obtenu pour π^*, $[0{,}285;\ 0{,}999]$, est bien contenu dans $[0,1]$. Ce sera toujours le cas, puisque g^{-1} prend ses valeurs dans $(0,1)$ par construction, quelles que soient les bornes de l'intervalle sur η^*.

b) Pour une transformation h dérivable, la méthode du delta donne $Var[h(\hat{\eta}^*)] \approx [h'(\hat{\eta}^*)]^2\, Var[\hat{\eta}^*]$, d'où, avec $h = g^{-1}$, $\hat{\mathrm{V}}(\hat{\pi}^*) \approx [g^{-1\prime}(\hat{\eta}^*)]^2\, \hat{\mathrm{V}}(\hat{\eta}^*)$. La dérivée vaut $g^{-1\prime}(\eta) = \pi(1-\pi)$ avec $\pi = g^{-1}(\eta)$ (propriété de la fonction logistique, voir l'exercice 3 du niveau moyen du chapitre 2). Comme $\hat{\pi}^* = g^{-1}(3) \approx 0{,}9526$, on a :

$$g^{-1\prime}(3) = 0{,}9526 \times (1 - 0{,}9526) \approx 0{,}0452.$$

La variance estimée est donc $\hat{\mathrm{V}}(\hat{\pi}^*) \approx (0{,}0452)^2 \times 4 \approx 0{,}00816$, et l'erreur-type $\sqrt{\hat{\mathrm{V}}(\hat{\pi}^*)} \approx 0{,}0904$.

L'intervalle de Wald direct sur $\hat{\pi}^*$ est alors

$$[0{,}9526 - 1{,}96 \times 0{,}0904;\ 0{,}9526 + 1{,}96 \times 0{,}0904] \approx [0{,}775;\ 1{,}130].$$

La borne supérieure, 1,130, dépasse 1 : cet intervalle n'est donc pas valide pour une probabilité. C'est pour cette raison qu'on préfère construire l'intervalle sur l'échelle du prédicteur linéaire η^* (sans borne), puis transformer ses bornes par g^{-1}, plutôt que d'appliquer la méthode du delta directement sur $\hat{\pi}^*$. La première approche respecte automatiquement le domaine $(0, 1)$ de π^*, contrairement à la seconde.

Exercices appliqués

Exercice appliqué 1. Retour à l'énoncé

a) On calcule le taux observé pour 1000 naissances :

```
library(dplyr)
library(ggplot2)
library(MASS)
library(GLMsData)
data(downs)

downs <- downs |>
  mutate(age.c = Age - 30,
         taux.1000 = 1000 * DS / Births)
downs |>
  summarise(naissances = sum(Births), cas = sum(DS),
            taux.min = min(taux.1000), taux.max = max(taux.1000))
```

```
  naissances cas  taux.min taux.max
1     354880 519 0.4864799 36.91275
```

Les 354 880 naissances comptent 519 cas de trisomie 21. Le taux observé va d'environ 0,5 cas pour 1000 naissances (mères de 22,5 ans en moyenne) à environ 36,9 pour 1000 (mères de 44,4 ans). Le nombre de cas brut est trompeur, parce que les groupes n'ont pas la même taille : le plus grand compte 24 807 naissances, et celui des mères les plus âgées, 249 seulement. Ce qui varie avec l'âge, c'est le risque par naissance, et le nombre de cas attendu est proportionnel au nombre de naissances, $\mu_i = m_i \lambda_i$, avec m_i le nombre de naissances et λ_i le taux. En prenant le

logarithme, $\log \mu_i = \log m_i + \log \lambda_i$: le terme $\log m_i$ entre dans le prédicteur linéaire avec un coefficient fixé à 1, ce qui en fait une variable offset.

Comme $\lambda_i \leq 0{,}037$ et que les m_i sont grands, on est dans le régime où la loi binomiale $\text{Bin}(m_i, \lambda_i)$ est bien approchée par la loi de Poisson (Section 3.6).

b) On ajuste le modèle linéaire en âge :

```
m1.downs <- glm(DS ~ age.c + offset(log(Births)), family = poisson, data =
 ↪  downs)
summary(m1.downs)$coefficients
```

```
             Estimate  Std. Error   z value     Pr(>|z|)
(Intercept) -6.4382556 0.046631254 -138.0674  0.00000e+00
age.c        0.1371908 0.006463156   21.2266 5.42442e-100
```

```
adequation <- function(m) {
  X2 <- sum(residuals(m, type = "pearson")^2)
  ddl <- df.residual(m)
  c(D = deviance(m), X2 = X2, ddl = ddl,
    p.D = pchisq(deviance(m), ddl, lower.tail = FALSE),
    p.X2 = pchisq(X2, ddl, lower.tail = FALSE))
}
signif(adequation(m1.downs), 3)
```

```
       D       X2      ddl      p.D     p.X2
1.83e+02 2.27e+02 2.80e+01 1.06e-24 4.49e-33
```

```
range(fitted(m1.downs))
```

```
[1]  3.642879 25.863551
```

Les valeurs ajustées $\hat{\mu}_i$ vont de 3,6 à 25,9 cas, de sorte que l'approximation χ^2 de la déviance et de X^2_P est raisonnable. Les deux tests rejettent très nettement le modèle : la déviance, 183,0, est plus de six fois plus grande que ses 28 degrés de liberté (valeur-p $\approx 1{,}06 \times 10^{-24}$). Le graphique de c) montre pourquoi.

c) On ajoute le terme quadratique, puis on compare les deux ajustements aux taux observés :

```r
m2.downs <- glm(DS ~ age.c + I(age.c^2) + offset(log(Births)),
                family = poisson, data = downs)
summary(m2.downs)$coefficients
```

```
                Estimate   Std. Error    z value     Pr(>|z|)
(Intercept) -6.907034413 0.0658247162 -104.93071 0.000000e+00
age.c        0.102825367 0.0057453659   17.89710 1.242323e-71
I(age.c^2)   0.008499131 0.0006704505   12.67675 7.956781e-37
```

```r
anova(m1.downs, m2.downs, test = "Chisq")
```

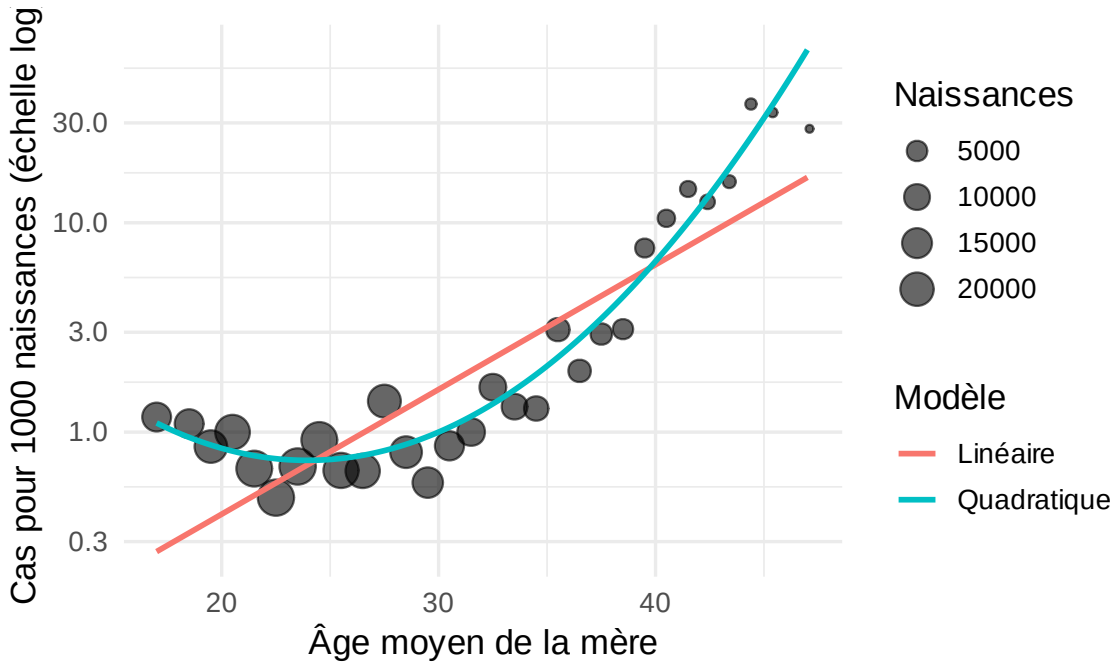
```
Analysis of Deviance Table

Model 1: DS ~ age.c + offset(log(Births))
Model 2: DS ~ age.c + I(age.c^2) + offset(log(Births))
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        28     183.03
2        27      46.05  1   136.98 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
signif(adequation(m2.downs), 3)
```

```
      D      X2     ddl     p.D    p.X2
46.1000 46.1000 27.0000  0.0126  0.0124
```

```r
# taux prédits pour 1000 naissances (Births = 1000 dans l'offset)
grille <- data.frame(Age = seq(17, 47, length.out = 200), Births = 1000) |>
  mutate(age.c = Age - 30)
courbes <- bind_rows(
  grille |> mutate(modele = "Linéaire", taux = predict(m1.downs, grille, type
 ↪  = "response")),
  grille |> mutate(modele = "Quadratique", taux = predict(m2.downs, grille,
 ↪  type = "response"))
)

ggplot(downs, aes(Age, taux.1000)) +
```

```
  geom_point(aes(size = Births), alpha = 0.6) +
  geom_line(data = courbes, aes(y = taux, colour = modele), linewidth = 1) +
  scale_y_log10() +
  labs(x = "Âge moyen de la mère", y = "Cas pour 1000 naissances (échelle
   ↪  log)",
       size = "Naissances", colour = "Modèle") +
  theme_minimal(base_size = 13)
```



Sur l'échelle log, les taux observés sont presque constants jusque vers 30 ans, puis augmentent rapidement. La droite du modèle linéaire ne peut pas suivre cette courbure : elle surestime le taux au milieu de l'étendue des âges et le sous-estime aux deux extrémités. Le terme quadratique est très significatif ($G^2 \approx 137{,}0$ à 1 degré de liberté) et la déviance tombe à 46,1 pour 27 degrés de liberté. La courbe quadratique suit bien les points sur toute l'étendue des âges. Son minimum est atteint vers 24 ans, et le taux estimé remonte légèrement chez les mères les plus jeunes, comme les taux observés.

Les tests d'adéquation rejettent encore le modèle au seuil de 5 % (valeurs-p $\approx 0{,}013$ pour la déviance et 0,012 pour X^2_P), mais l'écart est beaucoup plus modeste.

d) On estime ϕ et on fait le test de surdispersion (Section 3.6.5) :

```
phi.downs <- adeq2[["X2"]] / adeq2[["ddl"]]
phi.downs
```

```
[1] 1.70794
```

```
nb.downs <- glm.nb(DS ~ age.c + I(age.c^2) + offset(log(Births)), data =
 ↪  downs)
nb.downs$theta
```

```
[1] 31.26431
```

```
Q.obs <- max(2 * as.numeric(logLik(nb.downs) - logLik(m2.downs)), 0)
valeur.p <- if (Q.obs > 0) pchisq(Q.obs, df = 1, lower.tail = FALSE) / 2 else
 ↪  1
c(Q.obs = Q.obs, valeur.p = valeur.p)
```

```
     Q.obs   valeur.p
3.61699079 0.02859612
```

On obtient $\tilde{\phi}_P \approx 1{,}71$. Le test du rapport de vraisemblance de $H_0 : \alpha = 0$ donne $Q_{obs} \approx 3{,}62$, et comme $\alpha = 0$ est à la frontière de l'espace des paramètres, la valeur-p est la moitié de celle d'une χ^2_1, soit environ 0,029. On rejette donc l'absence de surdispersion au seuil de 5 %. L'estimation $\hat{\theta} \approx 31{,}3$ correspond à $\hat{\alpha} = 1/\hat{\theta} \approx 0{,}032$, une surdispersion faible. Elle compte surtout pour les groupes où μ_i est grand, puisque $Var[Y_i] = \mu_i(1 + \alpha\mu_i)$.

Notez qu'une surdispersion apparente peut aussi venir d'une moyenne mal spécifiée. Les âges sont des moyennes de groupes, et la forme quadratique n'est qu'une approximation de la vraie courbe. Le modèle quasi-Poisson est un choix prudent dans les deux cas : il garde les mêmes estimations et élargit les intervalles d'un facteur $\sqrt{\tilde{\phi}_P} \approx 1{,}31$.

```
q2.downs <- update(m2.downs, family = quasipoisson)
round(summary(q2.downs)$coefficients, 4)
```

```
            Estimate Std. Error  t value Pr(>|t|)
(Intercept)  -6.9070     0.0860 -80.2908        0
age.c         0.1028     0.0075  13.6945        0
I(age.c^2)    0.0085     0.0009   9.7000        0
```

```
nouv <- data.frame(Age = c(25, 35, 40, 45), Births = 1000) |>
  mutate(age.c = Age - 30)
pred <- predict(q2.downs, nouv, type = "link", se.fit = TRUE)
t.crit <- qt(0.975, df.residual(q2.downs))
nouv |>
  mutate(taux.1000 = exp(pred$fit),
         borne.inf = exp(pred$fit - t.crit * pred$se.fit),
         borne.sup = exp(pred$fit + t.crit * pred$se.fit)) |>
  dplyr::select(Age, taux.1000, borne.inf, borne.sup)   # dplyr:: : MASS a
 ↪  aussi une fonction select
```

```
  Age  taux.1000  borne.inf  borne.sup
1  25  0.7401313  0.6293791  0.8703726
2  35  2.0695392  1.7451669  2.4542022
3  40  6.5462189  5.5913782  7.6641179
4  45 31.6711377 24.2226678 41.4100120
```

L'intervalle est construit sur l'échelle du prédicteur linéaire avec le quantile de Student à 27 degrés de liberté, puis transformé par l'exponentielle, comme au Chapitre 3. Le taux estimé passe d'environ 0,74 cas pour 1000 naissances à 25 ans à 2,1 à 35 ans, 6,5 à 40 ans et 31,7 à 45 ans. Le taux d'une mère de 40 ans est environ 6,5 fois celui d'une mère de 30 ans, dont le taux estimé est d'environ 1,00 cas pour 1000 naissances. Comme le modèle est quadratique, ce rapport de taux dépend des deux âges comparés, et non seulement de leur différence. L'intervalle à 45 ans est le plus large, parce que peu de naissances (et donc peu de cas) viennent des mères de cet âge.

e) On ajuste le modèle binomial correspondant :

```
b2.downs <- glm(cbind(DS, Births - DS) ~ age.c + I(age.c^2),
                family = binomial, data = downs)
round(cbind(Poisson = coef(m2.downs), binomial = coef(b2.downs)), 4)
```

```
            Poisson binomial
(Intercept) -6.9070  -6.9110
age.c        0.1028   0.1033
I(age.c^2)   0.0085   0.0086
```

Les coefficients sont presque identiques. C'est attendu : avec des probabilités au plus de 0,037, la cote $\pi/(1-\pi)$ est pratiquement égale à π, de sorte que le logit et le logarithme du taux coïncident presque, et les deux lois donnent la même vraisemblance à peu de chose près. Les

coefficients du modèle de Poisson s'interprètent directement comme des rapports de taux (ou de risques), ce qui est plus parlant en épidémiologie que des rapports de cotes. Le modèle binomial resterait valide si les probabilités n'étaient pas petites, alors que l'approximation de Poisson ne le serait plus.

Chapitre 4 : Extensions des GLM

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

a) Avec $K = 2$ et $\beta_1 = 0$, on a $\pi_{i2} = e^{X_i^\top \beta_2}/(1 + e^{X_i^\top \beta_2})$ et $\pi_{i1} = 1 - \pi_{i2}$, de sorte que $\text{logit}(\pi_{i2}) = X_i^\top \beta_2$. La log-vraisemblance $\sum_i \{y_{i2} \log \pi_{i2} + (1 - y_{i2}) \log(1 - \pi_{i2})\}$ est celle de la régression logistique de $y_{i2} = \mathbf{1}\{Y_i = 2\}$.

b) On a $\log(\pi_{ik}/\pi_{il}) = \log(\pi_{ik}/\pi_{i1}) - \log(\pi_{il}/\pi_{i1}) = X_i^\top(\beta_k - \beta_l)$. Avec la catégorie l comme référence, les nouveaux coefficients sont donc $\beta_k^{(l)} = \beta_k - \beta_l$, pour tout k (en particulier, $\beta_l^{(l)} = 0$ et $\beta_1^{(l)} = -\beta_l$). Les probabilités ne changent pas, puisqu'on divise numérateur et dénominateur de π_{ik} par le même facteur $e^{X_i^\top \beta_l}$. Le changement de référence est une simple reparamétrisation : la vraisemblance maximale, les probabilités ajustées et les tests globaux sont les mêmes.

Exercice 2. Retour à l'énoncé

a) $P(Y = 0) = \pi + (1 - \pi)e^{-\mu} = 0{,}2 + 0{,}8\, e^{-2} \approx 0{,}2 + 0{,}108 = 0{,}308$, et $E[Y] = (1 - \pi)\mu = 0{,}8 \times 2 = 1{,}6$.

b) Sous une loi de Poisson d'espérance 1,6, $P(Y = 0) = e^{-1{,}6} \approx 0{,}202$. Le modèle ZIP prévoit donc environ 50 % de zéros de plus que la loi de Poisson de même moyenne. Parmi ses zéros, deux sur trois sont structurels (0,2/0,308).

Exercice 3. Retour à l'énoncé

a) Avec $\eta = \beta x$, on a $P(Y \le k) = \text{logit}^{-1}(\alpha_k - \eta)$. Pour $x = 10$, $\eta = 3$, et

$$P(Y \le 1) = \text{logit}^{-1}(-2) \approx 0{,}119, \qquad P(Y \le 2) = \text{logit}^{-1}(0) = 0{,}5,$$

d'où $(\pi_1, \pi_2, \pi_3) \approx (0{,}119;\ 0{,}381;\ 0{,}500)$. Pour $x = 20$, $\eta = 6$, $P(Y \le 1) = \text{logit}^{-1}(-5) \approx 0{,}007$ et $P(Y \le 2) = \text{logit}^{-1}(-3) \approx 0{,}047$, d'où $(\pi_1, \pi_2, \pi_3) \approx (0{,}007;\ 0{,}041;\ 0{,}953)$.

b) La cote de $\{Y > k\}$ vaut $e^{\eta - \alpha_k}$. Pour $k = 1$, elle passe de e^2 à e^5, et pour $k = 2$, de $e^0 = 1$ à e^3. Dans les deux cas, elle est multipliée par $e^3 = e^{\beta \times 10} \approx 20{,}1$, le facteur $e^{\beta \Delta x}$ annoncé par la propriété de cotes proportionnelles.

Niveau moyen

Exercice 1. Retour à l'énoncé

a) Avec $\beta_1 = 0$, $\log \pi_{ik} = X_i^\top \beta_k - \log\left(1 + \sum_{l=2}^{K} e^{X_i^\top \beta_l}\right)$, et le second terme est le même pour toutes les catégories. Comme $\sum_k y_{ik} = 1$, il apparaît une seule fois pour chaque i, et le terme $k = 1$ du premier est nul, d'où le résultat.

b) La dérivée du premier terme par rapport à β_k est $y_{ik} X_i$. Celle du second est

$$\frac{e^{X_i^\top \beta_k} X_i}{1 + \sum_l e^{X_i^\top \beta_l}} = \pi_{ik} X_i,$$

d'où $\partial \ell / \partial \beta_k = \sum_i (y_{ik} - \pi_{ik}) X_i$. C'est exactement le score de la régression logistique (chapitre 2), avec y_{ik} et π_{ik} à la place de y_i et π_i, et une équation par catégorie autre que la référence.

c) La composante du score qui correspond à l'ordonnée à l'origine de β_k est $\sum_i (y_{ik} - \pi_{ik})$. Elle s'annule à l'estimateur, de sorte que $\sum_i \hat{\pi}_{ik} = \sum_i y_{ik}$ pour $k = 2, \dots, K$. Pour $k = 1$, on l'obtient par différence, puisque $\sum_k \hat{\pi}_{ik} = \sum_k y_{ik} = 1$ pour chaque i.

Exercice 2. Retour à l'énoncé

Avec $Y = (1 - S)W$ et S indépendant de W, $P(Y = 0) = P(S = 1) + P(S = 0)P(W = 0) = \pi + (1 - \pi)e^{-\mu}$ et, pour $y \geq 1$, $P(Y = y) = (1 - \pi)P(W = y)$: c'est bien la loi ZIP. Comme $(1 - S)^2 = 1 - S$,

$$E[Y] = E[1 - S]\, E[W] = (1 - \pi)\mu, \qquad E[Y^2] = E[1 - S]\, E[W^2] = (1 - \pi)(\mu + \mu^2).$$

Ainsi,

$$Var[Y] = (1 - \pi)(\mu + \mu^2) - (1 - \pi)^2 \mu^2 = (1 - \pi)\mu \left\{1 + \mu - (1 - \pi)\mu\right\} = (1 - \pi)\mu(1 + \pi\mu).$$

Le rapport vaut $Var[Y]/E[Y] = 1 + \pi\mu$, qui dépasse 1 dès que $\pi > 0$ (et $\mu > 0$). Les zéros structurels créent donc de la surdispersion, d'autant plus forte que μ est grand.

Exercice 3. Retour à l'énoncé

a) Notons $d_i = \mathbf{1}\{y_i > 0\}$. La log-vraisemblance vaut

$$\ell = \sum_{i=1}^{n} \left\{(1 - d_i)\log(1 - p_i) + d_i \log p_i\right\} + \sum_{i:\, d_i = 1} \left\{\log f(y_i; \mu_i) - \log\left(1 - f(0; \mu_i)\right)\right\}.$$

La première somme ne dépend que de γ : c'est la log-vraisemblance de la régression logistique de d_i. La seconde ne dépend que de β : c'est la log-vraisemblance d'une loi tronquée en zéro, calculée sur les seules observations positives. On maximise donc chaque partie séparément.

b) La partie binaire est une régression logistique de d_i. Si z_i contient une ordonnée à l'origine, la composante correspondante du score s'annule à l'estimateur (même argument qu'à l'exercice 1 c)) : $\sum_i (d_i - \hat{p}_i) = 0$. Ainsi, $\sum_i (1 - \hat{p}_i) = \sum_i (1 - d_i)$, qui est le nombre de zéros observé. Le modèle ZIP n'a pas cette propriété, parce que ses zéros dépendent aussi de la partie de comptage.

Niveau difficile

Exercice 1. Retour à l'énoncé

a) $P(Y_i \leq k) = P(x_i^\top \beta + \sigma\epsilon_i \leq \alpha_k) = F\{(\alpha_k - x_i^\top \beta)/\sigma\}$, où F est la fonction de répartition logistique. Cette probabilité ne dépend que des rapports α_k/σ et β/σ, qui ne changent pas si on multiplie tous les paramètres par $c > 0$. Les données ne peuvent donc pas distinguer ces paramétrisations, et on fixe l'échelle de la variable latente par $\sigma = 1$. Les coefficients s'interprètent alors en unités du paramètre d'échelle de la loi logistique standard. Comme l'écart-type de cette loi vaut $\pi/\sqrt{3} \approx 1{,}81$, $\beta_j\sqrt{3}/\pi$ mesure l'effet de x_j en écarts-types de l'erreur latente.

b) Si $x_i^\top \beta$ contient une ordonnée à l'origine β_0, alors $\alpha_k - \beta_0 - \tilde{x}_i^\top \tilde{\beta}$ ne change pas si on ajoute la même constante à β_0 et à tous les seuils. Seules les différences $\alpha_k - \beta_0$ sont identifiables, et on fixe $\beta_0 = 0$ (ou, dans d'autres logiciels, $\alpha_1 = 0$).

c) Pour $K = 3$, $\pi_{i1} = F(\alpha_1 - \eta_i)$ et $\pi_{i3} = 1 - F(\alpha_2 - \eta_i)$, avec $\eta_i = x_i^\top \beta$, de sorte que

$$\log \frac{\pi_{i1}}{\pi_{i3}} = \log F(\alpha_1 - \eta_i) - \log\{1 - F(\alpha_2 - \eta_i)\}.$$

En régression multinomiale, ce logarithme vaut $X_i^\top(\beta_1 - \beta_3)$: il est linéaire en x_i, et une régression logistique ajustée sur les seules observations des catégories 1 et 3 redonne les mêmes coefficients. C'est le sens concret de l'indépendance des alternatives non pertinentes. Pour le modèle ordinal, l'expression ci-dessus n'est pas linéaire en η_i et dépend des deux seuils à la fois. Restreindre les données aux catégories 1 et 3 ne donne donc pas un modèle logistique de mêmes coefficients, et la propriété ne vaut pas. Ce n'est pas un défaut : sur une échelle ordonnée, retirer la catégorie du milieu n'a pas de sens, et la question pertinente est plutôt la fusion de catégories adjacentes.

Exercice 2. Retour à l'énoncé

a) On compare les deux ajustements de la partie binaire :

```
library(pscl)
data(bioChemists, package = "pscl")
m.hurdle.p <- hurdle(art ~ ., data = bioChemists, dist = "poisson")
m.binaire <- glm(I(art > 0) ~ ., data = bioChemists, family = binomial)
round(rbind(hurdle = coef(m.hurdle.p, model = "zero"), glm =
 ↪  coef(m.binaire)), 5)
```

```
       (Intercept) femWomen marMarried     kid5     phd    ment
hurdle      0.2368 -0.25115    0.32623 -0.28525 0.02222 0.08012
glm         0.2368 -0.25115    0.32623 -0.28525 0.02222 0.08012
```

Les coefficients sont identiques : la partie binaire du modèle à barrière est la régression logistique de l'indicatrice de publier au moins un article.

b) La loi de Poisson tronquée en zéro a pour probabilités $f(y;\mu)/(1-e^{-\mu})$, $y \geq 1$:

```
positifs <- filter(bioChemists, art > 0)
X.pos <- model.matrix(art ~ ., positifs)
moins.logvrais <- function(b) {
  mu <- exp(as.vector(X.pos %*% b))
  -sum(dpois(positifs$art, mu, log = TRUE) - log(1 - exp(-mu)))
}
opt <- optim(rep(0, ncol(X.pos)), moins.logvrais, method = "BFGS",
             control = list(maxit = 500, reltol = 1e-12))
round(rbind(optim = opt$par, hurdle = coef(m.hurdle.p, model = "count")), 4)
```

```
       (Intercept) femWomen marMarried    kid5     phd   ment
optim       0.6712  -0.2286     0.0965 -0.1421 -0.0127 0.0187
hurdle      0.6711  -0.2286     0.0965 -0.1422 -0.0127 0.0187
```

Les estimations coïncident aux erreurs d'optimisation numérique près : la partie de comptage n'utilise que les observations positives.

c) On compare le nombre de zéros :

```
c(predits = sum(predict(m.hurdle.p, type = "prob")[, 1]),
  observes = sum(bioChemists$art == 0))
```

```
 predits observes
     275      275
```

Les deux valeurs sont égales, comme le prévoit l'exercice 3 du niveau moyen.

Chapitre 5 : Prédiction, mesures de performance et validation croisée

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

a) Les équations normales (exercice 2 du niveau facile du chapitre 0) imposent $\sum_i e_i = 0$ et $\sum_i x_i e_i = 0$. On a bien $1{,}0 - 1{,}4 + 0{,}8 - 1{,}4 + 1{,}0 = 0$ et $1{,}0 - 2{,}8 + 2{,}4 - 5{,}6 + 5{,}0 = 0$. Avec $\bar{x} = 3$ et $\sum_j (x_j - \bar{x})^2 = 10$, les leviers valent

$$h_i = 0{,}2 + \frac{(i-3)^2}{10}, \qquad \text{soit} \qquad (h_1, \dots, h_5) = (0{,}6;\ 0{,}3;\ 0{,}2;\ 0{,}3;\ 0{,}6).$$

Leur somme vaut $2 = p + 1$, la trace de la matrice chapeau. Les observations aux extrémités du domaine des x_i ont le plus grand levier.

b) L'erreur d'entraînement vaut

$$\frac{SSE}{n} = \frac{1 + 1{,}96 + 0{,}64 + 1{,}96 + 1}{5} = \frac{6{,}56}{5} = 1{,}312.$$

Pour $VC_{(n)}$, on calcule les résidus de validation croisée $e_i/(1-h_i)$: 2,5, -2, 1, -2 et 2,5. Leurs carrés valent 6,25, 4, 1, 4 et 6,25, d'où

$$VC_{(n)} = \frac{21{,}5}{5} = 4{,}3.$$

Le critère est plus de trois fois plus grand que l'erreur d'entraînement. Les observations 1 et 5 n'ont pas les plus grands résidus, mais leur levier élevé multiplie leur résidu par 2,5, et elles fournissent à elles seules 58 % de la somme. Ce sont des points qui « tirent » la droite vers eux, de sorte que leur résidu ordinaire sous-estime l'erreur qu'on ferait en les prédisant sans elles.

Exercice 2. Retour à l'énoncé

a) La matrice de design est la colonne 1_n, de sorte que $X^\top X = n$ et $H = 1_n 1_n^\top / n$, dont tous les éléments diagonaux valent $1/n$.

b) Comme $\bar{Y}_{(-i)} = (n\bar{Y} - Y_i)/(n-1)$,

$$Y_i - \bar{Y}_{(-i)} = \frac{(n-1)Y_i - n\bar{Y} + Y_i}{n-1} = \frac{n}{n-1}(Y_i - \bar{Y}) = \frac{Y_i - \bar{Y}}{1 - 1/n},$$

ce qui est bien $(Y_i - \hat{Y}_i)/(1-h_i)$.

c) En élevant au carré et en faisant la moyenne,

$$VC_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \frac{n^2}{(n-1)^2}(Y_i - \bar{Y})^2 = \frac{n}{(n-1)^2}\sum_{i=1}^{n}(Y_i - \bar{Y})^2 = \frac{n}{n-1}\, s^2.$$

C'est la valeur de référence du modèle nul. Un modèle plus riche n'a d'intérêt pour la prédiction que s'il fait mieux.

Exercice 3. Retour à l'énoncé

a) On écrit $Y - p = (Y - \pi) + (\pi - p)$. Comme $E[Y - \pi] = 0$, le terme croisé a une espérance nulle et

$$E[(Y-p)^2] = E[(Y-\pi)^2] + (\pi - p)^2 = \pi(1-\pi) + (\pi - p)^2.$$

Le premier terme ne dépend pas de p et le second est minimal, et nul, en $p = \pi$. L'espérance du score est donc minimale lorsque la probabilité prédite est la vraie probabilité : le score de Brier est propre (et même strictement propre, puisque le minimum est unique).

b) Si $p > 1/2$, on prédit 1 et on se trompe avec probabilité $P(Y = 0) = 1 - \pi$. Si $p \leq 1/2$, on prédit 0 et on se trompe avec probabilité π. L'espérance de la perte ne dépend donc de p que par le côté de $1/2$ où il tombe. Par exemple, si $\pi = 0{,}8$, toutes les valeurs $p \in (1/2, 1]$ donnent la même perte espérée, 0,2, qu'elles soient bien calibrées ou non. Le minimum est atteint, mais pas seulement en $p = \pi$: la perte n'est pas strictement propre et ne récompense pas une probabilité juste.

Niveau moyen

Exercice 1. Retour à l'énoncé

La prédiction de l'observation i par le modèle ajusté sans elle est $\hat{Y}_{i,(-i)} = X_i^\top \hat{\beta}_{(-i)}$. En substituant la formule de mise à jour,

$$\hat{Y}_{i,(-i)} = X_i^\top \hat{\beta} - \frac{X_i^\top (X^\top X)^{-1} X_i \, e_i}{1 - h_i} = \hat{Y}_i - \frac{h_i e_i}{1 - h_i},$$

puisque $X_i^\top (X^\top X)^{-1} X_i = h_i$. On en déduit

$$Y_i - \hat{Y}_{i,(-i)} = e_i + \frac{h_i e_i}{1 - h_i} = \frac{e_i(1 - h_i) + h_i e_i}{1 - h_i} = \frac{e_i}{1 - h_i}.$$

En faisant la moyenne des carrés, on obtient $VC_{(n)} = \frac{1}{n}\sum_i \{e_i/(1-h_i)\}^2$, calculé à partir d'un seul ajustement.

Exercice 2. Retour à l'énoncé

a) L'espérance vaut $R(p) = -2[\pi \log p + (1-\pi)\log(1-p)]$, dont la dérivée est

$$R'(p) = -2\left[\frac{\pi}{p} - \frac{1-\pi}{1-p}\right] = -2\,\frac{\pi - p}{p(1-p)}.$$

Elle est négative pour $p < \pi$ et positive pour $p > \pi$, de sorte que R est minimale en $p = \pi$. La valeur minimale, $-2[\pi \log \pi + (1-\pi)\log(1-\pi)]$, est deux fois l'entropie de la loi de Bernoulli : c'est la part de la perte qu'aucune prédiction ne peut éliminer.

b) Seuls les termes $-2Y \log \hat{\mu} + 2\hat{\mu}$ dépendent de $\hat{\mu}$. Leur espérance est $R(\hat{\mu}) = -2\mu \log \hat{\mu} + 2\hat{\mu}$, de dérivée $R'(\hat{\mu}) = 2(1 - \mu/\hat{\mu})$. Cette dérivée est négative pour $\hat{\mu} < \mu$ et positive pour $\hat{\mu} > \mu$, d'où un minimum en $\hat{\mu} = \mu$.

Exercice 3. Retour à l'énoncé

a) Comme $0 \le h_i < 1$, on a $1/(1-h_i)^2 \ge 1$, d'où $\{e_i/(1-h_i)\}^2 \ge e_i^2$ pour chaque i, avec égalité si et seulement si $e_i = 0$ ou $h_i = 0$. En faisant la moyenne, $VC_{(n)} \ge SSE/n$.

b) En remplaçant h_i par $(p+1)/n$ dans chaque terme, le dénominateur devient constant et

$$GCV = \frac{1}{n}\sum_i \frac{e_i^2}{\{1-(p+1)/n\}^2} = \frac{SSE/n}{\{1-(p+1)/n\}^2}.$$

Avec $(1-x)^{-2} = 1 + 2x + O(x^2)$ pour x petit, on obtient l'approximation annoncée. Le facteur $1 + 2(p+1)/n$ corrige l'optimisme de l'erreur d'entraînement : chaque paramètre ajusté réduit artificiellement la SSE, et la correction croît avec le nombre de paramètres. On retrouve la forme d'une pénalité de type C_p ou AIC, que l'exercice 2 du niveau difficile justifie.

Niveau difficile

Exercice 1. Retour à l'énoncé

a) Notons $c = 1 - v^\top J^{-1} u$. Le produit vaut

$$(J - uv^\top)\left(J^{-1} + \frac{J^{-1}uv^\top J^{-1}}{c}\right) = I - uv^\top J^{-1} + \frac{uv^\top J^{-1} - u(v^\top J^{-1}u)v^\top J^{-1}}{c}.$$

Comme $v^\top J^{-1} u$ est un scalaire, la fraction vaut $uv^\top J^{-1}(1 - v^\top J^{-1}u)/c = uv^\top J^{-1}$, qui annule le deuxième terme. Il reste I.

b) La i^{e} ligne de $W^{1/2}X$ est $\sqrt{w_i}\,X_i^\top$, de sorte que l'élément (i,i) de H est $\sqrt{w_i}\,X_i^\top J^{-1} X_i \sqrt{w_i} = w_i X_i^\top J^{-1} X_i$.

c) On applique a) avec $u = w_i X_i$ et $v = X_i$, de sorte que $v^\top J^{-1} u = h_i$ par b) :

$$\left(J - w_i X_i X_i^\top\right)^{-1} X_i = J^{-1} X_i + \frac{J^{-1} w_i X_i \left(X_i^\top J^{-1} X_i\right)}{1 - h_i} = J^{-1} X_i \left(1 + \frac{h_i}{1 - h_i}\right) = \frac{J^{-1} X_i}{1 - h_i}.$$

Retirer l'observation i revient à retirer le terme $w_i X_i X_i^\top$ de l'information J, et la formule montre que cette mise à jour se fait sans nouvelle inversion. Dans le modèle linéaire normal ($w_i = 1$), elle permet de retrouver, en tenant compte aussi du terme $-X_i Y_i$ retiré de $X^\top Y$, la formule de mise à jour utilisée à l'exercice 1 du niveau moyen (le calcul complet est fait à l'exercice 5 des exercices théoriques du chapitre 8).

Exercice 2. Retour à l'énoncé

a) L'exercice 3 du niveau moyen du chapitre 0 a établi que $E[SSE] = \sigma^2(n - (p+1))$. En divisant par n, on obtient $E[SSE/n] = \sigma^2\{1 - (p+1)/n\}$.

b) Comme Y_i^0 est indépendante de $\hat{Y}_i$ (qui ne dépend que des Y_j) et que $E[Y_i^0] = E[\hat{Y}_i] = X_i^\top \beta$ (l'estimateur est sans biais),

$$E[(Y_i^0 - \hat{Y}_i)^2] = Var[Y_i^0] + Var[\hat{Y}_i] = \sigma^2 + \sigma^2 h_i,$$

puisque $Var[\hat{Y}] = \sigma^2 H$. En faisant la moyenne et en utilisant $\sum_i h_i = \mathrm{Tr}(H) = p + 1$, on obtient $\sigma^2(1 + (p+1)/n)$.

c) La différence vaut

$$\sigma^2 \left(1 + \frac{p+1}{n}\right) - \sigma^2 \left(1 - \frac{p+1}{n}\right) = \frac{2\sigma^2(p+1)}{n}.$$

L'erreur d'entraînement sous-estime donc l'erreur de prédiction d'une quantité qui croît avec le nombre de paramètres. En lui ajoutant une estimation de cet optimisme, on obtient le critère C_p, dont l'espérance est l'erreur de prédiction si $\hat{\sigma}^2$ est sans biais pour σ^2, par exemple estimé sur le modèle le plus riche qu'on considère. L'exercice 3 du niveau moyen montre que $VC_{(n)}$, ou sa version GCV, applique la même correction à un ordre près, sans avoir à estimer σ^2 séparément : $E[GCV] \approx \sigma^2\{1-(p+1)/n\}\{1+2(p+1)/n\} \approx \sigma^2\{1 + (p+1)/n\}$.

Exercices appliqués

Exercice appliqué 1. Retour à l'énoncé

a) On prépare les deux échantillons, puis on ajuste les quatre modèles sur l'échantillon d'entraînement :

```r
library(dplyr)
library(MASS)
library(boot)

pima.ent  <- Pima.tr |> mutate(diabete = as.numeric(type == "Yes"))
pima.test <- Pima.te |> mutate(diabete = as.numeric(type == "Yes"))
c(entrainement = mean(pima.ent$diabete), test = mean(pima.test$diabete))
```

```
entrainement         test
   0.3400000    0.3283133
```

```r
m1 <- glm(diabete ~ glu, family = binomial, data = pima.ent)
m2 <- glm(diabete ~ glu + bmi + ped + age, family = binomial, data =
 ↪  pima.ent)
m3 <- glm(diabete ~ npreg + glu + bp + skin + bmi + ped + age,
          family = binomial, data = pima.ent)
m4 <- glm(diabete ~ (npreg + glu + bp + skin + bmi + ped + age)^2,
          family = binomial, data = pima.ent)
modeles <- list(M1 = m1, M2 = m2, M3 = m3, M4 = m4)

# fonctions de perte du @sec-cv : y observé, p probabilité prédite
cout.classif <- function(y, p) mean(y != (p > 0.5))
cout.brier   <- function(y, p) mean((y - p)^2)
cout.dev     <- function(y, p) {
  p <- pmin(pmax(p, 1e-8), 1 - 1e-8)   # évite log(0)
  -2 * mean(y * log(p) + (1 - y) * log(1 - p))
}
couts <- list(classif = cout.classif, brier = cout.brier, deviance =
 ↪  cout.dev)

entrainement <- t(sapply(modeles, function(m)
  c(parametres = length(coef(m)), AIC = AIC(m),
    deviance.moy = cout.dev(pima.ent$diabete, fitted(m)))))
round(entrainement, 3)
```

```
   parametres     AIC deviance.moy
M1          2 211.373        1.037
M2          5 191.082        0.905
M3          8 194.391        0.892
M4         29 212.813        0.774
```

La proportion de femmes diabétiques est d'environ 0,34 dans l'échantillon d'entraînement et 0,33 dans l'échantillon test. Les modèles sont emboîtés, de sorte que la déviance d'entraînement par observation diminue toujours avec le nombre de paramètres, de 1,037 pour `M1` à 0,774 pour `M4`. Elle mesure l'adéquation aux 200 femmes de l'échantillon, pas la qualité des prédictions pour de nouvelles patientes. L'AIC, qui pénalise le nombre de paramètres, est minimal pour `M2` (191,1) et nettement plus grand pour `M4` (212,8).

b) On calcule $VC_{(10)}$ pour chaque modèle et chaque perte, avec la même partition :

```
# warning: false masque l'avertissement de séparation de M4 dans certains
 ↪ blocs
vc10 <- sapply(couts, function(cout)
  sapply(modeles, function(m) {
    set.seed(4300)   # même partition en blocs pour tous les modèles
    cv.glm(pima.ent, m, cost = cout, K = 10)$delta[1]
  }))
round(vc10, 4)
```

```
   classif  brier deviance
M1   0.255 0.1768   1.0617
M2   0.245 0.1662   0.9888
M3   0.265 0.1665   0.9869
M4   0.285 0.2117   1.3311
```

Avec cette partition, les pertes ne désignent pas toutes le même modèle. La déviance retient `M3`, de très peu devant `M2` (0,9869 contre 0,9888), alors que le score de Brier retient `M2`. `M2` et `M3` sont donc pratiquement à égalité, et c) vérifie si ce classement dépend de la partition. `M1` est nettement moins bon : la glycémie seule ne suffit pas. `M4` est le pire des quatre selon les trois pertes, alors qu'il avait la plus petite erreur d'entraînement. Ses 29 paramètres pour 200 observations (dont 68 cas de diabète seulement) mènent au surajustement. Des probabilités prédites très proches de 0 ou de 1 pour des femmes du bloc test qui ont l'autre statut font exploser la déviance (voir la remarque « Quelle perte choisir ? » du Chapitre 5).

Le taux de mauvaise classification retient lui aussi `M2`, mais il est trop grossier pour départager des modèles aussi proches. Sur 200 observations, un écart de 0,005 correspond à une seule femme mal classée de plus ou de moins, et ce critère ignore la qualité des probabilités prédites.

c) On répète le calcul avec 20 partitions différentes :

```
vc.graines <- sapply(1:20, function(graine)
  sapply(modeles, function(m) {
    set.seed(graine)
    cv.glm(pima.ent, m, cost = cout.dev, K = 10)$delta[1]
  }))
round(apply(vc.graines, 1, range), 3)
```

```
        M1    M2    M3    M4
[1,] 1.050 0.943 0.964 1.303
[2,] 1.085 0.983 1.038 1.565
```

```
table(factor(rownames(vc.graines)[apply(vc.graines, 2, which.min)],
             levels = names(modeles)))
```

```
M1 M2 M3 M4
 0 19  1  0
```

La valeur de $VC_{(10)}$ d'un même modèle varie d'une partition à l'autre (de 0,943 à 0,983 pour `M2`, par exemple). `M2` est retenu dans 19 partitions sur 20, et `M3` dans 1. Le choix est donc assez stable. Comme tous les modèles sont évalués sur la même partition pour une graine donnée, une partie de la variabilité se compense dans la comparaison : c'est l'intérêt de fixer la graine avant chaque appel. Avec seulement 200 observations, il vaut quand même mieux regarder plusieurs partitions avant de conclure, surtout quand deux modèles sont proches comme `M2` et `M3`.

d) On évalue les modèles ajustés sur l'échantillon d'entraînement sur les 332 femmes de l'échantillon test :

```
erreur.test <- sapply(couts, function(cout)
  sapply(modeles, function(m)
    cout(pima.test$diabete, predict(m, newdata = pima.test, type =
     ↪  "response"))))
round(erreur.test, 4)
```

```
   classif  brier deviance
M1  0.2259 0.1603   0.9875
M2  0.2078 0.1417   0.8968
M3  0.1988 0.1393   0.8814
M4  0.2470 0.1728   1.2532
```

```r
round(cbind(VC10 = vc10[, "deviance"], test = erreur.test[, "deviance"]), 3)
```

```
    VC10  test
M1 1.062 0.988
M2 0.989 0.897
M3 0.987 0.881
M4 1.331 1.253
```

Sur l'échantillon test, la déviance est la plus petite pour `M3`, de très peu devant `M2` (0,881 contre 0,897). Le classement est le même que celui de la validation croisée pour les modèles éloignés (`M1` et surtout `M4` sont nettement moins bons), mais `M2` et `M3` sont pratiquement à égalité, ce que la validation croisée laissait déjà voir. Le choix de `M2`, retenu avec la plupart des partitions en c), est donc défendable : il fait presque aussi bien que `M3` avec trois variables de moins (le nombre de grossesses, la pression diastolique et le pli cutané).

Les erreurs test sont ici plus petites que les erreurs de validation croisée, pour les quatre modèles. Ce n'est pas une contradiction : l'échantillon test est un autre échantillon aléatoire, et son erreur dépend aussi des femmes qui le composent. On ne peut pas en conclure que la validation croisée est pessimiste en général. Notez que le minimum de $VC_{(10)}$ est en principe optimiste pour le modèle retenu, puisque le modèle a été choisi parce qu'il minimise ce critère ; seul l'échantillon test, qui n'a servi à aucun choix, donne une estimation sans biais de l'erreur du modèle retenu (voir la remarque sur la fuite d'information au Chapitre 5).

En pratique, `M2` dit que la probabilité de diabète augmente avec la glycémie, l'indice de masse corporelle, l'âge et la fonction de prédisposition familiale (`ped`). Pour le dépistage, on choisirait plutôt un seuil u inférieur à $1/2$, parce qu'un diabète non détecté coûte plus cher qu'un test de confirmation inutile (voir Section 2.5).

Chapitre 6 : Sélection de variables et régularisation

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

En développant, $Q_\lambda^{(R)}(\beta) = Y^\top Y - 2\beta^\top X^\top Y + \beta^\top (X^\top X + \lambda I_p)\beta$, dont le gradient est

$$\frac{\partial Q_\lambda^{(R)}}{\partial \beta} = -2X^\top Y + 2(X^\top X + \lambda I_p)\beta.$$

Il s'annule en $\hat{\beta}_\lambda^{(R)} = (X^\top X + \lambda I_p)^{-1} X^\top Y$, qui est bien un minimum puisque la matrice hessienne, $2(X^\top X + \lambda I_p)$, est définie positive.

Pour tout $v \neq 0$, on a $v^\top (X^\top X + \lambda I_p) v = \|Xv\|^2 + \lambda \|v\|^2 \geq \lambda \|v\|^2 > 0$. La matrice est donc définie positive, donc inversible, quel que soit le rang de X. Lorsque $p > n$, $X^\top X$ est singulière et les moindres carrés n'ont pas de solution unique, mais ridge en a une.

Exercice 2. Retour à l'énoncé

Avec $X^\top X = I_p$, l'estimateur des moindres carrés est $\hat{\beta} = X^\top Y$, et

$$\hat{\beta}_\lambda^{(R)} = (I_p + \lambda I_p)^{-1} X^\top Y = \frac{1}{1+\lambda}\, \hat{\beta}.$$

Dans ce cas, ridge multiplie chaque coefficient des moindres carrés par le même facteur $1/(1+\lambda) \in (0, 1]$: il rétrécit tous les coefficients proportionnellement vers 0, sans en annuler aucun. Avec des variables corrélées, le rétrécissement n'est plus uniforme (voir l'exercice 1 du niveau difficile).

Exercice 3. Retour à l'énoncé

a) En remplaçant x_i par $c x_i$,

$$\tilde{\beta}_\lambda^{(R)} = \frac{c \sum_i x_i Y_i}{c^2 \sum_i x_i^2 + \lambda} = \frac{1}{c} \cdot \frac{\sum_i x_i Y_i}{\sum_i x_i^2 + \lambda / c^2}.$$

C'est l'estimateur ridge sur l'échelle d'origine, divisé par c, mais avec le paramètre de pénalité λ/c^2 plutôt que λ. Donc $\tilde{\beta}_\lambda^{(R)} \neq \hat{\beta}_\lambda^{(R)}/c$ dès que $\lambda > 0$.

b) Pour $\lambda = 0$, on a exactement $\tilde{\beta} = \hat{\beta}/c$: les moindres carrés sont équivariants par changement d'échelle, et la prédiction $\tilde{\beta}\,\tilde{x} = \hat{\beta} x$ ne dépend pas de l'unité. Ridge perd cette propriété. Avec c grand (des millimètres plutôt que des mètres), la pénalité effective λ/c^2 devient négligeable et la variable n'est presque pas pénalisée. Avec c petit, elle est fortement rétrécie. Pour que la pénalité traite toutes les variables de la même façon, on les ramène à la même échelle en les standardisant, ce que fait `glmnet` par défaut.

Niveau moyen

Exercice 1. Retour à l'énoncé

En écrivant $Y - X\beta = (Y - X\hat{\beta}) + X(\hat{\beta} - \beta)$, le terme croisé vaut $2(\hat{\beta} - \beta)^\top X^\top (Y - X\hat{\beta}) = 0$ par les équations normales, et $\|X(\hat{\beta} - \beta)\|^2 = \|\hat{\beta} - \beta\|^2$ puisque $X^\top X = I_p$. Le critère se sépare donc en une constante et une somme de p problèmes indépendants :

$$Q_\lambda^{(L)}(\beta) = \|Y - X\hat{\beta}\|^2 + \sum_{j=1}^{p} \left\{ (\beta_j - \hat{\beta}_j)^2 + \lambda |\beta_j| \right\}.$$

Fixons j et supposons $\hat{\beta}_j > 0$ (le cas négatif est symétrique). Un $\beta_j < 0$ ne peut pas être optimal, puisque $-\beta_j$ donne la même pénalité et un premier terme plus petit. Sur $\beta_j \geq 0$, la fonction $(\beta_j - \hat{\beta}_j)^2 + \lambda\beta_j$ est une parabole de minimum en $\beta_j = \hat{\beta}_j - \lambda/2$. Si $\hat{\beta}_j > \lambda/2$, ce point est admissible et c'est la solution. Sinon, la parabole est croissante sur $[0, \infty)$ et le minimum est en $\beta_j = 0$. Dans les deux cas, $\hat{\beta}_{j,\lambda}^{(L)} = (\hat{\beta}_j - \lambda/2)_+$, ce qui donne la formule annoncée.

Ridge multiplie chaque $\hat{\beta}_j$ par $1/(1+\lambda)$, ce qui ne donne jamais exactement 0 si $\hat{\beta}_j \neq 0$. Le lasso soustrait une quantité fixe $\lambda/2$ et met à 0 tous les coefficients dont $|\hat{\beta}_j| \leq \lambda/2$: les petits coefficients sont éliminés, ce qui est une sélection de variables. En contrepartie, les grands coefficients sont tous réduits de $\lambda/2$, ce qui introduit un biais même pour les variables importantes.

Exercice 2. Retour à l'énoncé

a) L'espérance est $\beta_j/(1+\lambda)$, d'où un biais de $\beta_j/(1+\lambda) - \beta_j = -\lambda\beta_j/(1+\lambda)$. La variance est $\sigma^2/(1+\lambda)^2$. Ainsi,

$$EQM(\lambda) = \frac{\lambda^2\beta_j^2 + \sigma^2}{(1+\lambda)^2}.$$

b) La dérivée vaut

$$EQM'(\lambda) = \frac{2\lambda\beta_j^2(1+\lambda) - 2(\lambda^2\beta_j^2 + \sigma^2)}{(1+\lambda)^3} = \frac{2(\lambda\beta_j^2 - \sigma^2)}{(1+\lambda)^3},$$

négative pour $\lambda < \sigma^2/\beta_j^2$ et positive au-delà : le minimum est en $\lambda^* = \sigma^2/\beta_j^2$. En substituant,

$$EQM(\lambda^*) = \frac{\sigma^4/\beta_j^2 + \sigma^2}{(1+\sigma^2/\beta_j^2)^2} = \frac{\sigma^2}{1+\sigma^2/\beta_j^2} = \frac{\sigma^2\beta_j^2}{\beta_j^2+\sigma^2} < \sigma^2.$$

Comme $EQM'(0) = -2\sigma^2 < 0$, toute petite pénalité améliore d'ailleurs l'erreur quadratique moyenne des moindres carrés.

c) La pénalité optimale est grande lorsque le coefficient est petit par rapport au bruit ($\beta_j^2 \ll \sigma^2$) : il vaut alors mieux rétrécir fortement un estimateur peu informatif. Elle est petite lorsque le signal domine. Mais λ^* dépend de β_j et de σ^2, qui sont inconnus, et il diffère d'un coefficient à l'autre, alors que ridge utilise un seul λ. C'est pourquoi on choisit λ par validation croisée.

Exercice 3. Retour à l'énoncé

La somme des carrés sur les données augmentées se sépare en deux blocs :

$$\|\tilde{Y} - \tilde{X}\beta\|^2 = \|Y - X\beta\|^2 + \|0_p - \sqrt{\lambda}\,E\beta\|^2 = \|Y - X\beta\|^2 + \lambda\sum_{j=1}^{p}\beta_j^2,$$

puisque $E\beta = (\beta_1, \dots, \beta_p)^\top$. C'est exactement $Q_\lambda^{(R)}(\beta)$. On peut aussi le vérifier sur les équations normales, puisque $\tilde{X}^\top \tilde{X} = X^\top X + \lambda E^\top E = X^\top X + \lambda \tilde{I}_{p+1}$ et $\tilde{X}^\top \tilde{Y} = X^\top Y$.

Chaque observation ajoutée « observe » une réponse nulle pour un coefficient β_j, avec un poids qui croît avec λ. Ridge revient donc à ajouter aux données p pseudo-observations qui tirent chaque coefficient vers 0. Ce point de vue permet d'ajuster ridge avec n'importe quel logiciel de moindres carrés, et il annonce l'interprétation bayésienne de la pénalité (exercice 4 des exercices théoriques du chapitre 9).

Niveau difficile

Exercice 1. Retour à l'énoncé

a) On a $X^\top X = VD^2V^\top$ et $X^\top X + \lambda I_p = V(D^2 + \lambda I_p)V^\top$, dont l'inverse est $V(D^2 + \lambda I_p)^{-1}V^\top$ puisque V est orthogonale. Ainsi,

$$\hat{\beta}_\lambda^{(R)} = V(D^2 + \lambda I_p)^{-1}V^\top VDU^\top Y = V(D^2 + \lambda I_p)^{-1}DU^\top Y,$$

dont l'écriture colonne par colonne donne la première formule. En multipliant par $X = UDV^\top$, on obtient $X\hat{\beta}_\lambda^{(R)} = UD^2(D^2 + \lambda I_p)^{-1}U^\top Y$, soit la seconde.

b) La matrice chapeau est $H_\lambda = \sum_j \frac{d_j^2}{d_j^2+\lambda} u_j u_j^\top$, et comme $\mathrm{Tr}(u_j u_j^\top) = u_j^\top u_j = 1$, sa trace vaut $\sum_j d_j^2/(d_j^2 + \lambda)$. Chaque terme décroît strictement en λ, vaut 1 en $\lambda = 0$ et tend vers 0 quand $\lambda \to \infty$. On retrouve p paramètres pour les moindres carrés et aucun à la limite.

c) Comme les v_j sont orthonormés,

$$\|\hat{\beta}_\lambda^{(R)}\|^2 = \sum_{j=1}^{p} \frac{d_j^2\,(u_j^\top Y)^2}{(d_j^2 + \lambda)^2},$$

une somme de termes positifs dont chacun décroît en λ.

d) Ridge exprime les moindres carrés dans la base des composantes principales u_j de X et multiplie la j^{e} composante par $d_j^2/(d_j^2 + \lambda)$. Les directions de petite valeur singulière d_j, celles où les variables explicatives varient peu, sont les plus rétrécies. Ce sont exactement les directions où l'estimateur des moindres carrés est le plus variable, puisque $Var[\hat{\beta}] = \sigma^2 VD^{-2}V^\top$. Avec des variables fortement corrélées, comme les statistiques de carrière de `Hitters`, la direction qui oppose deux variables presque égales (par exemple `CHits` moins `CRuns`) a un d_j très petit. Les moindres carrés y produisent des coefficients de signes opposés qui se compensent, et ridge les écrase, ce qui répartit l'effet entre les variables corrélées.

Exercice 2. Retour à l'énoncé

a) Le score de la régression logistique est $X^\top(y-\mu)$ et sa dérivée est $-X^\top W X$. Comme la dérivée de $\lambda\beta^\top \tilde{I}_{p+1}\beta$ est $2\lambda\tilde{I}_{p+1}\beta$,

$$\nabla Q_\lambda^{(R)}(\beta) = -2X^\top(y-\mu) + 2\lambda\tilde{I}_{p+1}\beta, \qquad \nabla^2 Q_\lambda^{(R)}(\beta) = 2\left(X^\top W X + \lambda\tilde{I}_{p+1}\right).$$

b) Notons $M = X^\top W^{(t)} X + \lambda\tilde{I}_{p+1}$. L'itération de Newton s'écrit

$$\beta^{(t+1)} = \beta^{(t)} - \left[\nabla^2 Q_\lambda^{(R)}\right]^{-1} \nabla Q_\lambda^{(R)} = \beta^{(t)} + M^{-1}\left\{X^\top(y-\mu^{(t)}) - \lambda\tilde{I}_{p+1}\beta^{(t)}\right\}.$$

En écrivant $\beta^{(t)} = M^{-1}M\beta^{(t)}$, les termes en λ s'annulent :

$$\beta^{(t+1)} = M^{-1}\left\{X^\top W^{(t)} X\beta^{(t)} + X^\top(y-\mu^{(t)})\right\} = M^{-1}X^\top W^{(t)}\left\{X\beta^{(t)} + (W^{(t)})^{-1}(y-\mu^{(t)})\right\},$$

ce qui est la formule annoncée. Comme le lien logit est canonique, la hessienne ne dépend pas de y, de sorte que Newton-Raphson et Fisher scoring coïncident.

Exercice 3. Retour à l'énoncé

a) Le critère ne dépend de (β_1, β_2) que par $(\beta_1+\beta_2)x_1$ dans l'ajustement et par $\beta_1^2+\beta_2^2$ dans la pénalité, deux quantités symétriques en β_1 et β_2. Si $\hat{\beta}_\lambda^{(R)}$ est le minimum, le vecteur obtenu en permutant ses deux premières composantes donne la même valeur du critère, donc est aussi un minimum. Comme le minimum est unique (exercice 1 du niveau facile), les deux vecteurs coïncident, d'où $\hat{\beta}_{1,\lambda}^{(R)} = \hat{\beta}_{2,\lambda}^{(R)}$. Autrement dit, ridge partage l'effet également entre les deux copies. C'est d'ailleurs ce qui minimise $\beta_1^2+\beta_2^2$ à somme fixée.

b) Le terme d'ajustement ne dépend de (β_1, β_2) que par leur somme, qui reste s. Pour la pénalité, $|a| + |s-a| = |s|$ lorsque a est entre 0 et s, puisque a et $s-a$ ont alors le signe de s, alors que $|\hat{\beta}_1| + |\hat{\beta}_2| \geq |\hat{\beta}_1 + \hat{\beta}_2| = |s|$ par l'inégalité du triangle. Le nouveau vecteur donne donc une valeur du critère inférieure ou égale à celle du minimum, ce qui en fait aussi un minimum. Le lasso a une infinité de solutions, dont $(s, 0)$ et $(0, s)$, où une seule des deux copies est retenue.

c) Lorsque deux variables sont presque identiques, la situation est proche du cas b). Le lasso tend à en garder une et à annuler l'autre, et le choix dépend de petites différences dans les données. C'est ce qui se produit avec `CHits` et `CRuns`, presque parfaitement corrélées : le lasso annule `CHits` et garde `CRuns`. Ridge, comme en a), répartit l'effet entre les deux. Cette instabilité du lasso face aux variables corrélées a motivé le filet élastique (*elastic net*), qui combine les deux pénalités (argument `alpha` de `glmnet` strictement entre 0 et 1).

Exercices appliqués

Exercice appliqué 1. Retour à l'énoncé

a) On commence par repérer les valeurs impossibles :

```
library(dplyr)
library(ggplot2)
library(glmnet)
data(fat, package = "faraway")

fat |>
  mutate(id = row_number()) |>
  filter(height < 60 | brozek <= 0) |>
  dplyr::select(id, brozek, age, weight, height, abdom)
```

```
     id brozek age weight height abdom
42   42   31.7  44  205.0   29.5 104.3
182 182    0.0  40  118.5   68.0  69.4
```

L'homme 42 mesurerait 29,5 pouces (75 cm) pour 205 livres, ce qui est une erreur de saisie, et l'homme 182 aurait 0 % de gras, ce qui est physiquement impossible. On retire ces deux observations. On retire aussi `siri`, `density` et `free`, qui sont calculées à partir de la pesée hydrostatique (la mesure qu'on cherche justement à éviter), ainsi que l'indice d'adiposité `adipos`, calculé à partir du poids et de la taille.

```
gras <- fat |>
  filter(height > 60, brozek > 0) |>
  dplyr::select(-siri, -density, -free, -adipos)
dim(gras)
```

```
[1] 250  14
```

```
round(cor(gras[, c("weight", "neck", "chest", "abdom", "hip", "thigh")]), 2)
```

```
        weight neck chest abdom  hip thigh
weight    1.00 0.83  0.89  0.89 0.94  0.87
neck      0.83 1.00  0.78  0.76 0.74  0.70
chest     0.89 0.78  1.00  0.91 0.83  0.73
abdom     0.89 0.76  0.91  1.00 0.87  0.76
hip       0.94 0.74  0.83  0.87 1.00  0.89
thigh     0.87 0.70  0.73  0.76 0.89  1.00
```

```r
mc.gras <- lm(brozek ~ ., data = gras)
round(summary(mc.gras)$coefficients, 3)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -13.855     20.771  -0.667    0.505
age            0.056      0.030   1.853    0.065
weight        -0.078      0.058  -1.344    0.180
height        -0.057      0.166  -0.341    0.734
neck          -0.428      0.219  -1.957    0.052
chest         -0.037      0.097  -0.380    0.704
abdom          0.890      0.084  10.647    0.000
hip           -0.201      0.136  -1.477    0.141
thigh          0.215      0.137   1.576    0.116
knee          -0.021      0.230  -0.091    0.928
ankle          0.154      0.206   0.749    0.455
biceps         0.168      0.160   1.053    0.293
forearm        0.415      0.185   2.246    0.026
wrist         -1.490      0.496  -3.005    0.003
```

Le poids et les circonférences du tronc et des cuisses sont fortement corrélés entre eux (toutes les corrélations du tableau dépassent 0,70), parce qu'ils augmentent tous avec la corpulence. Cette colinéarité gonfle les erreurs-types des moindres carrés. Seuls 3 coefficients sont significatifs au seuil de 5 % (ceux de `abdom`, `forearm` et `wrist`), alors que le poids, le tour de poitrine et le tour de hanches, très liés au pourcentage de gras pris un à un, ne le sont pas. Plusieurs signes sont aussi contre-intuitifs : les coefficients du poids, du cou et des hanches sont négatifs. Une partie s'interprète (à tour de taille fixé, un homme plus lourd a plus de muscle), mais avec des variables aussi corrélées, les coefficients individuels sont instables. C'est la situation où la régularisation est utile.

b) On sépare les données, puis on choisit λ par validation croisée sur l'échantillon d'entraînement seulement, avec les mêmes blocs pour ridge et le lasso :

```r
x <- model.matrix(brozek ~ ., gras)[, -1]
y <- gras$brozek

set.seed(4300)
entrainement <- sample(nrow(x), 166)
test <- -entrainement
y.test <- y[test]
eqm <- function(prediction) mean((y.test - prediction)^2)
```

```
grille.lambda <- 10^seq(2, -4, length.out = 100)
set.seed(4300)
blocs <- sample(rep(1:10, length.out = length(entrainement)))
cv.ridge <- cv.glmnet(x[entrainement, ], y[entrainement], alpha = 0,
                      lambda = grille.lambda, foldid = blocs)
cv.lasso <- cv.glmnet(x[entrainement, ], y[entrainement], alpha = 1,
                      lambda = grille.lambda, foldid = blocs)

mc.ent <- lm(brozek ~ ., data = gras[entrainement, ])
erreurs <- c(
  "Modèle nul"        = eqm(mean(y[entrainement])),
  "Moindres carrés"   = eqm(predict(mc.ent, gras[test, ])),
  "Ridge, lambda.min" = eqm(predict(cv.ridge, newx = x[test, ], s =
   ↪  "lambda.min")),
  "Ridge, lambda.1se" = eqm(predict(cv.ridge, newx = x[test, ], s =
   ↪  "lambda.1se")),
  "Lasso, lambda.min" = eqm(predict(cv.lasso, newx = x[test, ], s =
   ↪  "lambda.min")),
  "Lasso, lambda.1se" = eqm(predict(cv.lasso, newx = x[test, ], s =
   ↪  "lambda.1se"))
)
round(erreurs, 2)
```

```
       Modèle nul   Moindres carrés Ridge, lambda.min Ridge, lambda.1se
            61.12             20.03             20.02             19.08
Lasso, lambda.min Lasso, lambda.1se
            20.01             17.82
```

```
data.frame(
  methode = c("Ridge", "Lasso"),
  lambda.min.au.bord = c(cv.ridge$lambda.min, cv.lasso$lambda.min) ==
   ↪  min(grille.lambda),
  non.nuls.1se = c(cv.ridge$nzero[cv.ridge$lambda == cv.ridge$lambda.1se],
                   cv.lasso$nzero[cv.lasso$lambda == cv.lasso$lambda.1se])
)
```

```
  methode lambda.min.au.bord non.nuls.1se
1   Ridge               TRUE           13
2   Lasso               TRUE            5
```

Avec `lambda.min`, ridge et le lasso donnent pratiquement la même erreur test que les moindres carrés (environ 20,0, en points de pourcentage de gras au carré). C'est cohérent avec la validation croisée : pour les deux méthodes, `lambda.min` est la plus petite valeur de la grille, de sorte que l'erreur de validation croisée diminue encore quand $\lambda \to 0$. Prolonger la grille ne changerait rien, puisque les deux estimateurs tendent alors vers les moindres carrés. Avec 166 observations pour 13 variables, la variance des moindres carrés n'est pas assez grande pour qu'une pénalité améliore nettement la prédiction, malgré la colinéarité.

Avec `lambda.1se`, le lasso ne garde que 5 variables et son erreur test est un peu plus faible (17,8). Cet écart n'est pas significatif : comme les deux méthodes sont évaluées sur les mêmes 84 hommes, on peut calculer l'erreur-type de la différence des erreurs quadratiques, qui vaut environ 1,6 pour un écart de 2,2. Toutes les méthodes font beaucoup mieux que le modèle nul (61,1) : la racine de l'erreur quadratique moyenne passe d'environ 7,8 à 4,5 points de pourcentage.

c) Les coefficients du lasso retenus avec `lambda.1se` :

```
coef.lasso <- as.matrix(coef(cv.lasso, s = "lambda.1se"))[, 1]
round(coef.lasso[coef.lasso != 0], 3)
```

```
(Intercept)         age      height       abdom     forearm       wrist
      6.166       0.062      -0.352       0.583       0.243      -1.430
```

Le lasso retient `age`, `height`, `abdom`, `forearm` et `wrist`. Le tour de taille porte l'essentiel de l'information : chaque centimètre de plus augmente le pourcentage de gras estimé d'environ 0,58 point, les autres variables restant fixes. Le tour de poignet, qui mesure l'ossature plutôt que le gras, a un coefficient négatif : à tour de taille égal, un homme à l'ossature plus forte a moins de gras. Le poids, le tour de poitrine et le tour de hanches sont écartés, parce que l'information qu'ils apportent est déjà presque entièrement contenue dans le tour de taille. Ce n'est pas la preuve qu'ils n'ont pas d'effet : avec des variables aussi corrélées, le lasso choisit l'une d'elles un peu arbitrairement, et ridge, lui, répartit l'effet entre elles (voir le Chapitre 6). Comme la sélection a été faite par le lasso, les erreurs-types usuelles ne s'appliquent pas aux coefficients retenus (remarque « Inférence » du chapitre).

Pour l'application, c'est une bonne nouvelle : un ruban à mesurer, une balance et l'âge suffisent pour une estimation dont l'erreur typique (racine de l'erreur quadratique moyenne test) est d'environ 4,2 points de pourcentage, comparable à celle du modèle complet.

d) On reprend les mêmes étapes avec toutes les interactions d'ordre 2 :

```
x2 <- model.matrix(brozek ~ .^2, gras)[, -1]
ncol(x2)
```

```
[1] 91
```

```
mc2.ent <- lm(y[entrainement] ~ x2[entrainement, ])
cv.ridge2 <- cv.glmnet(x2[entrainement, ], y[entrainement], alpha = 0,
                       lambda = grille.lambda, foldid = blocs)
cv.lasso2 <- cv.glmnet(x2[entrainement, ], y[entrainement], alpha = 1,
                       lambda = grille.lambda, foldid = blocs)

erreurs2 <- c(
  "Moindres carrés"   = eqm(cbind(1, x2[test, ]) %*% coef(mc2.ent)),
  "Ridge, lambda.min" = eqm(predict(cv.ridge2, newx = x2[test, ], s =
  ↪  "lambda.min")),
  "Lasso, lambda.min" = eqm(predict(cv.lasso2, newx = x2[test, ], s =
  ↪  "lambda.min")),
  "Lasso, lambda.1se" = eqm(predict(cv.lasso2, newx = x2[test, ], s =
  ↪  "lambda.1se"))
)
round(erreurs2, 2)
```

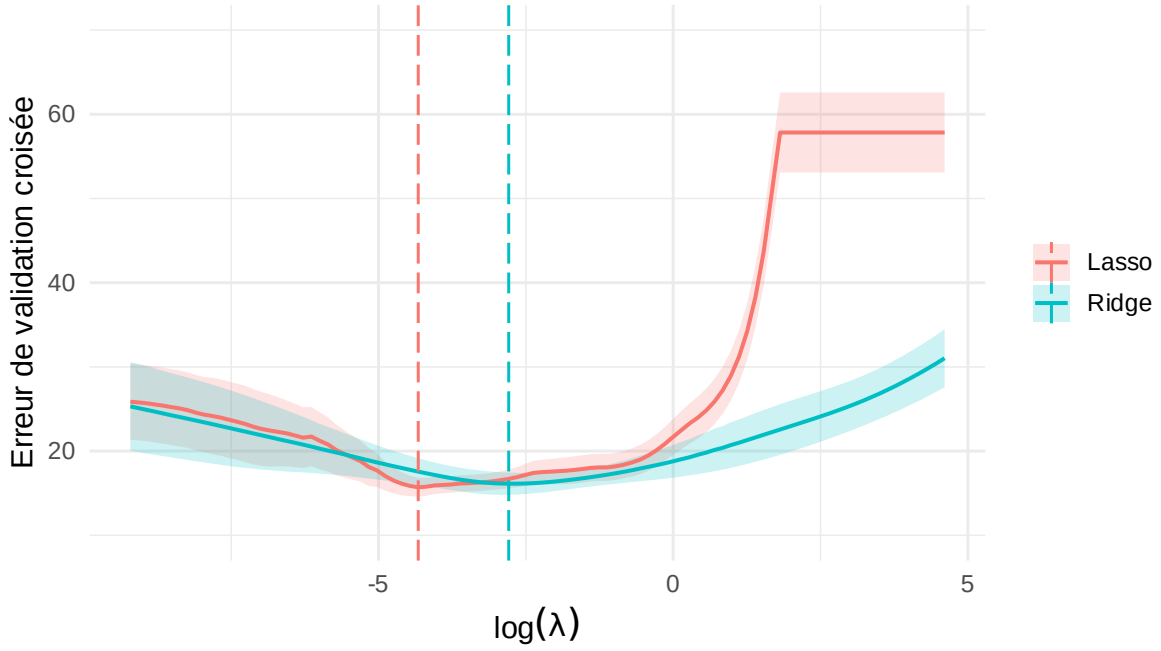
```
  Moindres carrés Ridge, lambda.min Lasso, lambda.min Lasso, lambda.1se
            42.45             18.71             18.64             18.33
```

```
c(non.nuls.min = unname(cv.lasso2$nzero[cv.lasso2$lambda ==
↪  cv.lasso2$lambda.min]),
  non.nuls.1se = unname(cv.lasso2$nzero[cv.lasso2$lambda ==
  ↪  cv.lasso2$lambda.1se]))
```

```
non.nuls.min non.nuls.1se
          17           13
```

```
courbes.cv <- bind_rows(
  data.frame(methode = "Ridge", log.lambda = log(cv.ridge2$lambda),
             erreur = cv.ridge2$cvm, et = cv.ridge2$cvsd),
  data.frame(methode = "Lasso", log.lambda = log(cv.lasso2$lambda),
```

```
            erreur = cv.lasso2$cvm, et = cv.lasso2$cvsd)
)
choix.lambda <- data.frame(methode = c("Ridge", "Lasso"),
                           log.lambda = log(c(cv.ridge2$lambda.min,
                            ↪  cv.lasso2$lambda.min)))

ggplot(courbes.cv, aes(log.lambda, erreur, colour = methode, fill = methode))
 ↪  +
  geom_ribbon(aes(ymin = erreur - et, ymax = erreur + et), alpha = 0.2,
   ↪  colour = NA) +
  geom_line(linewidth = 0.8) +
  geom_vline(data = choix.lambda, aes(xintercept = log.lambda, colour =
   ↪  methode),
             linetype = "longdash") +
  coord_cartesian(ylim = c(10, 70)) +
  labs(x = expression(log(lambda)), y = "Erreur de validation croisée",
       colour = NULL, fill = NULL) +
  theme_minimal(base_size = 13)
```



Avec 91 variables pour 166 observations d'entraînement, les moindres carrés sont encore calculables, mais leur variance est énorme : l'erreur test monte à 42,4, soit 2,1 fois celle des moindres carrés sans interactions. Ridge et le lasso ramènent l'erreur test à environ 18,6, au

niveau du modèle sans interactions. La figure montre cette fois une vraie courbe en U : le minimum de la validation croisée est atteint à l'intérieur de la grille, et l'erreur remonte quand $\lambda \to 0$, où les deux estimateurs rejoignent les moindres carrés. Le lasso garde 17 des 91 variables avec `lambda.min`.

Les interactions n'améliorent donc pas la prédiction de façon nette. Les deux analyses mènent à la même conclusion pratique : un modèle additif simple, dominé par le tour de taille, prédit aussi bien que des modèles beaucoup plus riches. La régularisation est indispensable dès que le nombre de variables approche le nombre d'observations, alors qu'avec 13 variables et 166 observations, elle sert surtout à simplifier le modèle (`lambda.1se`) sans perte de précision.

Chapitre 7 : k plus proches voisins et estimateur à noyau

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

a) Les distances à $x = 4$ sont $3, 2, 1, 1, 4, 5$. Pour $k = 2$, $d_{(2)}(4) = 1$, et les voisins sont les observations en $X = 3$ et $X = 5$, d'où $\hat{f}(4) = (3 + 7)/2 = 5$. Pour $k = 3$, $d_{(3)}(4) = 2$, et on ajoute l'observation en $X = 2$, d'où $\hat{f}(4) = (4 + 3 + 7)/3 = 14/3 \approx 4{,}67$.

b) Avec $h = 2{,}5$, les observations telles que $|X_i - 4| < 2{,}5$ sont celles en $X = 2, 3, 5$, d'où $\hat{f}(4) = 14/3$, la même valeur que le kNN avec $k = 3$. Avec $h = 1$, aucune observation ne vérifie $|X_i - 4| < 1$ (les deux plus proches sont exactement à distance 1). Le numérateur et le dénominateur sont nuls, et l'estimateur n'est pas défini en $x = 4$. C'est un défaut d'une fenêtre de largeur fixe, que le kNN n'a pas : il trouve toujours k voisins.

c) Les poids non normalisés $e^{-d_i^2/2}$ valent environ 0,0111, 0,1353, 0,6065, 0,6065, 0,0003 et 0,0000. Leur somme est 1,3598 et

$$\hat{f}(4) = \frac{0{,}0111 \times 2 + 0{,}1353 \times 4 + 0{,}6065 \times 3 + 0{,}6065 \times 7 + 0{,}0003 \times 6}{1{,}3598} \approx \frac{6{,}631}{1{,}3598} \approx 4{,}88.$$

Les deux voisins à distance 1 portent près de 90 % du poids total.

Exercice 2. Retour à l'énoncé

a) Conditionnellement aux X_i, les poids $w_i(x)$ sont des constantes et $\hat{f}(x) = \sum_i w_i(x) Y_i$ est une combinaison linéaire de variables indépendantes de variance σ^2, d'où $Var[\hat{f}(x)] = \sigma^2 \sum_i w_i(x)^2$.

b) Avec le noyau uniforme, $w_i(x) = 1/n_h(x)$ pour les $n_h(x)$ observations dans la fenêtre et 0 ailleurs, de sorte que $\sum_i w_i(x)^2 = n_h(x) \cdot 1/n_h(x)^2 = 1/n_h(x)$. On retrouve la variance σ^2/k du kNN, avec $n_h(x)$ à la place de k.

c) Par Cauchy-Schwarz, $1 = \left(\sum_{i=1}^m w_i \cdot 1\right)^2 \leq \sum_{i=1}^m w_i^2 \cdot \sum_{i=1}^m 1 = m \sum_i w_i^2$, d'où $\sum_i w_i^2 \geq 1/m$, avec égalité si et seulement si les w_i sont proportionnels à 1, c'est-à-dire tous égaux à $1/m$. Un noyau non uniforme, comme le noyau gaussien, paie donc un peu de variance pour réduire le biais en donnant plus de poids aux observations les plus proches de x.

Exercice 3. Retour à l'énoncé

a) Le plus proche voisin de X_i est l'observation i elle-même, à distance 0, de sorte que $\hat{f}(X_i) = Y_i$ et que l'erreur d'entraînement est nulle.

b) Avec $k = n$, toutes les observations sont voisines de tout x, et $\hat{f}(x) = \bar{Y}$ est constant. C'est le modèle nul.

c) L'erreur d'entraînement est nulle pour $k = 1$ et ne peut pas être plus petite, de sorte que ce critère choisit toujours le k le plus flexible, qui reproduit le bruit des observations. Le problème vient de ce que chaque Y_i sert à se prédire lui-même. La validation croisée prédit chaque Y_i par $\hat{f}_{-i}(X_i)$, calculé sans l'observation i. Pour $k = 1$, cette prédiction est la réponse de l'observation voisine, et l'erreur reflète alors le bruit des deux observations.

Niveau moyen

Exercice 1. Retour à l'énoncé

Posons $K_{ij} = K\{(X_j - X_i)/h\}$ et $S_i = \sum_{j=1}^n K_{ij}$, de sorte que $\hat{f}(X_i) = \sum_j K_{ij} Y_j / S_i$ et $w_i(X_i) = K_{ii}/S_i$. Retirer l'observation i revient à retirer le terme $j = i$ du numérateur et du dénominateur :

$$\hat{f}_{-i}(X_i) = \frac{\sum_{j \neq i} K_{ij} Y_j}{\sum_{j \neq i} K_{ij}} = \frac{S_i \hat{f}(X_i) - K_{ii} Y_i}{S_i - K_{ii}} = \frac{\hat{f}(X_i) - w_i(X_i) Y_i}{1 - w_i(X_i)},$$

en divisant le numérateur et le dénominateur par S_i. Ainsi,

$$Y_i - \hat{f}_{-i}(X_i) = \frac{Y_i\{1 - w_i(X_i)\} - \hat{f}(X_i) + w_i(X_i) Y_i}{1 - w_i(X_i)} = \frac{Y_i - \hat{f}(X_i)}{1 - w_i(X_i)}.$$

La formule ne s'applique pas au kNN. Pour l'estimateur à noyau, retirer l'observation i ne change pas les poids relatifs des autres observations. Pour le kNN, le voisinage change : sans l'observation i, le $(k+1)^{\text{e}}$ plus proche voisin de X_i entre dans $N_k(X_i)$. C'est pourquoi le Chapitre 7 calcule les prédictions LOOCV du kNN en refaisant le calcul pour chaque i.

Exercice 2. Retour à l'énoncé

Notons $K_i = K\{(X_i - x)/h\} \geq 0$. Le critère est un polynôme de degré 2 en a, dont la dérivée est $-2\sum_i K_i(Y_i - a)$. Elle s'annule en $a = \sum_i K_i Y_i / \sum_i K_i$, qui est un minimum puisque la dérivée seconde, $2\sum_i K_i$, est positive. On reconnaît $\hat{f}(x)$.

Autour de chaque point x, on ajuste donc une constante par moindres carrés, en donnant plus de poids aux observations proches de x : c'est un estimateur localement constant. L'estimateur localement linéaire remplace la constante par une droite centrée en x, en minimisant

$$\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) \{Y_i - a - b(X_i - x)\}^2$$

en (a, b), et pose $\hat{f}(x) = \hat{a}$. Près des bords, où les voisins sont tous du même côté de x, la droite corrige le biais du premier ordre de l'estimateur localement constant (voir l'exercice 1 du niveau difficile). C'est ce que fait `locpoly` avec `degree = 1`.

Exercice 3. Retour à l'énoncé

a) Avec $\Sigma^{-1} = \mathrm{diag}(1/s_1^2, \ldots, 1/s_p^2)$,

$$d(z_i, z)^2 = \sum_{j=1}^{p} \frac{(z_{ij} - z_j)^2}{s_j^2} = \sum_{j=1}^{p} \left(\frac{z_{ij}}{s_j} - \frac{z_j}{s_j}\right)^2,$$

le carré de la distance euclidienne entre les vecteurs standardisés.

b) On a $\tilde{z}_i - \tilde{z} = B(z_i - z)$, la constante c s'éliminant, et

$$(\tilde{z}_i - \tilde{z})^\top (B\Sigma B^\top)^{-1} (\tilde{z}_i - \tilde{z}) = (z_i - z)^\top B^\top (B^\top)^{-1} \Sigma^{-1} B^{-1} B (z_i - z) = (z_i - z)^\top \Sigma^{-1} (z_i - z).$$

La distance est donc la même pour toute transformation affine inversible des variables explicatives : changement d'unités, de point d'origine, ou même remplacement de deux variables par des combinaisons linéaires de celles-ci. Les voisins, et donc l'estimateur kNN, ne dépendent pas de ces choix, alors que ceux de la distance euclidienne changent dès qu'on change d'unité.

Niveau difficile

Exercice 1. Retour à l'énoncé

a) Posons $M = (k-1)/2$. Les voisins sont $x + m\delta$, pour $m = -M, \ldots, M$, et le développement de Taylor donne

$$E[\hat{f}(x)] = \frac{1}{k} \sum_{m=-M}^{M} f(x + m\delta) \approx f(x) + f'(x)\frac{\delta}{k} \sum_{m=-M}^{M} m + \frac{f''(x)\delta^2}{2k} \sum_{m=-M}^{M} m^2.$$

La première somme est nulle par symétrie, et la seconde vaut $2\sum_{m=1}^{M} m^2 = M(M+1)(2M+1)/3$. Avec $M = (k-1)/2$, on a $M+1 = (k+1)/2$ et $2M+1 = k$, d'où $\sum m^2 = k(k^2-1)/12$, et

$$\text{biais}[\hat{f}(x)] \approx \frac{f''(x)\delta^2}{2k} \cdot \frac{k(k^2-1)}{12} = \frac{f''(x)\,\delta^2\,(k^2-1)}{24}.$$

b) Les voisins sont $x + m\delta$ pour $m = 0, \ldots, k-1$, et le terme du premier ordre ne s'annule plus :

$$E[\hat{f}(x)] - f(x) \approx f'(x)\frac{\delta}{k}\sum_{m=0}^{k-1} m = f'(x)\frac{\delta}{k} \cdot \frac{k(k-1)}{2} = \frac{f'(x)\,\delta\,(k-1)}{2}.$$

Par symétrie, au point le plus à droite, le biais vaut $-f'(x)\,\delta\,(k-1)/2$, qui est négatif si f est croissante. Le biais au bord est d'ordre $k\delta$, celui de l'intérieur d'ordre $(k\delta)^2$. Comme $k\delta$, la largeur du voisinage, est petit, le biais au bord est beaucoup plus grand. On retrouve l'observation du Chapitre 7 : les plus gros arbres n'ont que des voisins plus petits, et l'estimateur sous-estime leur hauteur.

Exercice 2. Retour à l'énoncé

a) Avec $\delta = 1/n$ et $k^2 - 1 \approx k^2$, le biais vaut $f''(x)k^2/(24n^2)$, et

$$EQM(k) \approx \frac{f''(x)^2\,k^4}{576\,n^4} + \frac{\sigma^2}{k}.$$

b) Comme $\frac{d}{dk}\frac{k^4}{576} = \frac{4k^3}{576} = \frac{k^3}{144}$, la dérivée vaut

$$EQM'(k) \approx \frac{f''(x)^2 k^3}{144\,n^4} - \frac{\sigma^2}{k^2},$$

qui est négative pour les petites valeurs de k et positive pour les grandes. Elle s'annule lorsque $f''(x)^2k^5 = 144\,\sigma^2 n^4$, soit

$$k^* = \left\{\frac{144\,\sigma^2 n^4}{f''(x)^2}\right\}^{1/5} \propto n^{4/5}.$$

Le nombre optimal de voisins augmente avec n, mais moins vite que n, de sorte que la largeur du voisinage, $k^*/n \propto n^{-1/5}$, diminue.

c) En substituant $k^* \propto n^{4/5}$, les deux termes sont du même ordre : $\sigma^2/k^* \propto n^{-4/5}$ et $k^{*4}/n^4 \propto n^{16/5-4} = n^{-4/5}$. L'erreur quadratique moyenne optimale décroît donc comme $n^{-4/5}$, moins vite que le n^{-1} d'un modèle paramétrique bien spécifié. C'est le prix de la flexibilité : ne rien supposer sur la forme de f coûte de la précision, et ce prix augmente avec le nombre de variables explicatives (le fléau de la dimension du Chapitre 9).

Exercice 3. Retour à l'énoncé

a) Si $c(x) = 1$, on se trompe lorsque $Y = 0$, ce qui arrive avec probabilité $1 - \pi(x)$ conditionnellement à $X = x$. Si $c(x) = 0$, on se trompe lorsque $Y = 1$, avec probabilité $\pi(x)$.

b) Pour chaque x, l'erreur conditionnelle de c^* est $\min\{\pi(x), 1 - \pi(x)\}$, la plus petite des deux valeurs possibles d'après a). Pour tout autre classifieur c, on a donc $P\{Y \neq c(X) \mid X = x\} \geq P\{Y \neq c^*(X) \mid X = x\}$ pour tout x. En prenant l'espérance par rapport à X, $P\{Y \neq c(X)\} \geq P\{Y \neq c^*(X)\}$.

c) Le classifieur de Bayes demande de connaître $\pi(x)$. Le kNN l'estime par $\hat{\pi}(x)$, la proportion de voisins pour lesquels $Y_i = 1$, et classe selon que $\hat{\pi}(x) > 1/2$: il remplace π par son estimation dans la règle optimale. Le seuil $1/2$ est optimal parce que les deux erreurs coûtent la même chose. Si une erreur coûte a lorsqu'on prédit 0 alors que $Y = 1$ et b dans le cas contraire, le même raisonnement mène à prédire 1 lorsque $a\,\pi(x) > b\{1 - \pi(x)\}$, soit au seuil $u = b/(a + b)$. Par exemple, en dépistage, on accepte des faux positifs pour éviter les faux négatifs, et on prend $u < 1/2$.

Chapitre 8 : Polynômes, fonctions par morceaux et splines

Retour aux exercices

Exercices appliqués

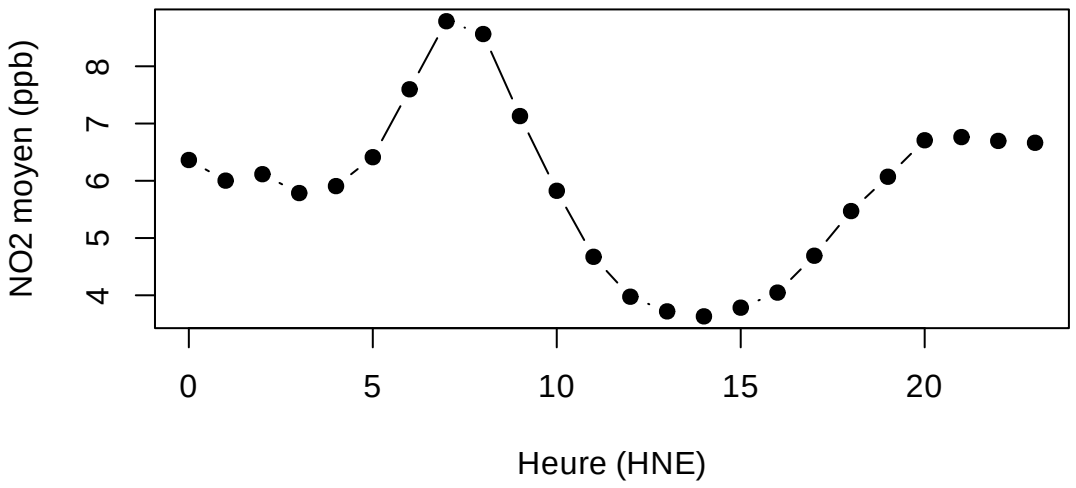
Exercice 1. Retour à l'énoncé

a) Les moyennes horaires s'obtiennent avec `group_by` et `summarise` :

```
moy.no2 <- no2 |>
  group_by(heure) |>
  summarise(NO2 = mean(NO2)) |>
  pull(NO2, name = heure)
round(moy.no2, 1)
```

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
6.4 6.0 6.1 5.8 5.9 6.4 7.6 8.8 8.6 7.1 5.8 4.7 4.0 3.7 3.6 3.8 4.0 4.7 5.5 6.1
 20  21  22  23
6.7 6.8 6.7 6.7
```

```
plot(0:23, moy.no2, type = "b", pch = 19,
     xlab = "Heure (HNE)", ylab = "NO2 moyen (ppb)")
```



Le cycle a deux sommets : un premier vers 7 h-8 h, puis un second en soirée, à partir de 20 h, avec un creux en début d'après-midi. Le dioxyde d'azote vient surtout de la combustion, et en particulier de la circulation automobile, d'où les pointes du matin et du soir. En après-midi, la couche de mélange de l'atmosphère est plus épaisse, ce qui dilue les polluants, et une partie du NO2 est consommée par la photochimie qui produit l'ozone. C'est pourquoi le creux du NO2 coïncide avec le sommet de l'ozone.

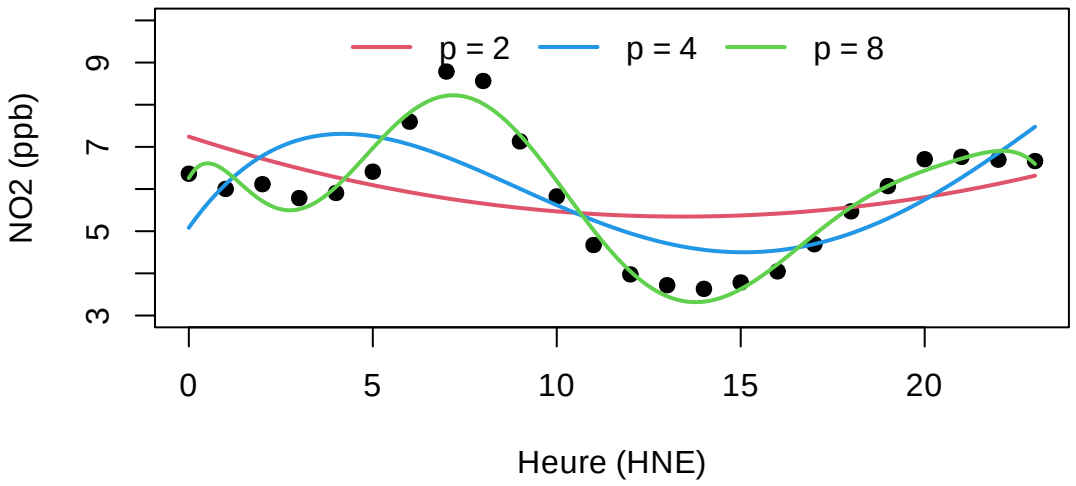
b) On ajuste les douze polynômes et on calcule les deux critères :

```
degres <- 1:12
modeles.no2 <- lapply(degres, function(p) lm(NO2 ~ poly(heure, p), data =
 ↪  no2))
crit.no2 <- rbind(degre = degres,
                  vc.n = round(sapply(modeles.no2, vc.n), 2),
                  BIC = round(sapply(modeles.no2, BIC), 1))
crit.no2
```

```
          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
degre     1.00     2.00     3.00     4.00     5.00     6.00     7.00     8.00
vc.n     29.58    29.39    28.75    28.69    28.05    27.99    27.89    27.83
BIC   38884.90 38850.60 38719.10 38713.90 38578.50 38572.70 38556.40 38549.80
          [,9]   [,10]    [,11]    [,12]
```

```
degre     9.00    10.0    11.00    12.00
vc.n     27.84    27.8    27.78    27.79
BIC   38558.30 38556.5 38559.40 38568.00
```

```r
grille.h <- data.frame(heure = seq(0, 23, length.out = 300))
plot(0:23, moy.no2, pch = 19, ylim = c(3, 10), xlab = "Heure (HNE)", ylab =
 ↪  "NO2 (ppb)")
couleurs <- c(2, 4, 3)
for (j in 1:3) {
  p <- c(2, 4, 8)[j]
  lines(grille.h$heure, predict(modeles.no2[[p]], grille.h), lwd = 2, col =
   ↪  couleurs[j])
}
legend("top", legend = paste("p =", c(2, 4, 8)), col = couleurs, lwd = 2,
 ↪  horiz = TRUE, bty = "n")
```



Le BIC est minimal pour $p = 8$, et $VC_{(n)}$ diminue jusqu'à environ ce degré, puis devient pratiquement plat. Le polynôme de degré 2 ne voit qu'un creux au milieu de la journée, et celui de degré 4 place la pointe du matin trop tôt et l'aplatit. Le degré 8 reproduit les deux pointes. Les écarts de $VC_{(n)}$ entre les degrés sont petits en valeur absolue (de 29,58 à 27,78) : l'amplitude du cycle, environ 5 ppb, est faible devant la variabilité d'un jour à l'autre, qui dépend de la météo et de la circulation.

c) La courbe change quatre fois de direction (creux de nuit, pointe du matin, creux de l'après-midi, pointe du soir). La dérivée d'un polynôme de degré p a au plus $p-1$ racines, de sorte qu'il faut au moins le degré 5, et un peu plus pour que la forme des pointes soit juste. Pour l'ozone, une seule bosse suffisait.

Avec `raw = TRUE`, la matrice de design contient les colonnes $x, x^2, \ldots, x^{12}$, et $23^{12} \approx 2 \times 10^{16}$. Les colonnes ont des ordres de grandeur si différents que $X^\top X$ est numériquement singulière, et `lm` peut écarter des colonnes (coefficients `NA`) ou donner des estimations imprécises. La fonction `poly` sans `raw = TRUE` construit des polynômes orthogonaux, qui engendrent le même espace de fonctions, donc les mêmes valeurs ajustées, avec une matrice parfaitement conditionnée. C'est le même argument que pour les B-splines face à la base des puissances tronquées.

Exercice 2. Retour à l'énoncé

a) On reprend le code du Chapitre 8 :

```
Ks <- c(2, 3, 4, 6, 8, 12, 24)
vc.cut.no2 <- sapply(Ks, function(K) {
  bornes <- seq(0, 24, length.out = K + 1)
  vc.n(lm(NO2 ~ cut(heure, bornes, right = FALSE), data = no2))
})
rbind(K = Ks, vc.n = round(vc.cut.no2, 2))
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
K     2.00  3.00  4.00  6.00  8.00 12.00 24.00
vc.n 29.16 29.31 28.32 28.38 27.95 27.89 27.87
```

Le critère n'est pas monotone : $K = 3$ fait moins bien que $K = 2$, parce que des intervalles de 8 heures mélangent la pointe du matin et le creux de la journée. Il faut aller jusqu'à $K = 8$ ou plus pour que les intervalles isolent les pointes, et les moyennes horaires ($K = 24$) ne font pas mieux que $K = 12$.

b) Avec des bornes choisies d'après le cycle :

```
bornes5 <- c(0, 6, 10, 16, 20, 24)
m.cut5 <- lm(NO2 ~ cut(heure, bornes5, right = FALSE), data = no2)
vc.n(m.cut5)
```

```
[1] 28.0406
```

Avec seulement cinq paramètres, ce découpage donne $VC_{(n)} \approx 28{,}04$, mieux que les découpages réguliers à $K = 4$ et $K = 6$. Placer les bornes là où la courbe change vaut mieux qu'augmenter leur nombre au hasard. La même idée vaut pour les nœuds d'une spline de régression : on peut en mettre davantage là où la courbe varie rapidement.

c) Sans ordonnée à l'origine, chaque coefficient correspond à un intervalle :

```
m.cut5b <- lm(NO2 ~ cut(heure, bornes5, right = FALSE) - 1, data = no2)
moyennes.cut5 <- no2 |>
  group_by(intervalle = cut(heure, bornes5, right = FALSE)) |>
  summarise(moyenne = mean(NO2))
cbind(coefficient = coef(m.cut5b), moyenne = moyennes.cut5$moyenne)
```

```
                                          coefficient  moyenne
cut(heure, bornes5, right = FALSE)[0,6)      6.097661 6.097661
cut(heure, bornes5, right = FALSE)[6,10)     8.019527 8.019527
cut(heure, bornes5, right = FALSE)[10,16)    4.267783 4.267783
cut(heure, bornes5, right = FALSE)[16,20)    5.069399 5.069399
cut(heure, bornes5, right = FALSE)[20,24)    6.707707 6.707707
```

Les deux colonnes sont identiques : $\hat{\beta}_k$ est la moyenne des Y_i dont le X_i tombe dans le k^{e} intervalle (voir aussi l'exercice 1 des exercices théoriques).

Exercice 3. Retour à l'énoncé

a) et b) On ajuste les trois familles et on regroupe le critère et les prédictions dans une même table :

```
nouv <- data.frame(diameter_cm = c(40, 45, 50, 60))
resume <- function(m) c(vc.n = vc.n(m), predict(m, nouv))
tab.extrap <- rbind(
  t(sapply(1:5, function(p) resume(lm(height_m ~ poly(diameter_cm, p), data =
   ↪  arbres)))),
  t(sapply(3:6, function(d) resume(lm(height_m ~ bs(diameter_cm, df = d),
   ↪  data = arbres)))),
  t(sapply(1:6, function(d) resume(lm(height_m ~ ns(diameter_cm, df = d),
   ↪  data = arbres))))
)
rownames(tab.extrap) <- c(paste0("poly ", 1:5), paste0("bs df=", 3:6),
 ↪  paste0("ns df=", 1:6))
colnames(tab.extrap) <- c("vc.n", paste0("D=", nouv$diameter_cm))
round(tab.extrap, 1)
```

```
         vc.n D=40 D=45 D=50  D=60
poly 1    6.2 25.5 27.9 30.4  35.4
poly 2    6.2 23.5 24.6 25.4  26.0
poly 3    6.3 24.3 27.0 30.6  40.9
poly 4    6.6 24.1 23.7 19.6  -9.2
poly 5    6.9 23.9 27.1 35.6 102.0
bs df=3   6.3 24.3 27.0 30.6  40.9
bs df=4   6.4 24.2 26.6 29.5  37.4
bs df=5   6.5 24.1 25.4 26.3  26.3
bs df=6   6.6 24.0 24.9 24.7  20.3
ns df=1   6.2 25.5 27.9 30.4  35.4
ns df=2   6.2 23.7 25.3 26.9  30.2
ns df=3   6.2 24.2 26.1 28.0  31.9
ns df=4   6.3 24.4 26.5 28.6  32.9
ns df=5   6.3 24.4 26.5 28.6  32.9
ns df=6   6.4 24.5 26.8 29.0  33.5
```

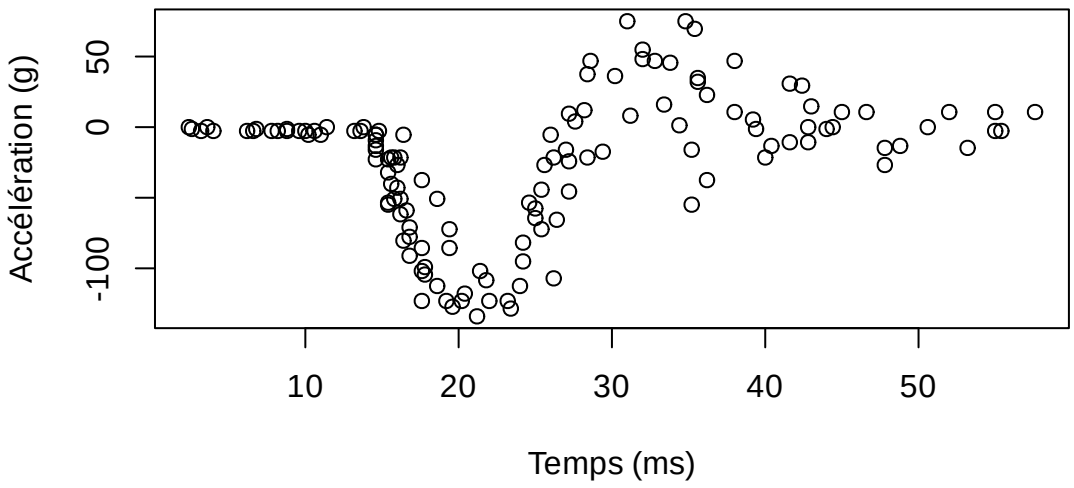
L'avertissement de `bs` (masqué ici) signale justement qu'on prédit au-delà des nœuds frontières. Sur l'étendue des données, tous les modèles se valent : $VC_{(n)}$ varie peu, et la spline naturelle à 2 degrés de liberté est légèrement meilleure. À 40 cm, toutes les prédictions sont proches. Au-delà, elles divergent complètement. Le polynôme de degré 4 prédit une hauteur négative à 60 cm et celui de degré 5, une hauteur de plus de 100 m. Les splines cubiques `bs` avec beaucoup de degrés de liberté se mettent à décroître. Seules les splines naturelles, avec 2 degrés de liberté ou plus, donnent des prédictions stables d'un modèle à l'autre.

c) Au-delà des nœuds frontières, une spline naturelle est linéaire par construction, et la droite prolonge la pente estimée au bord du domaine. Les polynômes et les splines cubiques prolongent plutôt leurs termes de degré élevé, qui dominent dès qu'on s'éloigne des données. La prédiction à 60 cm reste une extrapolation : rien dans les données ne dit que la hauteur continue de croître au même rythme au-delà de 41,5 cm. En fait, la croissance en hauteur ralentit fortement chez les gros arbres, et les quatre espèces de l'extrait dépassent rarement 25 à 30 m. La spline naturelle évite les prédictions absurdes, mais elle ne remplace pas des données dans la zone qui nous intéresse.

Exercice 4. Retour à l'énoncé

a) Le nuage de points montre trois phases : une accélération nulle et très peu bruitée jusqu'à environ 14 ms, puis une forte décélération suivie d'un rebond entre 15 et 35 ms, et enfin un retour progressif vers 0 avec des oscillations amorties.

```
plot(accel ~ times, data = mcycle, xlab = "Temps (ms)", ylab = "Accélération
 ↪  (g)")
```




```
vc.poly.mc <- sapply(1:12, function(p) vc.n(lm(accel ~ poly(times, p), data =
 ↪  mcycle)))
round(rbind(degre = 1:12, vc.n = vc.poly.mc))
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
degre    1    2    3    4    5    6    7    8    9    10    11    12
vc.n  2162 2057 1633 1666 1245 1162 1411  812 2000  1723  3323  1556
```

Un polynôme est une seule fonction analytique sur tout le domaine : pour suivre la chute brutale après 14 ms, il doit osciller aussi dans la zone plate du début et dans la queue. Le critère $VC_{(n)}$ le montre. Il ne décroît pas régulièrement, mais saute d'un degré à l'autre, parce que chaque terme ajouté déplace les oscillations. Le meilleur degré ($p = 8$) reste instable : les degrés voisins sont nettement moins bons.

b) Avec les splines :

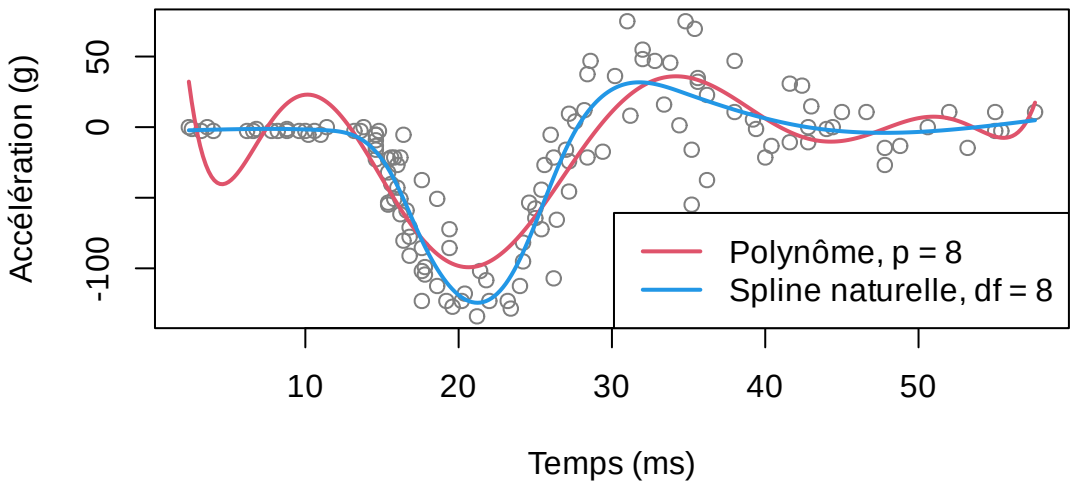
```
dfs <- 3:15
vc.bs.mc <- sapply(dfs, function(d) vc.n(lm(accel ~ bs(times, df = d), data =
 ↪  mcycle)))
vc.ns.mc <- sapply(dfs, function(d) vc.n(lm(accel ~ ns(times, df = d), data =
 ↪  mcycle)))
round(rbind(df = dfs, bs = vc.bs.mc, ns = vc.ns.mc))
```

```
   [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
df    3    4    5    6    7    8    9   10   11    12    13    14    15
bs 1633 1631 1093 1029  597  549  578  535  532   535   549   554   559
ns 1483 1341  713  583  576  532  530  534  547   552   556   555   562
```

Le critère des splines décroît régulièrement jusqu'à 8 ou 9 degrés de liberté, puis remonte lentement. La spline naturelle à 9 degrés de liberté a le plus petit $VC_{(n)}$ (530), soit environ 35 % de moins que le meilleur polynôme. Pour les petites valeurs de `df`, `ns` fait nettement mieux que `bs`, parce que les deux degrés de liberté que `bs` consacre aux bords sont mieux utilisés en nœuds intérieurs dans la zone de l'impact. Comme le minimum est plat, on peut retenir `ns(times, df = 8)`.

c) La figure suivante superpose les deux ajustements :

```
grille.t <- data.frame(times = seq(min(mcycle$times), max(mcycle$times),
 ↪  length.out = 300))
m.poly8 <- lm(accel ~ poly(times, 8), data = mcycle)
m.ns8 <- lm(accel ~ ns(times, df = 8), data = mcycle)
plot(accel ~ times, data = mcycle, col = "grey50", xlab = "Temps (ms)", ylab
 ↪  = "Accélération (g)")
lines(grille.t$times, predict(m.poly8, grille.t), lwd = 2, col = 2)
lines(grille.t$times, predict(m.ns8, grille.t), lwd = 2, col = 4)
legend("bottomright", c("Polynôme, p = 8", "Spline naturelle, df = 8"), col =
 ↪  c(2, 4), lwd = 2)
```



Le polynôme ondule dans la zone plate du début, où les observations sont pourtant presque toutes nulles, et il sous-estime la profondeur de la chute. La spline naturelle reste presque plate au début, descend plus bas au fond de la chute, suit le rebond, puis s'amortit.

d) Le dernier essai a lieu à 57,6 ms :

```
nouv.t <- data.frame(times = c(60, 65))
m.bs8 <- lm(accel ~ bs(times, df = 8), data = mcycle)
round(rbind(poly8 = predict(m.poly8, nouv.t), bs8 = predict(m.bs8, nouv.t),
            ns8 = predict(m.ns8, nouv.t)), 1)
```

```
          1      2
poly8 198.9 2483.1
bs8   -18.0  -84.1
ns8     8.2   15.0
```

Après l'impact, l'accélération doit revenir vers 0. Le polynôme de degré 8 prédit des valeurs de plusieurs centaines, voire milliers de g, ce qui est absurde. La spline cubique s'écarte elle aussi rapidement. La spline naturelle prolonge linéairement la tendance au bord et reste dans l'ordre de grandeur des données, même si elle ne prévoit pas le retour à 0 que la physique suggère. Aucune méthode de régression n'extrapole bien, mais la spline naturelle limite les dégâts.

Exercice 5. Retour à l'énoncé

a) On ajuste la spline de lissage :

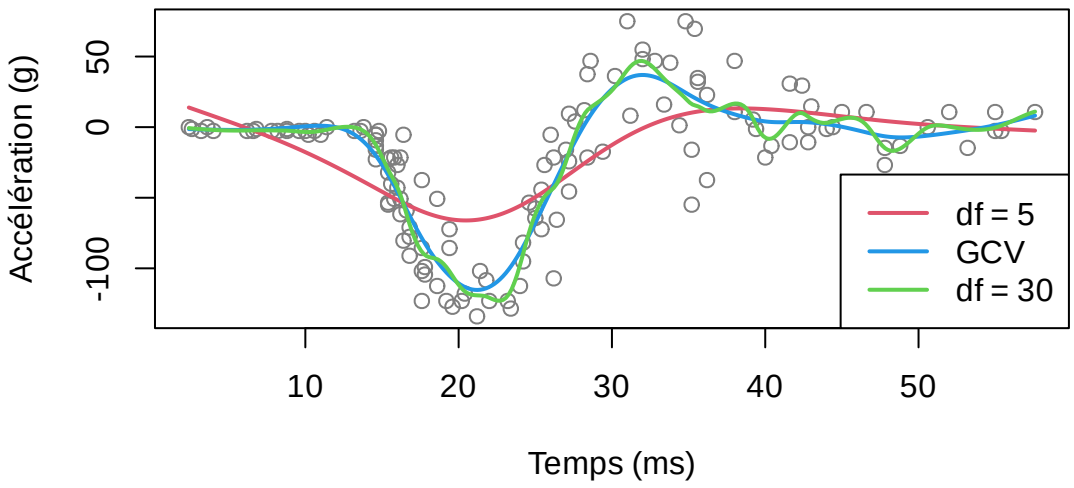
```
sp.mc <- smooth.spline(mcycle$times, mcycle$accel)
sp.mc
```

```
Call:
smooth.spline(x = mcycle$times, y = mcycle$accel)

Smoothing Parameter  spar= 0.6598558  lambda= 0.00011075 (14 iterations)
Equivalent Degrees of Freedom (Df): 12.20876
Penalized Criterion (RSS): 38650.54
GCV: 565.4513
```

La GCV retient environ 12,2 degrés de liberté effectifs. C'est plus que les 9 paramètres de `ns(times, df = 8)` (ordonnée à l'origine comprise), mais la comparaison est indirecte : la spline de lissage dispose d'un nœud par valeur distincte de `times` (94 nœuds) et répartit sa flexibilité là où les données l'exigent, alors que `ns` place ses nœuds aux quantiles de `times`. Le critère GCV (565) est du même ordre que le $VC_{(n)}$ de la spline naturelle.

b) On compare trois niveaux de lissage :

```
plot(accel ~ times, data = mcycle, col = "grey50", xlab = "Temps (ms)", ylab
 ↪  = "Accélération (g)")
lines(predict(smooth.spline(mcycle$times, mcycle$accel, df = 5),
 ↪  grille.t$times), lwd = 2, col = 2)
lines(predict(sp.mc, grille.t$times), lwd = 2, col = 4)
lines(predict(smooth.spline(mcycle$times, mcycle$accel, df = 30),
 ↪  grille.t$times), lwd = 2, col = 3)
legend("bottomright", c("df = 5", "GCV", "df = 30"), col = c(2, 4, 3), lwd =
 ↪  2)
```



Avec 5 degrés de liberté, la pénalité est trop forte : la courbe n'atteint pas le fond de la chute et commence à descendre avant 14 ms (biais élevé). Avec 30 degrés de liberté, elle suit les fluctuations des observations individuelles, surtout dans la queue (variance élevée). La GCV choisit un intermédiaire.

c) On calcule l'écart-type des résidus par zone :

```
res.mc <- mcycle$accel - predict(sp.mc, mcycle$times)$y
ecarts.zone <- tibble(residu = res.mc,
                      zone = cut(mcycle$times, c(0, 14, 40, 60),
                                 labels = c("< 14 ms", "14-40 ms", "> 40
                                  ↪  ms"))) |>
  group_by(zone) |>
  summarise(ecart.type = sd(residu))
ecarts.zone
```

```
# A tibble: 3 x 2
  zone     ecart.type
  <fct>         <dbl>
1 < 14 ms        3.67
2 14-40 ms      25.4
3 > 40 ms       13.5
```

La dispersion des résidus est d'environ 4 g avant l'impact, mais de 25 g pendant l'impact : la variance n'est pas constante. Or la spline de lissage, comme la GCV, suppose implicitement des erreurs de même variance. Le λ choisi est un compromis global, trop peu lissant pour la zone bruitée et sans conséquence dans la zone calme, où les données sont presque exactes. Les intervalles de confiance usuels, qui utilisent une seule estimation de σ, seraient trop larges avant 14 ms et trop étroits pendant l'impact. On pourrait corriger en pondérant les observations par l'inverse de leur variance (argument `w` de `smooth.spline`, ou `weights` de `lm`), estimée par exemple à partir de ces écarts-types par zone.

Exercice 6. Retour à l'énoncé

a) On suit la recette du Chapitre 1 :

```
f.vraie <- function(x) sin(2 * pi * x)
x0 <- c(0.02, 0.5, 0.98)

set.seed(4300)
B <- 500
pred <- replicate(B, {
  x <- runif(50)
  y <- f.vraie(x) + rnorm(50, sd = 0.3)
  m.bs <- lm(y ~ bs(x, df = 6, Boundary.knots = c(0, 1)))
  m.ns <- lm(y ~ ns(x, df = 6, Boundary.knots = c(0, 1)))
  c(predict(m.bs, data.frame(x = x0)), predict(m.ns, data.frame(x = x0)))
})
```

b) Chaque ligne de `pred` correspond à une méthode et à un point x_0 :

```
biais <- rowMeans(pred) - rep(f.vraie(x0), 2)
variance <- apply(pred, 1, var)
tab.mc <- rbind(biais = biais, variance = variance, EQM = biais^2 + variance)
colnames(tab.mc) <- paste(rep(c("bs", "ns"), each = 3), "x0 =", rep(x0, 2))
round(tab.mc, 4)
```

```
         bs x0 = 0.02 bs x0 = 0.5 bs x0 = 0.98 ns x0 = 0.02 ns x0 = 0.5
biais          0.0123      0.0024       0.0133       0.0156      0.0041
variance       0.0878      0.0093       0.0932       0.0425      0.0109
EQM            0.0879      0.0093       0.0933       0.0427      0.0110
         ns x0 = 0.98
biais          0.0017
variance       0.0483
EQM            0.0483
```

Au centre ($x_0 = 0{,}5$), les deux estimateurs sont équivalents. Aux bords, la variance de `bs` est environ 2 fois plus grande que celle de `ns`. Le biais reste faible dans les deux cas, de sorte que l'erreur quadratique moyenne est dominée par la variance et nettement plus petite pour la spline naturelle. L'erreur-type de Monte Carlo d'une variance estimée est d'environ $\hat{v}\sqrt{2/(B-1)}$, soit 6 % de sa valeur, ce qui est largement assez précis pour conclure.

c) Les deux modèles ont exactement le même nombre de paramètres (6 plus l'ordonnée à l'origine). La spline cubique `bs(x, df = 6)` a 3 nœuds intérieurs, et elle laisse la courbure libre aux deux bords, où peu d'observations la contraignent. La spline naturelle impose $f'' = 0$ aux bords, ce qui libère deux paramètres qu'elle réinvestit en nœuds intérieurs (5 au lieu de 3). Elle gagne en stabilité aux bords et en flexibilité au centre.

d) On reprend le même code avec la nouvelle fonction et la même graine :

```
f.cos <- function(x) cos(2 * pi * x)
set.seed(4300)
pred.cos <- replicate(B, {
  x <- runif(50)
  y <- f.cos(x) + rnorm(50, sd = 0.3)
  m.bs <- lm(y ~ bs(x, df = 6, Boundary.knots = c(0, 1)))
  m.ns <- lm(y ~ ns(x, df = 6, Boundary.knots = c(0, 1)))
  c(predict(m.bs, data.frame(x = x0)), predict(m.ns, data.frame(x = x0)))
})
biais.cos <- rowMeans(pred.cos) - rep(f.cos(x0), 2)
variance.cos <- apply(pred.cos, 1, var)
tab.cos <- rbind(biais = biais.cos, variance = variance.cos, EQM =
 ↪  biais.cos^2 + variance.cos)
colnames(tab.cos) <- colnames(tab.mc)
round(tab.cos, 4)
```

```
         bs x0 = 0.02 bs x0 = 0.5 bs x0 = 0.98 ns x0 = 0.02 ns x0 = 0.5
biais          0.0064      0.0047       0.0037       0.0488      0.0055
variance       0.0849      0.0094       0.0906       0.0446      0.0111
EQM            0.0850      0.0094       0.0907       0.0469      0.0111
         ns x0 = 0.98
biais          0.0459
variance       0.0498
EQM            0.0520
```

Cette fois, la contrainte $f''(0) = f''(1) = 0$ est fausse, puisque $f''(0) = -4\pi^2$. La spline naturelle a donc un biais aux bords, environ dix fois plus grand que celui de la spline cubique. Ce biais reste toutefois petit devant l'écart-type de l'estimateur, et le gain en variance l'emporte encore :

avec $n = 50$, l'erreur quadratique moyenne de `ns` reste nettement plus petite aux bords. Le compromis pourrait s'inverser avec un n beaucoup plus grand, puisque la variance diminue avec n alors que le biais de la contrainte, lui, ne disparaît pas (il diminue seulement à mesure que les nœuds se rapprochent des bords).

Exercice 7. Retour à l'énoncé

a) et b) La base contient $1, x, x^2, x^3$ et une fonction $(x - c_k)_+^3$ par nœud :

```
base.pt <- function(x, noeuds) {
  cbind(1, x, x^2, x^3, sapply(noeuds, function(c) pmax(x - c, 0)^3))
}

comparaison <- t(sapply(c(3, 6, 10, 15), function(K) {
  noeuds <- quantile(mcycle$times, (1:K) / (K + 1))
  X.pt <- base.pt(mcycle$times, noeuds)
  m.pt <- lm(mcycle$accel ~ X.pt - 1)
  m.bs <- lm(accel ~ bs(times, knots = noeuds), data = mcycle)
  c(K = K,
    ecart.max = max(abs(fitted(m.pt) - fitted(m.bs))),
    kappa.pt = kappa(crossprod(X.pt), exact = TRUE),
    kappa.bs = kappa(crossprod(model.matrix(m.bs)), exact = TRUE))
}))
signif(comparaison, 3)
```

```
      K ecart.max kappa.pt kappa.bs
[1,]  3  9.10e-12 6.42e+11      430
[2,]  6  3.60e-11 2.05e+12      747
[3,] 10  8.87e-11 7.96e+12     1130
[4,] 15  4.62e-10 2.94e+13     1800
```

c) Les valeurs ajustées des deux bases sont identiques, à l'erreur d'arrondi près (10^{-10} au plus), parce que les deux bases engendrent le même espace de splines cubiques avec les mêmes nœuds. L'estimateur des moindres carrés, $\hat{Y} = HY$, ne dépend que de cet espace (la matrice chapeau est la projection orthogonale sur l'espace engendré par les colonnes de X). Le nombre de condition est cependant de l'ordre de 10^{12} à 10^{13} pour les puissances tronquées, contre quelques centaines ou milliers pour les B-splines. Avec `times` jusqu'à 57,6 ms, la colonne x^3 dépasse 10^5 alors que la première vaut 1, et les fonctions $(x - c_k)_+^3$ de nœuds voisins sont presque colinéaires. En double précision, un nombre de condition de 10^{13} fait perdre environ 13 des 16 chiffres significatifs dans la résolution des équations normales. Ici, `lm` s'en sort parce qu'il utilise une décomposition QR plutôt que d'inverser $X^\top X$, mais les coefficients eux-mêmes sont mal déterminés et la situation se dégrade avec plus de nœuds ou des valeurs de x plus grandes.

Exercices théoriques

Exercice 1. Retour à l'énoncé

a) L'élément (k,l) de $X^\top X$ est $\sum_i b_k(X_i)b_l(X_i)$. Comme les intervalles sont disjoints, $b_k(X_i)b_l(X_i) = 0$ si $k \neq l$, et $b_k(X_i)^2 = b_k(X_i)$, dont la somme est n_k. De même, la k^{e} composante de $X^\top Y$ est $\sum_i b_k(X_i)Y_i$, la somme des Y_i de l'intervalle k. Ainsi,

$$\hat{\beta}_k = \frac{\sum_i b_k(X_i)Y_i}{n_k},$$

la moyenne des réponses de l'intervalle.

b) On a $Var[\hat{\beta}] = \sigma^2(X^\top X)^{-1} = \sigma^2 \text{diag}(1/n_1, \ldots, 1/n_K)$. Ses éléments diagonaux sont σ^2/n_k et ses éléments hors diagonale sont nuls. Chaque $\hat{\beta}_k$ n'utilise que les observations de son intervalle, si bien qu'un intervalle peu peuplé a une estimation très variable. Avec des intervalles de même largeur, c'est ce qui arrive là où les X_i sont rares, par exemple aux extrémités du domaine.

Exercice 2. Retour à l'énoncé

a) Pour $x < c$, h est identiquement nulle, ainsi que toutes ses dérivées. Pour $x > c$,

$$h'(x,c) = 3(x-c)^2, \qquad h''(x,c) = 6(x-c), \qquad h'''(x,c) = 6.$$

Quand $x \to c^+$, h, h' et h'' tendent vers 0, la valeur commune à gauche : elles sont continues en c. En revanche, h''' vaut 0 à gauche et 6 à droite, ce qui est un saut.

b) Sur chaque intervalle entre deux nœuds consécutifs, f est un polynôme de degré au plus 3, puisque chaque $(x-c_k)_+^3$ y est soit nul, soit un polynôme cubique. Par a), f, f' et f'' sont continues partout, et seule f''' saute aux nœuds (de $6\theta_k$ en c_k). C'est la définition d'une spline cubique. Elle compte 4 coefficients polynomiaux et $K-1$ coefficients θ_k, soit $K+3$ paramètres, ce qui correspond au compte $K+p$ du Chapitre 8 avec $p=3$. Les θ_k mesurent les sauts de la dérivée troisième, d'où leur interprétation peu parlante.

Exercice 3. Retour à l'énoncé

a) Sur $(-\infty, \xi_1]$, tous les termes tronqués sont nuls et $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$. Ce polynôme est linéaire si et seulement si ses coefficients de degrés 2 et 3 sont nuls.

b) Pour $x \geq \xi_m$, tous les termes tronqués sont actifs et, avec $\beta_2 = \beta_3 = 0$,

$$f(x) = \beta_0 + \beta_1 x + \sum_k \theta_k(x^3 - 3\xi_k x^2 + 3\xi_k^2 x - \xi_k^3).$$

Le coefficient de x^3 est $\sum_k \theta_k$ et celui de x^2 est $-3\sum_k \theta_k \xi_k$. La fonction est linéaire sur $[\xi_m, \infty)$ si et seulement si ces deux coefficients sont nuls.

c) La spline cubique compte $m+4$ paramètres, et la linéarité aux deux bouts impose les quatre contraintes linéaires indépendantes $\beta_2 = 0$, $\beta_3 = 0$, $\sum_k \theta_k = 0$ et $\sum_k \theta_k \xi_k = 0$. Il reste $m + 4 - 4 = m$ degrés de liberté. Au Chapitre 8, les nœuds sont $c_0, \ldots, c_K$, soit $m = K + 1$ nœuds en comptant les nœuds frontières, d'où $K + 1$ degrés de liberté. La fonction `ns(x, df = d)` place ainsi $d - 1$ nœuds intérieurs entre les deux nœuds frontières, soit $d + 1$ nœuds et $d + 1$ paramètres avec l'ordonnée à l'origine.

Exercice 4. Retour à l'énoncé

a) Posons $g(\beta_1, \beta_2) = -\beta_1/(2\beta_2)$. Son gradient est

$$\nabla g = \left(\frac{\partial g}{\partial \beta_1}, \frac{\partial g}{\partial \beta_2}\right)^\top = \left(-\frac{1}{2\beta_2},\ \frac{\beta_1}{2\beta_2^2}\right)^\top = -\frac{1}{2\beta_2}\,(1,\ 2x^*)^\top,$$

puisque $\beta_1/(2\beta_2^2) = -x^*/\beta_2$. La méthode du delta multivariée donne $Var[\hat{x}^*] \approx \nabla g^\top \Sigma \nabla g$, où Σ est la matrice de variance-covariance de $(\hat{\beta}_1, \hat{\beta}_2)$:

$$Var[\hat{x}^*] \approx \frac{1}{4\beta_2^2}\,(1,\ 2x^*)\,\Sigma \begin{pmatrix} 1 \\ 2x^* \end{pmatrix} = \frac{1}{4\beta_2^2}\left\{Var[\hat{\beta}_1] + 4x^* Cov[\hat{\beta}_1, \hat{\beta}_2] + 4x^{*2} Var[\hat{\beta}_2]\right\}.$$

En pratique, on remplace les paramètres par leurs estimations.

b) Le facteur $1/(4\beta_2^2)$ explose quand $\beta_2 \to 0$. La courbe est alors presque une droite, et son extremum est rejeté très loin, souvent hors du domaine des données, où il n'a pas de sens (pour les arbres du Chapitre 8, la parabole concave culmine bien au-delà du plus gros diamètre observé). De plus, $\hat{x}^*$ est un rapport. Si $(\hat{\beta}_1, \hat{\beta}_2)$ suit exactement une loi normale, la densité de $\hat{\beta}_2$ est positive en 0 et le rapport n'a même pas de variance finie. La méthode du delta n'est donc utile que si $P(\hat{\beta}_2 \approx 0)$ est négligeable, ce qui n'est plus le cas lorsque β_2 est proche de 0 par rapport à son erreur-type. On ne devrait interpréter $\hat{x}^*$ que lorsque $\hat{\beta}_2$ est clairement non nul et que $\hat{x}^*$ tombe dans le domaine des données.

Exercice 5. Retour à l'énoncé

a) Notons $G = X^\top X + \lambda M$. Sans l'observation i, la matrice devient $G - X_i X_i^\top$ et le second membre $X^\top Y - X_i Y_i$. Par Sherman-Morrison avec $u = v = X_i$, et puisque $X_i^\top G^{-1} X_i = A_{ii}$,

$$(G - X_i X_i^\top)^{-1} = G^{-1} + \frac{G^{-1} X_i X_i^\top G^{-1}}{1 - A_{ii}}.$$

En multipliant par $X^\top Y - X_i Y_i$ et en utilisant $G^{-1} X^\top Y = \hat{\beta}_\lambda$ et $X_i^\top \hat{\beta}_\lambda = \hat{Y}_i$,

$$\hat{\beta}_{\lambda,(-i)} = \hat{\beta}_\lambda - G^{-1} X_i Y_i + \frac{G^{-1} X_i (\hat{Y}_i - A_{ii} Y_i)}{1 - A_{ii}} = \hat{\beta}_\lambda - \frac{G^{-1} X_i (Y_i - \hat{Y}_i)}{1 - A_{ii}}.$$

En prémultipliant par $X_i^\top$, on obtient $X_i^\top \hat{\beta}_{\lambda,(-i)} = \hat{Y}_i - A_{ii}(Y_i - \hat{Y}_i)/(1 - A_{ii})$, d'où

$$Y_i - X_i^\top \hat{\beta}_{\lambda,(-i)} = (Y_i - \hat{Y}_i)\left(1 + \frac{A_{ii}}{1 - A_{ii}}\right) = \frac{Y_i - \hat{Y}_i}{1 - A_{ii}}.$$

b) L'égalité de a) est exacte pour tout λ fixé, comme pour la régression linéaire ($\lambda = 0$, où l'on retrouve l'exercice 1 du niveau moyen du chapitre 5). La moyenne des carrés donne donc exactement le critère LOOCV, à condition que λ reste fixé lorsqu'on retire une observation. C'est ce que suppose le critère OCV du Chapitre 9 : on calcule $VC_{(n)}(\lambda)$ pour chaque λ, puis on minimise.

Exercice 6. Retour à l'énoncé

a) On écrit $X^\top X + \lambda M = R^\top (I + \lambda R^{-\top} M R^{-1}) R = R^\top U (I + \lambda D) U^\top R$, dont l'inverse est $R^{-1} U (I + \lambda D)^{-1} U^\top R^{-\top}$. Par invariance cyclique de la trace,

$$\mathrm{Tr}(A_\lambda) = \mathrm{Tr}\left\{(I + \lambda D)^{-1} U^\top R^{-\top} X^\top X R^{-1} U\right\} = \mathrm{Tr}\left\{(I + \lambda D)^{-1}\right\} = \sum_{k=1}^{q} \frac{1}{1 + \lambda d_k},$$

puisque $R^{-\top} X^\top X R^{-1} = I$ et $U^\top U = I$.

b) Comme R et U sont inversibles, le nombre de valeurs propres nulles de $R^{-\top} M R^{-1}$ est la dimension du noyau de M. Or $\beta^\top M \beta = \int f''(t)^2 dt$ pour $f = \sum_k \beta_k b_k$, et comme M est semi-définie positive, $M\beta = 0$ si et seulement si $\beta^\top M \beta = 0$, c'est-à-dire $f'' \equiv 0$ sur $[A, B]$. Les fonctions de la base dont la dérivée seconde est nulle sont les fonctions linéaires $a + bx$, qui appartiennent à l'espace des splines cubiques. Cet ensemble est de dimension 2.

c) Pour $\lambda = 0$, chaque terme vaut 1 et la somme vaut q. Chaque terme $1/(1 + \lambda d_k)$ décroît en λ si $d_k > 0$ et reste égal à 1 si $d_k = 0$. Quand $\lambda \to \infty$, les termes avec $d_k > 0$ tendent vers 0 et il reste les deux termes avec $d_k = 0$. Les degrés de liberté effectifs décroissent donc de q à 2, et la limite correspond à la droite des moindres carrés, que la pénalité ne touche pas.

Chapitre 9 : Modèles additifs et modèles additifs généralisés (GAM)

Retour aux exercices

Exercices appliqués

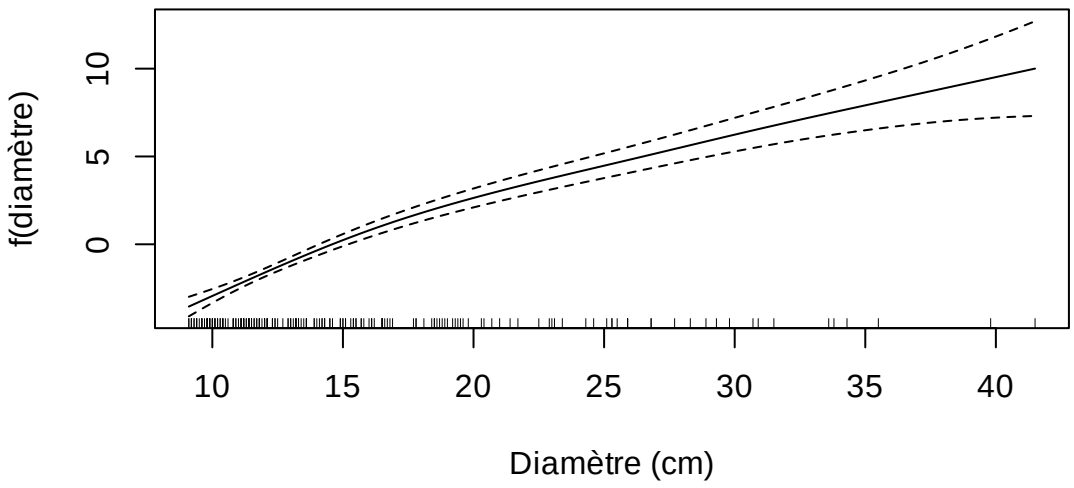
Exercice 1. Retour à l'énoncé

a) On ajuste le modèle et on trace la fonction lisse :

```
g.arbres <- gam(height_m ~ s(diameter_cm) + espece, data = arbres)
summary(g.arbres)
```

```
Family: gaussian
Link function: identity

Formula:
height_m ~ s(diameter_cm) + espece

Parametric coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           14.0580     0.3283  42.821  < 2e-16 ***
especeÉpinette noire  -1.5989     0.4658  -3.433 0.000731 ***
especeÉrable rouge     0.3866     0.4785   0.808 0.420121
especeSapin baumier   -1.5775     0.4663  -3.383 0.000867 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                 edf Ref.df     F p-value
s(diameter_cm) 2.676   3.36 95.38  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.674   Deviance explained = 68.3%
GCV = 5.5246  Scale est. = 5.3402    n = 200
```

```r
plot(g.arbres, shade = TRUE, xlab = "Diamètre (cm)", ylab = "f(diamètre)")
```



La fonction du diamètre utilise environ 2,7 degrés de liberté effectifs. Elle est croissante et concave : la hauteur augmente rapidement avec le diamètre pour les petits arbres, puis de plus en plus lentement, et la bande s'élargit au-delà de 30 cm, où les arbres sont rares. On retrouve la relation allométrique du Chapitre 5. À diamètre égal, l'épinette noire et le sapin baumier sont plus courts que le bouleau à papier.

b) On compare les deux modèles :

```
l.arbres <- lm(height_m ~ diameter_cm + espece, data = arbres)
c(GAM = AIC(g.arbres), lineaire = AIC(l.arbres))
```

```
     GAM lineaire
911.1896 919.0261
```

L'AIC du modèle additif est plus petit d'environ 8 points, malgré ses degrés de liberté supplémentaires : la courbure est réelle, sans être forte. Contrairement à l'exemple logistique du Chapitre 9, où la fonction lisse s'était réduite à une droite, le modèle additif apporte ici quelque chose.

c) L'AIC compare des log-vraisemblances, c'est-à-dire des densités de la réponse. Le modèle log-log donne la densité de $\log H$, et pas celle de H. Si $Z = \log H$ a la densité f_Z, la

densité de H est $f_H(h) = f_Z(\log h)/h$ (changement de variable). Ainsi,

$$\ell_H = \sum_i \log f_H(H_i) = \ell_Z - \sum_i \log H_i, \qquad AIC_H = -2\ell_H + 2k = AIC_Z + 2\sum_i \log H_i.$$

On fait la correction :

```
l.loglog <- lm(log(height_m) ~ log(diameter_cm) + espece, data = arbres)
c(brut = AIC(l.loglog), corrige = AIC(l.loglog) + 2 *
 ↪  sum(log(arbres$height_m)),
  GAM = AIC(g.arbres))
```

```
     brut   corrige       GAM
-96.59814 922.30493 911.18960
```

L'AIC brut du modèle log-log est négatif et n'a rien à voir avec les deux autres. Une fois ramené à l'échelle de H, il vaut environ 922, ce qui est moins bon que le modèle additif et même un peu moins bon que la droite. Sur l'échelle de H, la loi de puissance impose une forme concave trop rigide pour ces données. Notez que les deux modèles diffèrent aussi par la structure de l'erreur, additive pour le modèle additif et multiplicative pour le modèle log-log.

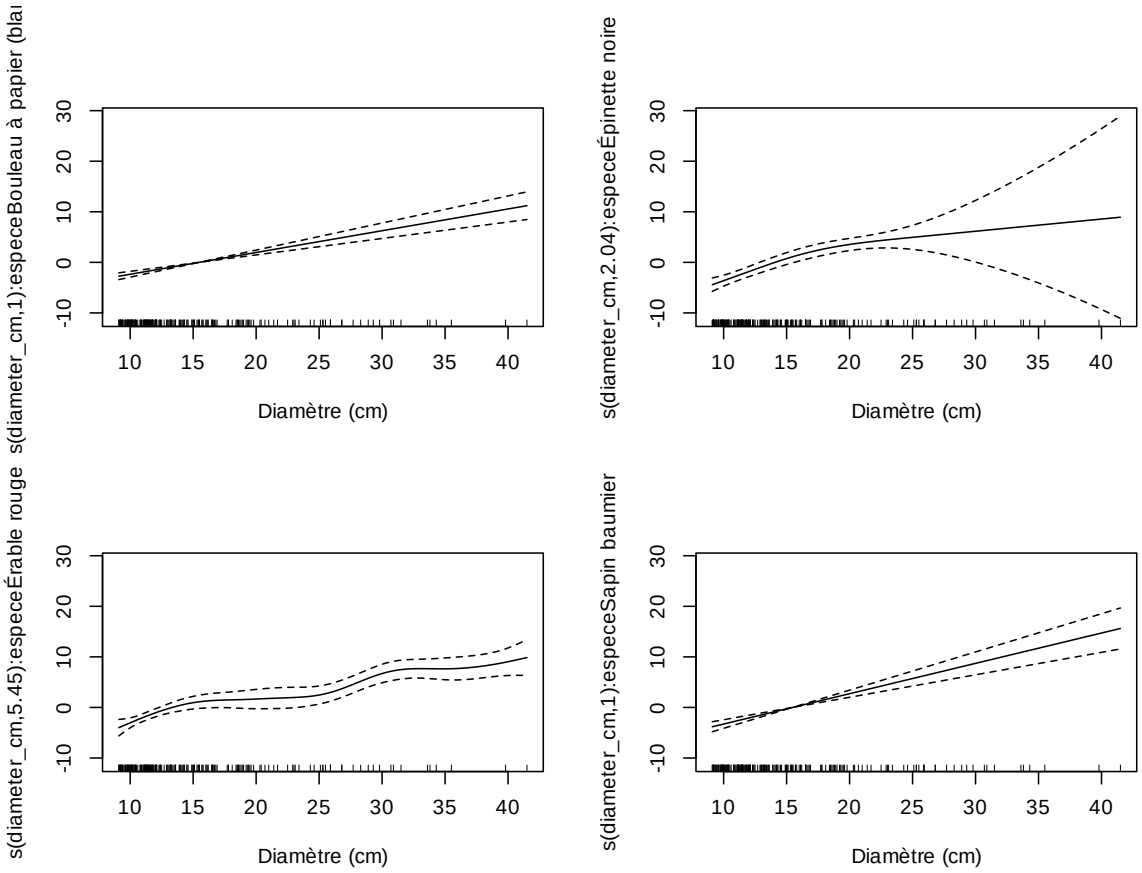
Exercice 2. Retour à l'énoncé

a) L'argument `by` crée une fonction lisse par niveau du facteur. Le terme paramétrique `espece` reste nécessaire, parce que chaque fonction est centrée :

```
g.by <- gam(height_m ~ s(diameter_cm, by = espece) + espece, data = arbres)
round(summary(g.by)$s.table, 3)
```

```
                                                edf Ref.df      F p-value
s(diameter_cm):especeBouleau à papier (blanc) 1.000  1.000 64.334       0
s(diameter_cm):especeÉpinette noire           2.042  2.560 33.033       0
s(diameter_cm):especeÉrable rouge             5.450  6.476 23.188       0
s(diameter_cm):especeSapin baumier            1.000  1.000 57.042       0
```

```
par(mfrow = c(2, 2))
plot(g.by, shade = TRUE, xlab = "Diamètre (cm)")
```



Les fonctions du bouleau à papier et du sapin baumier se réduisent à des droites (1 degré de liberté effectif), celle de l'épinette noire est légèrement courbée, alors que celle de l'érable rouge utilise plus de 5 degrés de liberté et présente des ondulations.

b) On compare les deux modèles :

```
c(commune = AIC(g.arbres), par.espece = AIC(g.by))
```

```
   commune par.espece
  911.1896   905.6466
```

```
anova(g.arbres, g.by, test = "F")
```

```
Analysis of Deviance Table
```

```
Model 1: height_m ~ s(diameter_cm) + espece
Model 2: height_m ~ s(diameter_cm, by = espece) + espece
  Resid. Df Resid. Dev     Df Deviance      F  Pr(>F)
1    192.64    1032.39
2    184.96     938.01 7.6764    94.38 2.4447 0.01687 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

L'AIC est plus petit pour le modèle avec une fonction par espèce, et le test F approximatif rejette le modèle à fonction commune au seuil de 5 %.

c) On programme la validation croisée à 10 blocs, en réutilisant la même partition pour les trois modèles :

```
set.seed(4300)
blocs <- sample(rep(1:10, length.out = nrow(arbres)))
vc.gam <- function(formule) {
  mean(sapply(1:10, function(k) {
    m <- gam(formule, data = arbres[blocs != k, ])
    mean((arbres$height_m[blocs == k] - predict(m, arbres[blocs == k, ]))^2)
  }))
}
c(lineaire = vc.gam(height_m ~ diameter_cm + espece),
  commune = vc.gam(height_m ~ s(diameter_cm) + espece),
  par.espece = vc.gam(height_m ~ s(diameter_cm, by = espece) + espece))
```

```
  lineaire    commune par.espece
  5.809622   5.665079   5.910435
```

La validation croisée donne la conclusion inverse : le modèle avec une fonction par espèce prédit moins bien que le modèle à fonction commune, et même moins bien que le modèle linéaire. Deux explications se combinent. D'abord, l'AIC et le test F traitent les paramètres de lissage comme connus, alors qu'ils sont estimés. Avec 50 arbres par espèce, la courbe de l'érable rouge suit des particularités de l'échantillon que la sélection de λ par GCV n'a pas su pénaliser, et ces deux critères ne tiennent pas compte de l'incertitude sur ce choix. Ensuite, la validation croisée réestime tout, y compris les λ, sur chaque échantillon d'entraînement, et mesure directement la qualité des prédictions. Avec un échantillon de cette taille, la fonction commune, qui emprunte l'information des quatre espèces, est le choix le plus sûr. Notez que la validation croisée à 10 blocs a elle-même une variabilité, qu'on pourrait évaluer en répétant le calcul avec d'autres partitions.

Exercice 3. Retour à l'énoncé

a) Avec des splines ordinaires :

```
m.ord <- gam(O3 ~ s(heure) + s(jour), data = longueuil)
predict(m.ord, data.frame(heure = 12, jour = c(1, 365)))
```

```
       1        2
31.30688 37.49108
```

Les deux prédictions diffèrent de 6,2 ppb, alors que deux jours consécutifs devraient donner des valeurs voisines.

b) Une spline cyclique impose que la fonction et ses deux premières dérivées se raccordent aux bornes du cycle. Les bornes $-0{,}5$ et 23,5 (ou 0,5 et 365,5) font que 23 h et 0 h, ou le 365e et le 1er jour, sont à la même distance que deux autres valeurs consécutives :

```
bornes <- list(heure = c(-0.5, 23.5), jour = c(0.5, 365.5))
m.cyc <- gam(O3 ~ s(heure, bs = "cc", k = 12) + s(jour, bs = "cc", k = 20),
             data = longueuil, knots = bornes)
predict(m.cyc, data.frame(heure = 12, jour = c(1, 365)))
```

```
       1        2
33.24509 33.32258
```

```
c(ordinaire = AIC(m.ord), cyclique = AIC(m.cyc))
```

```
ordinaire  cyclique
 63782.33  63503.20
```

Les deux prédictions sont maintenant presque égales, et l'AIC diminue d'environ 279 points. L'amélioration vient de la contrainte cyclique, mais aussi de la base plus riche pour le jour ($k = 20$ plutôt que 10).

c) La fonction `gam.check` produit des graphiques de diagnostic des résidus et un test de la dimension de base :

```
set.seed(1)
gam.check(m.cyc)
```

```
Method: GCV   Optimizer: magic
Smoothing parameter selection converged after 8 iterations.
The RMS GCV score gradient at convergence was 0.0001168144 .
The Hessian was positive definite.
Model rank =  29 / 29

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

            k'   edf k-index p-value
s(heure) 10.00  6.95    0.98   0.025 *
s(jour)  18.00 17.65    0.58  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pour `s(jour)`, les degrés de liberté effectifs atteignent presque le maximum k' et l'indice `k-index` est nettement inférieur à 1, ce qui suggère que `k` est trop petit. Avec `k = 40` :

```
m.cyc40 <- gam(O3 ~ s(heure, bs = "cc", k = 12) + s(jour, bs = "cc", k = 40),
               data = longueuil, knots = bornes)
round(summary(m.cyc40)$s.table, 2)
```

```
           edf Ref.df      F p-value
s(heure)  7.02     10 253.42       0
s(jour)  36.86     38  87.88       0
```

```
AIC(m.cyc40)
```

```
[1] 63039.8
```

La fonction du jour prend presque tous les degrés de liberté disponibles, et l'AIC diminue encore. La courbe suit maintenant des épisodes météorologiques de quelques jours. Ce n'est pas faux, puisque ces épisodes existent, mais ce n'est plus un cycle saisonnier. Le problème vient des erreurs : les heures consécutives d'une même journée sont fortement corrélées, de sorte qu'une journée anormale fournit 24 résidus de même signe. La GCV et l'AIC, qui supposent des erreurs indépendantes, interprètent ces blocs de résidus comme du signal et choisissent un lissage trop faible. Le test de `gam.check` repose sur la même hypothèse et réclamera toujours plus de flexibilité. Pour un cycle saisonnier, il vaut mieux fixer `k` d'après ce qu'on veut représenter (une vingtaine de degrés de liberté, soit une résolution d'environ trois semaines) que de suivre aveuglément ces critères.

Exercice 4. Retour à l'énoncé

a) Le terme `ti` ne contient que l'interaction, sans les effets principaux, ce qui permet de garder `s(heure)` et `s(jour)` et de lire séparément ce que l'interaction ajoute :

```
m.int <- gam(O3 ~ s(heure, bs = "cc", k = 12) + s(jour, bs = "cc", k = 20) +
               ti(heure, jour, bs = c("cc", "cc"), k = c(10, 12)),
             data = longueuil, knots = bornes)
round(summary(m.int)$s.table, 2)
```

```
                 edf Ref.df      F p-value
s(heure)        7.05     10 260.52       0
s(jour)        17.67     18 161.76       0
ti(heure,jour) 30.17     80   9.06       0
```

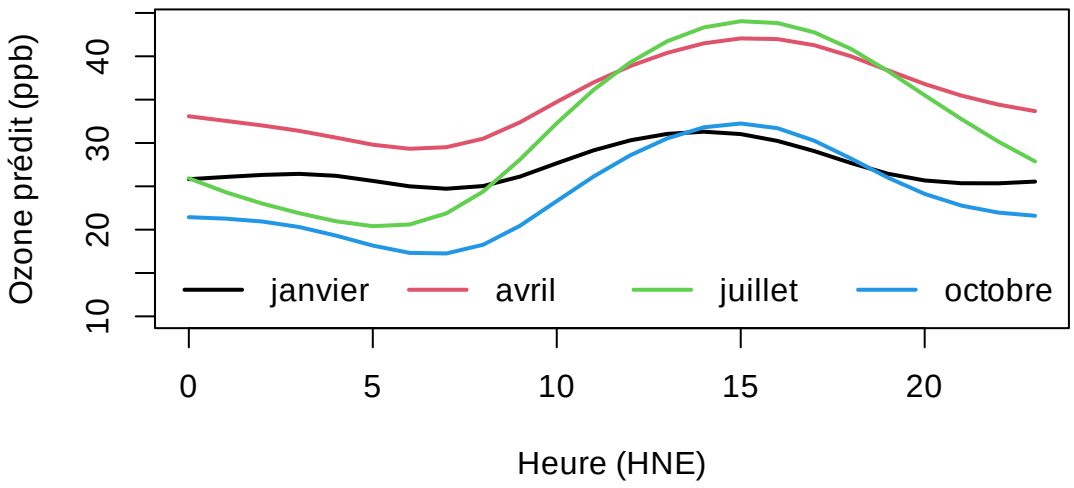
```
c(additif = AIC(m.cyc), interaction = AIC(m.int))
```

```
    additif interaction
   63503.20    62817.29
```

L'interaction utilise environ 30 degrés de liberté effectifs, et l'AIC diminue d'environ 686 points. La forme du cycle journalier change donc au cours de l'année.

b) On prédit le cycle sur une grille d'heures pour les quatre jours :

```
jours <- c(janvier = 15, avril = 105, juillet = 196, octobre = 288)
cycles <- sapply(jours, function(j) predict(m.int, data.frame(heure = 0:23,
 ↪  jour = j)))
matplot(0:23, cycles, type = "l", lty = 1, lwd = 2, col = 1:4, ylim = c(10,
 ↪  max(cycles)),
        xlab = "Heure (HNE)", ylab = "Ozone prédit (ppb)")
legend("bottom", names(jours), col = 1:4, lwd = 2, bty = "n", horiz = TRUE)
```




```
amplitude <- apply(cycles, 2, function(v) diff(range(v)))
round(amplitude, 1)
```

```
janvier   avril juillet octobre
    6.6    12.7    23.6    15.0
```

L'amplitude du cycle journalier est d'environ 7 ppb en janvier et de 24 ppb en juillet. Le modèle additif impose la même amplitude toute l'année et la sous-estime donc en été et la surestime en hiver. C'est cohérent avec la photochimie : la production d'ozone en après-midi dépend du rayonnement solaire, beaucoup plus intense en été. On remarque aussi que le niveau moyen est plus élevé au printemps qu'en été, mais avec un cycle moins marqué.

c) La valeur-p du terme `ti` est pratiquement nulle. Elle repose toutefois sur l'hypothèse d'erreurs indépendantes, alors que les mesures horaires sont fortement corrélées (exercice précédent). Le nombre effectif d'observations indépendantes est bien inférieur à 8 727, de sorte que les erreurs-types sont trop petites et les valeurs-p trop optimistes. Ici, l'effet est assez grand, et assez cohérent avec la physique, pour qu'on le retienne, mais une interaction plus subtile mériterait une méthode qui tient compte de la corrélation, par exemple un modèle à erreurs autocorrélées (`gamm` dans `mgcv`) ou une validation croisée par blocs de jours.

Exercice 5. Retour à l'énoncé

a) On agrège les données par jour, puis on ajuste le modèle :

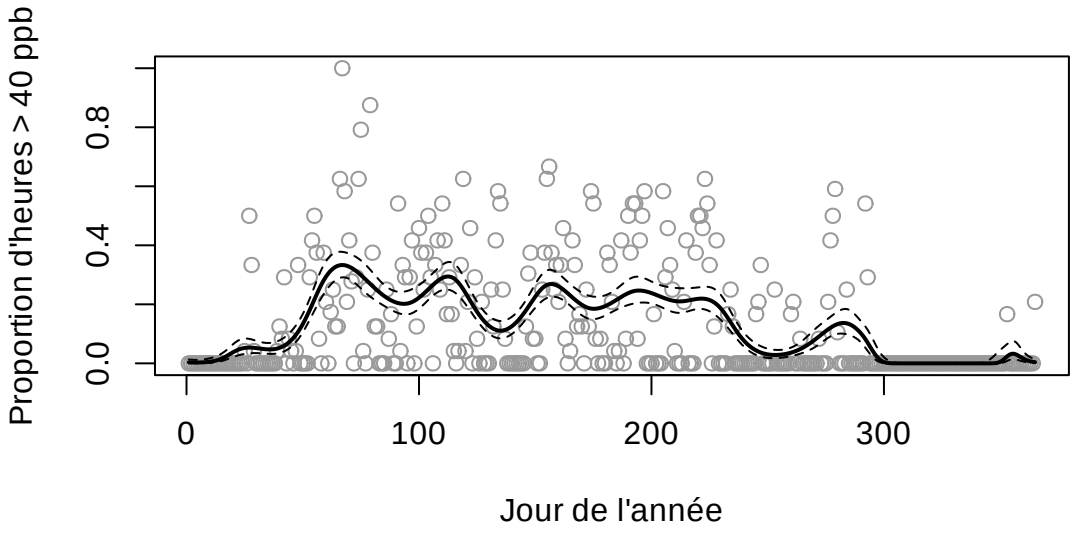
```
par.jour <- longueuil |>
  mutate(depasse = as.integer(O3 > 40)) |>
  group_by(jour) |>
  summarise(n = n(), dep = sum(depasse))
summary(par.jour$n)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.00   24.00   24.00   23.91   24.00   24.00
```

```
g.dep <- gam(cbind(dep, n - dep) ~ s(jour, bs = "cc", k = 20), family =
 ↪  binomial,
             data = par.jour, knots = list(jour = c(0.5, 365.5)))
round(summary(g.dep)$s.table, 2)
```

```
          edf Ref.df Chi.sq p-value
s(jour) 17.89     18 380.58       0
```

```
grille.j <- data.frame(jour = 1:365)
pr <- predict(g.dep, grille.j, se.fit = TRUE)
plot(par.jour$jour, par.jour$dep / par.jour$n, col = "grey60",
     xlab = "Jour de l'année", ylab = "Proportion d'heures > 40 ppb")
lines(grille.j$jour, plogis(pr$fit), lwd = 2)
lines(grille.j$jour, plogis(pr$fit + 2 * pr$se.fit), lty = 2)
lines(grille.j$jour, plogis(pr$fit - 2 * pr$se.fit), lty = 2)
```



Les dépassements sont pratiquement absents de novembre à janvier. Ils deviennent fréquents de mars à août, avec plusieurs sommets, puis se raréfient en septembre et en octobre. La proportion journalière observée est très dispersée autour de la courbe : de nombreux jours n'ont aucune heure de dépassement, alors que d'autres en comptent plus de la moitié. La courbe elle-même ondule beaucoup, avec presque tous ses degrés de liberté, ce qui annonce le problème de la question suivante.

b) On calcule la dispersion :

```
phi.dep <- sum(residuals(g.dep, type = "pearson")^2) / df.residual(g.dep)
phi.dep
```

```
[1] 5.893197
```

La dispersion estimée vaut environ 5,9, alors que le modèle binomial la fixe à 1 : il y a une forte surdispersion. Le modèle binomial traite les 24 heures d'une journée comme 24 essais indépendants de même probabilité. En réalité, une journée chaude et ensoleillée produit des dépassements pendant plusieurs heures consécutives, et une journée nuageuse n'en produit aucun : les heures d'une même journée sont fortement corrélées, et la variance du nombre de dépassements dépasse largement $m_i \pi_i (1 - \pi_i)$.

c) On réajuste le modèle avec la famille quasibinomiale :

```
g.dep.q <- gam(cbind(dep, n - dep) ~ s(jour, bs = "cc", k = 20), family =
 ↪  quasibinomial,
               data = par.jour, knots = list(jour = c(0.5, 365.5)))
round(summary(g.dep.q)$s.table, 2)
```

```
          edf Ref.df    F p-value
s(jour) 13.55     18 4.03       0
```

```
g.dep.q$scale
```

```
[1] 6.359484
```

Avec la famille quasibinomiale, le paramètre de dispersion est estimé (environ 6,4) et le critère de lissage passe de l'UBRE, qui suppose $\phi = 1$, à la GCV. Les degrés de liberté effectifs diminuent, de 17,9 à 13,6 : en supposant à tort $\phi = 1$, l'UBRE prend la variabilité d'un jour à l'autre pour du signal et sous-lisse. Les erreurs-types sont multipliées par environ $\sqrt{\hat{\phi}} \approx 2{,}5$, de sorte que les bandes sont nettement plus larges. C'est le même mécanisme que pour la surdispersion dans les GLM (Chapitre 3), avec en plus un effet sur le choix du lissage.

Exercice 6. Retour à l'énoncé

a) On place la pénalité de la fonction lisse dans une matrice nulle de la taille de β :

```
X <- model.matrix(g.arbres)
y <- arbres$height_m
lisse <- g.arbres$smooth[[1]]
indices <- lisse$first.para:lisse$last.para
S <- matrix(0, ncol(X), ncol(X))
S[indices, indices] <- g.arbres$sp[1] * lisse$S[[1]]

beta <- solve(crossprod(X) + S, crossprod(X, y))
max(abs(beta - coef(g.arbres)))
```

```
[1] 3.126677e-11
```

L'écart maximal est de l'ordre de l'erreur d'arrondi : pour λ fixé, l'estimateur d'un modèle additif gaussien est bien l'estimateur de type ridge du Chapitre 9. Les coefficients paramétriques (ordonnée à l'origine et espèces) ne sont pas pénalisés, ce qui correspond aux lignes et colonnes nulles de S.

b) On calcule la matrice d'influence :

```
A <- X %*% solve(crossprod(X) + S, t(X))
c(trace = sum(diag(A)), edf = sum(g.arbres$edf))
```

```
   trace      edf
6.675875 6.675875
```

Les deux valeurs coïncident : `gam` calcule les degrés de liberté effectifs comme la trace de la matrice d'influence, répartie entre les coefficients. Les 4 paramètres non pénalisés comptent chacun pour 1, et le reste revient à la fonction du diamètre.

c) On applique les définitions du Chapitre 9 :

```
n <- length(y)
y.chap <- as.vector(A %*% y)
gcv <- n * sum((y - y.chap)^2) / (n - sum(diag(A)))^2
ocv <- mean(((y - y.chap) / (1 - diag(A)))^2)
c(gcv = gcv, gam = g.arbres$gcv.ubre, ocv = ocv)
```

```
      gcv gam.GCV.Cp       ocv
 5.524598   5.524598   5.614425
```

La GCV calculée à la main est exactement celle que rapporte `gam`, qui l'a minimisée pour choisir λ. Le critère OCV est un peu plus grand, parce que quelques observations ont un levier A_{ii} nettement supérieur à la moyenne : les deux plus gros arbres ont un levier sept à dix fois plus grand que la moyenne $\mathrm{Tr}(A)/n$. La GCV, qui remplace chaque A_{ii} par cette moyenne, atténue leur poids.

Exercices théoriques

Exercice 1. Retour à l'énoncé

a) Dans les deux cas, $\alpha + f_1(x_{i1}) + f_2(x_{i2})$ est inchangé : $c - c = 0$ dans le premier, $-c + c = 0$ dans le second. Une infinité de paramétrisations décrivent donc le même modèle, et les données ne peuvent pas les distinguer.

b) En sommant $E[Y_i] = \alpha + f_1(x_{i1}) + f_2(x_{i2})$ sur i et en utilisant les contraintes, on obtient $\sum_i E[Y_i] = n\alpha$, d'où $\alpha = \frac{1}{n}\sum_i E[Y_i]$. La contrainte fixe donc la constante : on ne peut plus la déplacer d'une fonction à l'autre. De même, si $\sum_i \hat{Y}_i = \sum_i Y_i$ (ce qui est le cas lorsque l'ordonnée à l'origine n'est pas pénalisée, puisque les équations d'estimation pour α imposent que les résidus somment à 0), alors $n\hat{\alpha} = \sum_i \hat{Y}_i = \sum_i Y_i$, soit $\hat{\alpha} = \bar{Y}$.

c) Chaque $\hat{f}_j$ est une fonction centrée : elle représente l'écart à la moyenne attribuable à x_j, les autres variables étant fixées. Seules sa forme et ses différences entre deux valeurs de x_j s'interprètent ($\hat{f}_j(b) - \hat{f}_j(a)$ est l'effet estimé du passage de a à b), et non son niveau absolu. Pour obtenir une prédiction, on additionne $\hat{\alpha}$ et toutes les fonctions.

Exercice 2. Retour à l'énoncé

a) Avec $\eta_i = \log\{\pi_i/(1-\pi_i)\}$, on a $\partial\eta_i/\partial\pi_i = 1/\{\pi_i(1-\pi_i)\}$, donc $\partial\pi_i/\partial\eta_i = \pi_i(1-\pi_i)$. Comme $V(\pi_i) = \pi_i(1-\pi_i)$,

$$z_i = \eta_i + \frac{y_i - \pi_i}{\pi_i(1-\pi_i)}, \qquad w_i = \frac{\{\pi_i(1-\pi_i)\}^2}{\pi_i(1-\pi_i)} = \pi_i(1-\pi_i).$$

On retrouve les poids de la régression logistique (exercice 3 du niveau facile du chapitre 2).

b) Avec $\eta_i = \log\mu_i$, on a $\partial\eta_i/\partial\mu_i = 1/\mu_i$ et $V(\mu_i) = \mu_i$, d'où

$$z_i = \eta_i + \frac{y_i - \mu_i}{\mu_i}, \qquad w_i = \frac{\mu_i^2}{\mu_i} = \mu_i.$$

c) Avec le lien identité et la loi normale, $\eta_i = \mu_i$, $\partial\eta_i/\partial\mu_i = 1$ et $V(\mu_i) = 1$. La réponse de travail est donc $z_i = \mu_i^{(m)} + (y_i - \mu_i^{(m)}) = y_i$ et les poids valent $w_i = 1$, quelle que soit l'itération. La première itération ajuste donc le modèle additif pénalisé aux y_i avec des poids égaux, ce qui donne directement $\hat{\beta} = (X^\top X + \sum_j \lambda_j S_j)^{-1} X^\top Y$. L'itération suivante reproduit la même solution.

Exercice 3. Retour à l'énoncé

a) Avec $A_{ii} = \mathrm{Tr}(A)/n$ pour tout i, chaque dénominateur devient $\{1 - \mathrm{Tr}(A)/n\}^2 = \{n - \mathrm{Tr}(A)\}^2/n^2$, et

$$\frac{1}{n}\sum_i \frac{(y_i - \hat{y}_i)^2\, n^2}{\{n - \mathrm{Tr}(A)\}^2} = \frac{n\, RSS}{\{n - \mathrm{Tr}(A)\}^2} = GCV.$$

b) On écrit $GCV = \frac{RSS}{n}\{1 - \mathrm{Tr}(A)/n\}^{-2}$ et on utilise $(1-x)^{-2} \approx 1 + 2x$ pour x petit. Le critère UBRE de la déviance gaussienne ($D = RSS$) vaut

$$UBRE = \frac{RSS}{n} - \sigma^2 + \frac{2\sigma^2\, \mathrm{Tr}(A)}{n}.$$

Au terme constant $-\sigma^2$ près, qui ne change pas le λ choisi, les deux critères ont la même forme : l'erreur d'entraînement plus une pénalité proportionnelle aux degrés de liberté effectifs. La GCV remplace σ^2, inconnu, par l'estimation RSS/n. C'est la même correction d'optimisme que celle de l'exercice 2 du niveau difficile du chapitre 5, avec $\mathrm{Tr}(A)$ à la place de $p+1$.

c) La GCV ne dépend des A_{ii} que par leur somme, qui se calcule sans former la matrice A de taille $n \times n$. Elle est aussi moins sensible aux observations de levier très élevé, dont le terme $1/(1-A_{ii})^2$ peut dominer $VC_{(n)}$. Enfin, contrairement à $VC_{(n)}$, elle est invariante par rotation des données ($Y \mapsto QY$, $X \mapsto QX$ avec Q orthogonale), ce qui a motivé son introduction.

Exercice 4. Retour à l'énoncé

a) Par le théorème de Bayes, la densité a posteriori est proportionnelle au produit de la vraisemblance et de la densité a priori :

$$\pi(\beta \mid Y) \propto \exp\left\{-\frac{\|Y - X\beta\|^2}{2\sigma^2}\right\} \exp\left\{-\frac{\beta^\top P \beta}{2\sigma^2}\right\},$$

d'où le résultat. Maximiser cette densité revient à minimiser les moindres carrés pénalisés : l'estimateur pénalisé est le mode a posteriori.

b) Notons $G = X^\top X + P$ et $\hat{\beta} = G^{-1} X^\top Y$. En développant,

$$\|Y - X\beta\|^2 + \beta^\top P \beta = \beta^\top G \beta - 2\beta^\top X^\top Y + Y^\top Y = (\beta - \hat{\beta})^\top G (\beta - \hat{\beta}) + Y^\top Y - \hat{\beta}^\top G \hat{\beta},$$

puisque $G\hat{\beta} = X^\top Y$. Les deux derniers termes ne dépendent pas de β, de sorte que

$$\pi(\beta \mid Y) \propto \exp\left\{-\frac{1}{2}(\beta - \hat{\beta})^\top \left(\sigma^2 G^{-1}\right)^{-1} (\beta - \hat{\beta})\right\},$$

le noyau d'une loi $\mathcal{N}(\hat{\beta}, \sigma^2 G^{-1})$.

c) C'est la matrice $V_\beta = \sigma^2 (X^\top X + \sum_j \lambda_j S_j)^{-1}$ du Chapitre 9, plus simple que la variance fréquentiste de l'estimateur pénalisé. Les intervalles $\hat{f}_j(x) \pm 1{,}96\sqrt{\hat{\mathrm{V}}(\hat{f}_j(x))}$ tracés par `plot` en découlent. Ce sont des intervalles de crédibilité a posteriori, ce qui explique qu'ils tiennent compte du biais dû à la pénalité et qu'ils aient une bonne couverture fréquentiste en moyenne sur les x. L'exercice 3 du niveau moyen du chapitre 6 montrait que ridge revient à ajouter des pseudo-observations nulles pour chaque coefficient : la loi a priori centrée en 0 joue exactement ce rôle. Ridge, les splines de lissage et les modèles additifs partagent donc la même structure : une estimation par moindres carrés pénalisés, qui est aussi une moyenne a posteriori.

Chapitre 10 : Données corrélées

Retour aux exercices

Niveau facile

Exercice 1. Retour à l'énoncé

On a $Var[\bar{Y}] = N^{-2} \sum_{i,j} \sum_{l,k} Cov[Y_{ij}, Y_{lk}]$. Les covariances entre groupes sont nulles. Dans chaque groupe, il y a c variances σ_Y^2 et $c(c-1)$ covariances $\rho\sigma_Y^2$, de sorte que

$$Var[\bar{Y}] = \frac{m\left\{c\sigma_Y^2 + c(c-1)\rho\sigma_Y^2\right\}}{N^2} = \frac{\sigma_Y^2}{N}\left\{1 + (c-1)\rho\right\},$$

puisque $mc = N$. Avec $c = 10$ et $\rho = 0{,}3$, l'effet du plan vaut $1 + 9 \times 0{,}3 = 3{,}7$, et la taille effective est $N/3{,}7$: 200 observations valent environ 54 observations indépendantes.

Exercice 2. Retour à l'énoncé

a) $Var[Y_{ij}] = \sigma_b^2 + \sigma^2 = 4 + 16 = 20$ et $Cov[Y_{ij}, Y_{ik}] = Var[b_i] = 4$, puisque b_i est le seul terme aléatoire commun aux deux observations. La corrélation intraclasse vaut $4/20 = 0{,}2$.

b) Les deux observations ne partagent aucun terme aléatoire, et b_i, b_l, ϵ_{ij} et ϵ_{lk} sont indépendants, de sorte que la covariance est nulle.

Exercice 3. Retour à l'énoncé

Le facteur de rétrécissement vaut $n_i\sigma_b^2/(n_i\sigma_b^2 + \sigma^2)$. Pour le premier groupe, il vaut $1/(1+4) = 0{,}2$, d'où $\widehat{b}_1 = 0{,}4$. Pour le second, il vaut $16/(16+4) = 0{,}8$, d'où $\widehat{b}_2 = 1{,}6$. Le résidu moyen du premier groupe repose sur une seule observation, dont la variance d'erreur ($\sigma^2 = 4$) est grande devant la variance des effets de groupe ($\sigma_b^2 = 1$). Le modèle juge donc que l'écart observé est surtout du bruit et le ramène fortement vers 0. Le second groupe fournit une moyenne de 16 observations, de variance $4/16 = 0{,}25$, et son écart est presque entièrement attribué à l'effet du groupe.

Exercice 4. Retour à l'énoncé

a) $Var[Y_{ij}] = d_{00} + 2td_{01} + t^2 d_{11} + \sigma^2$. Au jour 0, elle vaut $612 + 655 = 1\,267$, soit un écart-type d'environ 36 ms. Au jour 9, elle vaut $612 + 2 \times 9 \times 9{,}6 + 81 \times 35{,}1 + 655 \approx 4\,283$, soit un écart-type d'environ 65 ms. La pente aléatoire fait croître la dispersion entre sujets avec le temps, puisque les droites individuelles s'écartent les unes des autres.

b) $Cov[Y_{i0}, Y_{i9}] = Cov[b_{0i},\, b_{0i} + 9b_{1i}] = d_{00} + 9d_{01} = 612 + 86{,}4 = 698{,}4$, d'où la corrélation

$$\frac{698{,}4}{\sqrt{1\,267 \times 4\,283}} \approx 0{,}30.$$

Pour les jours 8 et 9, $Cov = d_{00} + 17d_{01} + 72d_{11} \approx 3\,302$, $Var[Y_{i8}] \approx 3\,667$, et la corrélation vaut environ 0,83. Avec une ordonnée à l'origine aléatoire seule, la corrélation entre deux mesures d'un même sujet est la même pour toutes les paires de jours : $1\,378/(1\,378 + 960) \approx 0{,}59$, une valeur intermédiaire. Avec une pente aléatoire, elle

dépend des deux jours : la corrélation des jours 8 et 9 est beaucoup plus forte que celle des jours 0 et 9, parce que la pente du sujet, qui domine la variance en fin d'étude, agit presque de la même façon sur les deux.

Exercice 5. Retour à l'énoncé

a) Avec la loi χ^2_1, $P(\chi^2_1 \geq 3{,}2) \approx 0{,}074$, et on ne rejette pas H_0 au seuil de 5 %. Avec le mélange, la valeur-p vaut $\frac{1}{2} \times 0 + \frac{1}{2} \times 0{,}074 = 0{,}037$ (la masse en 0 ne contribue pas, puisque $3{,}2 > 0$), et on rejette H_0. Le test usuel est conservateur : il conclut moins souvent à un effet de groupe qu'il ne le devrait.

b) On cherche c tel que $\frac{1}{2}P(\chi^2_1 \geq c) = 0{,}05$, soit $P(\chi^2_1 \geq c) = 0{,}10$. La valeur critique est donc le quantile d'ordre 0,90 de la loi χ^2_1, environ 2,71, au lieu de 3,84.

Niveau moyen

Exercice 1. Retour à l'énoncé

a) Notons $a = \sigma_b^2/(\sigma^2 + n_i\sigma_b^2)$ et utilisons $1^\top 1 = n_i$:

$$\left(\sigma^2 I + \sigma_b^2 11^\top\right) \frac{1}{\sigma^2}\left(I - a11^\top\right) = I + \left(\frac{\sigma_b^2}{\sigma^2} - a - \frac{n_i\sigma_b^2 a}{\sigma^2}\right) 11^\top.$$

Le coefficient entre parenthèses vaut $\sigma_b^2/\sigma^2 - a(\sigma^2 + n_i\sigma_b^2)/\sigma^2 = \sigma_b^2/\sigma^2 - \sigma_b^2/\sigma^2 = 0$, d'où le résultat. C'est un cas particulier de la formule de Sherman-Morrison (exercice 1 du niveau difficile du chapitre 5).

b) On calcule

$$1^\top V_i^{-1} = \frac{1}{\sigma^2}\left(1^\top - an_i 1^\top\right) = \frac{1}{\sigma^2} \cdot \frac{\sigma^2}{\sigma^2 + n_i\sigma_b^2} 1^\top = \frac{1^\top}{\sigma^2 + n_i\sigma_b^2}.$$

Comme $1^\top(y_i - X_i\hat{\beta}) = n_i(\bar{y}_i - \bar{x}_i^\top\hat{\beta})$,

$$\hat{b}_i = \frac{\sigma_b^2\, n_i(\bar{y}_i - \bar{x}_i^\top\hat{\beta})}{\sigma^2 + n_i\sigma_b^2} = \frac{n_i\sigma_b^2}{n_i\sigma_b^2 + \sigma^2}\left(\bar{y}_i - \bar{x}_i^\top\hat{\beta}\right).$$

Exercice 2. Retour à l'énoncé

La dérivée du critère par rapport à b vaut $-2\sum_j (r_j - b) + 2\lambda b = -2n_i\bar{r} + 2(n_i + \lambda)b$, qui s'annule en $\hat{b} = n_i\bar{r}/(n_i + \lambda)$ (un minimum, puisque la dérivée seconde $2(n_i + \lambda)$ est positive). Avec $\lambda = \sigma^2/\sigma_b^2$,

$$\hat{b} = \frac{n_i\bar{r}}{n_i + \sigma^2/\sigma_b^2} = \frac{n_i\sigma_b^2}{n_i\sigma_b^2 + \sigma^2}\,\bar{r},$$

le BLUP de l'exercice précédent. Quand $\sigma_b^2 \to \infty$, $\lambda \to 0$ et $\hat{b} \to \bar{r}$: sans pénalité, chaque groupe a son propre effet fixe. Quand $\sigma_b^2 \to 0$, $\lambda \to \infty$ et $\hat{b} \to 0$: tous les groupes sont confondus, comme dans le modèle linéaire ordinaire. Le modèle mixte est l'intermédiaire entre ces deux extrêmes, et les données choisissent le degré de mise en commun par l'estimation de σ_b^2.

Exercice 3. Retour à l'énoncé

a) Conditionnellement à b_i, $P(Y_{ij} = 1 \mid b_i) = P(Z \leq \eta_{ij} + b_i \mid b_i)$. En prenant l'espérance par rapport à b_i,

$$P(Y_{ij} = 1) = P(Z \leq \eta_{ij} + b_i) = P(Z - b_i \leq \eta_{ij}).$$

Comme Z et b_i sont des variables normales indépendantes, $Z - b_i \sim \mathcal{N}(0, 1 + \sigma_b^2)$, d'où $P(Y_{ij} = 1) = \Phi\left(\eta_{ij}/\sqrt{1 + \sigma_b^2}\right)$.

b) La probabilité marginale suit donc encore un modèle probit, de prédicteur linéaire $x_{ij}^\top \beta / \sqrt{1 + \sigma_b^2}$, ce qui donne les coefficients marginaux annoncés. Pour $\sigma_b = 1$, le facteur vaut $1/\sqrt{2} \approx 0{,}71$: l'effet d'une variable sur la probabilité moyenne de la population est d'environ 30 % plus faible que son effet pour un groupe donné. Ce n'est pas une contradiction, mais deux questions différentes.

Exercice 4. Retour à l'énoncé

a) À des constantes près qui ne dépendent pas de b,

$$h_i(b) = -\frac{1}{2\sigma^2} \sum_{j=1}^{n_i} (r_j - b)^2 - \frac{b^2}{2\sigma_b^2},$$

un polynôme de degré 2 en b. Son développement de Taylor d'ordre 2 autour du mode est donc exact, et l'intégrale d'une exponentielle de polynôme du second degré se calcule exactement : c'est la formule de l'approximation de Laplace. Ce résultat explique pourquoi l'estimation d'un modèle linéaire mixte ne demande aucune approximation numérique de l'intégrale.

b) On a

$$h_i'(b) = \frac{1}{\sigma^2} \sum_j (r_j - b) - \frac{b}{\sigma_b^2} = \frac{n_i \bar{r}}{\sigma^2} - b\left(\frac{n_i}{\sigma^2} + \frac{1}{\sigma_b^2}\right),$$

qui s'annule en

$$\tilde{b}_i = \frac{n_i \bar{r}/\sigma^2}{n_i/\sigma^2 + 1/\sigma_b^2} = \frac{n_i \sigma_b^2}{n_i \sigma_b^2 + \sigma^2}\, \bar{r},$$

le BLUP. Ensuite, $-h_i''(b) = n_i/\sigma^2 + 1/\sigma_b^2$, d'où

$$\frac{1}{-h_i''(\tilde{b}_i)} = \frac{\sigma^2 \sigma_b^2}{n_i \sigma_b^2 + \sigma^2}.$$

Par ailleurs, la formule générale du Chapitre 10 donne $Var[b_i \mid y_i] = \sigma_b^2 - \sigma_b^4 1^\top V_i^{-1} 1$, et l'exercice 1 du niveau moyen donne $1^\top V_i^{-1} 1 = n_i/(\sigma^2 + n_i\sigma_b^2)$, d'où

$$Var[b_i \mid y_i] = \sigma_b^2 - \frac{n_i\sigma_b^4}{\sigma^2 + n_i\sigma_b^2} = \frac{\sigma^2\sigma_b^2}{n_i\sigma_b^2 + \sigma^2}.$$

Les deux expressions coïncident : dans le cas normal, la loi a posteriori de b_i est exactement la loi normale de l'approximation de Laplace. Pour un GLMM, cette loi normale n'est qu'une approximation de la loi conditionnelle de b_i.

Niveau difficile

Exercice 1. Retour à l'énoncé

a) Avec le lien identité, $\mu_i = X_i\beta$ et $\Delta_i = X_i$. Avec $V(\mu) = 1$ et $R_i = I$, $V_i = \phi I$, et les équations deviennent $\phi^{-1}\sum_i X_i^\top(y_i - X_i\beta) = 0$, soit $X^\top(y - X\beta) = 0$: ce sont les équations normales, dont la solution est l'estimateur des moindres carrés.

b) Ici, $B = \phi^{-1}\sum_i X_i^\top X_i = \phi^{-1}X^\top X$, et le terme central vaut $\phi^{-2}\sum_i X_i^\top e_i e_i^\top X_i$. Les facteurs ϕ s'annulent, ce qui donne la formule annoncée. L'estimateur usuel suppose $Var[y_i] = \sigma^2 I$ et remplace le terme central par $\hat\sigma^2 X^\top X$. Le sandwich estimé, lui, la variance de $X_i^\top y_i$ par le produit observé $X_i^\top e_i e_i^\top X_i$, sans hypothèse sur la structure de $Var[y_i]$. Comme ces termes sont indépendants d'un groupe à l'autre, leur somme estime de façon convergente la variance du score quand $m \to \infty$.

c) Avec $n_i = 1$, $X_i^\top e_i e_i^\top X_i = e_i^2 x_i x_i^\top$, et on obtient l'estimateur de White,

$$(X^\top X)^{-1}\left(\sum_{i=1}^{n} e_i^2 x_i x_i^\top\right)(X^\top X)^{-1},$$

qui reste valide lorsque la variance des erreurs n'est pas constante. Il relâche l'hypothèse d'homoscédasticité, alors que la version par groupes relâche en plus l'indépendance à l'intérieur des groupes.

Exercice 2. Retour à l'énoncé

a) On suit la recette du Chapitre 1 :

```
library(lme4)
set.seed(4300)
B <- 300; m <- 20; c.grp <- 10
couv <- t(replicate(B, {
  groupe <- rep(1:m, each = c.grp)
  x <- rep(rnorm(m), each = c.grp)
```

```
  y <- x + rep(rnorm(m, sd = sqrt(0.3)), each = c.grp) + rnorm(m * c.grp, sd
 ↪  = sqrt(0.7))
  ic.lm <- confint(lm(y ~ x))["x", ]
  mod <- lmer(y ~ x + (1 | groupe))
  est <- fixef(mod)[["x"]]
  et <- sqrt(vcov(mod)["x", "x"])
  c(lm = ic.lm[[1]] <= 1 & 1 <= ic.lm[[2]],
    lmer.z = abs(est - 1) <= qnorm(0.975) * et,
    lmer.t = abs(est - 1) <= qt(0.975, df = m - 2) * et)
}))
couverture <- colMeans(couv)
rbind(couverture = couverture, erreur.type.MC = sqrt(couverture * (1 -
 ↪  couverture) / B))
```

```
                       lm     lmer.z     lmer.t
couverture     0.66666667 0.90333333 0.93333333
erreur.type.MC 0.02721655 0.01706089 0.01440165
```

b) L'intervalle de `lm` a une couverture d'environ 67 %, très loin de 95 %. La variable explicative est constante dans chaque groupe, et toute l'information sur la pente vient des 20 moyennes de groupe. L'effet du plan vaut $1+(c-1)\rho = 1+9\times 0{,}3 = 3{,}7$, de sorte que l'erreur-type naïve est trop petite d'un facteur proche de $\sqrt{3{,}7} \approx 1{,}9$. Le modèle mixte corrige l'essentiel du problème. Sa couverture avec le quantile normal (90 %) reste un peu faible, parce qu'avec 20 groupes seulement, l'incertitude sur les composantes de variance n'est pas négligeable. Le quantile de Student à $m-2$ degrés de liberté, qui reflète le nombre de groupes plutôt que le nombre d'observations, ramène la couverture près du niveau nominal (93 %, compte tenu de l'erreur de Monte Carlo).

Exercice 3. Retour à l'énoncé

a) et b) On ajuste les deux modèles, puis on calcule les probabilités marginales du GLMM par intégration numérique.

```
library(lme4)
library(geepack)
data(ohio, package = "geepack")
glmm <- glmer(resp ~ age + smoke + (1 | id), family = binomial, data = ohio,
 ↪  nAGQ = 25)
gee  <- geeglm(resp ~ age + smoke, id = id, family = binomial,
               corstr = "exchangeable", data = ohio)
s.b <- attr(VarCorr(glmm)$id, "stddev")[[1]]
```

```r
grille <- expand.grid(age = -2:1, smoke = 0:1)
X <- model.matrix(~ age + smoke, grille)
eta.glmm <- as.vector(X %*% fixef(glmm))
# proportions observées, dans l'ordre de la grille (age varie le plus vite)
observe <- ohio |>
  group_by(smoke, age) |>
  summarise(observe = mean(resp), .groups = "drop")
grille <- grille |>
  mutate(GLMM.marginal = sapply(eta.glmm, function(e)
           integrate(function(u) plogis(e + u) * dnorm(u, 0, s.b), -Inf,
             ↪ Inf)$value),
         GEE = plogis(as.vector(X %*% coef(gee))),
         observe = observe$observe,
         GLMM.typique = plogis(eta.glmm))
round(grille, 3)
```

```
  age smoke GLMM.marginal   GEE observe GLMM.typique
1  -2     0         0.162 0.161   0.160        0.060
2  -1     0         0.147 0.146   0.149        0.051
3   0     0         0.133 0.132   0.143        0.043
4   1     0         0.120 0.120   0.106        0.036
5  -2     1         0.199 0.200   0.166        0.087
6  -1     1         0.182 0.182   0.209        0.074
7   0     1         0.166 0.166   0.187        0.063
8   1     1         0.151 0.151   0.139        0.053
```

Les probabilités marginales du GLMM et celles du GEE ne diffèrent que de quelques millièmes, et toutes deux suivent de près les proportions observées. Les deux modèles décrivent donc la même moyenne marginale, même si leurs coefficients sont très différents. Les petits écarts avec les proportions observées viennent de ce que les deux modèles imposent un effet linéaire de l'âge et un effet du tabagisme maternel identique à tous les âges, sur l'échelle logit.

c) La probabilité de l'enfant typique, $\text{logit}^{-1}(x^\top \hat{\beta})$, est de 4 à 9 % environ, alors que les proportions observées sont de 11 à 21 %. Avec $\hat{\sigma}_b \approx 2,2$, la loi des effets aléatoires est très étalée sur l'échelle logit. La plupart des enfants ont une probabilité inférieure à celle de l'enfant typique ou proche d'elle, mais les enfants du haut de la distribution ont des probabilités beaucoup plus grandes. Plus précisément, pour $\eta < 0$, on a $\{\text{logit}^{-1}(\eta + u) + \text{logit}^{-1}(\eta - u)\}/2 \geq \text{logit}^{-1}(\eta)$ pour tout u. Comme la loi de b est symétrique, la moyenne des probabilités dépasse donc la probabilité évaluée en $b = 0$, et nettement quand σ_b est grand. La probabilité de l'enfant typique n'est donc pas une estimation de la fréquence de la respiration sifflante dans la population.

Références complémentaires

Je vous suggère les ouvrages suivants si vous voulez approfondir le contenu ou voir les mêmes idées présentées autrement. La première ligne de chaque ouvrage indique les chapitres du livre concernés.

Quelques ouvrages ont été suggérés par Claude, le modèle de langage qui a participé à la rédaction de ce livre. Les autres sont des suggestions personnelles.

Les petits textes descriptifs sont des commentaires sur le contenu et l'utilité de chaque ouvrage. Ces textes ont été rédigés par Claude, puis relus et validés par moi. Ils ne sont pas des citations des auteurs.

Dunn et Smyth (2018). *Generalized Linear Models with Examples in R*

Chapitres 0, 2 et 3.

Springer (Dunn et Smyth 2018). Le livre part de la régression linéaire et de ses diagnostics, puis couvre les GLM binomiaux, de Poisson et gamma, la déviance, les diagnostics, la quasi-vraisemblance et la surdispersion, avec des exemples tirés des jeux de données de la librairie `GLMsData`, celle-là même qui sert dans les notes.

Cornillon, Hengartner, Matzner-Løber et Rouvière (2023). *Régression avec R*

Chapitres 0, 2, 3, 5, 6, 7 et 8.

3e édition, EDP Sciences (Cornillon et al. 2023). Un des rares manuels en français sur le sujet. Il couvre la régression linéaire et sa validation, la sélection de variables, la régularisation, les régressions logistique et de Poisson (avec les courbes ROC et le choix du seuil) et la régression non paramétrique (splines, noyaux et plus proches voisins), avec le code R. C'est le meilleur point de départ si vous préférez lire en français.

Morris, White et Crowther (2019). « Using simulation studies to evaluate statistical methods »

Chapitre 1.

Statistics in Medicine, 38(11), 2074–2102 (Morris et al. 2019). Article en libre accès. Il explique comment planifier une étude par simulation : choix des scénarios et du nombre de répétitions, mesures de performance (biais, erreur quadratique moyenne, couverture) et erreur de Monte Carlo de chacune.

McCullagh et Nelder (1989). *Generalized Linear Models*

Chapitres 2, 3 et 4.

2e édition, Chapman & Hall (McCullagh et Nelder 1989). C'est l'ouvrage fondateur des GLM, à lire si vous voulez la théorie à la source. Il consacre aussi un chapitre aux données binaires et un autre aux réponses polytomiques, nominales et ordinales.

Faraway (2016). *Extending the Linear Model with R*

Chapitres 2, 3, 4, 7, 8, 9 et 10.

2e édition, Chapman & Hall/CRC (Faraway 2016). Le livre aborde les GLM, les réponses multinomiales et ordinales, la régression non paramétrique, les GAM, les modèles mixtes et les GEE par l'analyse de données en R, avec un minimum de théorie.

Agresti (2013). *Categorical Data Analysis*

Chapitres 2, 3, 4 et 10.

3e édition, Wiley (Agresti 2013). La référence sur les réponses catégorielles : introduction aux GLM, régression logistique, modèles multinomiaux et ordinaux, et modèles pour réponses catégorielles corrélées (GEE au chapitre 12, GLMM au chapitre 13). La 2e édition (2002) traite déjà de ces deux derniers sujets, mais pas la 1re (1990).

James, Witten, Hastie et Tibshirani (2021). *An Introduction to Statistical Learning : with Applications in R*

Chapitres 2, 3 et 5 à 9.

2e édition, Springer (James et al. 2021). Gratuit en ligne à statlearning.com. Le livre présente de façon accessible, avec peu de mathématiques, la régression logistique et la régression de Poisson, la validation croisée, ridge et le lasso, les kNN, les polynômes, les splines et les GAM. Je vous le conseille comme point de départ si un chapitre de ce livre vous paraît trop abstrait ; notez qu'il en existe aussi une version avec Python.

Wood (2017). *Generalized Additive Models : An Introduction with R*

Chapitres 3 (présentation compacte des GLM), 8, 9 et 10 (lien entre lissage pénalisé et effets aléatoires).

2e édition, Chapman & Hall/CRC (Wood 2017). Écrit par l'auteur de la librairie `mgcv`, il traite des bases de splines, de la pénalisation, du choix du paramètre de lissage, des degrés de liberté effectifs et de l'inférence sur les fonctions lisses. Il est plus exigeant mathématiquement que les précédents.

Heritier, Cantoni, Copt et Victoria-Feser (2009). *Robust Methods in Biostatistics*

Chapitres 3, 10 et 12.

Wiley (Heritier et al. 2009). Une présentation appliquée des méthodes robustes pour les modèles de ce livre : régression linéaire, GLM, modèles mixtes, GEE et modèle de Cox. C'est la suite naturelle du chapitre sur la robustesse.

Zeileis, Kleiber et Jackman (2008). « Regression Models for Count Data in R »

Chapitre 4.

Journal of Statistical Software, 27(8) (Zeileis et al. 2008). Gratuit en ligne. L'article qui accompagne la librairie `pscl` compare les modèles de Poisson, binomial négatif, à zéros excédentaires et à barrière sur le même jeu de données `bioChemists` qu'au Chapitre 4.

Agresti (2010). *Analysis of Ordinal Categorical Data*

Chapitres 4 et 10.

2e édition, Wiley (Agresti 2010). Le modèle à cotes proportionnelles en détail, avec ses variantes (cotes partiellement proportionnelles, liens probit et log-log complémentaire) et les modèles ordinaux pour données corrélées.

Hastie, Tibshirani et Friedman (2009). *The Elements of Statistical Learning*

Chapitres 5 à 8 et 9.

2e édition, Springer (Hastie et al. 2009). Gratuit en ligne à hastie.su.domains/ElemStatLearn. C'est la version avancée du livre de James et coll., surtout utile pour le compromis biais-variance, la validation croisée, la régularisation, les méthodes à noyau, les splines et les modèles additifs.

Hastie, Tibshirani et Wainwright (2015). *Statistical Learning with Sparsity : The Lasso and Generalizations*

Chapitre 6.

CRC Press (Hastie et al. 2015). Gratuit en ligne à hastie.su.domains/StatLearnSparsity. Le prolongement naturel du chapitre sur la régularisation : lasso et ses variantes, filet élastique, algorithmes de descente de coordonnées et inférence après sélection.

Wasserman (2006). *All of Nonparametric Statistics*

Chapitres 7 et 8.

Springer (Wasserman 2006). Une présentation concise et rigoureuse de la régression non paramétrique : estimateurs à noyau et polynômes locaux, splines, choix du paramètre de lissage et vitesses de convergence.

Ruppert, Wand et Carroll (2003). *Semiparametric Regression*

Chapitres 8, 9 et 10.

Cambridge University Press (Ruppert et al. 2003). Le livre développe le lien entre splines pénalisées et modèles mixtes vu au Chapitre 10, et s'en sert pour ajuster et interpréter les modèles additifs.

Fitzmaurice, Laird et Ware (2011). *Applied Longitudinal Analysis*

Chapitre 10.

2e édition, Wiley (Fitzmaurice et al. 2011). La référence la plus pédagogique sur les données longitudinales : modèles mixtes linéaires et généralisés, GEE, structures de covariance et distinction entre interprétations marginale et conditionnelle.

Pinheiro et Bates (2000). *Mixed-Effects Models in S and S-PLUS*

Chapitre 10.

Springer (Pinheiro et Bates 2000). La référence technique sur l'estimation des modèles mixtes (ML, REML, BLUP) et sur les erreurs corrélées dans le temps, avec la librairie `nlme`. Pour `lme4`, la librairie utilisée au Chapitre 10, l'article de Bates et al. (2015), gratuit en ligne, décrit la syntaxe des formules et les méthodes d'estimation.

Agresti, Alan. 2010. *Analysis of Ordinal Categorical Data*. 2 éd. Wiley.

Agresti, Alan. 2013. *Categorical Data Analysis*. 3 éd. Wiley.

Arel-Bundock, Vincent, Noah Greifer, et Andrew Heiss. s. d. *marginaleffects : Predictions, Comparisons, Slopes, Marginal Means, and Hypothesis Tests*. https://CRAN.R-project.org/package=marginaleffects.

Bates, Douglas, Martin Mächler, Ben Bolker, et Steve Walker. 2015. « Fitting Linear Mixed-Effects Models Using lme4 ». *Journal of Statistical Software* 67 (1) : 1-48. https://doi.org/10.18637/jss.v067.i01.

Belsley, David A., Edwin Kuh, et Roy E. Welsch. 1980. *Regression Diagnostics : Identifying Influential Data and Sources of Collinearity*. Wiley.

Cantoni, Eva, et Elvezio Ronchetti. 2001. « Robust Inference for Generalized Linear Models ». *Journal of the American Statistical Association* 96 (455) : 1022-30. https://doi.org/10.1198/016214501753209031.

Cornillon, Pierre-André, Nicolas Hengartner, Eric Matzner-Løber, et Laurent Rouvière. 2023. *Régression avec R*. 3 éd. EDP Sciences.

Dunn, Peter K., et Gordon K. Smyth. 2018. *Generalized Linear Models with Examples in R*. Springer Texts in Statistics. Springer. https://doi.org/10.1007/978-1-4419-0118-7.

Faraway, Julian. 2025. *faraway : Functions and Datasets for Books by Julian Faraway*. https://CRAN.R-project.org/package=faraway.

Faraway, Julian J. 2016. *Extending the Linear Model with R : Generalized Linear, Mixed Effects and Nonparametric Regression Models*. 2 éd. Chapman ; Hall/CRC. https://doi.org/10.1201/9781315382722.

Fitzmaurice, Garrett M., Nan M. Laird, et James H. Ware. 2011. *Applied Longitudinal Analysis*. 2 éd. Wiley.

Gelman, Andrew, et Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.

Hartig, Florian. s. d. *DHARMa : Residual Diagnostics for Hierarchical (Multi-Level/Mixed) Regression Models*. https://CRAN.R-project.org/package=DHARMa.

Hastie, Trevor, Robert Tibshirani, et Jerome Friedman. 2009. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. 2 éd. Springer Series in Statistics. Springer. https://doi.org/10.1007/978-0-387-84858-7.

Hastie, Trevor, Robert Tibshirani, et Martin Wainwright. 2015. *Statistical Learning with Sparsity : The Lasso and Generalizations*. CRC Press.

Hauck, Walter W., et Allan Donner. 1977. « Wald's Test as Applied to Hypotheses in Logit Analysis ». *Journal of the American Statistical Association* 72 (360) : 851-53. https://doi.org/10.1080/01621459.1977.10479969.

Heritier, Stephane, Eva Cantoni, Samuel Copt, et Maria-Pia Victoria-Feser. 2009. *Robust Methods in Biostatistics*. Wiley.

Hosmer, David W., Stanley Lemeshow, et Rodney X. Sturdivant. 2013. *Applied Logistic Regression*. 3 éd. Wiley.

Huber, Peter J., et Elvezio M. Ronchetti. 2009. *Robust Statistics*. 2 éd. Wiley. https://doi.org/10.1002/9780470434697.

James, Gareth, Daniela Witten, Trevor Hastie, et Robert Tibshirani. 2021. *An Introduction to Statistical Learning : with Applications in R*. 2 éd. Springer Texts in Statistics. Springer. https://doi.org/10.1007/978-1-0716-1418-1.

Jolliffe, Ian T. 1972. « Discarding Variables in a Principal Component Analysis. I : Artificial Data ». *Journal of the Royal Statistical Society, Series C (Applied Statistics)* 21 (2) : 160-73.

Jolliffe, Ian T. 1973. « Discarding Variables in a Principal Component Analysis. II : Real Data ». *Journal of the Royal Statistical Society, Series C (Applied Statistics)* 22 (1) : 21-31.

Koenker, Roger. 2005. *Quantile Regression*. Cambridge University Press.

Maron, Martine. 2007. « Threshold effect of eucalypt density on an aggressive avian competitor ». *Biological Conservation* 136 : 100-107.

Maronna, Ricardo A., R. Douglas Martin, Victor J. Yohai, et Matías Salibián-Barrera. 2019. *Robust Statistics : Theory and Methods (with R)*. 2 éd. Wiley. https://doi.org/10.1002/9781119214656.

McCullagh, Peter, et John A. Nelder. 1989. *Generalized Linear Models*. 2 éd. Chapman ; Hall.

Miron, Julien, Benjamin Poilane, et Eva Cantoni. 2022. « Robust Polytomous Logistic Regression ». *Computational Statistics & Data Analysis* 176 : 107564. https://doi.org/10.1016/j.csda.2022.107564.

Morris, Tim P., Ian R. White, et Michael J. Crowther. 2019. « Using simulation studies to evaluate statistical methods ». *Statistics in Medicine* 38 (11) : 2074-102. https://doi.org/10.1002/sim.8086.

Pan, Wei. 2001. « Akaike's Information Criterion in Generalized Estimating Equations ». *Biometrics* 57 (1) : 120-25.

Pinheiro, José C., et Douglas M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. Springer.

Ruppert, David, M. P. Wand, et R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge University Press.

Stasinopoulos, Mikis D., Robert A. Rigby, Gillian Z. Heller, Vlasios Voudouris, et Fernanda De Bastiani. 2017. *Flexible Regression and Smoothing : Using GAMLSS in R*. CRC Press.

Therneau, Terry M., et Patricia M. Grambsch. 2000. *Modeling Survival Data : Extending the Cox Model*. Springer.

Ver Hoef, Jay M., et Peter L. Boveng. 2007. « Quasi-Poisson vs. Negative Binomial Regression : How Should We Model Overdispersed Count Data ? » *Ecology* 88 (11) : 2766-72.

Wasserman, Larry. 2006. *All of Nonparametric Statistics*. Springer.

Wood, Simon N. 2003. « Thin Plate Regression Splines ». *Journal of the Royal Statistical Society : Series B* 65 (1) : 95-114.

Wood, Simon N. 2017. *Generalized Additive Models : An Introduction with R*. 2 éd. Chapman ; Hall/CRC. https://doi.org/10.1201/9781315370279.

Yuan, Ming, et Yi Lin. 2006. « Model Selection and Estimation in Regression with Grouped Variables ». *Journal of the Royal Statistical Society : Series B* 68 (1) : 49-67.

Zeileis, Achim, Christian Kleiber, et Simon Jackman. 2008. « Regression Models for Count Data in R ». *Journal of Statistical Software* 27 (8) : 1-25. https://doi.org/10.18637/jss.v027.i08.

Zou, Hui, et Trevor Hastie. 2005. « Regularization and Variable Selection via the Elastic Net ». *Journal of the Royal Statistical Society : Series B* 67 (2) : 301-20.

Index thématique

Approximation de l'erreur de validation croisée, 144
Approximation de Laplace, 271

BLUP et rétrécissement, 255

Cas ϕ fixé, 99
Cas ϕ inconnu, 99
Convergence de D et X^2_P, 67
Convergence et estimateur sandwich, 277
Corrélation intraclasse, 252
Cote (odds), 36
Cotes proportionnelles, 125
Courbe ROC et AUC, 78

Déviance, 61
DFBETA et DFFIT, 16
DFBETAS et DFFITS, 16
Distance de Cook, 16
Distribution asymptotique de $G^2 = D_{\text{réduit}} - D_{\text{complet}}$, 62

Effet du plan, 251
Effet multiplicatif sur la cote, 36
Elastic net, 167
Équations de Henderson, 266
Erreur de Monte Carlo, 22
Existence sous séparation, 176

Famille exponentielle, 86
Fonction d'influence, 300
Fonctions de perte pour un dénombrement, 143
Fonctions de perte pour une réponse binaire, 142
Formulation sous contrainte, 163
Formule explicite pour la régression linéaire, 139

g^{-1} est une fonction de répartition, 35
GAMLSS, 290
GEE, 276
GLMM, 270

Indices de conditionnement et proportions de décomposition de la variance, 169
Interprétation des coefficients, 120
Interprétation par une variable latente, 125
IRLS pénalisé (ridge), 176

Lien canonique, 91
Loi asymptotique de l'estimateur du maximum de vraisemblance, 93
Loi asymptotique de la déviance, 94
Loi binomiale négative, 108
Loi du rapport de vraisemblance à la frontière, 256

M-estimateur de régression, 300
Matrice chapeau et leviers, 13
Matrice de design, 12
Modèle à barrière (hurdle), 128
Modèle à cotes proportionnelles, 124
Modèle à risques proportionnels de Cox, 292
Modèle de Poisson à zéros excédentaires (ZIP), 127
Modèle intra-inter, 261
Modèle linéaire généralisé, 91
Modèle linéaire mixte, 253

Modèle linéaire mixte à ordonnée à l'origine aléatoire, 252
Modèle saturé, 60
Modèles emboîtés, 14
Moindres carrés généralisés, 254
Moments d'une variable de la famille exponentielle, 87
Moments du modèle ZIP, 128

Non-négativité de la déviance, 61
Non-négativité de la déviance individuelle, 94

Point de rupture, 299
Prédicteur linéaire et fonction de lien, 34

Régression logistique multinomiale, 119
Régression quantile, 291
Résidus de déviance et résidus de Pearson, 67
Résidus, résidus standardisés et studentisés, 15
Résumés usuels, 23

Score, 38
Séparation complète et quasi complète, 39
Simulation de Monte Carlo, 21
Spécificité et sensibilité, 78
Surdispersion (overdispersion), 72

Test de Hosmer-Lemeshow, 69
Test de surdispersion, 108
Test du score, 54

Variable offset, 102
VIF et valeurs propres, 169
Vraisemblances ML et REML, 254

Index des fonctions R

abline, 72, 74, 111
aes, 24, 43, 45, 50, 57, 70, 79, 103, 122, 161, 186, 189, 191, 192, 196, 200, 207, 209, 211, 214, 217, 220, 224, 225, 243, 259, 264, 269, 274, 283, 315, 329, 330, 333, 342, 366
after_stat, 24
AIC, 17
AIC, 63, 65, 99, 126, 129, 324, 325, 354, 390–392, 394–396
all, 324
anova, 53, 54, 110, 111, 255, 258, 262, 263, 268, 285, 342, 392
arrange, 79, 282, 334
as.Date, 228, 238, 246
as_tibble, 282
avg_comparisons, 59, 123

bind_rows, 45, 70, 342, 366
binomial, 44
bs, 220, 221, 223, 227, 230, 295, 376–380, 383–385

coalesce, 282
coef, 23, 26, 50, 73, 103, 106, 122, 127, 157, 159, 160, 179, 217, 219, 244, 258, 261, 263, 264, 267, 279, 280, 282, 283, 303, 317, 326, 328, 334, 345, 348, 349, 354, 365, 366, 376, 400, 408
colMeans, 161, 407
concurvity, 240
confint, 47–49, 81, 317, 328, 407
confint.default, 48, 81, 244
cooks.distance, 75, 333
coord_cartesian, 161, 366
coord_equal, 259, 333
cor, 156, 171, 173, 362
corAR1, 268
cos, 384
count, 121, 327
cout, 356
cut, 214, 316, 375, 376, 382
cv.glm, 29, 139, 140, 143, 145–147, 149, 152, 313, 355
cv.glmnet, 159, 161, 165–167, 177, 178, 181, 185, 363, 366

data, 129, 272, 273, 315, 324, 329, 340, 348, 362, 408
datagrid, 59
desc, 334
deviance, 62, 67, 106, 331, 341
df.residual, 62, 67, 331, 341, 344, 399
dfbetas, 16, 18, 75
dffits, 16, 18, 75
diff, 80, 397
dnbinom, 129
dnorm, 195, 196, 274, 408
dpois, 129, 349
drop1, 327, 333

ecart.type, 317
eigen, 171, 173
expand.grid, 243, 282, 408

facet_wrap, 70, 103, 189, 200, 264
filter, 189, 209, 228, 238, 246, 264, 268, 274, 282, 331, 349, 362
first, 32
fitted, 70, 74, 79, 106, 129, 147, 315, 326, 333, 341, 354, 385

fixef, 264, 267, 272, 273, 279, 282, 283, 407, 408
formula, 64, 325

gam, 270
gam, 238, 241, 245, 248, 249, 267, 268, 295, 388, 391, 393–396, 398, 400, 401
gam.check, 247, 394, 395
gam.vcomp, 267
gamm, 270
gamm, 253, 268, 285, 397
geeglm, 278–280, 282, 285, 296, 408
geom_abline, 79, 259, 264, 283, 333
geom_histogram, 24
geom_hline, 70, 161, 259, 315, 330
geom_jitter, 209, 211, 214, 217, 224, 225, 243
geom_line, 45, 50, 57, 79, 103, 122, 161, 191, 192, 196, 200, 207, 211, 217, 220, 224, 225, 243, 269, 274, 329, 342, 366
geom_point, 43, 45, 70, 103, 186, 189, 191, 192, 196, 207, 259, 264, 274, 283, 315, 329, 330, 333, 342
geom_ribbon, 57, 70, 207, 211, 269, 366
geom_step, 214
geom_text, 333
geom_vline, 24, 161, 189, 191, 196, 217, 220, 366
get.knn, 191
ggplot, 24, 43, 45, 50, 57, 70, 79, 103, 122, 161, 186, 189, 191, 192, 196, 200, 207, 209, 211, 214, 217, 220, 224, 225, 243, 259, 264, 269, 274, 283, 315, 329, 330, 333, 342, 366
glm, 35, 40, 44, 48–50, 52, 53, 62–64, 70, 72, 73, 77, 96, 102, 105, 114, 115, 129, 139, 146, 147, 149, 178, 181, 238, 244, 245, 272, 324, 325, 330, 332, 341, 345, 348, 354
glm.control, 326
glm.nb, 108–110, 115, 129, 294, 344
glmer, 266, 271–273, 285, 296, 408
glmnet, 157
glmnet, 155, 157, 159–161, 164, 167, 176, 177, 181, 182, 295, 358, 361
group_by, 32, 70, 189, 259, 261, 274, 282, 283, 316, 372, 376, 382, 398, 408

hat, 24, 50, 74, 259, 333
hatvalues, 75, 113, 139, 144, 147, 173, 207, 210, 214, 217, 219, 222, 223, 228, 318, 333
hurdle, 129, 132, 295, 348

if_else, 121, 136, 282, 327
install_github, 11
integrate, 274, 408
interaction, 217

kable, 157, 166, 167, 180
kappa, 219, 222, 385
knn.reg, 188, 190–192, 194, 202, 295

labs, 24, 43, 45, 50, 57, 70, 79, 103, 122, 161, 186, 189, 191, 192, 196, 200, 207, 209, 211, 214, 217, 220, 224, 225, 243, 259, 264, 269, 274, 283, 315, 329, 330, 333, 342, 366
lag, 282
legend, 374, 379, 381, 396
lines, 40, 374, 379, 381, 398
lm, 23, 26, 31, 138, 157, 159, 173, 207, 210, 214, 217, 219, 221, 223, 225, 227, 257, 260, 261, 263, 264, 289, 294, 299, 303, 315, 317–319, 362, 363, 366, 373, 375, 376, 378–380, 383–385, 390, 391, 407, 408
lmer, 253–255, 257, 259, 261–263, 265, 268, 282, 285, 289, 295, 407
logLik, 50, 109, 344

map_dfr, 50
match, 166, 167
matplot, 396
median, 189

model.matrix, 155, 156, 168, 178, 180, 222, 349, 363, 365, 385, 400, 408
multinom, 120, 121, 132, 294
mutate, 32, 43, 45, 50, 57, 70, 100, 103, 121, 125, 136, 145, 178, 189, 192, 196, 209, 217, 228, 238, 241, 243, 246, 257, 259, 261, 268, 279, 282, 315, 316, 324, 327, 329, 330, 333, 340, 342, 344, 353, 362, 398, 408

n, 70, 259, 282, 316, 398
n1, 199
na.omit, 156
ns, 223, 225, 230, 295, 376, 378, 379, 381, 383–385
ntile, 70

offset, 102, 109, 147, 341, 344
optim, 134, 349

par, 51, 76, 164, 178, 239, 391
pchisq, 62, 67, 70, 106, 107, 109, 263, 331, 341, 344
pivot_longer, 50
plogis, 50, 56, 57, 70, 274, 282, 398, 408
plot, 40, 51, 72, 74–76, 111, 112, 114, 164, 178, 239, 242, 245, 246, 249, 372, 374, 377, 379, 381, 389, 391, 398, 403
pnorm, 318
polr, 124–126, 132, 294
poly, 136, 207, 210, 217, 373, 375, 376, 378, 379
predict, 40, 45, 56, 57, 103, 122, 129, 138, 144, 149, 159, 161, 180, 207, 211, 214, 217, 224, 225, 243, 269, 317, 342, 344, 349, 356, 363, 366, 374, 376, 379–384, 393, 394, 396, 398
predictions, 59
pull, 372

QIC, 279, 280
qnorm, 25, 26, 56, 274, 407
qt, 25, 26, 318, 344, 407

ranef, 259
rank, 199
read.csv, 100, 121, 136, 145, 178, 186, 207, 209, 228, 238, 241, 246, 257, 268
recode, 50
residuals, 67, 72, 73, 111, 112, 136, 139, 173, 207, 210, 214, 217, 219, 222, 223, 228, 259, 315, 330, 331, 341, 399
rlm, 302–304
row_number, 32, 330, 333, 362
rowMeans, 317, 383, 384
rowSums, 172, 173, 191, 195
rstandard, 75
rstudent, 75

s, 238, 241, 245, 267, 268, 388, 391, 393–396, 398, 400
sample, 138, 159, 167, 178, 363, 393
scale_colour_manual, 189
scale_linetype_manual, 264
scale_y_continuous, 274
scale_y_log10, 342
select, 32, 59, 123, 331, 334, 344, 362
setdiff, 173
sigma, 317
simulateResiduals, 114
sin, 383
smooth.spline, 224, 227, 229, 295, 381, 383
split, 264
step, 17
step, 17, 18, 64, 65, 84, 325, 328
summarise, 70, 189, 259, 274, 283, 316, 340, 372, 376, 382, 398, 408
summary, 23, 26, 40, 44, 45, 47, 50, 52, 54, 66, 73, 74, 81, 84, 102, 121, 125, 126, 130, 156, 178, 221, 236–241, 245, 246, 257–259, 261, 263, 272, 279, 280, 315, 326, 332, 341, 344, 362, 388, 391, 395, 396, 398, 400
sweep, 172, 173

terms, 65, 325, 328
testDispersion, 114
testZeroInflation, 114
theme, 264, 269, 283
theme_minimal, 24, 43, 45, 50, 57, 70, 79, 103, 122, 161, 186, 189, 191, 192, 196, 200, 207, 209, 211, 214, 217, 220, 224, 225, 243, 259, 264, 269, 274, 283, 315, 329, 330, 333, 342, 366
ti, 396, 397
tilde, 74, 333
title, 164, 178

uncount, 32
ungroup, 32, 261, 282
unname, 139, 366
update, 110, 111, 272, 273, 326, 344

VarCorr, 262, 273, 408
vcov, 279, 407

with, 219

zeroinfl, 129, 132, 295