Projet statistiques bayésiennes

Garance MALNOË et Matthias MAZET

2026-01-26

Contents


```
# Packages nécessaires
library(ggplot2)
library(ggpubr)
library(GGally)
```

1 Introduction

À partir d'un processus gamma non homogène simulé, nous voulons construire une inférence bayésienne permettant de retrouver les paramètres du dit processus. Nous supposons que ce processus possède un *paramètre de forme* $a(t) = \alpha t^{\beta}$, avec $\alpha, \beta > 0$, et un *paramètre d'échelle* $\theta > 0$. Nous posons aussi, pour tout $t > s \geq 0$, $X(t) - X(s) \sim \mathcal{G}(a(t) - a(s), \theta)$ et, pour tout $x \in \mathbb{R}_+$, $f_{a(t)-a(s),\theta}(x) = \frac{x^{a(t)-a(s)-1}}{\theta^{a(t)-a(s)}\Gamma(a(t)-a(s))} e^{-\frac{x}{\theta}}$.
L'inférence bayésienne va donc nous servir à retrouver 3 paramètres : α, β et θ. Afin d'analyser la pertinence de l'approche bayésienne face à un problème de ce type, nous avons séparé notre analyse en x phases : i) la simulation du processus selon différentes lois pour α, β et θ afin d'observer leur impact respectif, ii) la mise en place de l'inférence bayésienne et iii) l'étude de l'impact du pas d'inspection.
Afin de se fixer un cadre d'étude, nous restreignons les possibilités pour les lois théoriques de chaque paramètre :

- α et β suivront chacun soit une loi *lognormale*, soit une loi *Gamma*.
- θ suivra une loi *Inversegamma*.

2 Simulations

```r
# Fonction pour simuler un processus gamma
simulations <- function(horizon = 100, dt_sim = 0.01, dt_insp = 5, alpha, beta, theta){
  # Simulations du processus continu, avec un pas de simulation dt_sim
  grille_sim <- seq(0, horizon, by = dt_sim)
  a <- function(t) {alpha*t^beta}  # val de a(t)
  shape <- diff(a(grille_sim))     # a(t) - a(s)
  increments <- rgamma(length(shape), shape=shape, scale=theta)
  X_sim <- c(0, cumsum(increments))
  # Calcul des inspections, avec un pas d'inspection dt_insp
  grille_insp <- seq(0, horizon, by = dt_insp)
  n <- dt_insp / dt_sim
  X_insp <- X_sim[seq(1, length(X_sim), by = n)]
  # Résultats
  list(
    grille_sim = grille_sim, X_sim = X_sim,     # Résultats simulés
    grille_insp = grille_insp, X_insp = X_insp  # Résultats inspectés
  )
}

# Fonction graphique d'un processus
plot_simulations <- function(simulations, subtitle = "") {
  ggplot() +
    geom_line(
      aes(x = simulations$grille_sim, y = simulations$X_sim, col = "Simulées")
      ) +
    geom_point(
      aes(x = simulations$grille_insp, y = simulations$X_insp, col = "Inspectées"),
      shape = 18, size = 2
      ) +
    scale_color_manual(
      name = "Type de données",
      values = c("Simulées" = "#2d0569", "Inspectées" = "#db2e0b"),
    ) +
    labs(x = "t", y = expression(X[t]), subtitle = subtitle) +
    theme_light() +
    theme(
      axis.title = element_text(size = 10, face = "bold"),
      axis.text = element_text(size = 8),
      panel.border = element_blank(),
      axis.line = element_line(colour = "darkgrey"),
```

```
      panel.grid.minor = element_blank(),
      panel.grid.major = element_line(color = "grey85"),
      legend.title = element_text(size = 10, face = "bold"),
      legend.text = element_text(size = 8),
      legend.position = "right"
    )
}
```

2.1 *Variations de alpha*

```
# Seed pour la reproductibilité
set.seed(1)

# Paramètres de simulations
dt_sim <- 0.01
dt_insp <- 5
beta <- 2
theta <- 2
# Valeurs de alpha
list_alpha <- c(.1, 1, 10, 20)

# Simulations selon les valeurs de alpha de list_alpha et les paramètres fixés
sim1 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = list_alpha[1], beta = beta, theta = theta
)
plot_sim1 <- plot_simulations(
  sim1, subtitle = paste(expression(alpha), "=", list_alpha[1])
)
sim2 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = list_alpha[2], beta = beta, theta = theta
)
plot_sim2 <- plot_simulations(
  sim2, subtitle = paste(expression(alpha), "=", list_alpha[2])
)
sim3 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = list_alpha[3], beta = beta, theta = theta
)
plot_sim3 <- plot_simulations(
  sim3, subtitle = paste(expression(alpha), "=", list_alpha[3])
)
sim4 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = list_alpha[4], beta = beta, theta = theta
)
plot_sim4 <- plot_simulations(
  sim4, subtitle = paste(expression(alpha), "=", list_alpha[4])
)

ggarrange(plot_sim1, plot_sim2, plot_sim3, plot_sim4, common.legend = TRUE)
```

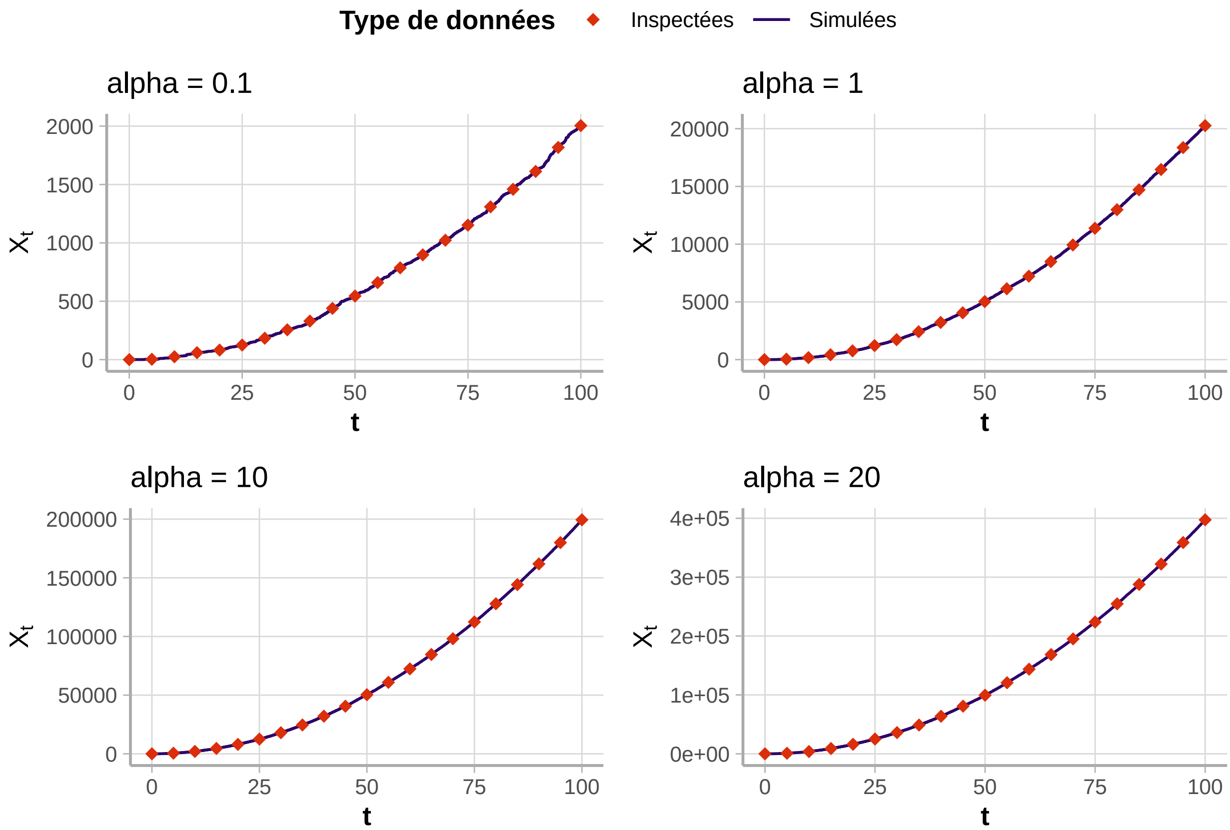

Figure 1: Simulations d'un processus gamma non-homogène ($\beta = 2$, $\theta = 2$) pour différents α, avec un pas de simulation de 0.01 et un pas d'inspection de 5

Commentaire : Pas d'impact sur la forme de la courbe seulement sur les valeurs prises par Xt. Proportionnel à alpha : 0.1 -> 2000 1 -> 20.000 10 -> 200.000 20 -> 400.000 Faire le lien avec les maths / la théorie. Dégradations + rapide avec alpha + grand.

2.2 *Variations de beta*

```r
# Seed pour la reproductibilité
set.seed(1)

# Paramètres de simulations
dt_sim <- 0.01
dt_insp <- 5
alpha <- 1
theta <- 2
# Valeurs de beta
list_beta <- c(0.5, 1, 2, 3)

# Simulations selon les valeurs de alpha de list_beta et les paramètres fixés
sim1 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = alpha, beta = list_beta[1], theta = theta
)
plot_sim1 <- plot_simulations(
  sim1, subtitle = paste(expression(beta), "=", list_beta[1])
)
sim2 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = alpha, beta = list_beta[2], theta = theta
)
plot_sim2 <- plot_simulations(
  sim2, subtitle = paste(expression(beta), "=", list_beta[2])
```

```r
)
sim3 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = alpha, beta = list_beta[3], theta = theta
)
plot_sim3 <- plot_simulations(
  sim3, subtitle = paste(expression(beta), "=", list_beta[3])
)
sim4 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = alpha, beta = list_beta[4], theta = theta
)
plot_sim4 <- plot_simulations(
  sim4, subtitle = paste(expression(beta), "=", list_beta[4])
)

ggarrange(plot_sim1, plot_sim2, plot_sim3, plot_sim4, common.legend = TRUE)
```

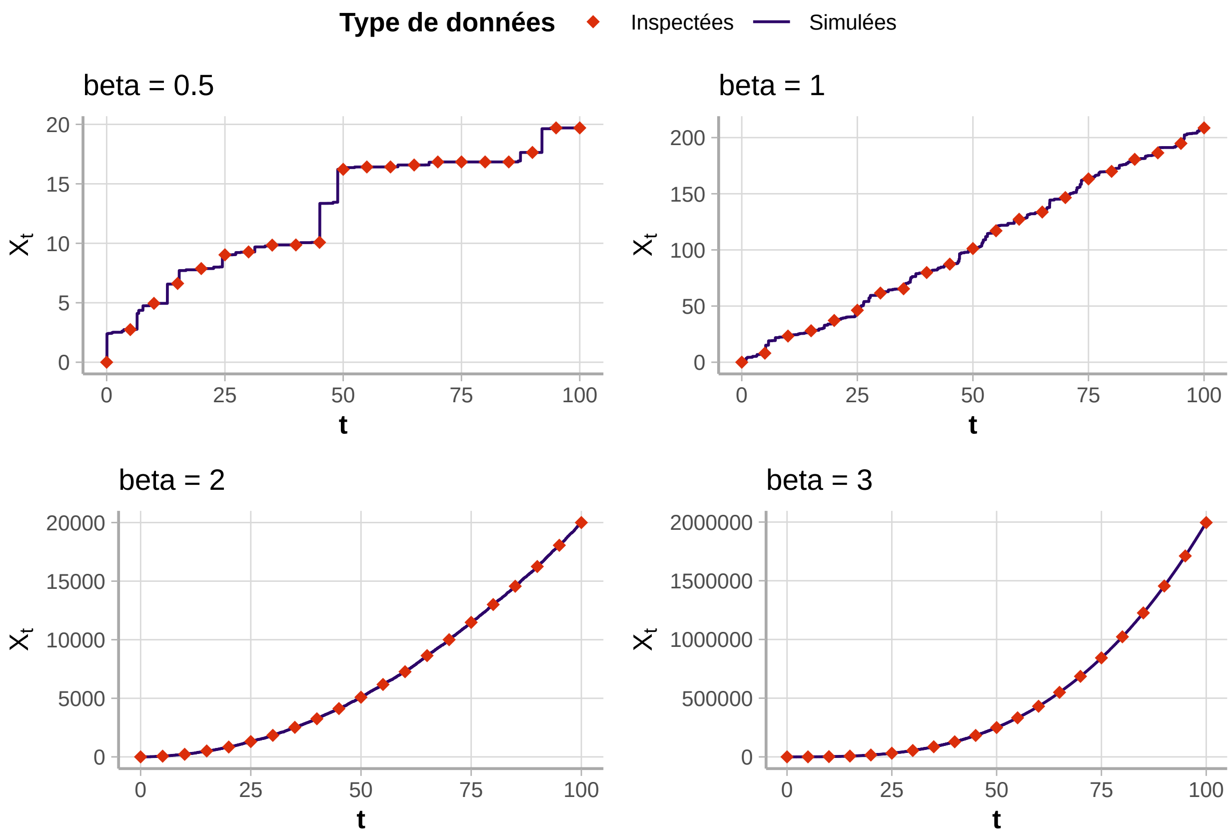

Figure 2: Simulations d'un processus gamma non-homogène ($\alpha = 1$, $\theta = 2$) pour différents β, avec un pas de simulation de 0.01 et un pas d'inspection de 5

Commentaire : beta $<$ 1, dégradation qui ralenti, forme concave beta $=$ 1, dégradation linéaire, forme linéaire beta $>$ 1, dégradation accélérée, forme convexe

2.3 *Variations de theta*

```r
# Seed pour la reproductibilité
set.seed(1)

# Paramètres de simulations
dt_sim <- 0.01
dt_insp <- 5
```

```r
alpha <- 1
beta <- 1
# Valeurs de theta
list_theta <- c(0.5, 1, 2, 5)

# Simulations selon les valeurs de alpha de list_beta et les paramètres fixés
sim1 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = alpha, beta = beta, theta = list_theta[1]
)
plot_sim1 <- plot_simulations(
  sim1, subtitle = paste(expression(theta), "=", list_theta[1])
)
sim2 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = alpha, beta = beta, theta = list_theta[2]
)
plot_sim2 <- plot_simulations(
  sim2, subtitle = paste(expression(theta), "=", list_theta[2])
)
sim3 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = alpha, beta = beta, theta = list_theta[3]
)
plot_sim3 <- plot_simulations(
  sim3, subtitle = paste(expression(theta), "=", list_theta[3])
)
sim4 <- simulations(
  dt_sim = dt_sim, dt_insp = dt_insp, alpha = alpha, beta = beta, theta = list_theta[4]
)
plot_sim4 <- plot_simulations(
  sim4, subtitle = paste(expression(theta), "=", list_theta[4])
)

ggarrange(plot_sim1, plot_sim2, plot_sim3, plot_sim4, common.legend = TRUE)
```

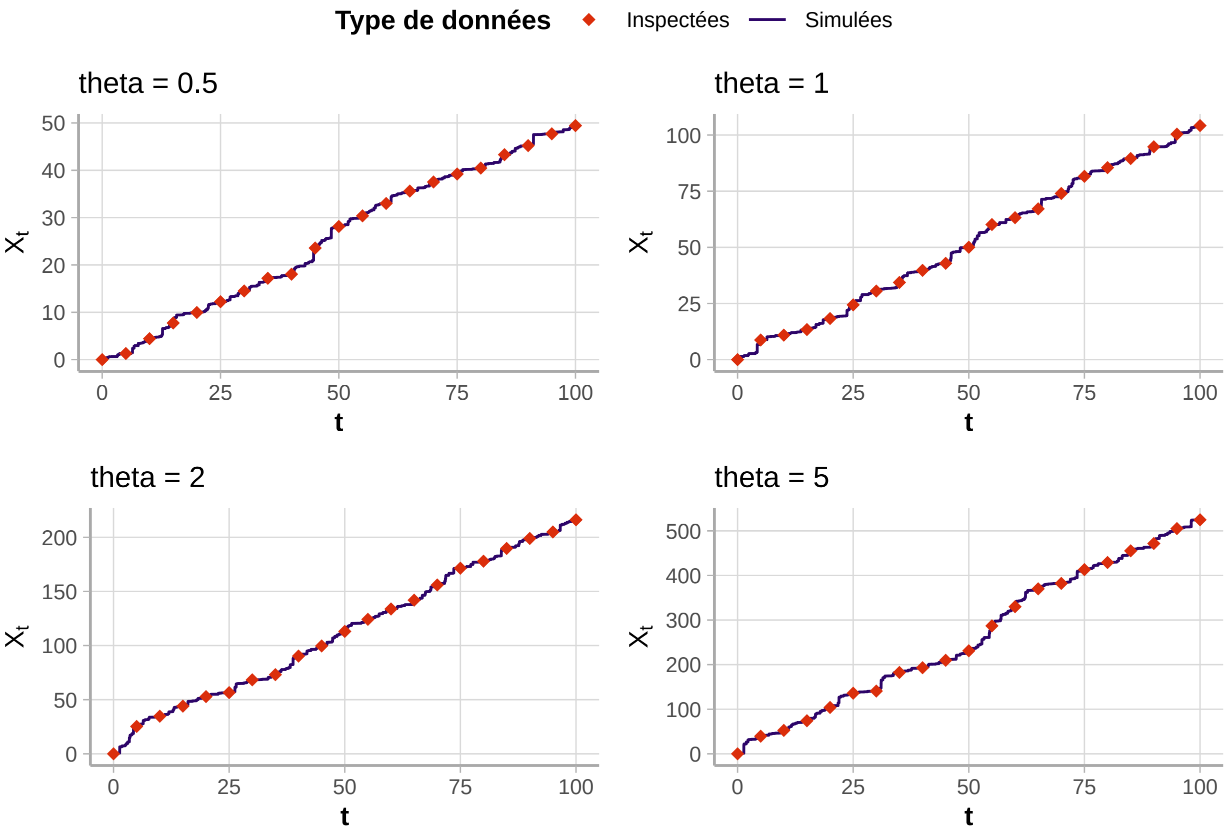

Figure 3: Simulations d'un processus gamma non-homogène ($\alpha = 1$, $\beta = 1$) pour différents θ, avec un pas de simulation de 0.01 et un pas d'inspection de 5

Commentaire : Pas d'impact sur la forme de la courbe seulement sur les valeurs prises par Xt. Proportionnel à theta : 0.5 -> 50 1 -> 100 2 -> 200 Dégradation + rapide avec theta + grand Faire le lien avec les maths / la théorie.

3 Etude de l'inférence bayésienne

4 Conclusion