Section 2

Études de simulation

Séance du vendredi 22 janvier 2027
Lectures : Morris, White et Crowther (2019) ; complément : Koehler, Brown et Haneuse (2009)

Julien Miron – notes rédigées à partir du matériel de Lajmi Lakhal-Chaieb et Alexandre Bureau

Objectifs de la séance

À la fin de cette séance, vous devriez être en mesure de :

1. utiliser un générateur de nombres pseudo-aléatoires de façon reproductible ;
2. estimer par la méthode de Monte Carlo une probabilité, une espérance ou un quantile, et quantifier la précision de cette estimation (erreur type Monte Carlo) ;
3. valider un estimateur (biais, variance, estimateur de la variance, erreur quadratique moyenne) et la couverture d'un intervalle de confiance ;
4. valider un test : uniformité des valeurs p sous H_0, erreur de première espèce, courbe de puissance ;
5. générer des données d'une loi quelconque par inversion de la fonction de répartition, y compris une loi tronquée ;
6. planifier, programmer et rapporter une étude de simulation selon le cadre ADEMP, en choisissant le nombre de réplications en fonction de la précision voulue.

Mise en contexte

À la séance 1, nous avons obtenu des résultats *asymptotiques* : l'EMV est approximativement normal, la variance sandwich est convergente, les statistiques de Wald suivent approximativement une loi du khi-deux. Ces résultats ne disent rien de la qualité de l'approximation à $n = 50$ patients, ni de ce qui se passe quand les hypothèses sont violées. Une *étude de simulation* répond à ces questions de façon expérimentale : on génère un grand nombre de jeux de données sous un mécanisme connu, on applique la méthode à chacun, et on compare les résultats à la vérité, connue par construction.

Les études de simulation sont omniprésentes en biostatistique méthodologique : presque tous les articles qui proposent une méthode en contiennent une, et les trois études de cas du cours (Neuhaus et McCulloch, 2006 ; Tuglus et van der Laan, 2011 ; Breslow et Chatterjee, 1999) s'appuient sur des

simulations ou des rééchantillonnages. Vous en programmerez dans les deux devoirs et dans le travail de synthèse. L'objectif de la séance est que ces études soient *planifiées* et *rapportées* avec la même rigueur qu'une étude clinique, ce qui inclut la quantification de leur propre erreur d'échantillonnage.

1 Nombres pseudo-aléatoires

Les générateurs « aléatoires » des logiciels produisent une suite *déterministe* de nombres qui a les propriétés statistiques d'une suite i.i.d. uniforme sur $(0, 1)$: ce sont des nombres pseudo-aléatoires. Toutes les autres lois sont obtenues par transformation de ces uniformes (section 5). Le générateur par défaut de R est le Mersenne-Twister (Matsumoto et Nishimura, 1998), de période $2^{19937} - 1$.

L'état initial du générateur est fixé par `set.seed()`, à laquelle on passe un entier quelconque. Deux exécutions qui partent du même état produisent exactement les mêmes nombres :

```
set.seed(7360); runif(3)
set.seed(7360); runif(3)
```

```
[1] 0.8407097 0.3748649 0.1218414
[1] 0.8407097 0.3748649 0.1218414
```

Bonnes pratiques :

— Fixer la graine *une fois*, au début du script (ou au début de chaque scénario), et non dans la boucle de réplications : rappeler `set.seed(1)` à chaque itération produirait B fois le même jeu de données.

— Noter la version de R et le générateur (`RNGkind()`) : l'algorithme de `sample()` a changé avec R 3.6.0, de sorte que d'anciens scripts ne reproduisent plus exactement leurs résultats sans `RNGkind(sample.kind = "Rounding")`.

— Pour paralléliser, utiliser des flux indépendants conçus à cette fin (`RNGkind("L'Ecuyer-CMRG")` avec `parallel::clusterSetRNGStream`) plutôt que des graines « au hasard » sur chaque processeur.

— Pouvoir régénérer un jeu de données particulier (par exemple celui où l'algorithme n'a pas convergé) : conserver `.Random.seed` au début de chaque réplication, ou utiliser une graine déterministe par réplication.

2 Estimer les caractéristiques d'une loi par Monte Carlo

Le principe de Monte Carlo est la loi des grands nombres utilisée comme outil de calcul. Toute caractéristique qui s'écrit comme une espérance, $\theta = \mathrm{E}\{h(Y)\}$, s'estime par

$$\widehat{\theta}_N = \frac{1}{N}\sum_{j=1}^{N} h(Y_j), \qquad Y_1, \ldots, Y_N \overset{\text{i.i.d.}}{\sim} F,$$

et le théorème central limite fournit sa précision : $\widehat{\theta}_N \approx \mathcal{N}(\theta, \mathrm{Var}\{h(Y)\}/N)$. L'écart type de $\widehat{\theta}_N$, estimé par $s_h/\sqrt{N}$ où s_h est l'écart type empirique des $h(Y_j)$, est l'*erreur type Monte Carlo* (ETMC).

Elle décroît en $1/\sqrt{N}$: pour gagner une décimale, il faut cent fois plus de tirages. Contrairement à l'erreur d'échantillonnage d'une étude réelle, nous contrôlons N : il n'y a aucune excuse pour une ETMC trop grande ou non rapportée.

Exemple fil conducteur. Soit $Z \sim \mathcal{N}(0,1)$ et $Y = \log|Z|$. Pour $y \in \mathbb{R}$,

$$F_Y(y) = \mathbb{P}(\log|Z| \le y) = \mathbb{P}(-e^y \le Z \le e^y) = 2\Phi(e^y) - 1, \qquad f_Y(y) = 2e^y\phi(e^y),$$

où ϕ est la densité normale standard. Nous connaissons donc la vérité, ce qui permet de juger les estimations.

2.1 Estimer une probabilité

Pour $\theta = \mathbb{P}(Y \in A)$, on prend $h(y) = \mathbf{1}\{y \in A\}$: $\widehat{\theta} = N^{-1}\sum_j \mathbf{1}\{Y_j \in A\}$ est une proportion, de variance $\theta(1-\theta)/N$ estimée par $\widehat{\theta}(1-\widehat{\theta})/N$, et $\widehat{\theta} \pm 1{,}96\sqrt{\widehat{\theta}(1-\widehat{\theta})/N}$ est un intervalle de confiance (IC) à 95 % pour θ. Avec $A = (-\infty, -1]$, la vraie valeur est $\theta = 2\Phi(e^{-1}) - 1 = 0{,}287$.

```
set.seed(7360)
N <- 1000
Y <- log(abs(rnorm(N)))
th <- mean(Y < -1); se.th <- sqrt(th*(1-th)/N)
round(c(theta.chapeau = th, se = se.th, IC = th + c(-1,1)*1.96*se.th), 4)
```

```
theta.chapeau            se           IC1           IC2
       0.2990        0.0145        0.2706        0.3274
```

2.2 Estimer une espérance

Pour $\mu = \mathrm{E}(Y)$, $\widehat{\mu} = \bar{Y}$ et $\mathrm{ETMC}(\widehat{\mu}) = s/\sqrt{N}$ avec $s^2 = (N-1)^{-1}\sum_j (Y_j - \bar{Y})^2$. La vraie valeur s'obtient par intégration numérique de $y f_Y(y)$ ou, ici, en forme close : $\mathrm{E}\log|Z| = -(\gamma + \log 2)/2 \approx -0{,}6352$, où γ est la constante d'Euler.

```
mu <- mean(Y); se.mu <- sd(Y)/sqrt(N)
round(c(mu.chapeau = mu, se = se.mu, IC = mu + c(-1,1)*1.96*se.mu), 4)
f.Y <- function(y) 2*exp(y)*dnorm(exp(y))
vrai.mu <- integrate(function(y) y*f.Y(y), lower = -30, upper = 10)$value
round(c(vrai.mu = vrai.mu, forme.close = -(-digamma(1) + log(2))/2), 5)
```

```
mu.chapeau         se        IC1        IC2
   -0.6644     0.0383    -0.7394    -0.5894
    vrai.mu forme.close
   -0.63518    -0.63518
```

Intégration numérique

`integrate(..., -Inf, Inf)` échoue ici : pour y grand, `exp(y)` déborde et le produit `Inf * 0` vaut `NaN`. Il faut examiner le support effectif de la densité (tracer f_Y) et choisir des bornes finies qui couvrent toute la masse : la densité de $\log|Z|$ a une longue queue à gauche.

2.3 Estimer un quantile

Le quantile d'ordre β est $q_\beta = F^{-1}(\beta)$; pour notre exemple, $2\Phi(e^{q_\beta}) - 1 = \beta$ donne $q_\beta = \log \Phi^{-1}\{(1+\beta)/2\}$. On l'estime par la statistique d'ordre $Y_{(\lceil N\beta \rceil)}$ (où $\lceil x \rceil$ est le plus petit entier supérieur ou égal à x), et on obtient un IC *sans hypothèse sur* F grâce à la loi binomiale.

Proposition 2.1. *Soit $Y_1, \ldots, Y_N$ i.i.d. de loi continue F, $K \sim \text{Bin}(N, \beta)$ et des entiers $L < R$. Alors*

$$\mathbb{P}(Y_{(L)} \leq q_\beta \leq Y_{(R)}) = \mathbb{P}(L \leq K \leq R-1).$$

En prenant L le quantile $\alpha/2$ et $R-1$ le quantile $1-\alpha/2$ de K, l'intervalle $[Y_{(L)}, Y_{(R)}]$ a une couverture d'au moins (environ) $1-\alpha$.

Démonstration. Soit $K = \#\{j : Y_j \leq q_\beta\} \sim \text{Bin}(N, \beta)$. L'événement $\{Y_{(L)} \leq q_\beta\}$ signifie qu'au moins L observations sont inférieures ou égales à q_β, soit $\{K \geq L\}$. De même, $\{Y_{(R)} \geq q_\beta\}$ équivaut (loi continue) à $\{K \leq R-1\}$. L'intersection est $\{L \leq K \leq R-1\}$. □

```r
set.seed(7362)
beta <- 0.75; q <- log(qnorm((1+beta)/2))       # valeur théorique
N <- 1e5
Ys <- sort(log(abs(rnorm(N))))
L <- qbinom(0.025, N, beta); R <- qbinom(0.975, N, beta) + 1
round(c(q.vrai = q, q.chapeau = Ys[ceiling(N*beta)], IC = Ys[c(L, R)]), 4)
pbinom(R-1, N, beta) - pbinom(L-1, N, beta)     # couverture exacte
```

```
   q.vrai q.chapeau       IC1       IC2
   0.1401    0.1373    0.1316    0.1432
[1] 0.9505275
```

La proposition donne la couverture *exacte* de l'intervalle pour tout N. Pour un petit échantillon ($N = 50$), elle vaut 0,967 avec $L = 31$ et $R = 44$, et une simulation de 5000 échantillons donne une couverture empirique de 0,965, en accord avec la théorie.

Remarque 2.2 (Corrections au document H26). Le document H26 définissait la partie entière $\lfloor x \rfloor$ comme « le plus petit entier inférieur ou égal à x » : c'est le *plus grand*. Il utilisait aussi $R =$ `qbinom(1-alpha/2, N, beta)` comme borne supérieure ; d'après la proposition 2.1, l'indice R doit dépasser de 1 le quantile binomial pour garantir la couverture (l'écart est négligeable pour $N = 10^5$, pas pour $N = 50$).

À retenir

- Monte Carlo = moyenne de N tirages ; précision = ETMC = $s_h/\sqrt{N}$; IC = estimation $\pm 1{,}96$ ETMC.
- Probabilité : ETMC = $\sqrt{\widehat{\theta}(1-\widehat{\theta})/N}$; quantile : IC par les statistiques d'ordre et la loi binomiale.
- Toujours confronter à la vérité quand elle est calculable (forme close, intégration numérique).

3 Valider un estimateur

3.1 Mesures de performance et leur précision

Soit $\widehat{\theta}$ un estimateur de $\theta \in \mathbb{R}$, calculé sur un échantillon de taille n issu de F_θ, et $v(\widehat{\theta})$ un estimateur de sa variance (par exemple l'inverse de l'information, ou la variance sandwich). On génère B échantillons indépendants, on calcule $\widehat{\theta}_b$ et $v_b = v(\widehat{\theta}_b)$ pour $b = 1, \ldots, B$, puis

$$\bar{\theta} = \frac{1}{B}\sum_b \widehat{\theta}_b, \qquad S^2_{\widehat{\theta}} = \frac{1}{B-1}\sum_b (\widehat{\theta}_b - \bar{\theta})^2, \qquad \bar{v} = \frac{1}{B}\sum_b v_b, \qquad \widehat{\mathrm{EQM}} = \frac{1}{B}\sum_b (\widehat{\theta}_b - \theta)^2.$$

Notez la distinction entre deux tailles d'échantillon : n, celle de chaque jeu de données simulé (elle détermine la performance de la méthode) et B, le nombre de réplications (il détermine la précision de l'étude de simulation). Le tableau 1 résume les mesures usuelles et leur ETMC (Morris et coll., 2019).

Table 1 – Mesures de performance d'un estimateur et erreurs types Monte Carlo (ETMC).

Mesure	Estimation	ETMC
Biais	$\bar{\theta} - \theta$	$S_{\widehat{\theta}}/\sqrt{B}$
Écart type empirique	$S_{\widehat{\theta}}$	$S_{\widehat{\theta}}/\sqrt{2(B-1)}$ (approx. normale)
Erreur quadratique moyenne	$\widehat{\mathrm{EQM}}$	$\sqrt{\sum_b \{(\widehat{\theta}_b - \theta)^2 - \widehat{\mathrm{EQM}}\}^2/\{B(B-1)\}}$
Moyenne des variances estimées	$\bar{v}$	$S_v/\sqrt{B}$ (S_v : écart type des v_b)
Couverture d'un IC	$\widehat{C} = B^{-1}\sum_b \mathbf{1}\{\theta \in \mathrm{IC}_b\}$	$\sqrt{\widehat{C}(1-\widehat{C})/B}$
Taux de rejet (niveau, puissance)	$\widehat{P} = B^{-1}\sum_b \mathbf{1}\{p_b \leq \alpha\}$	$\sqrt{\widehat{P}(1-\widehat{P})/B}$

Interprétation :

— $\widehat{\theta}$ est (approximativement) sans biais si $|\bar{\theta} - \theta|$ est petit *relativement à son ETMC et à* $S_{\widehat{\theta}}$. Un biais « statistiquement significatif » mais minuscule devant l'écart type n'a pas d'importance pratique ; le biais relatif $(\bar{\theta} - \theta)/\theta$ et le biais standardisé $(\bar{\theta} - \theta)/S_{\widehat{\theta}}$ aident à juger.

— $v(\widehat{\theta})$ est un bon estimateur de la variance si $\bar{v} \approx S^2_{\widehat{\theta}}$; on rapporte souvent le rapport $\sqrt{\bar{v}}/S_{\widehat{\theta}}$ (erreur type moyenne sur erreur type empirique).

— EQM = biais^2 + variance : c'est le critère naturel pour comparer des estimateurs qui n'ont pas le même biais.

3.2 Exemple : l'EMV de la loi de Weibull

On reprend la loi de Weibull de la séance 1, de forme $a = 2$ et d'échelle $b = 10$, avec $n = 1000$ et $B = 1000$. La fonction `MASS::fitdistr` calcule l'EMV et l'inverse de l'information observée (`vcov`). Plutôt qu'une boucle qui remplit cinq vecteurs, on écrit une fonction qui traite *une* réplication et on la répète avec `replicate` ; le résultat est une matrice $B \times 5$ qu'on conserve.

```
library(MASS)
```

```r
set.seed(7360)
B <- 1000; N <- 1000; shape <- 2; scale <- 10
sim1 <- function() {
  X <- rweibull(N, shape = shape, scale = scale)
  fit <- suppressWarnings(fitdistr(X, "weibull"))
  c(fit$estimate, diag(fit$vcov), fit$vcov[1, 2])
}
res <- t(replicate(B, sim1()))
colnames(res) <- c("a", "b", "var.a", "var.b", "cov.ab")
moy <- colMeans(res)
round(rbind(
  moyenne     = moy[c("a", "b")],
  biais       = moy[c("a", "b")] - c(shape, scale),
  var.emp     = c(var(res[, "a"]), var(res[, "b"])),
  moy.var.est = moy[c("var.a", "var.b")],
  EQM         = c(mean((res[, "a"] - shape)^2), mean((res[, "b"] - scale)^2))), 5)
round(c(cov.emp = cov(res[, "a"], res[, "b"]), moy.cov.est = unname(moy["cov.ab"])), 5)
round(c(etmc.biais.a = sd(res[, "a"])/sqrt(B),
        etmc.se.emp.a = sd(res[, "a"])/sqrt(2*(B - 1))), 5)
```

```
                a       b
moyenne     2.00192 10.00836
biais       0.00192  0.00836
var.emp     0.00223  0.02595
moy.var.est 0.00244  0.02775
EQM         0.00223  0.02599
    cov.emp moy.cov.est
    0.00188     0.00257
 etmc.biais.a etmc.se.emp.a
      0.00149       0.00106
```

(Les avertissements de `fitdistr`, supprimés ici, proviennent d'évaluations de la densité hors de l'espace des paramètres pendant l'optimisation ; ils sont sans conséquence lorsque l'algorithme converge.)

Lecture. Le biais de $\widehat{a}$ (0,0019) est de l'ordre de son ETMC (0,0015) : aucune preuve de biais. La moyenne des variances estimées (0,00244) dépasse la variance empirique (0,00223) d'environ 9 %. L'ETMC de la variance empirique est d'environ $0{,}00223\sqrt{2/(B-1)} \approx 0{,}0001$, de sorte que l'écart représente environ deux ETMC. La théorie tranche : la variance asymptotique de l'EMV de la forme d'une loi de Weibull est $6a^2/(\pi^2 n) = 0{,}00243$, très proche de $\bar{v}$. C'est donc la variance empirique qui est un peu basse par le hasard de la simulation. Leçon : un écart de deux ETMC sur l'un des nombreux indicateurs d'une étude de simulation n'a rien d'exceptionnel, et $B = 1000$ ne permet pas de détecter un biais relatif inférieur à une dizaine de pour cent dans un estimateur de variance.

3.3 Couverture d'un intervalle de confiance

L'IC de Wald à 95 % construit sur le b-ième échantillon est $\widehat{\theta}_b \pm 1{,}96\sqrt{v_b}$, et la couverture empirique est la proportion des B intervalles qui contiennent la vraie valeur. C'est une proportion : on lui associe l'ETMC et l'IC du tableau 1. Avec les résultats déjà stockés, le calcul est vectoriel :

```r
inf <- res[, "a"] - qnorm(0.975)*sqrt(res[, "var.a"])
sup <- res[, "a"] + qnorm(0.975)*sqrt(res[, "var.a"])
couv <- mean(inf < shape & shape < sup)
```

```
round(c(couverture = couv, IC = couv + c(-1, 1)*1.96*sqrt(couv*(1-couv)/B)), 4)
```

```
couverture         IC1         IC2
    0.9600      0.9479      0.9721
```

Pour l'échelle b, on obtient de même 0,957 [0,944; 0,970].

Correction de l'interprétation H26

Les notes H26 concluaient que « la couverture est plus grande que le taux spécifié, donc les intervalles de confiance sont trop larges ». Avec $B = 1000$, l'IC de la couverture contient 0,95 : les données de la simulation sont compatibles avec une couverture nominale, et la théorie (séance 1) prédit d'ailleurs une couverture correcte avec $n = 1000$. Une conclusion sur l'écart à 0,95 exige de tenir compte de l'ETMC (environ 0,007 pour $B = 1000$).

Le même raisonnement s'applique à l'IC de l'espérance de $\log|Z|$ de la section précédente. Ici, on génère d'un coup une matrice $B \times N$ dont chaque ligne est un échantillon, ce qui évite la boucle :

```
set.seed(7361)
B <- 1000; N <- 1000
Ymat <- matrix(log(abs(rnorm(N*B))), nrow = B)  # une ligne = un échantillon
m <- rowMeans(Ymat); s <- apply(Ymat, 1, sd)
couv <- mean(abs(m - vrai.mu) <= 1.96*s/sqrt(N))
round(c(couverture = couv, IC = couv + c(-1,1)*1.96*sqrt(couv*(1-couv)/B)), 4)
```

```
couverture         IC1         IC2
    0.9560      0.9433      0.9687
```

À retenir

- Valider un estimateur : biais, écart type empirique, moyenne des variances estimées, EQM, couverture, chacun accompagné de son ETMC.
- Comparer $\bar{v}$ à $S^2_{\hat{\theta}}$ valide l'*estimateur de la variance*, pas l'estimateur lui-même.
- Une couverture n'est « trop grande » ou « trop petite » que si l'écart à 0,95 dépasse nettement l'ETMC.

4 Valider un test d'hypothèses

4.1 Uniformité des valeurs p sous H_0

Proposition 4.1. *Soit T une statistique de test de fonction de répartition F_0* continue *sous H_0, et la valeur p $P = 1 - F_0(T)$ (on rejette pour les grandes valeurs de T). Alors $P \sim \mathcal{U}(0,1)$ sous H_0, et le test qui rejette lorsque $P \leq \alpha$ a une erreur de première espèce exactement égale à α.*

Démonstration. Pour $u \in (0,1)$, $\mathbb{P}(P \leq u) = \mathbb{P}\{F_0(T) \geq 1-u\} = 1 - \mathbb{P}\{F_0(T) < 1-u\} = 1-(1-u) = u$, car $F_0(T)$ suit une loi uniforme lorsque F_0 est continue (transformation intégrale de probabilité, voir le théorème 5.1). □

Pour un test asymptotique, F_0 n'est que la loi limite ($\chi^2_{p_1}$ ou $\mathcal{N}(0,1)$), et l'uniformité n'est qu'approximative : c'est précisément ce que la simulation vérifie. (Pour une statistique discrète, avec la valeur p $P = 1 - F_0(T^-) = \mathbb{P}_{H_0}(T' \geq t)\big|_{t=T}$, on a seulement $\mathbb{P}(P \leq u) \leq u$: le test est conservateur.)

Démarche. On génère B échantillons sous H_0, on calcule les valeurs p $p_1, \ldots, p_B$, puis :

1. on examine leur distribution : histogramme (doit être plat) et graphique quantile-quantile contre la loi uniforme, de préférence sur l'échelle $-\log_{10}$ qui agrandit la région des petites valeurs p, la seule qui compte pour la décision ;
2. on peut ajouter un test formel de Kolmogorov-Smirnov (`ks.test(p, "punif")`), en gardant en tête qu'il est peu sensible aux écarts dans les queues ;
3. surtout, on estime l'erreur de première espèce aux seuils usuels, $\widehat{P} = B^{-1} \sum_b \mathbf{1}\{p_b \leq \alpha\}$, avec son IC binomial.

Sur le graphique quantile-quantile, une bande de confiance ponctuelle s'obtient en remarquant que la i-ème statistique d'ordre de B uniformes suit une loi Bêta$(i, B - i + 1)$.

Exemple 4.2 (Test de Wald dans le modèle de Weibull)**.** On teste $H_0 : (a, b) = (2, 10)$ par la statistique de Wald à 2 degrés de liberté de la séance 1, avec $n = 200$.

```
wald_weib <- function(X, theta0) {
  fit <- suppressWarnings(fitdistr(X, "weibull"))
  d <- fit$estimate - theta0
  Q <- drop(t(d) %*% solve(fit$vcov) %*% d)
  pchisq(Q, df = 2, lower.tail = FALSE)          # valeur p
}
set.seed(7360)
B <- 1000; N <- 200; theta0 <- c(2, 10)
p.H0 <- replicate(B, wald_weib(rweibull(N, 2, 10), theta0))
ks.test(p.H0, "punif")$p.value
for (a in c(0.01, 0.05)) {
  e <- mean(p.H0 <= a)
  print(round(c(alpha = a, erreur1 = e, IC = e + c(-1,1)*1.96*sqrt(e*(1-e)/B)), 4))
}
## graphique quantile-quantile sur l'échelle -log10, avec bande ponctuelle
u <- (1:B)/(B + 1); i <- 1:B
plot(-log10(u), -log10(sort(p.H0)), pch = 16, cex = 0.5)
abline(0, 1, col = "firebrick")
lines(-log10(u), -log10(qbeta(0.025, i, B - i + 1)), lty = 2)
lines(-log10(u), -log10(qbeta(0.975, i, B - i + 1)), lty = 2)
```

```
[1] 0.3478107
  alpha erreur1     IC1     IC2
 0.0100  0.0110  0.0045  0.0175
  alpha erreur1     IC1     IC2
 0.0500  0.0410  0.0287  0.0533
```

Le test de Kolmogorov-Smirnov ne rejette pas l'uniformité, les erreurs de première espèce estimées sont compatibles avec les niveaux nominaux, et les points du graphique quantile-quantile restent dans la bande (figure 1). Avec $B = 1000$, l'IC de l'erreur de type I au seuil de 1 % va pourtant de

0,45 % à 1,75 % : il faut un B beaucoup plus grand pour détecter une inflation modérée à ce seuil (section 6.2).

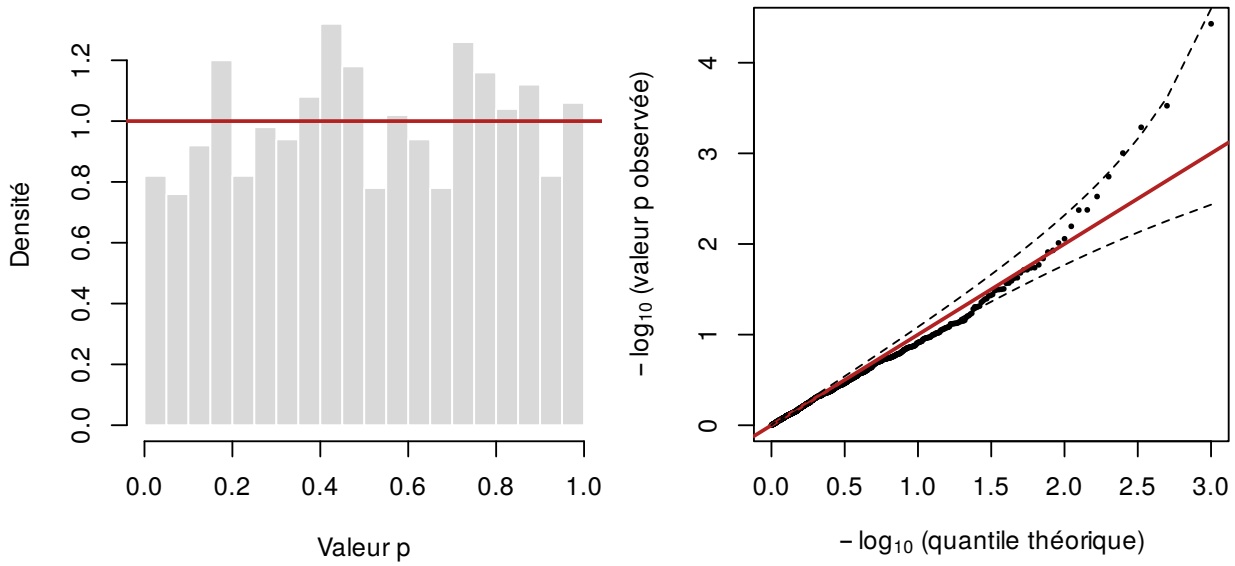


FIGURE 1 – Valeurs p du test de Wald de $H_0 : (a, b) = (2, 10)$ dans 1000 échantillons de Weibull simulés sous H_0 ($n = 200$). Gauche : histogramme et densité uniforme. Droite : graphique quantile-quantile sur l'échelle $-\log_{10}$, avec bande ponctuelle à 95 % (pointillés).

4.2 Puissance

On génère ensuite les données sous une alternative et on estime la puissance au seuil α par $B^{-1} \sum_b \mathbf{1}\{p_b \leq \alpha\}$. On vérifie que la puissance croît lorsqu'on s'éloigne de H_0 et lorsque n augmente. Comme dans les notes H26, générons sous $(a, b) = (2, 10)$ et testons $H_0 : (a, b) = (1{,}7, 10)$:

```
set.seed(7361)
p.H1 <- replicate(B, wald_weib(rweibull(N, 2, 10), c(1.7, 10)))
pw <- mean(p.H1 < 0.05)
round(c(puissance = pw, IC = pw + c(-1,1)*1.96*sqrt(pw*(1-pw)/B)), 4)
```

```
puissance       IC1       IC2
   0.7870    0.7616    0.8124
```

Une *courbe de puissance* répète ce calcul sur une grille d'alternatives. La présentation la plus naturelle fixe $H_0 : (a, b) = (2, 10)$ et fait varier la vraie forme a (figure 2) ; au point $a = 2$, la courbe donne l'erreur de type I.

```
set.seed(7362)
grille <- seq(1.6, 2.4, by = 0.05); Bp <- 500
puiss <- sapply(c(100, 200), function(n)
  sapply(grille, function(a)
    mean(replicate(Bp, wald_weib(rweibull(n, a, 10), theta0)) < 0.05)))
round(cbind(shape = grille, n100 = puiss[, 1], n200 = puiss[, 2]), 3)[c(1, 3, 5, 9, 13,
    15, 17), ]
```

```
     shape  n100  n200
[1,]   1.6 0.798 0.968
[2,]   1.7 0.508 0.784
[3,]   1.8 0.256 0.444
[4,]   2.0 0.064 0.046
[5,]   2.2 0.174 0.384
[6,]   2.3 0.338 0.642
[7,]   2.4 0.536 0.890
```

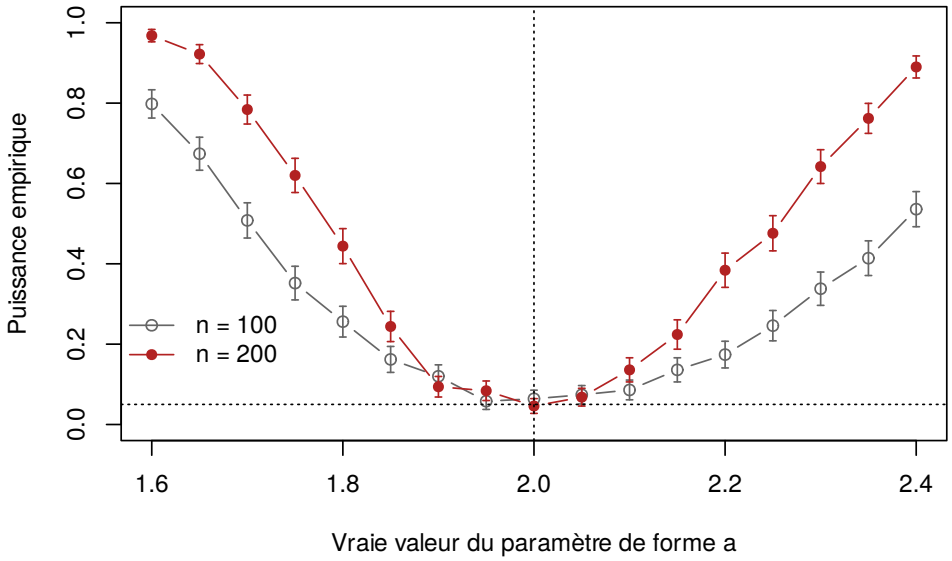


FIGURE 2 – Puissance empirique du test de Wald de $H_0 : (a, b) = (2, 10)$ au seuil de 5 % en fonction de la vraie forme a (échelle $b = 10$), pour $n = 100$ et $n = 200$; $B_p = 500$ réplications par point, barres : IC à 95 % (ETMC au plus 0,022).

La courbe n'est pas symétrique autour de $a = 2$: une même distance en a ne correspond pas à la même distance statistique (la variance de $\widehat{a}$ est proportionnelle à a^2). Les petites irrégularités (par exemple 0,120 pour $n = 100$ contre 0,094 pour $n = 200$ à $a = 1{,}9$, un écart de 0,026 pour une ETMC de la différence d'environ 0,02) sont des fluctuations Monte Carlo, que les barres d'erreur rendent visibles. Le point de vue H26 (générer sous $(2, 10)$, tester $(1{,}7, 10)$) et celui de la courbe (générer sous $(1{,}7, 10)$, tester $(2, 10)$) ne sont pas identiques, mais donnent ici des puissances semblables (0,787 et 0,784).

À retenir

- Sous H_0, les valeurs p d'un test valide sont uniformes : histogramme, QQ-plot en $-\log_{10}$, KS, et surtout erreur de type I aux seuils usuels avec IC.
- Puissance : taux de rejet sous des alternatives ; courbe en fonction de l'écart à H_0 et de n, avec barres d'ETMC.
- Un test dont l'erreur de type I est gonflée n'a pas de puissance interprétable : on valide le niveau d'abord.

5 Générer des données

R offre des générateurs pour les lois usuelles : `runif`, `rnorm`, `rexp`, `rweibull`, `rnbinom`, ou encore `MASS::mvrnorm` pour la loi normale multivariée. Pour une loi qui n'y figure pas, la méthode générale est l'inversion.

5.1 Méthode d'inversion

Théorème 5.1. *Soit F une fonction de répartition et $F^{-1}(u) = \inf\{x : F(x) \geq u\}$ son inverse généralisée. Si $U \sim \mathcal{U}(0,1)$, alors $X = F^{-1}(U)$ a pour fonction de répartition F. Réciproquement, si $X \sim F$ avec F continue, alors $F(X) \sim \mathcal{U}(0,1)$.*

Démonstration. Par continuité à droite de F, $F^{-1}(u) \leq x \iff u \leq F(x)$. Donc $\mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x)$. Pour la réciproque, si F est continue, pour $u \in (0,1)$, $\mathbb{P}(F(X) \geq u) = \mathbb{P}(X \geq F^{-1}(u)) = 1 - F(F^{-1}(u)) = 1 - u$, car $F(F^{-1}(u)) = u$ lorsque F est continue. □

Algorithme : générer $U_1, \ldots, U_n \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$ et poser $X_i = F^{-1}(U_i)$. La méthode est exacte et vectorielle, mais exige une forme explicite (ou un calcul numérique rapide) de F^{-1}.

Exemple 5.2 (Loi triangulaire sur $[0,2]$)**.** La densité est $f(x) = x$ sur $[0,1]$ et $2 - x$ sur $[1,2]$. En intégrant,

$$F(x) = \begin{cases} x^2/2, & 0 \leq x \leq 1, \\ 2x - x^2/2 - 1, & 1 \leq x \leq 2, \end{cases} \qquad F^{-1}(u) = \begin{cases} \sqrt{2u}, & 0 \leq u \leq 1/2, \\ 2 - \sqrt{2(1-u)}, & 1/2 \leq u \leq 1. \end{cases}$$

```
set.seed(7360)
N <- 5000
U <- runif(N)
X <- ifelse(U < 1/2, sqrt(2*U), 2 - sqrt(2*(1 - U)))
round(c(moyenne = mean(X), variance = var(X)), 4)     # théorie : 1 et 1/6
ks.test(X, function(x) ifelse(x < 1, x^2/2, 2*x - x^2/2 - 1))$p.value
hist(X, breaks = 30, freq = FALSE)
lines(c(0, 1, 2), c(0, 1, 0), col = "firebrick", lwd = 2)
```

```
moyenne variance
 1.0011   0.1662
[1] 0.9945854
```

La moyenne et la variance empiriques sont proches des valeurs théoriques 1 et $1/6 \approx 0{,}1667$, et l'histogramme épouse la densité (figure 3, gauche).

5.2 Loi tronquée

On veut générer $X \sim F$ *conditionnellement* à $a \leq X \leq b$ ($a < b$). La fonction de répartition conditionnelle est

$$F_{[a,b]}(x) = \mathbb{P}(X \leq x \mid a \leq X \leq b) = \frac{F(x) - F(a)}{F(b) - F(a)}, \qquad a \leq x \leq b,$$

(0 avant a, 1 après b), dont l'inverse est $F^{-1}_{[a,b]}(u) = F^{-1}\{F(a) + u[F(b) - F(a)]\}$. On génère donc $U \sim \mathcal{U}(0,1)$ et on pose $X = F^{-1}\{F(a) + U[F(b) - F(a)]\}$: c'est l'inversion appliquée à une uniforme restreinte à l'intervalle $[F(a), F(b)]$.

Exemple 5.3 (Loi $\mathcal{N}(2,1)$ tronquée à $[4,5]$)**.**

```
m <- 2; s <- 1; a <- 4; b <- 5
Fa <- pnorm(a, m, s); Fb <- pnorm(b, m, s)
T <- qnorm(Fa + runif(N)*(Fb - Fa), m, s)
round(c(min = min(T), max = max(T), moyenne = mean(T)), 4)
round(Fb - Fa, 5)            # proportion acceptée par un rejet simple
```

```
    min     max moyenne
 4.0001  4.9990  4.3104
[1] 0.0214
```

L'approche naïve (générer des $\mathcal{N}(2,1)$ et ne garder que celles qui tombent dans $[4,5]$) n'accepterait que 2,1 % des tirages ; l'inversion n'en perd aucun.

Queues extrêmes

Si $[a,b]$ est très loin dans la queue, $F(a)$ et $F(b)$ sont arrondis à 1 et l'inversion échoue (`qnorm(1)` vaut `Inf`). On travaille alors sur l'échelle logarithmique de la fonction de survie (`pnorm(..., lower.tail = FALSE, log.p = TRUE)`) ou on utilise un module spécialisé.

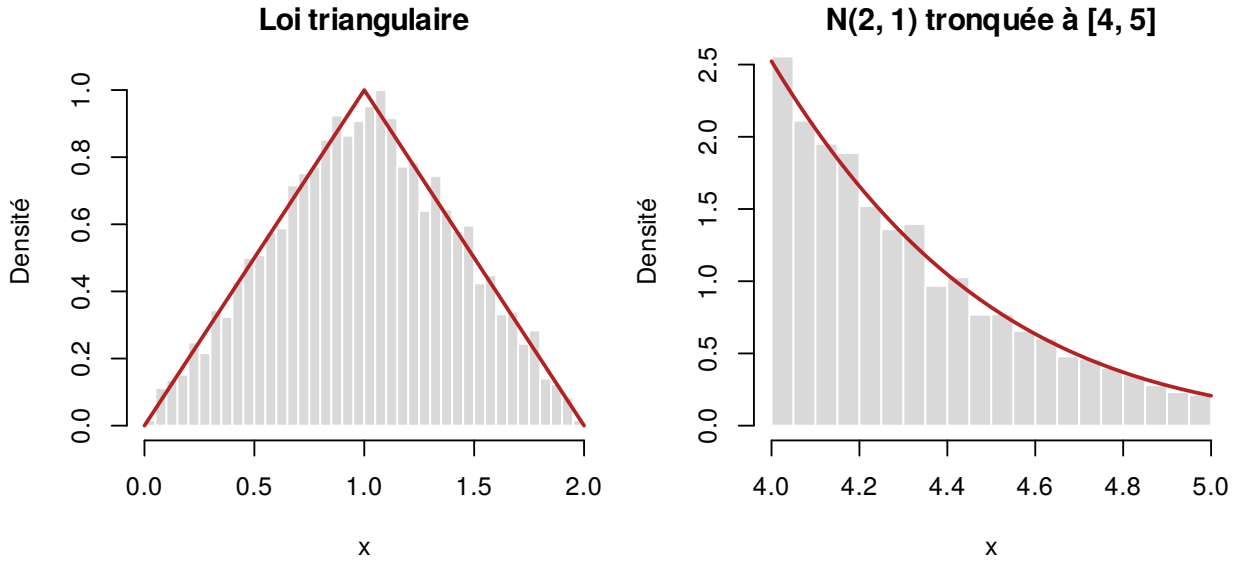


FIGURE 3 – Génération par inversion : loi triangulaire sur $[0,2]$ (gauche, $N = 5000$) et loi $\mathcal{N}(2,1)$ tronquée à $[4,5]$ (droite). En rouge, la densité théorique.

Niveau doctorat : méthode d'acceptation-rejet

Lorsque F^{-1} n'est pas disponible mais que la densité f l'est (à une constante près), on choisit une densité instrumentale g facile à simuler et une constante M telle que $f \le Mg$. On génère $Y \sim g$ et $U \sim \mathcal{U}(0,1)$ indépendants, et on accepte $X = Y$ si $U \le f(Y)/\{Mg(Y)\}$. Les valeurs acceptées suivent exactement f, et la probabilité d'acceptation vaut $1/M$ (si f et g sont

normalisées) : tout l'art consiste à trouver g proche de f. L'approche naïve de l'exemple précédent est le cas $g =$ densité non tronquée, $M = 1/\{F(b) - F(a)\}$.

Données plus complexes. Dans le cours, on générera aussi des covariables corrélées (avec `MASS::mvrnorm`), des effets aléatoires (séance 3), des réponses conditionnelles à des covariables (générer X, puis $Y \mid X$, comme dans l'exemple 6.1 ci-dessous), des données manquantes selon un mécanisme donné (séances 8 et 9). La logique est toujours de générer selon la factorisation de la loi conjointe.

6 Planifier une étude de simulation : le cadre ADEMP

Morris, White et Crowther (2019) proposent de planifier toute étude de simulation, *avant* d'écrire du code, selon cinq rubriques dont l'acronyme est ADEMP. Ce plan tient lieu de protocole ; on vous demandera de le rédiger dans les devoirs et le travail de synthèse.

6.1 Les cinq rubriques

A – Objectifs (*Aims*)
Quelle question précise l'étude doit-elle trancher ? Par exemple : « l'EMV est-il approximativement sans biais à $n = 30$? », « la variance sandwich donne-t-elle des IC de couverture nominale sous surdispersion ? », « quel estimateur a la plus petite EQM ? ». Des objectifs vagues mènent à des scénarios mal choisis.

D – Mécanismes générateurs (*Data-generating mechanisms*)
Le modèle selon lequel on simule et les *scénarios* : valeurs de n, des paramètres, du degré de violation d'une hypothèse (surdispersion, corrélation, proportion de données manquantes…). On choisit des valeurs réalistes (inspirées d'un jeu de données réel) et des valeurs extrêmes qui font échouer les méthodes. Plan factoriel complet (toutes les combinaisons) si le nombre de facteurs est petit ; sinon, faire varier un facteur à la fois autour d'un scénario de référence. Les données peuvent aussi être rééchantillonnées d'un jeu réel (simulation « plasmode »).

E – Quantité estimée (*Estimand*)
La cible, définie précisément et connue (ou calculable) dans chaque scénario : un coefficient de régression, un effet marginal, une probabilité… Attention aux méthodes qui ne visent pas la même quantité (par exemple, effet conditionnel d'un modèle mixte et effet marginal d'une GEE, séances 3 et 6) : la « vraie valeur » diffère alors d'une méthode à l'autre.

M – Méthodes
Les estimateurs, tests ou procédures comparés, y compris une méthode de référence (l'EMV du vrai modèle, la méthode usuelle), avec tous leurs réglages (logiciel, options, critère de convergence).

P – Mesures de performance
Choisies en fonction des objectifs (tableau 1) : biais, écart type empirique, erreur type moyenne, EQM, couverture, erreur de type I, puissance, temps de calcul, taux d'échec de convergence. Chaque mesure est accompagnée de son ETMC.

6.2 Choisir le nombre de réplications B

Le nombre de réplications se choisit comme une taille d'échantillon : on fixe l'ETMC tolérable pour la mesure de performance la plus importante, et on résout.

— *Couverture ou erreur de type I* : $\text{ETMC} = \sqrt{p(1-p)/B}$, d'où $B = p(1-p)/\,\text{ETMC}^2$. Pour une couverture attendue de 95 % et une ETMC de 0,5 %, $B = 0{,}95 \times 0{,}05/0{,}005^2 = 1900$. Si la couverture peut être loin de 95 %, le pire cas $p = 0{,}5$ donne $B = 0{,}25/\,\text{ETMC}^2 = 10\,000$. Pour une erreur de type I de 1 % estimée à ±0,2 % près (ETMC 0,1 %), il faut $B \approx 9900$.

— *Biais* : $\text{ETMC} = S_{\widehat{\theta}}/\sqrt{B}$, d'où $B = S^2_{\widehat{\theta}}/\,\text{ETMC}^2$, où $S_{\widehat{\theta}}$ provient d'une petite étude pilote ou de la théorie. Pour la forme de la loi de Weibull ($S_{\widehat{\theta}} \approx 0{,}049$ à $n = 1000$), une ETMC de 0,001 demande $B \approx 2400$.

— *Comparaisons* : appliquer toutes les méthodes aux *mêmes* jeux de données simulés. Les estimations sont alors positivement corrélées et la variance Monte Carlo de leur différence est bien plus petite que la somme des variances : on compare à précision égale avec moins de réplications.

On commence par une exécution avec un petit B (10 à 100) pour vérifier le code, mesurer le temps de calcul et obtenir les écarts types pilotes, puis on lance l'étude complète.

6.3 Programmer et rapporter

— Écrire une fonction qui génère *un* jeu de données, une fonction par méthode, et une fonction qui exécute une réplication ; conserver les estimations brutes de chaque réplication (pas seulement les résumés), pour pouvoir calculer de nouvelles mesures sans tout relancer.

— Vérifier le mécanisme générateur sur un très grand échantillon (les paramètres simulés sont-ils retrouvés ?).

— Prévoir les échecs (non-convergence, erreurs) avec `tryCatch`, les compter et les rapporter ; ne jamais les éliminer silencieusement, car ils ne surviennent pas au hasard (souvent dans les échantillons les plus difficiles).

— Rapporter le plan ADEMP, B et sa justification, la graine, le logiciel, et les résultats sous forme de tableaux ou de graphiques avec les ETMC. Les résultats de nombreux scénarios se présentent bien par des graphiques en treillis (un panneau par facteur).

6.4 Exemple complet : erreurs types fondées sur le modèle ou sandwich

Exemple 6.1. Reprenons l'exemple de la régression de Poisson de la séance 1 et planifions l'étude qui le valide.

A Évaluer, pour l'effet d'un traitement dans une régression de Poisson, la couverture des IC à 95 % fondés sur l'erreur type du modèle et sur l'erreur type sandwich, avec et sans surdispersion.

D $x_i \sim \text{Bernoulli}(0{,}5)$, $\mu_i = \exp(1 + 0{,}4x_i)$; $Y_i \mid x_i$ de Poisson (pas de surdispersion) ou binomiale négative de paramètre de forme 1,5 ($\text{Var} = \mu + \mu^2/1{,}5$) ; $n \in \{50, 200\}$. Plan factoriel : 4

scénarios.

E $\beta_1 = 0{,}4$, le log du rapport des taux, identique dans les deux lois puisque la moyenne conditionnelle est la même.

M EMV de Poisson (`glm`) avec (i) l'erreur type du modèle, (ii) l'erreur type sandwich.

P Biais, écart type empirique, erreurs types moyennes, couverture. La couverture est la mesure principale : $B = 2000$ donne une ETMC de $\sqrt{0{,}95 \times 0{,}05/2000} \approx 0{,}5$ %.

```
sandwich_se <- function(fit) {
  X <- model.matrix(fit); mu <- fitted(fit); r <- fit$y - mu
  Ainv <- solve(crossprod(X, X * mu))
  Bm <- crossprod(X * r)
  c(modele = sqrt(Ainv[2, 2]), sandwich = sqrt((Ainv %*% Bm %*% Ainv)[2, 2]))
}
une_rep <- function(n, taille, beta = c(1, 0.4)) {
  x <- rbinom(n, 1, 0.5)
  mu <- exp(beta[1] + beta[2]*x)
  y <- if (is.infinite(taille)) rpois(n, mu) else rnbinom(n, mu = mu, size = taille)
  fit <- glm(y ~ x, family = poisson)
  c(est = unname(coef(fit)[2]), sandwich_se(fit))
}
scenarios <- expand.grid(n = c(50, 200), taille = c(Inf, 1.5))
B <- 2000
set.seed(20270122)
resultats <- lapply(seq_len(nrow(scenarios)), function(s)
  t(replicate(B, une_rep(scenarios$n[s], scenarios$taille[s]))))
perf <- function(r, beta1 = 0.4) {
  B <- nrow(r); est <- r[, "est"]; se.emp <- sd(est)
  couv <- function(se) mean(abs(est - beta1) <= qnorm(0.975)*se)
  c(biais = mean(est) - beta1, etmc.biais = se.emp/sqrt(B),
    se.emp = se.emp, etmc.se.emp = se.emp/sqrt(2*(B - 1)),
    se.mod = mean(r[, "modele"]), se.sand = mean(r[, "sandwich"]),
    couv.mod = couv(r[, "modele"]), couv.sand = couv(r[, "sandwich"]))
}
cbind(scenarios, t(sapply(resultats, perf)))
```

L'exécution prend environ une minute. Le tableau 2 met en forme la sortie.

TABLE 2 – Résultats de l'étude de simulation ($B = 2000$ par scénario ; ETMC entre parenthèses ; ETMC des couvertures : 0,005). BN : binomiale négative de forme 1,5.

			Écart type (ÉT)			Couverture	
Loi	n	Biais	empirique	modèle	sandwich	modèle	sandwich
Poisson	50	−0,002 (0,004)	0,158 (0,002)	0,159	0,155	0,947	0,939
Poisson	200	0,000 (0,002)	0,079 (0,001)	0,079	0,078	0,952	0,952
BN	50	−0,003 (0,006)	0,285 (0,005)	0,161	0,271	0,732	0,928
BN	200	0,004 (0,003)	0,139 (0,002)	0,079	0,139	0,740	0,946

Interprétation. L'EMV de Poisson est sans biais dans les quatre scénarios, y compris sous surdispersion : l'équation d'estimation est sans biais dès que la moyenne est correctement spécifiée (séance 1). Sans surdispersion, les deux erreurs types sont correctes. Sous surdispersion, l'erreur type du

modèle est environ 45 % trop petite et la couverture chute à 73–74 %, que n soit petit ou grand : augmenter n ne corrige pas une variance mal spécifiée. L'erreur type sandwich suit l'écart type empirique, mais la couverture reste un peu sous 95 % à $n = 50$ (0,939 et 0,928, soit 2 à 4,5 ETMC sous la valeur nominale) : c'est le biais de petit échantillon connu de l'estimateur sandwich, qui motive les corrections de petits échantillons utilisées avec les GEE (séance 6).

Grille méthodologique : l'étude de simulation

1. **Objet d'intérêt** : la performance (biais, variance, couverture, niveau, puissance) d'une méthode sous un mécanisme générateur donné (A et E d'ADEMP).
2. **Nuisance et biais** : l'erreur Monte Carlo ; des scénarios non représentatifs ; des échecs de convergence non aléatoires.
3. **Approche** : plan ADEMP ; scénarios réalistes et extrêmes ; mêmes jeux de données pour toutes les méthodes.
4. **Critère** : les mesures de performance (P), définies avant de lancer l'étude.
5. **Calcul** : générateur pseudo-aléatoire reproductible, inversion, code vectoriel ou parallèle.
6. **Inférence** : chaque mesure de performance est une moyenne de B termes i.i.d. : ETMC et IC par le TCL ou la loi binomiale ; B choisi pour atteindre l'ETMC visée.

À retenir

— Une étude de simulation est une expérience : protocole (ADEMP), taille (B) justifiée, résultats avec incertitude (ETMC).

— $B = p(1-p)/\,\text{ETMC}^2$ pour une couverture ou un taux de rejet ; $B = S^2_{\widehat{\theta}}/\,\text{ETMC}^2$ pour un biais.

— Conserver les résultats bruts, compter les échecs, fixer la graine, comparer les méthodes sur les mêmes données.

7 Exercices

Les exercices 7.1 à 7.4 sont repris de la série « Simulations – Exercices » (H26, L. Lakhal-Chaieb). Pour chacun, rédigez d'abord un plan ADEMP succinct et justifiez B.

Exercice 7.1 (Simulations H26, exercice 1)**.** Soit $X_1, \ldots, X_n$ un échantillon de la loi de Poisson de paramètre θ. On propose trois estimateurs de $\gamma = e^{-\theta} = \mathbb{P}(X = 0)$:

$$\widehat{\gamma}_1 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{X_i = 0\}, \qquad \widehat{\gamma}_2 = e^{-\bar{X}}, \qquad \widehat{\gamma}_3 = \Big(1 - \frac{1}{n}\Big)^{\sum_i X_i}.$$

Proposer une étude de simulation (avec $n = 100$) afin de vérifier si ces trois estimateurs sont approximativement sans biais. Lequel est le meilleur ?

Exercice 7.2 (Simulations H26, exercice 2)**.** Soit $X_1, \ldots, X_n$ un échantillon de densité $f_\theta(x) = \theta x^{\theta-1}\mathbf{1}\{x \in [0,1]\}$. À l'exercice 5.3 des notes de la séance 1, on a obtenu l'estimateur des moments $\tilde{\theta} = \bar{X}/(1-\bar{X})$ et l'EMV $\widehat{\theta} = -n/\sum_i \log X_i$, ainsi que leurs variances asymptotiques.

a) Proposer un algorithme pour générer des données de cette loi.

b) Comparer la performance des deux estimateurs pour $n \in \{30, 100\}$ et $\theta \in \{3, 5\}$.

c) Proposer un algorithme pour valider les expressions des variances asymptotiques de $\tilde{\theta}$ et de $\widehat{\theta}$. Quelle est votre conclusion ?

Exercice 7.3 (Simulations H26, exercice 3)**.** Proposer une étude de simulation pour estimer $\mathrm{E}(X \mid X > 1)$ et $\mathrm{Var}(X \mid X > 1)$, où X suit une loi de Weibull de paramètres de forme et d'échelle $(2, 2)$. Donner la précision de vos estimations, et comparer aux valeurs théoriques obtenues par intégration numérique.

Exercice 7.4 (Simulations H26, exercice 4)**.** Soit $X_1, \ldots, X_n$ un échantillon de la loi exponentielle de taux λ. On propose la statistique de Wald

$$Q = \frac{\widehat{\lambda} - 1}{\sqrt{\widehat{\lambda}^2 / n}}, \qquad \widehat{\lambda} = 1/\bar{X},$$

pour tester $H_0 : \lambda = 1$ contre $H_1 : \lambda \neq 1$; on rejette H_0 au seuil de 5 % si $|Q| > 1{,}96$. Proposer une étude de simulation pour vérifier empiriquement la validité de ce test et pour tracer sa courbe de puissance lorsque $n = 100$. Comparer avec le test du rapport de vraisemblances.

Exercice 7.5 (Nouveau, H27)**.** Reprendre l'exercice 5.5 des notes de la séance 1 (modèle exponentiel ajusté à des données de Weibull) : planifier selon ADEMP puis programmer une étude qui compare la couverture des IC fondés sur la variance « modèle » $\widehat{\lambda}^2/n$ et sur la variance sandwich, pour $\alpha \in \{0{,}5; 1; 2\}$ et $n \in \{30, 100, 500\}$.

Références

— Burton, A., Altman, D. G., Royston, P. et Holder, R. L. (2006). The design of simulation studies in medical statistics. *Statistics in Medicine*, 25, 4279–4292.

— Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer.

— Koehler, E., Brown, E. et Haneuse, S. J.-P. A. (2009). On the assessment of Monte Carlo error in simulation-based statistical analyses. *The American Statistician*, 63, 155–162.

— Matsumoto, M. et Nishimura, T. (1998). Mersenne Twister : a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8, 3–30.

— Morris, T. P., White, I. R. et Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, 38, 2074–2102.

— Rizzo, M. L. (2019). *Statistical Computing with R*, 2e éd. Chapman & Hall/CRC.

— Robert, C. P. et Casella, G. (2004). *Monte Carlo Statistical Methods*, 2e éd. Springer.

— Lakhal-Chaieb, L. et Bureau, A. (2024). *STT-7360 : Simulations*, *Simulations – Exercices – Solutions* et script `simu.R`. Notes de cours, Université Laval.