

République Algérienne Démocratique et Populaire.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.
Université Mouloud Mammeri, Tizi-Ouzou.
Faculté des Sciences.
Département de Mathématiques.



Spécialité : Mathématiques
Option : Probabilités et statistique
Mémoire de fin de cycle intitulé :

Inférence statistique dans les modèles Bêta ARMA

Réalisé par :

Tareb Lydia

Encadré par :

M^{me} Atil Lynda

Soutenu devant le jury d'examen composé de :

M^r Fellag Hocine,	Professeur, UMMTO,	Président.
M^{me} Atil Lynda,	Maître de conférences A, UMMTO,	Rapporteur.
M^{me} Belkacem Cherifa,	Maître de conférences B, UMMTO,	Examinatrice.

Table des matières

Table des matières	1
Introduction générale	5
1 Les modèles ARMA	6
1.1 Introduction	6
1.2 Les modèles ARMA(p,q)	6
1.2.1 Définition	6
1.2.2 Exemples de modèles ARMA(p,q)	7
1.2.3 Propriétés statistiques	8
1.2.4 Les fonctions d'autocovariance et d'autocorrélation	10
1.2.5 Exemple d'application sous R	12
1.3 L'estimation dans les modèles ARMA	13
1.3.1 La méthode de Box-Jenkins	16
1.3.2 Les tests d'hypothèses	17
1.4 Les domaines d'application des modèles $ARMA$	18
1.5 Conclusion	19
2 Méthodes de Monte-Carlo	20
2.1 Introduction	20
2.2 Les cas d'application des méthodes de Monte-Carlo	20
2.3 Simulation des variables aléatoires	21
2.3.1 La simulation de la loi uniforme	22
2.3.2 La simulation de la loi normale	24
2.3.3 La méthode de Rejet	26
2.4 Les méthodes de Monte-Carlo par chaînes de Markov	29
2.4.1 Définition (Chaînes de Markov)	29
2.4.2 Les méthodes de Monte-Carlo par chaînes de Markov	29
2.4.3 Quelques méthodes MCMC	29
2.5 Conclusion	34
3 Les modèles bêta $ARMA$	35
3.1 Introduction	35
3.2 Les modèles bêta $ARMA$	36
3.2.1 Définition	36
3.2.2 Les propriétés des modèles $\beta ARMA$	37
3.2.3 L'estimation dans les modèles $\beta ARMA(p, q)$	38
3.2.4 Estimation par les méthodes MMC	39
3.3 Tests d'adéquation	43

3.3.1	Définition	43
3.4	Application sur données réelles	46
3.5	Conclusion	52
	Conclusion générale	53
	Bibliographie	54

Dédicace

*Je dédie cet événement marquant de ma vie et cet humble travail,
Tout d'abord, à mes chers parents qui m'ont soutenu pour arriver là où je suis au-
jourd'hui. Je leur adresse un merci tout particulier même si, peu importe combien je
les remercie, cela ne suffirait jamais compte tenu des efforts qu'ils ont déployé pour
m'élever et m'éduquer. Que Dieu les protège et les garde pour moi.*

*À ma chère encadrante Madame Atil Lynda et à tous mes enseignants, je les
remercie infiniment pour la qualité de leur enseignement.*

*À mes chers frères Arezki et Ali qui m'ont encouragé tout au long de mon par-
cours.*

*À tous les membres de ma famille, merci pour leur amour et leurs encourage-
ments.*

*À mes sœurs adorées Soussou, Nina, Celia, Lyza, Mounia qui n'ont jamais cessé
de me soutenir ainsi qu'aux autres, chacune en son nom.*

*Pour finir, à toutes les personnes qui m'ont soutenu de près ou de loin, je les
remercie beaucoup pour leur soutien.*

Tareb Lydia

Remerciements

Avec l'expression de ma reconnaissance, j'adresse mes sincères remerciements avant tout à Dieu le Tout-Puissant qui m'a accordé la patience et la volonté d'accomplir ce mémoire.

J'adresse un merci indescriptible à mes chers parents qui m'ont accompagné dans toutes les étapes de ma vie. Leur soutien est le secret de ma réussite, ils méritent donc tout le respect.

Je tiens à remercier tout particulièrement ma chère encadrante, madame Atil Lynda, qui m'a soutenue tout au long de cette période par ses précieux conseils et orientations. Elle a été mon bras droit à tout moment.

J'adresse un grand merci au meilleur professeur M. Fellag et à toutes mes chères enseignantes M^{me} Merabet, M^{me} Belkacem, M^{me} Boualem pour les efforts qu'ils ont déployés pour nous éduquer du mieux possible.

Je remercie infiniment mes amies et cousines que je considère comme mes sœurs, d'être restées à mes côtés, notamment dans les moments les plus difficiles, et qui m'ont encouragée à travailler et à persévérer pour atteindre mes objectifs.

Enfin, je suis reconnaissante à toutes les personnes qui se sont tenues à mes côtés et qui ont contribué à la réalisation de ce mémoire.

Introduction générale

L'analyse des séries chronologiques joue un rôle crucial dans de nombreux domaines, où la modélisation de l'évolution des données au fil du temps permet de comprendre des phénomènes complexes et d'établir des prévisions.

Les modèles bêta *ARMA*, qui reposent sur la distribution bêta, permettent de modéliser des séries temporelles où les variables prennent des valeurs dans l'intervalle $(0,1)$, comme c'est souvent le cas pour des données de proportions et de probabilités. Ces modèles offrent une modélisation plus précise des séries chronologiques présentant des caractéristiques spécifiques. Ce mémoire s'articule autour de l'étude des modèles bêta *ARMA*. Nous commencerons par un aperçu approfondi des modèles *ARMA* traditionnels dans le premier chapitre, où nous présenterons ces modèles et leurs propriétés, ainsi que les méthodes d'estimation classiques, une vue sur les tests d'hypothèses et quelques exemples d'application sous le langage R.

Le deuxième chapitre introduit les méthodes de simulation de Monte-Carlo, notamment les méthodes basées sur les chaînes de Markov (MCMC), qui sont indispensables pour l'estimation des modèles complexes tels que les modèles bêta *ARMA*. Le dernier chapitre s'intéresse spécifiquement aux modèles β *ARMA*, en présentant ces processus et leurs origines. Ensuite, proposons des méthodes d'estimation des modèles β *ARMA*, notamment la méthode numérique de Newton et la méthode MCMC adaptées à ces modèles. De plus, nous examinerons les tests d'adéquation, où nous nous intéressons aux tests de Portemanteau qui évaluent la performance des modèles β *ARMA*.

Enfin, nous illustrons l'efficacité de ces approches par une application aux données réelles. L'objectif de ce travail est d'offrir une vue complète des modèles bêta *ARMA*, tout en proposant des techniques d'estimation ainsi que des tests pour évaluer la performance de ces modèles. Cette contribution vise à enrichir les outils de modélisation des séries temporelles et à apporter des solutions adaptées à des problématiques plus complexes.

Chapitre 1

Les modèles ARMA

1.1 Introduction

En statistique, la théorie des séries chronologiques (ou temporelle) est une notion importante pour l'interprétation des données et la prédiction des valeurs futures d'une certaine variable. Autrement dit, une série temporelle représente l'évolution d'une variable à travers le temps. Mathématiquement, cette dernière peut être exprimée comme un processus aléatoire noté souvent $(X_t)_{t \in \mathbb{Z}}$ ou bien $\{X_t, t \in \mathbb{Z}\}$ lorsque ce processus est à temps discret. Cependant, quand $t \in \mathbb{R}$, le processus est à temps continu.

Dans ce chapitre, notre intérêt se dirige vers les modèles *ARMA* (Autoregressive Moving Average), qui sont reconnus comme un outil essentiel pour la modélisation et l'analyse des séries temporelles.

1.2 Les modèles ARMA(p,q)

1.2.1 Définition

Les modèles *ARMA* sont des outils statistiques spécialement conçus pour la prédiction et l'analyse des comportements observés dans les séries temporelles.

Ces modèles ont été créés en 1950 par le statisticien "Herman Wold", qui a montré la capacité de ces processus à modéliser n'importe quelle série temporelle stationnaire. Ce type de modèles représente un cas particulier d'un modèle *ARIMA* (Autoregressive Integrated Moving Average), dont le *I* est un terme qui signifie "intégration", tel que, *I*(*d*) indique que la série a été différenciée "*d*" fois pour qu'elle devienne stationnaire.

En effet, le processus *ARMA* est une combinaison de deux composantes,

- Une partie autorégressive (*AR*) d'ordre *p*, qui utilise les *p* observations précédentes pour prédire la prochaine valeur.
- Une autre partie moyenne mobile (*MA*) d'ordre *q*, qui utilise les *q* erreurs de prédiction précédentes pour prédire la prochaine observation.

Dans un modèle *ARIMA*(*p, d, q*), lorsque l'ordre d'intégration est égal à zéro, ce modèle se réduit à un *ARMA*(*p, q*).

Avant de donner la formule mathématique d'un modèle *ARMA*, nous allons tout

d'abord définir l'opérateur "B".

Définition

L'opérateur noté B est appelé "opérateur retard", et il est défini comme suit,

$$B^n X_t = X_{t-n}, \quad \forall n \in \mathbb{N}$$

Et donc,

$$B^n \varepsilon_t = \varepsilon_{t-n}, \quad \forall n \in \mathbb{N}$$

Dans ce qui suit, les résidus sont supposés gaussiens.

Mathématiquement, un modèle ARMA(p,q) peut être défini comme indiqué ci-dessous,

$$X_t = \sum_{i=1}^p \phi_i X_{t-i} + \varepsilon_t + \sum_{j=1}^q \theta_j \varepsilon_{t-j}, \quad \varepsilon_t \sim N(0, \sigma^2)$$

Ce qui équivaut à,

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}$$

Cela revient à,

$$X_t (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) = \varepsilon_t (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q)$$

Ou encore,

$$\phi(B)X_t = \Theta(B)\varepsilon_t$$

où,

- x_t est la valeur observée à l'instant t .
- (ϕ_1, \dots, ϕ_p) sont les coefficients d'un autorégressif d'ordre p .
- $(\theta_1, \dots, \theta_q)$ sont les coefficients d'une moyenne mobile d'ordre q .
- $(\varepsilon_t, \varepsilon_{t-1}, \dots, \varepsilon_{t-q})$ sont les termes d'erreurs qui sont souvent considérés comme des bruits blancs iid (indépendants identiquement distribués), tel que, $\mathbb{E}(\varepsilon_t) = 0$ et $\mathbb{V}(\varepsilon_t) = \sigma^2$.
- $\phi(B)$ est un polynôme d'ordre p .
- $\Theta(B)$ est un polynôme d'ordre q .

1.2.2 Exemples de modèles ARMA(p,q)

Exemple 1

Un modèle ARMA d'ordres $p = 2$ et $q = 1$, qui représente un ARMA(2,1), est donné par :

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Exemple 2

Un modèle ARMA d'ordres $p = 1$ et $q = 1$, qui réfère à un ARMA(1,1) est défini comme suit,

$$X_t = \phi_1 X_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Exemple d'un ARMA(1,1) sous langage R

Dans l'exemple qui suit nous avons utilisé les fonctions disponibles dans le package "stats", pour simuler un modèle ARMA(1,1) de paramètre autorégressif $\phi = 0.7$ et de paramètre de moyenne mobile $\theta = 0.3$ tel que, $n = 100$ désigne le nombre d'observation et $\sigma = 1$ représente l'écart-type du bruit blanc .

```

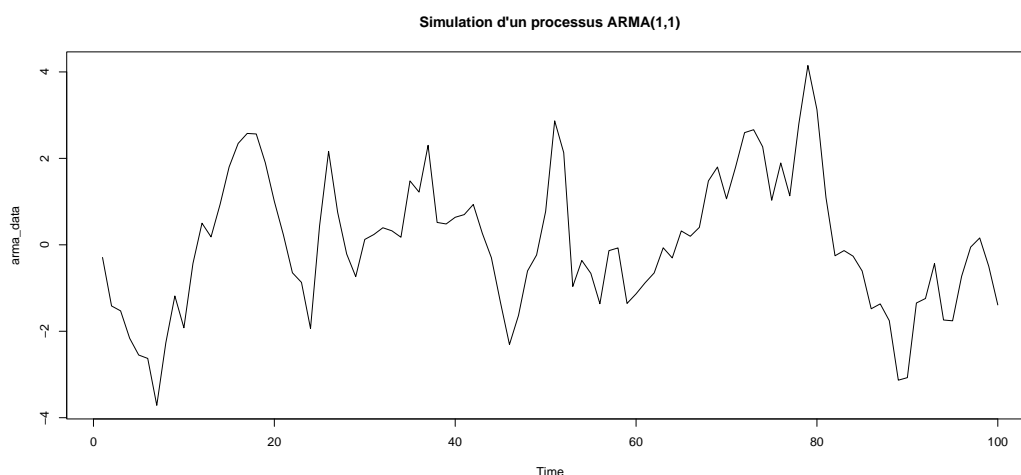
library(stats)

# Définir les paramètres du modèle ARMA(1,1)
phi <- 0.7 # Paramètre AR
theta <- 0.3 # Paramètre MA
sigma <- 1 # Écart-type du bruit blanc

# Générer des données ARMA(1,1)
set.seed(123) # pour la reproductibilité
n <- 100 # Nombre d'observations
arma_data <- arima.sim(n = n, list(ar = phi, ma = theta), sd = sigma)

# Afficher les données générées
plot(arma_data, type='l', main='Simulation d'un processus ARMA(1,1)')

```



• Interprétation

Ce graphique représente des variations naturelles et irrégulières d'une série temporelle, qui n'ont pas de direction précise.

1.2.3 Propriétés statistiques

1. La stationnarité

Un processus est dit stationnaire si ses propriétés statistiques (telles que, la moyenne et la covariance) sont stables dans le temps.

Un processus *ARMA* est dit stationnaire si sa composante $AR(p)$ est stationnaire. En effet, la partie autoregressive est stationnaire si les racines du polynôme caractéristique $\phi(z)$ associé à cette partie soient à l'extérieur du cercle unitaire (c'est-à-dire : $|z_i| > 1$). Cependant, la composante $MA(q)$ est toujours stationnaire. Il existe plusieurs types de stationnarité, en revanche, les plus utilisés en pratique sont la stationnarité forte et la stationnarité faible.

(a) La stationnarité forte (strictement stationnaire)

Une série temporelle $(X_t)_{t \in \mathbb{Z}}$ est strictement stationnaire si,

$\forall h \in \mathbb{Z}$ et $\forall t_1, t_2, \dots, t_k \in \mathbb{Z}$

$$(Y_{t_1+h}, Y_{t_2+h}, \dots, Y_{t_k+h}) \stackrel{\mathcal{D}}{=} (Y_{t_1}, Y_{t_2}, \dots, Y_{t_k})$$

où h est un décalage temporel.

(b) **La stationnarité faible (stationnarité au sens large)**

Une série temporelle $(X_t)_{t \in \mathbb{Z}}$ est dite faiblement stationnaire (stationnaire) si les conditions suivantes sont satisfaites,

- (i). $\forall t \in \mathbb{Z}, \mathbb{E}(X_t) = \mu$ (indépendante de t).
- (ii). $cov(X_t, X_{t+h}) = \gamma(h)$ (indépendante de t)

2. **Identification des ordres p et q**

L'identification des ordres p et q d'un processus *ARMA* est important en raison de son impact sur la modélisation du comportement des données (leurs évolutions) car un choix approprié permet de mieux représenter les relations temporelles dans les données.

De plus, des ordres plus élevés peuvent rendre le processus difficile à interpréter, tandis que les ordres réduits peuvent omettre des informations importantes sur le modèle.

3. **La causalité**

• **Théorème de Wold**

Ce théorème stipule que toute série temporelle stationnaire X_t peut être représentée comme une combinaison de sa moyenne μ et des chocs passés tel que,

$$X_t = \mu + \sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i}$$

, où, ε_t sont des erreurs (bruits blancs) iid supposés gaussiens, de moyenne nulle et de variance σ^2 ,

et les coefficients ψ_i satisfont la conditions suivante,

$$\sum_{i=0}^{\infty} |\psi_i| < \infty$$

- Un processus *ARMA* est dit causal si sa composante $AR(p)$ est causale.

Définition

Soit un $AR(p)$ tel que, $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t$

où,

X_t est la série temporelle observée.

ε_t est un bruit blanc, de moyenne nulle et de variance σ^2 .

ϕ_1, \dots, ϕ_p sont les coefficients du modèle $AR(p)$.

Un processus $AR(p)$ est causal si on peut écrire X_t en fonction d'une somme infinie des bruits blancs ε_t , c'est-à-dire, si $\sum_{i=0}^{\infty} |\psi_i| < \infty$,

et que, $X_t = \sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i}$

- Si la partie autorégressive est stationnaire alors elle est causale (le théorème de Wold le permet), d'où la causalité du modèle *ARMA*. Cependant, l'autre composante $MA(q)$ est toujours causale.

4. L'inversibilité

L'inversibilité des modèles *ARMA* est assurée lorsque la composante $MA(q)$ est inversible.

Définition

Soit un modèle $MA(q)$ défini comme suit,

$$X_t = \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \dots + \theta_q\varepsilon_{t-q}$$

où,

X_t est la série temporelle observée.

ε_t est un bruit blanc qui suit une loi $N(0, \sigma^2)$.

$\theta_1, \dots, \theta_q$ représentent les coefficients du modèle $MA(q)$.

• Un modèle $MA(q)$ est dit inversible si nous pouvons écrire les termes d'erreur ε_t en fonction d'une somme infinie des X_t telle que,

$$\varepsilon_t = \sum_{j=0}^{\infty} \varphi_j X_{t-j}, \quad \text{si,} \quad \sum_{j=0}^{\infty} |\varphi_j| < \infty$$

• La condition nécessaire et suffisante pour qu'un modèle $MA(q)$ soit inversible est que les racines du polynôme $\Theta(z)$ associé à ce processus se situent en dehors du cercle unitaire.

Exemple (Causalité et inversibilité)

Soit un $ARMA(1, 1)$ donné par la formule ci-dessous,

$$X_t = 0.5X_{t-1} + \varepsilon_t - 0.9\varepsilon_{t-1}$$

Vérifions la causalité et l'inversibilité de ce processus.

On a,

$$\begin{aligned} X_t &= 0.5X_{t-1} + \varepsilon_t - 0.9\varepsilon_{t-1} \\ \Leftrightarrow X_t - 0.5X_{t-1} &= \varepsilon_t - 0.9\varepsilon_{t-1} \\ \Leftrightarrow X_t - 0.5BX_t &= \varepsilon_t - 0.9B\varepsilon_t \\ \Leftrightarrow X_t(1 - 0.5B) &= \varepsilon_t(1 - 0.9B) \end{aligned}$$

Posons $B = z$ et notons $\phi(z) = 1 - 0.5z$, $\Theta(z) = 1 - 0.9z$.
 $\Rightarrow z_1 = \frac{1}{0.5} = 2$ et $z_2 = \frac{1}{0.9} \approx 1.11$.

Ce qui implique que $|z_1| > 1$ et $|z_2| > 1$.

Donc, le modèle considéré est un processus stationnaire, d'où il est causal et en même temps il est inversible.

1.2.4 Les fonctions d'autocovariance et d'autocorrélation

La fonction d'autocovariance

La fonction d'autocovariance est un outil essentiel pour la modélisation des séries chronologiques. Elle offre des informations significatives sur la structure des données. Cette fonction est définie comme suit,

$$\gamma(k) = \text{cov}(X_t, X_{t+k}) = \mathbb{E}((X_t - \mu)(X_{t+k} - \mu))$$

où,

▷ μ est la moyenne de la série temporelle.

▷ k est un décalage (retard).

Les propriétés de la fonction d'autocovariance

- Si la moyenne et la variance sont constantes, alors la covariance est constante.

Et on a,

$$\text{cov}(X_1, X_2) = \text{cov}(X_2, X_3) = \dots = \text{cov}(X_{n-1}, X_n)$$

$$\text{cov}(X_1, X_3) = \text{cov}(X_2, X_4)$$

$$\text{cov}(X_1, X_4) = \text{cov}(X_2, X_5)$$

(La condition est que le pas soit identique.)

- La fonction d'autocovariance est fréquemment utilisée dans le diagnostic des modèles de séries chronologiques, pour examiner si les résidus sont corrélés. Ceci est vérifié lorsque la fonction d'autocovariance des résidus montre des valeurs significativement différentes de zéro pour plusieurs retards.
- Elle donne des informations sur la variabilité des données, ce qui peut être pratique pour la prédiction du comportement futur de la série chronologique.
- Elle permet d'évaluer la dépendance linéaire entre les observations à divers décalages.

La fonction d'autocorrélation (ACF)

La fonction d'autocorrélation notée ρ_k mesure la corrélation linéaire entre une série de données X_t et une version d'elle-même décalée de k unités du temps X_{t+k} . Cette fonction est donnée par la formule suivante,

$$\begin{aligned} \rho_k &= \text{corr}(X_t, X_{t+k}) = \frac{\gamma_k}{\gamma_0} \\ &= \frac{\text{cov}(X_t, X_{t+k})}{\mathbb{V}(X_t)} \end{aligned}$$

La fonction d'autocorrélation partielle (PACF)

La fonction d'autocorrélation partielle évalue la corrélation directe entre une observation et une autre décalée, en prenant en compte les effets des autres points de données dans la série temporelle.

Propriétés

1. L'ACF et la PACF mesurent la corrélation entre les observations de manière standardisée, car elles sont toujours bornées entre -1 et 1 (c'est-à-dire, $-1 \leq \rho_k \leq 1$).
2. Une autocorrélation à un décalage de 0 est toujours égale à 1 (c'est-à-dire, $\rho_0 = 1$).
3. Pour une série temporelle stationnaire, la fonction d'autocorrélation est symétrique autour de zéro, telle que, $\rho_k = \rho_{-k}$.

4. En général, la fonction d'autocorrélation diminue à mesure que k augmente. Autrement dit, les observations sont moins corrélées à mesure qu'elles sont plus éloignées dans le temps.
5. Soient X et Y deux variables aléatoires, X et Y sont indépendantes $\Rightarrow X$ et Y ne sont pas corrélées. Donc, l'indépendance implique la non-corrélation mais le contraire n'est pas vérifié, sauf si les variables considérées suivent une loi gaussienne.
6. Pour déterminer l'ordre d'un autoregressif, on utilise le graphe de la PACF. Tandis que, pour déterminer l'ordre d'une moyenne mobile, on utilise le graphe de l'ACF.

1.2.5 Exemple d'application sous R

Ce programme génère une série chronologique de 100 observations, basée sur un modèle $ARMA(1,1)$ de paramètre autorégressif $\phi = 0.4$ et de paramètre de moyenne mobile $\theta = 0.3$ avec un écart-type du bruit blanc $\sigma = 1$, en utilisant la bibliothèque "stats" et la fonction "arima.sim". Puis examinons à travers les graphiques ses autocorrélations (ACF) et ses autocorrélations partielles (PACF).

```
library(stats) # la bibliothèque 'stats'
# Définir les paramètres du modèle ARMA(1,1)
phi <- 0.4 # Paramètre AR
theta <- 0.3 # Paramètre MA
sigma <- 1 # Écart-type du bruit blanc

# Générer des données ARMA(1,1)
set.seed(123) # pour la reproductibilité
n <- 100 # Nombre d'observations
arma_data <- arima.sim(n = n, list(ar = phi, ma = theta), sd = sigma)

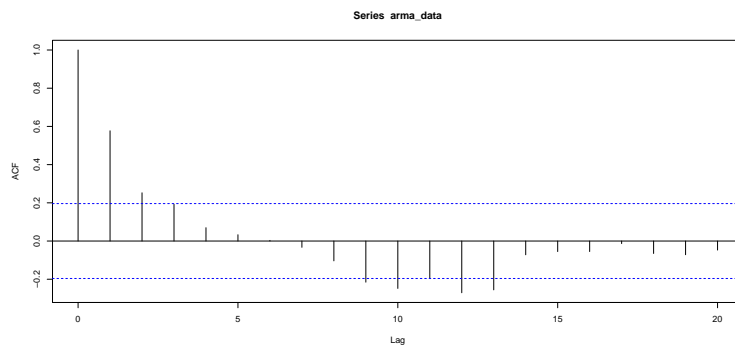
# Afficher les données générées
plot(arma_data, type='l', main='Simulation d\'un processus ARMA(1,1)')
install.packages("tseries")
library(tseries)

# Tracer l'ACF
acf(arma_data)

# Tracer la PACF
pacf(arma_data)
```

• Le graphique de l'ACF

Le graphique ci-dessous montre les autocorrélations de la série étudiée, ajustée selon un modèle $ARMA(1,1)$, permettant ainsi d'identifier l'influence des décalages successifs sur la variable étudiée, en observons le comportement de ses autocorrélations.

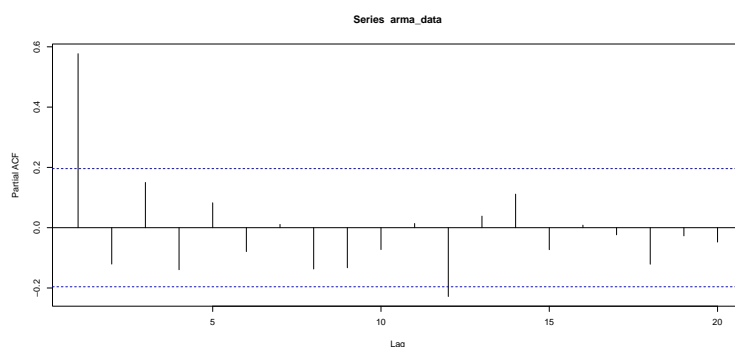
FIGURE 1.1 – Autocorrélations d'un $ARMA(1, 1)$

Interprétation

Nous constatons que l'autocorrélation pour le premier décalage dépasse de manière significative la bande de confiance, indiquant que l'ordre de moyenne mobile $q = 1$.

• Le graphique de la PACF

Ce graphique représente les autocorrélations partielles de la série modélisée par un modèle $ARMA(1, 1)$ détectons ainsi, l'ordre autorégressif en examinons jusqu'à quel décalage ces autocorrélations sont significatives.

FIGURE 1.2 – Autocorrélations partielles d'un $ARMA(1, 1)$

Interprétation

Nous observons une autocorrélation partielle significative au-dessus de la bande de confiance juste au premier décalage, ce qui montre que l'ordre autorégressif $p = 1$.

1.3 L'estimation dans les modèles ARMA

Dans le cadre de l'estimation des modèles ARMA, l'objectif est de déterminer les coefficients les plus adaptés pour une meilleure représentation des données observées. Pour cela, les méthodes d'estimation utilisent souvent des formules de vraisemblance qui permettent de maximiser la probabilité d'observer les données en fonction des paramètres du modèle. Deux approches sont possibles,

- Les méthodes exactes, qui offrent des estimations précises mais nécessitent des calculs complexes.
- Les méthodes approximatives, qui sont plus rapides et se basent sur des hypothèses

simplificatrices, comme les valeurs initiales pour les erreurs et les observations. Cela est particulièrement utile lorsque la quantité de données est limitée (lors d'un échantillon de petite taille).

Soit un modèle ARMA(p, q) défini comme suit,

$$Z_t = \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (1.1)$$

où a_t est un bruit blanc qui suit une loi $N(0, \sigma^2)$.

$\phi = (\phi_1, \dots, \phi_p)$ et $\theta = (\theta_1, \dots, \theta_q)$ sont les paramètres autorégressifs et de moyenne mobile, respectivement.

Z_t est la série temporelle considérée.

Pour estimer les paramètres ϕ , θ et σ^2 du modèle ARMA(p, q), nous allons considérer trois méthodes comme indiqué ci-dessous.

1. La méthode du maximum de vraisemblance conditionnelle

Cette méthode repose sur la maximisation de la fonction de vraisemblance conditionnelle pour les paramètres du modèle, en supposant une distribution normale pour les résidus a_t et $Z_t = \tilde{Z}_t - \mu$.

La densité de probabilité jointe pour $a = (a_1, a_2, \dots, a_n)'$ est définie par la formule suivante,

$$P(a|\phi, \mu, \theta, \sigma_a^2) = (2\pi\sigma_a^2)^{-n/2} \exp\left(-\frac{1}{2\sigma_a^2} \sum_{t=1}^n a_t^2\right) \quad (1.2)$$

En réécrivant l'équation (1.1), on obtient,

$$a_t = \theta_1 a_{t-1} + \dots + \theta_q a_{t-q} + Z_t - \phi_1 Z_{t-1} - \dots - \phi_p Z_{t-p}$$

Soit $Z = (Z_1, Z_2, \dots, Z_n)'$ en tenant compte des conditions initiales données comme suit :

$$\begin{cases} Z_* = (Z_{1-p}, \dots, Z_{-1}, Z_0)' \\ a_* = (a_{1-q}, \dots, a_{-1}, a_0)' \end{cases}$$

Ainsi, la fonction log-vraisemblance sera définie comme suit,

$$\log L_*(a|\phi, \mu, \theta, \sigma_a^2) = \frac{-n}{2} \log(2\pi\sigma_a^2) - \frac{S_*(\phi, \mu, \theta)}{2\sigma_a^2} \quad (1.3)$$

telle que $S_*(\phi, \mu, \theta) = \sum_{t=1}^n a_t^2(\phi, \mu, \theta|Z_*, a_*, Z)$.

Pour simplifier l'estimation de la vraisemblance conditionnelle du modèle ARMA, nous avons adopté l'hypothèse que les valeurs initiales de la série d'erreurs a_t sont nulles. Cette hypothèse permet de réduire l'influence des valeurs initiales inconnues, facilitant ainsi le calcul de la vraisemblance conditionnelle. Cette simplification est couramment utilisée dans les modèles de séries chronologiques pour rendre l'estimation plus stable (pour plus de détails, voir [11]).

2. La méthode conditionnelle proposée (PC)

Considérons un modèle ARMA(p, q) stationnaire tel que,

$$Z_t = \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

La méthode conditionnelle proposée comporte quatre étapes essentielles comme indiqué ci-dessous.

(a) **Les conditions initiales**

Pour spécifier les valeurs initiales des séries associées aux observations et aux erreurs, la méthode proposée utilise les moyennes harmoniques comme suit :

- La moyenne harmonique associée aux observations est définie par,

$$H_Z = \frac{n}{\sum_{t=1}^n \frac{1}{Z_t}} \quad (1.4)$$

- La moyenne harmonique associée à la série d'erreurs est définie comme suit,

$$H_a = \frac{n}{\sum_{t=1}^n \frac{1}{a_t}} \quad (1.5)$$

Ces moyennes harmoniques servent de valeurs initiales pour remplacer les valeurs inconnues des séries, tout en maintenant la stationnarité du modèle.

(b) **Le calcul des erreurs**

En fonction de la relation entre les ordres p et q du modèle ARMA, la méthode précise comment calculer les erreurs a_t pour garantir un bon ajustement.

- Quand $p \leq q$, les erreurs a_t sont calculées pour tout $t \geq (p + 1)$.
- Quand $q > p$, les erreurs sont calculées pour tout $t \geq (q + 1)$.

Dans ces deux cas, on utilise les moyennes harmoniques définies précédemment pour assurer que les conditions initiales sont cohérentes avec les hypothèses de stationnarité du modèle.

(c) **La fonction de somme des carrés conditionnelle**

L'objectif de la méthode est de minimiser la somme des carrés des erreurs, notée $S_*(\phi, \mu, \theta)$, pour obtenir les valeurs optimales des paramètres du modèle.

- Lorsque $p > q$, la fonction de somme des carrés conditionnelle est définie comme suit :

$$S_*(\phi, \mu, \theta) = \sum_{t=p+1}^n a_t^2(\phi, \mu, \theta | Z_H) \quad (1.6)$$

où Z_H est la nouvelle série d'observations après avoir substitué les valeurs inconnues de Z_* par la moyenne harmonique H_Z de la série d'observations. Dans ce cas, Z_H est utilisée pour garantir que les valeurs initiales des observations sont cohérentes. L'objectif de cette fonction est de fournir une mesure d'ajustement en minimisant les erreurs de la série temporelle, permettant ainsi d'estimer les paramètres de manière optimale.

(d) **La normalisation**

Après avoir ajouté les valeurs inconnues aux erreurs de la série associée à ces dernières, on effectue une normalisation qui consiste à :

- Soustraire la moyenne de la série des erreurs, afin d'éliminer les biais dans l'estimation.
- Diviser par l'écart-type de la série d'erreurs pour garantir la stationnarité et éviter les problèmes d'échelle.

Ce processus garantit une estimation robuste et stable des paramètres du modèle ARMA.

3. La méthode de Backcasting inconditionnelle

Cette méthode consiste à estimer les valeurs passées pour obtenir une estimation plus précise des paramètres du modèle considéré dans le futur. Pour effectuer le backcasting, les valeurs inconnues sont estimées comme suit,

$$\begin{cases} Z_* = (Z_{1-p}, \dots, Z_{-1}, Z_0)' \\ a_* = (a_{1-q}, \dots, a_{-1}, a_0)' \end{cases}$$

Les sommes des carrés nécessaires pour n'importe quel modèle ARMA peuvent être écrites sous la forme régressive suivante :

$$(1 - \phi_1 B - \dots - \phi_P B^P) Z_t = (1 - \theta_1 B - \dots - \theta_q B^q) a_t \quad (1.7)$$

ou sous la forme progressive donnée comme suit :

$$(1 - \phi_1 F - \dots - \phi_P F^P) Z_t = (1 - \theta_1 F - \dots - \theta_q F^q) a_t \quad (1.8)$$

où F est un opérateur de déplacement dans le futur, il sert à exprimer les valeurs futures de Z_t en fonction des valeurs futures des erreurs, tel que,

$$F^j Z_t = Z_{t+j}.$$

L'équation ci-dessous représente la fonction de log-vraisemblance introduite par Box et Jenkins. Maximiser cette fonction inconditionnelle permet d'estimer les paramètres du modèle ARMA,

$$\log L(\phi, \mu, \theta, \sigma_a^2) = \frac{-n}{2} \log(2\pi\sigma_a^2) - \frac{S(\phi, \mu, \theta)}{2\sigma_a^2} \quad (1.9)$$

où $S(\phi, \mu, \theta)$ est la fonction qui représente la somme inconditionnelle des carrés, et on a :

$$S(\phi, \mu, \theta) = \sum_{t=-\infty}^n (\mathbb{E}[a_t | \phi, \mu, \theta, Z])^2.$$

L'équation suivante est une approximation de l'équation ci-dessus, en limitant la somme à partir d'un grand nombre noté H . Cette approximation simplifie le calcul en conservant une bonne estimation de l'erreur totale. Le H est choisi assez grand pour que les valeurs au-delà de $-H$ aient un effet négligeable, tel que,

$$\mathbb{E}[Z_t | \phi, \mu, \theta, Z] \approx H.$$

Ainsi, pour tout $t \leq -(H+1)$, on a $\mathbb{E}[a_t | \phi, \mu, \theta, Z] \approx 0$.

1.3.1 La méthode de Box-Jenkins

Les deux statisticiens "George Box" et "Gwilym Jenkins" ont introduit une méthodologie connue sous le nom de Box-Jenkins (1976), qui consiste à modéliser les séries univariées au moyen des processus *ARMA* qui donnent souvent de bons résultats en prévision. Cette méthode peut se décomposer en quatre étapes,

- **Étape 1** : la première étape cherche à obtenir la stationnarité des données, en interprétant le graphique des autocovariances. En effet, le processus est considéré non stationnaire si les autocovariances décroissent très lentement.
- **Étape 2** : la seconde étape repose sur l'interprétation de graphiques d'autocorrélations (ACF) et d'autocorrélations partielles (PACF), pour choisir les ordres p et q .
- **Étape 3** : la troisième étape consiste à l'estimation des paramètres du modèle considéré.
- **Étape 4** : la dernière étape cherche à vérifier si le modèle est bien spécifié au moyen de tests.

1.3.2 Les tests d'hypothèses

Définition

Les tests d'hypothèses sont des outils statistiques indispensables qui permettent de tirer des conclusions objectives à partir des données disponibles. Un test d'hypothèse implique deux hypothèses :

- **L'hypothèse nulle (H_0)**

Cette hypothèse est formulée par l'idée que toute observation est due au hasard.

- **L'hypothèse alternative (H_1)**

Elle est considérée comme une hypothèse contraire à l'hypothèse nulle, et elle est généralement ce que nous voulons prouver.

Il existe divers tests d'hypothèses en statistiques ; cependant, nous ne mentionnerons que ceux qui sont généralement utilisés dans les modèles *ARMA*.

1. **Test de normalité des résidus**

Ce test cherche à vérifier si les résidus d'un modèle suivent une distribution gaussienne, comme le test de "Shapiro-Wilk" qui est couramment utilisé.

2. **Test de stationnarité des résidus**

Comme pour les modèles *ARMA*, il est important que les résidus soient stationnaires. Le test le plus utilisé à cet égard est celui de "Philips-Perron".

3. **Test de racine unitaire sur les résidus**

Ce test est utilisé pour vérifier la non-stationnarité des résidus (c'est-à-dire, la présence de racines unitaires dans les résidus).

4. **Test d'autocorrélation résiduelle**

Ce test est utilisé pour vérifier l'autocorrélation des résidus dans un modèle, par exemple, le test de "Ljung-Box" et le test de "Box-Pierce".

Un exemple d'application sous langage *R*

Cet exemple représente un test de normalité des résidus d'un modèle *ARMA*(1, 2) (test de Shapiro-Wilk) avec un seuil $\alpha = 0.05$.

```
> library(stats)
> # Exemple de données ARMA(1,2)
> set.seed(123)
> y <- arima.sim(list(order = c(1,0,2), ar = 0.7, ma = c(0.2, 0.5)), n = 100)
>
> # Ajuster le modèle ARMA(1,2)
> model <- arima(y, order = c(1,0,2))
>
> # Obtenir les résidus du modèle
> resid <- residuals(model)
>
> # Test de normalité des résidus
> shapiro.test(resid)
```

Shapiro-Wilk normality test

data: resid W = 0.99288, p-value = 0.8805

• Interprétation des résultats

On a,

$$\begin{cases} H_0 : & \text{les résidus suivent une distribution normale.} \\ H_1 : & \text{les résidus ne suivent pas une distribution normale.} \end{cases}$$

D'après les résultats du test de "Shapiro-Wilk", la p -value = 0.8805 > 0.05, ce qui signifie que les résidus de ce modèle $ARMA(1, 2)$ suivent une distribution normale. Alors, on accepte l'hypothèse H_0 .

1.4 Les domaines d'application des modèles $ARMA$

Les modèles $ARMA$, en raison de leur capacité à modéliser efficacement les séries chronologiques, trouvent une large gamme d'applications dans plusieurs domaines.

1. Le domaine médical

Dans ce secteur, les modèles $ARMA$ peuvent être utilisés pour l'étude et la prédiction des différents phénomènes tels que les variations saisonnières dans les données de santé publique et la propagation de maladies infectieuses.

2. Marketing

Les modèles $ARMA$ dans le domaine du marketing servent à planifier les campagnes publicitaires en fonction des tendances historiques des ventes, prévoir la demande de produits et à optimiser les stocks.

3. L'ingénierie

En ingénierie, les processus $ARMA$ sont conçus pour prédire le comportement des matériaux dans la structure et l'analyse de la consommation d'énergie.

4. Finance

En finance, ces modèles permettent de prévoir les rendements des actifs financiers et de modéliser la volatilité des prix, etc.

5. Météorologie

Au niveau de la météorologie, les processus $ARMA$ sont utilisés pour prédire les tendances à court terme, comme la température.

1.5 Conclusion

Les modèles *ARMA* représentent un pilier fondamental de l'analyse des séries temporelles, grâce à leurs capacités à modéliser à la fois les composantes autorégressives et moyennes mobiles pour capturer les tendances et les structures de dépendance dans les données. L'estimation des paramètres ainsi que les tests sur les résidus assurent la validité des prévisions et des conclusions tirées. L'approche *ARMA* est un concept essentiel qui permet de modéliser efficacement des processus stationnaires et offre une meilleure compréhension des données temporelles pour une prise de décision éclairée, d'où ces processus sont adaptés dans divers domaines, notamment en statistiques, en finance et en météorologie, etc.

Chapitre 2

Méthodes de Monte-Carlo

2.1 Introduction

Les méthodes de Monte-Carlo (MMC) sont des techniques d'échantillonnage. Elles ont été créées par 'Nicholas Metropolis' en 1947 et apparaissent pour la première fois en 1949 dans un article coécrit avec 'Stanislaw Ulam'.

Ces méthodes représentent un outil puissant pour la modélisation probabiliste et le traitement des problèmes complexes dans divers domaines. Nous distinguons plusieurs types de méthodes MMC, parmi elles la méthode de Monte-Carlo par chaînes de Markov (MCMC), qui est la plus utilisée, notamment en statistiques. En effet, elle est considérée comme une approche essentielle pour la simulation des variables aléatoires provenant de distributions complexes.

2.2 Les cas d'application des méthodes de Monte-Carlo

Dans cette section, nous allons aborder quelques applications des méthodes MMC.

1. Application au calcul d'une intégrale

Supposons que l'on cherche à calculer l'intégrale donnée par,

$$J = \int_0^1 g(x) dx$$

La méthodologie est la suivante,

- En premier lieu, l'intégrale J doit être représentée sous la forme d'une espérance. Soit (U_1, \dots, U_n) un échantillon de variables aléatoires (v.a) indépendantes, identiquement distribuées (iid), tel que U est une v.a. qui suit une loi $U[0,1]$.

Alors,

$$J = \mathbb{E}[g(U)]$$

- Ensuite, en utilisant la loi forte des grands nombres, on obtient la convergence presque sûre suivante,

$$\frac{1}{n} \sum_{j=1}^n g(u_j) \xrightarrow{\text{p.s.}} \mathbb{E}[g(U)]$$

Donc,

$$J \simeq \frac{1}{n} \sum_{j=1}^n g(u_j)$$

Ce qui veut dire que, $\frac{1}{n} \sum_{j=1}^n g(u_j)$ est une approximation de J .

Soit X une variable aléatoire de densité f . Si nous souhaitons calculer une intégrale de la forme $J = \int_{\mathbb{R}^n} g(x) f(x) dx$.

Alors, $J = \mathbb{E}[g(X)]$ et $\frac{1}{n} \sum_{i=1}^n g(x_i)$ est une approximation de J , où X_i sont des v.a. de densité f .

2. Approximation d'une intégrale par Monte-Carlo sur un support borné (un compact $[a, b]$)

Soit g une fonction telle que $g : [a, b] \mapsto \mathbb{R}$. L'objectif est de calculer l'intégrale suivante, $J = \int_a^b g(x) dx$.

On a $J = \int_a^b g(x) dx$, alors,

$$J = \int_a^b \frac{b-a}{b-a} g(x) dx = (b-a) \int_a^b \frac{1}{b-a} g(x) dx = (b-a) \int_a^b f(x) g(x) dx$$

donc, $J = (b-a) \mathbb{E}[g(X)]$

où f est la densité de la loi uniforme sur $[a, b]$ (i.e. $X \sim U[a, b]$)

Et on a, $\mathbb{E}[g(X)] \simeq \frac{1}{n} \sum_{i=1}^n g(x_i)$ (par les moindres carrés)

Donc, $J \simeq (b-a) \left(\frac{1}{n} \sum_{i=1}^n g(x_i) \right)$, où $X_i \sim U[a, b]$.

3. Approximation d'une intégrale sur un support non borné

Soit (X_1, \dots, X_n) un échantillon de v.a de densité f et considérons l'intégrale J telle que,

$J = \int_{\mathbb{R}} g(x) dx$ alors,

$$J = \int_{\mathbb{R}} g(x) \frac{f(x)}{f(x)} dx = \int_{\mathbb{R}} \frac{g(x)}{f(x)} f(x) dx$$

$$\Rightarrow J = \mathbb{E} \left(\frac{g(X)}{f(X)} \right)$$

où, X est une variable aléatoire qui suit une loi de densité f .

Et on a, $J \simeq \frac{1}{n} \sum_{i=1}^n \frac{g(x_i)}{f(x_i)}$

2.3 Simulation des variables aléatoires

Toute méthode MMC est basée sur la simulation d'une suite de variables aléatoires iid suivant une loi de probabilité quelconque.

2.3.1 La simulation de la loi uniforme

Soit $(X_n)_{n \geq 1}$ une suite de variables aléatoires iid suivant une loi uniforme sur l'intervalle $[0,1]$. Pour cela, nous considérons la méthode d'inversion.

- **La méthodologie d'inversion**

Soit X une variable aléatoire ayant G comme fonction de répartition telle que, $G(x) = P(X \leq x)$ et soit V une variable aléatoire uniformément distribuée sur $[0, 1]$, c'est-à-dire $V \sim \text{Uniform}(0, 1)$.

La méthode d'inversion consiste à obtenir des réalisations de X en utilisant V et l'inverse généralisée de G comme suit,

1. **Définition de l'inverse généralisée**

Soit X une variable aléatoire de fonction de répartition G . L'inverse généralisée de G (ou la fonction quantile de X) est définie par la formule suivante,

$$G^{-1}(v) = \inf\{x \in \mathbb{R} | G(x) \geq v\}, \quad \forall v \in [0, 1]$$

2. **Transformation de V**

Une réalisation de la variable aléatoire X est obtenue en posant,

$$X = G^{-1}(V)$$

En effet,

$$P(X \leq x) = P(G^{-1}(V) \leq x) = P(V \leq G(x)) = G(x), \quad \forall x \in \mathbb{R}$$

Cette définition indique que si V est une variable uniformément distribuée sur $[0, 1]$, alors $X = G^{-1}(V)$ suit la distribution définie par la fonction de répartition G .

- **Exemple**

Voici un simple exemple qui montre comment la méthode d'inversion fonctionne pour générer des échantillons d'une loi exponentielle.

Soit f une densité d'une loi exponentielle telle que, $f(x) = \lambda \exp(-\lambda x)$, $\forall x \geq 0$

- Calculons sa fonction de répartition F

$$F(x) = \int_{-\infty}^x f(t) dt = 1 - \exp(-\lambda x), \quad \forall x \geq 0$$

Posons, $u = 1 - \exp(-\lambda x)$

$$\Rightarrow \exp(-\lambda x) = 1 - u \Rightarrow -\lambda x = \log(1 - u)$$

$$\Rightarrow x = \frac{-\log(1 - u)}{\lambda}$$

Donc,

$$F^{-1}(u) = \frac{-\log(1 - u)}{\lambda}, \text{ tel que } U \sim U[0, 1]$$

- **Exemple d'application sous langage R**

Le programme qui suit permet de comprendre comment la méthode d'inversion fonctionne pour générer des échantillons de taille $n = 1000$ d'une loi exponentielle.

```
> # Définir le paramètre lambda
> lambda <- 1

> # Nombre d'échantillons à générer
> n <- 1000
>
> # Générer des valeurs uniformes aléatoires entre 0 et 1
> u <- runif(n)
>
> # Appliquer la méthode d'inversion
> samples <- -log(1 - u) / lambda
>
> # Visualiser les premiers échantillons générés
> head(samples)
[1] 0.7914795 1.0856496 0.1883781 1.0025443 0.3737771 1.2893649
>
> # Histogramme des échantillons générés
> hist(samples, breaks = 30, probability = TRUE, main = "Histogramme
  des échantillons générés", xlab = "Valeurs", ylab = "Fréquence",
  col.axis = "black", col.lab = "black")
> # Ajouter la courbe de densité exponentielle
> curve(dexp(x, rate = lambda), col = "red", lwd = 2, add = TRUE)
>
```

- Voici un histogramme qui permet de visualiser la distribution des échantillons générés d'une loi exponentielle .

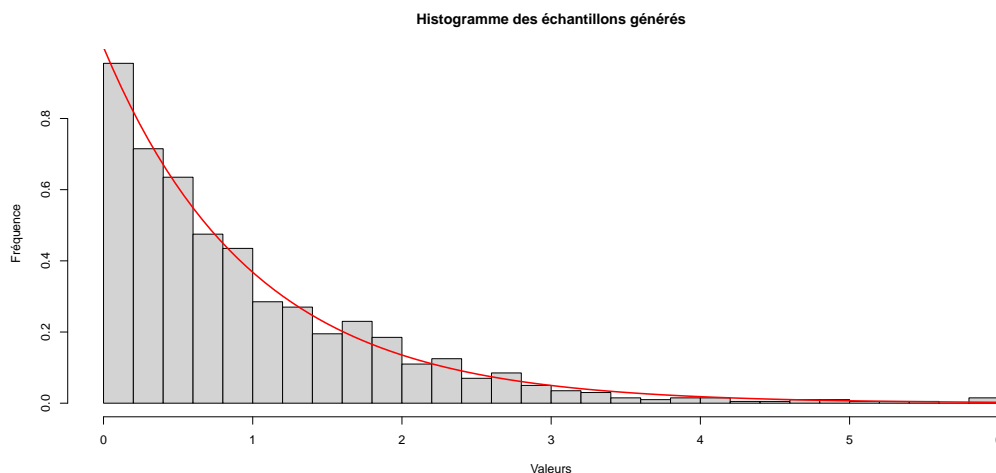


FIGURE 2.1 – Un histogramme des échantillons générés à partir d'une loi exponentielle

- **Interpretation graphique**

Nous remarquons que cette distribution a le comportement d'une loi exponentielle ; tel que, pour que la transformation fonctionne correctement, nous devons voir une

forme décroissante, typique de la loi exponentielle, et c'est ce que montre le graphique ci-dessus.

2.3.2 La simulation de la loi normale

Nous considérons l'algorithme de Box-Müller, afin de simuler une variable aléatoire qui suit une loi normale centrée, réduite $N(0, 1)$. Pour cela, il suffit d'avoir deux variables V et W iid telles que, $V, W \sim U[0, 1]$.

Posons

$$X = \sqrt{-2 \log V} \cos(2\pi W) \quad \text{ou} \quad X = \sqrt{-2 \log V} \sin(2\pi W)$$

En effet, soient V et W deux v.a de loi $U[0, 1]$ et considérons X, Y deux variables aléatoires iid de loi $N(0, 1)$. Soit $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ une fonction continue et bornée. Pour identifier la loi d'une v.a, il est suffisant de calculer cette espérance,

$$\begin{aligned} \mathbb{E}[g(X, Y)] &= \int_{\mathbb{R}} \int_{\mathbb{R}} g(x, y) \frac{\exp(-\frac{1}{2}(x^2 + y^2))}{2\pi} dx dy \\ \Rightarrow \mathbb{E}[g(X, Y)] &= \int_{\mathbb{R}^2} g(x, y) \frac{\exp(-\frac{1}{2}(x^2 + y^2))}{2\pi} dx dy \end{aligned}$$

Posons le changement de variables suivant,

$$\begin{cases} x = r \cos \theta \\ y = r \sin \theta, \quad \theta \in [0, 2\pi] \end{cases}$$

Puis, calculons la Jacobienne telle que,

$$\begin{aligned} J(x, y) &= \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{vmatrix} \\ \Rightarrow J(x, y) &= r \cos^2 \theta + r \sin^2 \theta = r(\cos^2 \theta + \sin^2 \theta) = r \\ \mathbb{E}[g(X, Y)] &= \int_{(r, \theta)} g(r \cos \theta, r \sin \theta) \frac{\exp(-\frac{r^2}{2})}{2\pi} |r| dr d\theta \\ &= \int_{\mathbb{R}_+ \times [0, 2\pi]} g(r \cos \theta, r \sin \theta) \frac{\exp(-\frac{r^2}{2})}{2\pi} |r| dr d\theta \end{aligned}$$

Posons un autre changement de variables tel que,

$$\begin{cases} r = \sqrt{2 \log v} \\ \theta = 2\pi w, \quad (v, w) \in ([0, 1])^2 \end{cases}$$

$$\begin{aligned} J(r, \theta) &= \begin{vmatrix} \frac{\partial r}{\partial v} & \frac{\partial r}{\partial w} \\ \frac{\partial \theta}{\partial v} & \frac{\partial \theta}{\partial w} \end{vmatrix} = \begin{vmatrix} \frac{-2}{2\sqrt{-2 \log v}} & 0 \\ 0 & 2\pi \end{vmatrix} \\ \Rightarrow J(r, \theta) &= \frac{-2\pi}{v\sqrt{-2 \log v}} \end{aligned}$$

Alors,

$$\mathbb{E}[g(X, Y)] = \int_0^1 \int_0^1 g(\sqrt{-2 \log v} \cos(2\pi w), \sqrt{-2 \log v} \sin(2\pi w))$$

$$\begin{aligned}
& \times \frac{\exp(\log v)}{2\pi} \frac{2\pi}{v\sqrt{-2\log v}} \sqrt{-2\log v} dv dw \\
& = \int_0^1 \int_0^1 g(\sqrt{-2\log v} \cos(2\pi w), \sqrt{-2\log v} \sin(2\pi w)) dv dw \\
& = \mathbb{E} \left(g(\sqrt{-2\log V} \cos(2\pi W), \sqrt{-2\log V} \sin(2\pi W)) \right)
\end{aligned}$$

où, $\sqrt{-2\log V} \cos(2\pi W)$ et $\sqrt{-2\log V} \sin(2\pi W)$ suivent une loi $N(0, 1)$
d'où, (X, Y) et $(\sqrt{-2\log V} \cos(2\pi W), \sqrt{-2\log V} \sin(2\pi W))$ ont la même loi.

• L'algorithme de Box-Müller

Posons $n = 200$ et prenons $X = \sqrt{-2\log V} \sin(2\pi W)$, on aura l'algorithme suivant,
 $n = 200$

$V = \text{runif}(n)$

$W = \text{runif}(n)$

$X = \sqrt{(-2 * \log(V)) * (\sin(2 * \pi * W))}$

$X \sim N(0, 1)$

• Exemple d'application sous langage R

L'exemple qui suit représente la simulation de deux échantillons $X1$ et $X2$ qui suivent une distribution normale $N(0, 1)$ à partir de deux autres échantillons V et W qui suivent une loi uniforme $U[0, 1]$.

```

> # Nombre d'échantillons à générer
> n <- 1000
> V <- runif(n)
> W <- runif(n)
> # Application de la méthode de Box-Muller
> X1 <- sqrt(-2 * log(V)) * cos(2 * pi * W)
> X2 <- sqrt(-2 * log(V)) * sin(2 * pi * W)
> # X1 et X2 sont deux échantillons indépendants suivant une distribution
normale standard
> # On peut combiner les deux échantillons pour obtenir 1000 échantillons gaussiens
> samples <- c(X1, X2)
> # Affichage des premiers échantillons générés
> head(samples)
[1] -1.4667442  0.1339494 -1.5782387 -0.5286462  0.7043456
1.1776513
> # Optionnel : visualisation de l'histogramme des échantillons
> hist(samples, breaks = 50, probability = TRUE, main = "Histogramme
des échantillons gaussiens ", xlab = "Valeurs", ylab = "Densité",
col = "gray", col.axis = "black", col.la = "black")
> # Ajouter la courbe de densité gaussienne
> curve(dnorm(x, mean = 0, sd = 1), col = "red", lwd = 2, add = TRUE)

```

• L'histogramme

Cet histogramme représente les échantillons générés par la méthode de Box-Müller.

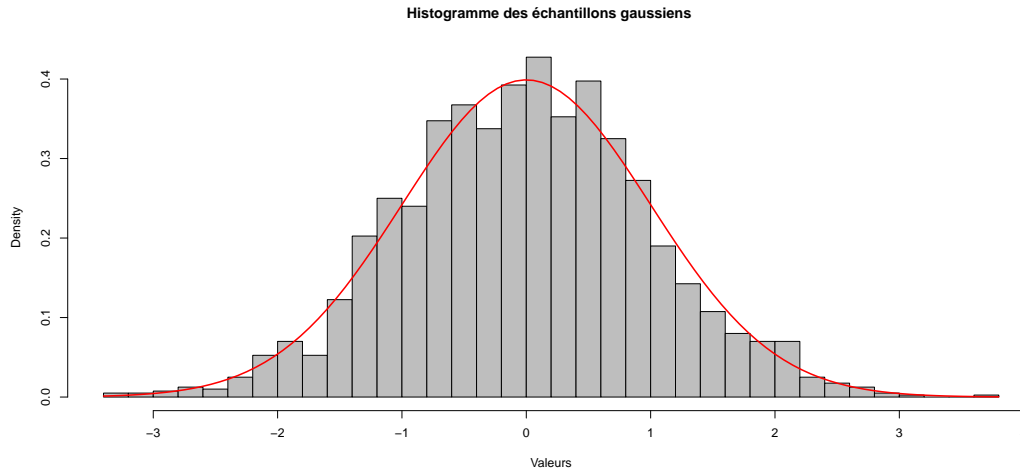


FIGURE 2.2 – L’histogramme des échantillons générés par la méthode de Box-Müller

• **Interpretation** Nous constatons que cette distribution a le comportement d’une loi gaussienne.

2.3.3 La méthode de Rejet

Cette méthode représente un outil basique pour la simulation des variables aléatoires, où la méthode d’inversion ne fonctionne pas. Elle consiste à l’approximation d’une densité de probabilités cible f par une autre densité h qui est considérée comme une distribution de proposition que l’on sait simuler simplement et posons une densité g telle que,

$g(x) = \frac{f(x)}{h(x)} \leq c$ où, $c = \max g(x)$ est une constante positive, telle que,

$$\forall x \in \mathbb{R}^d, \quad f(x) \leq h(x).c$$

Considérons les variables aléatoires suivantes,

X qui suit une loi de densité h , U suivant une loi uniforme sur l’intervalle $[0,1]$ et Y qui suit une loi uniforme sur l’intervalle $[0, c.h(x)]$, où x est l’observation spécifique de X .

• **L’algorithme de Rejet**

1. Par l’utilisation de la méthode d’inversion, on simule la variable aléatoire X selon la densité h .
2. On tire Y uniformément sur $[0, c.h(x)]$ en utilisant la formule

$$y = c.h(x).u \quad \text{telle que } U \sim U[0, 1].$$

3. Si la condition $y \leq f(x)$ est satisfaite, alors on accepte x , sinon on recommence les étapes précédentes.

Exemple

Soit X une variable aléatoire et f une densité de probabilités définie par,

$$f(x) = 2x(1 - x), \quad x \in [0, 1]$$

et de fonction de répartition F telle que,

$$\begin{aligned} F(x) &= \int_0^x f(t)dt = \int_0^x 2t - 2t^2 dt \\ &= 2 \left[\frac{1}{2}t^2 \right]_0^x - 2 \left[\frac{1}{3}t^3 \right]_0^x \\ F(x) &= x^2 - \frac{2}{3}x^3. \end{aligned}$$

Dans ce cas, la méthode d'inversion n'est pas applicable car le calcul de F^{-1} n'est pas possible, ce qui implique l'utilisation de la méthode de rejet.

On a, $f(x) = 2x - 2x^2$

en choisissant $h(x)$ comme densité d'une loi uniforme sur l'intervalle $[0, 1]$, ceci nous donne, $h(x) = 1$.

Soit,

$$\begin{aligned} g(x) &= \frac{f(x)}{h(x)} = \frac{2x - 2x^2}{1} \\ g(x) &= 2x - 2x^2 \end{aligned}$$

Calculons la première dérivée de g

$$g'(x) = 2 - 4x = 0 \implies x = \frac{2}{4}$$

Alors, $x = \frac{2}{4}$ est un extremum.

Calculons la seconde dérivée de g

$$g''(x) = -4 < 0 \implies g''\left(\frac{2}{4}\right) < 0$$

Donc, $x = \frac{2}{4}$ est un maximum.

D'où,

$$\begin{aligned} c &= g\left(\frac{2}{4}\right) = 2\left(\frac{2}{4}\right) - 2\left(\frac{2}{4}\right)^2 \\ &\implies c = \frac{1}{2} \end{aligned}$$

Par la suite, nous vérifions la condition $f(x) \leq c \cdot h(x) \iff f(x) \leq \frac{1}{2}$

• L'algorithme

1. X est une variable aléatoire qui suit la loi de densité h , ce qui veut dire ($X \sim U[0, 1]$).
2. La simulation de Y telle que, $y = c \cdot h(x) \cdot u$ cela implique, $y = \frac{1}{2}u$, $U \sim U[0, 1]$.
3. L'acceptation de x si $y \leq f(x) \implies y \leq 2x - 2x^2$, sinon toutes ces étapes seront répétées jusqu'à ce que la condition précédente soit vérifiée.

- **Exemple sous langage R**

Dans cet exemple, nous utilisons la méthode de rejet pour générer des échantillons à partir d'une densité f de distribution $\beta(2, 2)$ telle que $f(x) = 2x(1 - x)$, $x \in [0, 1]$ en utilisant une distribution uniforme sur l'intervalle $[0, 1]$ comme distribution de proposition.

```
> set.seed(123) # Pour la reproductibilité
>
> # Définir la fonction cible f(x)
> f <- fonction(x) {
  2 * x * (1 - x) }
>
> # Définir la taille de l'échantillon souhaité
> n <- 1000
>
> # Initialiser le vecteur pour les échantillons acceptés
> samples <- numeric(n)
> count <- 0
>
> # Générer un échantillon de la distribution de proposition
(uniforme entre 0 et 1)
> while (count < n) {
  x <- runif(1, min = 0, max = 1)
> # Générer un échantillon uniforme entre 0 et 1 pour la décision
d'acceptation
> y <- runif(1)
# Vérifier si la condition d'acceptation est satisfaite
if (y<=f(x)) { count <- count + 1
samples[count] <- x }
>}
> # Affichage des premiers échantillon générés
> head(samples)
[1] 0.9404673 0.5514350 0.2460877 0.2891597 0.7584595 0.3181810
>
> # Optionnel : visualisation de l'histogramme des échantillons
> hist(samples, breaks = 50, main = "Histogramme des échantillons générés
par la méthode de Rejet", xlab = "Valeurs", col = "lightblue",
probability = TRUE)
```

- **L'histogramme des échantillons générés**

L'histogramme suivant et la courbe de la fonction cible permettent de visualiser les résultats et de vérifier la qualité de l'échantillonnage.

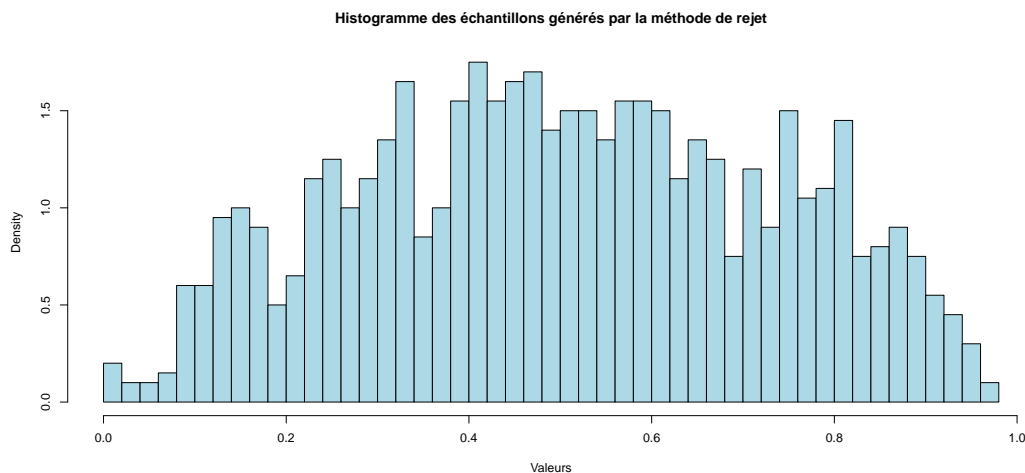


FIGURE 2.3 – Histogramme des échantillons générés par la méthode de Rejet

• Interprétation

Nous observons une forme parabolique qui représente le comportement d'une loi bêta de paramètres $a = 2$ et $b = 2$ indiquant ainsi, la réussite de cet échantillonnage en utilisant la méthode de rejet.

2.4 Les méthodes de Monte-Carlo par chaînes de Markov

2.4.1 Définition (Chaînes de Markov)

Une chaîne de Markov sur un espace d'états E est un processus $(X_t)_{t \geq 0}$ utilisé en statistiques pour décrire un système qui évolue dans le temps. Cette suite de variables aléatoires possède la propriété de mémoire limitée telle que, la probabilité de chaque état futur dépend uniquement de l'état présent et non des états précédents.

2.4.2 Les méthodes de Monte-Carlo par chaînes de Markov

Les méthodes de Monte-Carlo par chaînes de Markov (MCMC) représentent une sous-catégorie des méthodes MMC, qui utilise des chaînes de Markov comme mécanisme pour échantillonner des distributions complexes.

2.4.3 Quelques méthodes MCMC

Voici un aperçu des principales méthodes MCMC employées en statistiques.

- **Algorithme de Metropolis-Hastings** Soit f une densité cible et q une densité conditionnelle. L'algorithme de Metropolis-Hastings associé à ces deux densités fournit une chaîne de Markov $(X^{(t)})$ selon le noyau de transition comme suit,

L'algorithme

Considérons x_t ,

1. Générer $Y_t \sim q(y|x^{(t)})$

2. Choisir

$$X^{(t+1)} = \begin{cases} Y_t & \text{avec une probabilité } \mathbb{P}(x^{(t)}, Y_t), \\ x^{(t)} & \text{avec une probabilité } 1 - \mathbb{P}(x^{(t)}, Y_t). \end{cases}$$

tel que, $\mathbb{P}(x, y) = \min \left\{ \frac{f(y) q(x|y)}{f(x) q(y|x)}, 1 \right\}$.

• Implementation sous langage R

On dispose d'un générateur pour $q(y|x)$, qu'on note ici *geneq*(x).

Si $X[t]$ dénote la valeur de $X^{(t)}$ telle que,

```
> y = geneq(X[t])
> if (runif(1) < f(y) * q(y, X[t])/f(X[t]) * q(X[t], y)) {
> X[t+1] = y
> }else{
> X[t+1] = X[t]
> }
```

où, q est la loi de proposition et $\mathbb{P}(x, y)$ est la probabilité d'acceptation de Metropolis-Hastings.

• Exemple d'application sous langage R

L'exemple qui suit, montre comment implémenter la méthode de Metropolis en langage R pour simuler une distribution normale standard (de moyenne nulle et d'un écart-type qui égal à 1) par l'usage d'une distribution de proposition.

```
> # Fonction cible : Distribution normale standard (moyenne = 0, écart-type = 1)
> target_density <- fonction(x) {
  dnorm(x, mean = 0, sd = 1)
}
>
> # Fonction de la méthode de Metropolis
> metropolis <- fonction(n_iter, proposal_sd) {
> samples <- numeric(n_iter) # Vecteur pour stocker les échantillons
> current_sample <- 0 # Valeur initiale
>
> for (i in 1:n_iter) {
> # Proposition d'un nouveau point
> proposed_sample <- rnorm(1, mean = current_sample, sd = proposal_sd)
>
> # Calcul des densités cibles
> current_density <- target_density(current_sample)
> proposed_density <- target_density(proposed_sample)
>
> # Calcul du rapport d'acceptation
> acceptance_ratio <- proposed_density / current_density
>
> # Acceptation ou rejet de l'échantillon proposé
> if (runif(1) < acceptance_ratio) {
> current_sample <- proposed_sample
```

```

> }

> # Stockage de l'échantillon
> samples[i] <- current_sample
> }
> return(samples)
> }

> # Exécution de l'algorithme
> n_iter <- 1000          # Nombre d'itérations
> proposal_sd <- 1       # écart-type de la distribution de proposition
> samples <- metropolis(n_iter, proposal_sd)
> # Visualisation des résultats
> hist(samples, breaks = 30, main = "Histogramme des échantillons",
> xlab = "Valeurs", col = "lightblue")

```

• Représentation graphique

Ce graphique affiche les résultats de l'implémentation de l'algorithme de Metropolis pour la simulation d'une distribution normale standard, en utilisant une distribution de proposition.

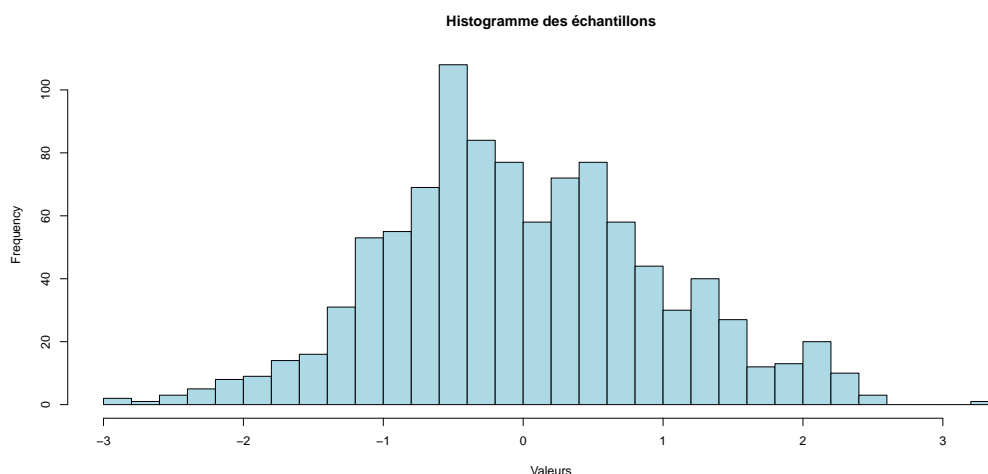


FIGURE 2.4 – Histogramme des échantillons générés par la méthode de Metropolis

• Interprétation

L'histogramme obtenu montre une bonne convergence de l'algorithme de Metropolis vers la distribution cible. La forme générale en cloche, centrée autour de 0, est conforme à la distribution normale standard. Les échantillons sont principalement situés entre -3 et 3, ce qui respecte la variabilité attendue et la fréquence des échantillons décroît à mesure qu'on s'éloigne de la moyenne, reflétant la symétrie et les caractéristiques typiques d'une distribution normale. Ces observations confirment que l'algorithme de Metropolis explore efficacement la distribution cible.

• L'échantillonnage de Gibbs

L'échantillonnage de Gibbs à plusieurs étapes est une méthode utilisée pour échantillonner une distribution conjointe complexe de plusieurs variables aléatoires, en

les mettant à jour une par une. À chaque étape, on échantillonne une variable en utilisant sa distribution conditionnelle par rapport aux valeurs actuelles des autres. En répétant ce processus, la méthode converge vers la distribution conjointe cible. Voici les principales étapes de cette méthode,

1. Mise à jour séquentielle

Considérons k variables aléatoires, notées X_1, X_2, \dots, X_k , à chaque itération, on met à jour chaque variable X_i en fonction de sa distribution conditionnelle, en utilisant les valeurs actuelles des autres variables telles que,

- Mettre à jour X_1 selon la distribution conditionnelle $P(X_1|X_2 = x_2^{(t)}, X_3 = x_3^{(t)}, \dots, X_k = x_k^{(t)})$.
- Puis mettre à jour X_2 selon $P(X_2|X_1 = x_1^{(t+1)}, X_3 = x_3^{(t)}, \dots, X_k = x_k^{(t)})$, et ainsi de suite jusqu'à X_k .

2. Convergence

Après plusieurs itérations, la chaîne atteint la distribution cible si les conditions de convergence sont satisfaites.

Remarque

L'avantage de cette méthode est de simplifier l'échantillonnage de distributions complexes, en traitant chaque variable séparément. Cependant, la convergence peut être lente en cas de forte dépendance entre les variables.

• L'échantillonnage de Gibbs à deux étapes

À partir d'une loi jointe donnée, l'échantillonnage de Gibbs à deux étapes crée une chaîne de Markov, si nous avons deux variables aléatoires X et Y qui ont une densité jointe $f(x, y)$ telle que, $f(Y|X)$ et $f(X|Y)$ représentent les densités conditionnelles correspondantes, alors l'échantillonneur de Gibbs génère une chaîne de Markov notée (X_t, Y_t) selon l'algorithme suivant,

L'algorithme d'échantillonnage de Gibbs à deux étapes

Considérons deux variables aléatoires X et Y de densités conditionnelles $f(X|Y)$ et $f(Y|X)$ respectivement, dans laquelle à chaque itération, on met à jour chaque variable en fonction de sa distribution conditionnelle, en utilisant les valeurs actuelles de l'autre variable telles que,

• Initialisation

Fixez une valeur initiale pour X_0 , notée x_0

• Étapes itératives

Pour chaque itération $t = 1, 2, \dots$

Cet algorithme itératif permet de générer une suite de paires (X_t, Y_t) qui convergent en distribution vers la distribution conjointe souhaitée de X et Y , sous certaines conditions de convergence.

Exemple d'application sous langage R

Considérons X et Y deux variables aléatoires de densité jointe $f(x, y)$, l'algorithme de Gibbs consiste à échantillonner alternativement X et Y à partir de leurs distributions conditionnelles données l'autre variable.

Supposons que nous avons une densité jointe donnée par,

$$f(x, y) = \exp(-x^2 - y^2),$$

ce qui représente une distribution gaussienne bivariée centrée à l'origine. Dans ce cas, les distributions conditionnelles sont également normales. L'algorithme de Gibbs associé à ces variables est le suivant,

```
n_iter <- 10000 # Nombre d'itérations
X <- numeric(n_iter) # Vecteur pour stocker les échantillons de X
Y <- numeric(n_iter) # Vecteur pour stocker les échantillons de Y

# Initialisation
X[1] <- 0 # Valeur initiale pour X
Y[1] <- 0 # Valeur initiale pour Y

# Algorithme de Gibbs
for (i in 2:n_iter) {
# Mise à jour de X | Y = y ~ N(0,1) (distribution conditionnelle de X donnée Y)
  X[i] <- rnorm(1, mean = 0, sd = 1)

# Mise à jour de Y | X = x ~ N(0,1) (distribution conditionnelle de Y donnée X)
  Y[i] <- rnorm(1, mean = 0, sd = 1)
}

# Tracer les résultats
par(mfrow = c(1, 2)) # Pour afficher deux graphiques côte à côte
plot(X, type = 'l', col = 'blue', main = "Échantillons de X",
xlab = "Itération", ylab = "X")
plot(Y, type = 'l', col = 'red', main = "Échantillons de Y",
xlab = "Itération", ylab = "Y")
```

• Représentation graphique

Les graphiques ci-dessous montrent les résultats de l'application de l'algorithme de Gibbs dans le cas de deux variables aléatoires X et Y suivant une densité jointe

$$f(x, y) = \exp(-x^2 - y^2)$$

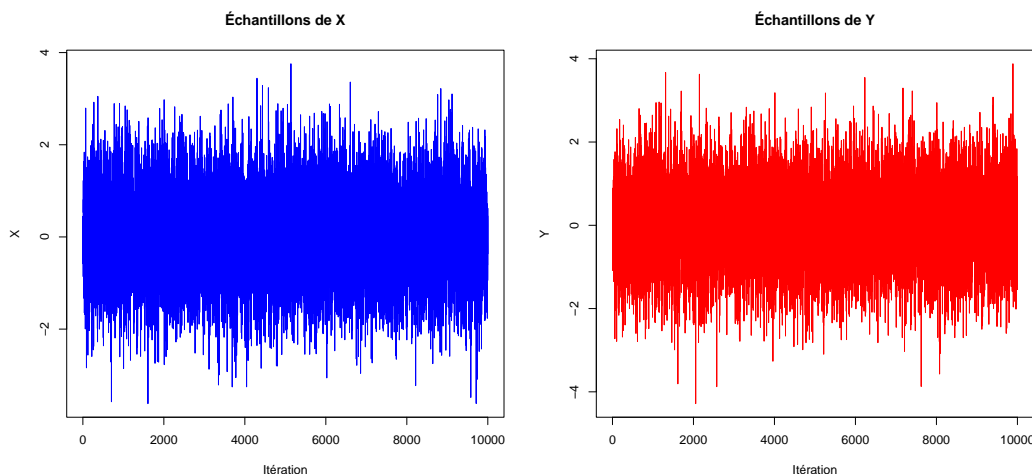


FIGURE 2.5 – Histogramme des échantillons générés par la méthode de Gibbs.

• Interprétation graphique

Les graphiques montrent les valeurs de X (en bleu) et celles de Y (en rouge) au fil des itérations où, nous observons que les valeurs oscillent autour de zéro, ce qui indique que l'algorithme converge correctement vers la distribution cible. En effet, ces graphiques confirment que l'algorithme de Gibbs a réussi à échantillonner efficacement la distribution cible. Cela démontre la capacité de cet algorithme à capturer la structure de dépendance de la distribution jointe en utilisant des distributions conditionnelles.

2.5 Conclusion

Les méthodes de Monte-Carlo sont des techniques indispensables en statistique où, les méthodes MCMC s'avèrent particulièrement, efficaces pour l'estimation des modèles complexes, qui n'ont pas de solutions analytiques simples. Ces approches offrent des estimations précises des paramètres, renforçant ainsi, leur utilité pour des modèles exigeants tels que, les modèles bêta *ARMA*.

Chapitre 3

Les modèles bêta *ARMA*

3.1 Introduction

La théorie de la régression des séries chronologiques s'est développée rapidement au fil de ces dernières années. Benjamin et al (2003) ont introduit l'approche *GARMA* (Generalized Autoregressive Moving Average)¹ qui sert à combiner la flexibilité du modèle linéaire généralisé (GLM) avec l'approche classique de la modélisation *ARMA*. Par la suite, la structure *GARMA* a évolué sous diverses formes, en supposant des distributions en dehors de la famille exponentielle en 2004.

Ferrari et Cribari-Neto (2004) ont proposé un modèle de régression dont la variable dépendante est de distribution bêta, ils comptaient sur le paramétrage suivant, Soit Y une variable aléatoire qui suit une distribution bêta, sa densité de probabilités est donnée par,

$$f(y; p, q) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} y^{p-1} (1-y)^{q-1}, \text{ si } : y \in]0, 1[$$

Posons,

$$\mathbb{E}(Y) = \frac{p}{p+q} = \mu \tag{3.1}$$

Si on pose,

$$\phi = p+q \tag{3.2}$$

On aura,

$$\mu = \frac{p}{\phi} \implies p = \mu\phi$$

1. Est une généralisation des modèles *ARMA*, conçu pour modéliser des séries chronologiques où les erreurs suivent n'importe quelle distribution. En effet, les modèles *ARMA* représentent un cas particulier des modèles *GARMA* dont les erreurs suivent une distribution gaussienne.

$$\implies p = \mu(p + q) \implies p - \mu p = \mu q \implies p(1 - \mu) = \mu q \implies q = \frac{p(1 - \mu)}{\mu} = \frac{p(1 - \mu)}{\frac{p}{\phi}}$$

$$q = \frac{p(1 - \mu)\phi}{p} \implies q = (1 - \mu)\phi, \text{ telque, } \phi > 0, \mu \in]0, 1[$$

et on a,

$$\mathbb{V}(Y) = \frac{pq}{(p + q)^2(p + q + 1)} = \frac{\mu\phi(1 - \mu)\phi}{\phi^2(\phi + 1)} = \frac{\mu(1 - \mu)}{\phi + 1} = \frac{\mathbb{V}(\mu)}{\phi + 1}$$

Pour toute valeur de μ , plus la valeur de ϕ augmente, plus la variance de Y diminue. Par l'utilisation de ce paramétrage, Ferrari et Cribari-Neto (2004) définissent un modèle de régression qui ressemble à la classe des modèles linéaires généralisés (GLM), c'est ce qui a conduit en 2009 à innover et introduire de nouveaux modèles, qui modélisent des séries chronologiques basés sur l'observation, nommés "Les modèles bêta *ARMA*" d'ordres (p, q) . Ces derniers sont apparus en raison de la nécessité d'avoir un modèle qui manipule des formes standards.

Ainsi, le terme "bêta" fait référence à la distribution bêta qui est adaptée pour modéliser des variables continues dans un intervalle fermé $[0, 1]$.

En utilisant la distribution bêta comme structure de base, ces processus peuvent fournir des prévisions plus adaptées pour les données qui présentent des caractéristiques particulières. De plus, les modèles $\beta ARMA$ sont pratiques pour la modélisation des séries temporelles stationnaires, qui ne répondent pas efficacement aux hypothèses de normalité des modèles *ARMA*.

3.2 Les modèles bêta *ARMA*

3.2.1 Définition

Un modèle bêta *ARMA* d'ordres (p, q) est un modèle dynamique² de séries chronologiques, qui intègre des distributions bêta dans le cadre des modèles *ARMA*, noté $\beta ARMA(p, q)$

et suggéré pour les variables aléatoires qui prennent des valeurs dans l'intervalle unitaire standard $(0, 1)$. Ce modèle est basé sur l'observation, dont la variable aléatoire considérée suit une distribution conditionnelle de type bêta.

Soit $(Y_t)_{t \in \mathbb{Z}}$ une série temporelle telle que, $\forall t \in \mathbb{Z}, Y_t \in (0, 1)$ et F_{t-1} la σ -algèbre qui représente ce qui est observé jusqu'à l'instant $(t-1)$. D'après le paramétrage de Ferrari et Cribari-Neto (2004), la construction du modèle $\beta ARMA(p, q)$

2. Est un modèle qui prend en compte l'évolution d'une variable ou d'un phénomène dans le temps. Il est conçu pour capturer les relations qui changent au fil du temps, en prenant en compte les dépendances temporelles (ie, la valeur actuelle d'une variable dépend des valeurs passées).

est faite comme suit,

$$f(y_t|F_{t-1}) = \frac{\Gamma(\phi)}{\Gamma(\phi\mu_t)\Gamma(\phi(1-\mu_t))} y_t^{\phi\mu_t-1} (1-y_t)^{(1-\mu_t)\phi-1}, \quad y_t \in]0, 1[$$

et on a,

$$\begin{aligned} \mathbb{E}(Y_t|F_{t-1}) &= \mu_t \\ \mathbb{V}(Y_t|F_{t-1}) &= \frac{\mu_t(1-\mu_t)}{1+\phi} = \frac{\mathbb{V}(\mu_t)}{1+\phi} \end{aligned}$$

D'après cette interprétation, nous avons constaté que le modèle est conditionnellement hétéroscédastique et nous avons également remarqué qu'à mesure que la valeur de ϕ

augmente, la variance conditionnelle diminue. Cependant, les valeurs très élevées de ϕ indiquent que le comportement d'homoscédasticité est respecté (la variance est constante).

Dans la classe des modèles de régression bêta, μ_t est lié à un prédicteur linéaire η_t via une fonction $g : (0, 1) \rightarrow \mathbb{R}$ qui est strictement monotone et deux fois différentiable. Rocha et Cribari-Neto (2009) ont proposé le modèle $\beta ARMA(p, q)$ suivant,

$$g(\mu_t) = \alpha + x_t' \beta + \sum_{i=1}^p \varphi_i \{g(y_{t-i}) - x_{t-i}' \beta\} + \sum_{j=1}^q \theta_j r_{t-j}$$

où, x_t représente les covariables non aléatoires telle que, $x_t = (x_{t_1}, \dots, x_{t_k})$.

$\beta = (\beta_1, \dots, \beta_k)'$ est l'ensemble des paramètres inconnus.

$(\varphi_1, \dots, \varphi_p)'$ et $(\theta_1, \dots, \theta_q)'$ sont les vecteurs des paramètres autoregressifs et moyenne mobile (respectivement).

r_t est le terme d'erreur qui est défini comme la différence entre une quantité observée (y_t ou $g(y_t)$) et la quantité basée sur le modèle correspondant (μ_t ou $g(\mu_t)$) (respectivement) tel que,

$[y_t - \mu_t]$ représente l'erreur sur l'échelle d'origine et $[g(y_t) - g(\mu_t) = g(y_t) - \eta_t]$ est l'erreur sur l'échelle des prédicteurs.

On a,

$\mathbb{E}(Y_t|F_{t-1}) = \mu_t$ ce qui implique que, $\mathbb{E}(r_t|F_{t-1}) = 0$. En effet,

$$\begin{aligned} \mathbb{E}(r_t|F_{t-1}) &= \mathbb{E}(Y_t - \mu_t|F_{t-1}) \\ &= \mathbb{E}(Y_t|F_{t-1}) - \mathbb{E}(\mu_t|F_{t-1}) \end{aligned}$$

puisque, μ_t est une moyenne donc, $\mathbb{E}(\mu_t|F_{t-1}) = \mu_t$

ce qui implique que, $\mathbb{E}(r_t|F_{t-1}) = 0$.

Rocha et Cribari-Neto (2009) ont montré que même à l'échelle de prédicteurs, la moyenne de r_t est approximativement nulle et sa variance est approximativement égale à $\left[\left(\frac{\partial \mu_t}{\partial \eta_t} \right)^2 \left(\frac{\mu_t(1-\mu_t)}{1+\phi} \right) \right]$.

3.2.2 Les propriétés des modèles $\beta ARMA$

Les modèles $\beta ARMA(p, q)$ présentent plusieurs propriétés importantes qui influent sur leur utilisation dans la modélisation des séries temporelles et parmi ces

propriétés voici quelques-unes,

1. La stationnarité

La stationnarité dans les modèles $\beta ARMA$ est assurée si les racines du polynôme autorégressif sont à l'extérieur du cercle unité, comme pour les modèles *ARMA*. Cependant, il faut également s'assurer que les valeurs restent dans l'intervalle $[0,1]$. Cela peut nécessiter des transformations pour que les séries générées soient stationnaires dans un cadre restreint.

2. L'inversibilité

L'inversibilité de ces modèles est généralement assurée quand les racines du polynôme caractéristique de la partie moyenne mobile (MA) se situent en dehors du cercle unité. Bien que, les paramètres de la composante *MA* doivent être ajustés de manière à ne pas générer des valeurs en dehors de $[0,1]$.

3. La causalité

Un modèle $\beta ARMA$ est dit causal s'il est stationnaire (i.e, les racines du polynôme caractéristique associé à la partie autorégressive soient à l'extérieur du cercle unité). Néanmoins, les coefficients de cette composante doivent être choisis avec soin pour respecter à la fois la causalité et les restrictions de l'intervalle.

4. Identification des paramètres

Le choix des ordres p et q peut avoir un impact significatif sur la performance du modèle, ainsi que l'estimation des coefficients θ et ϕ qui est considérée comme une étape importante dans la construction du modèle $\beta ARMA$. En effet, des ordres très élevés peuvent compliquer le respect des conditions de stationnarité tandis que, les ordres modérés sont souvent préférables pour une stabilité et une compatibilité avec les restrictions d'intervalle.

3.2.3 L'estimation dans les modèles $\beta ARMA(p, q)$

Globalement, selon l'approche fréquentiste, l'estimation des paramètres du modèle $\beta ARMA(p, q)$ est basée sur la méthode du maximum de vraisemblance conditionnelle. Cependant, cette estimation ne peut pas être exprimée sauf par l'utilisation des méthodes d'optimisation numériques, pour maximiser la fonction de log-vraisemblance.

Dans ce qui suit, nous allons utiliser l'algorithme d'optimisation non linéaire quasi-Newton BFGS, (voir Press et al. (1992)). Par l'usage de cet algorithme, Rocha et Cribari-Neto (2017) ont donné les dérivées de la log-vraisemblance associées au modèle $\beta ARMA(p, q)$ comme suit,

$$\frac{\partial \eta_t}{\partial \alpha} = 1 - \sum_{j=1}^q \theta_j \frac{\partial \eta_{t-j}}{\partial \alpha}$$

$$\frac{\partial \eta_t}{\partial \beta_j} = x'_t - \sum_{i=1}^p \varphi_i x'_{t-i} - \sum_{j=1}^q \theta_j \frac{\partial \eta_{t-j}}{\partial \beta_j}$$

$$\frac{\partial \eta_t}{\partial \varphi_i} = g(y_{t-i}) - x'_{t-i} \beta - \sum_{j=1}^q \theta_j \frac{\partial \eta_{t-j}}{\partial \varphi_i}, \quad i = 1, \dots, p$$

$$\frac{\partial \eta_t}{\partial \theta_l} = g(y_{t-l}) - \eta_{t-l} - \sum_{j=1}^q \theta_j \frac{\partial \eta_{t-j}}{\partial \theta_l}, \quad l = 1, \dots, q$$

3.2.4 Estimation par les méthodes MMC

Pour estimer les modèles bêta *ARMA* à l'aide des méthodes de Monte-Carlo, nous utilisons souvent les méthodes MCMC, qui permettent de générer des échantillons aléatoires à partir de distributions complexes. Ce qui est particulièrement utile pour l'estimation des paramètres des modèles bêta *ARMA*.

Des algorithmes spécifiques tels que, l'algorithme de Metropolis-Hastings ou celui de Gibbs peuvent être utilisés dans le cadre de cette approche.

Exemple d'application sous langage R (Algorithme de Metropolis-Hastings)

Dans cet exemple, nous allons estimer les paramètres ϕ et θ d'un modèle $\beta ARMA(1, 1)$ en utilisant la méthode de Metropolis-Hastings. Le programme qui suit génère une série temporelle de taille $n = 200$ à partir du modèle $\beta ARMA(1, 1)$ dont les valeurs réelles des paramètres sont $\phi - true = 0.8$ et $\theta - true = -0.7$ puis, après avoir initialiser les paramètres tels que $\phi = 0.6$ et $\theta = -0.4$ nous utilisons l'algorithme de Metropolis-Hastings pour générer 1000 itérations afin d'estimer les paramètres du modèle. ces paramètres estimés seront calculés à la fin du programme.

```
> # 1. Simulation des données BARMA(1,1)
> set.seed(123)
> n <- 200 # Taille de l'échantillon
> phi_true <- 0.8 # Paramètre AR(1) réel
> theta_true <- -0.7 # Paramètre MA(1) réel
>
> # Générer la série BARMA(1,1)
> y <- numeric(n)
> epsilon <- rnorm(n) # Bruit blanc
> y[1] <- runif(1) # Première observation dans [0,1]
>
> for (t in 2:n) {
+   y[t] <- phi_true * y[t-1] + epsilon[t] + theta_true * epsilon[t-1]
+   y[t] <- pmax(0, pmin(1, y[t])) # Contraindre les valeurs à [0,1]
+ }
>
> # 2. Fonction Metropolis-Hastings pour estimer phi et theta
> metropolis_hastings_BARMA <- fonction(y, n_iter = 1000) {
+   n <- length(y)
```

```
+
+ # Initialisation des paramètres
+ phi <- 0.6 # Paramètre AR initial
+ theta <- -0.4 # Paramètre MA initial
+
+ # Stockage des échantillons
+ phi_samples <- numeric(n_iter)
+ theta_samples <- numeric(n_iter)
+
+ # Fonction de vraisemblance
+ likelihood <- fonction(y, phi, theta) {
+   n <- length(y)
+   residuals <- numeric(n)
+ # La première observation est une donnée observée
+   residuals[1] <- y[1]
+
+   for (t in 2:n) {
+     residuals[t] <- y[t] - (phi * y[t-1] + theta * residuals[t-1])
+   }
+
+   return(-sum(residuals^2)) # Vraisemblance négative pour maximiser
+ }
+
+ # Algorithme de Metropolis-Hastings
+ for (i in 1:n_iter) {
+   # Proposer de nouvelles valeurs
+   phi_propose <- rnorm(1, mean = phi, sd = 0.1) # Proposition pour phi
+   theta_propose <- rnorm(1, mean = theta, sd = 0.1) # Proposition
+ pour theta
+ # Calcul des vraisemblances
+ likelihood_current <- likelihood(y, phi, theta)
+ likelihood_proposed <- likelihood(y, phi_propose, theta_propose)
+
+ # Calcul du rapport d'acceptation
+ acceptance_ratio <- exp(likelihood_proposed - likelihood_current)
+ # Acceptation ou rejet de la proposition
+ if (runif(1) < acceptance_ratio) {
+   phi <- phi_propose
+   theta <- theta_propose
+ }
+
+ # Sauvegarder les échantillons
+ phi_samples[i] <- phi
+ theta_samples[i] <- theta
+ }
+
+ # Retourner les échantillons
```

```

> list(phi_samples = phi_samples, theta_samples = theta_samples)
> }
>
> # 3. Exécution de l'échantillonneur Metropolis-Hastings
> resultats <- metropolis_hastings_BARMA(y)
>
> # 4. Calcul des valeurs estimées
> phi_est <- mean(resultats$phi_samples)
> theta_est <- mean(resultats$theta_samples)
>
> # 5. Afficher les valeurs estimées
> cat("Estimation de phi:", phi_est, "\n")
Estimation de phi: 0.9867228
> cat("Estimation de theta:", theta_est, "\n")
Estimation de theta: -0.8661483

```

• Interprétation des résultats

La valeur estimée pour ϕ est 0.9867228 et celle pour θ est -0.8661483 ce qui montre que les valeurs estimées pour les paramètres du modèle $\beta ARMA(1,1)$ en utilisant la méthode de Metropolis-Hastings sont relativement proches des valeurs réelles $\phi - true = 0.8$ et $\theta - true = -0.7$ avec un écart de "0.1" Ces résultats impliquent que l'algorithme de Metropolis-Hastings fonctionne bien pour estimer un modèle $\beta ARMA(1,1)$. Cependant, il est possible d'améliorer la précision en augmentant le nombre d'itérations.

Exemple d'application sous langage R (Algorithme de Gibbs)

L'exemple qui suit, représente une estimation des paramètres ϕ et θ d'un modèle $\beta ARMA(1,1)$ en utilisant la méthode de Gibbs. Ce programme génère une série temporelle de taille $n = 200$ à partir du modèle $\beta ARMA(1,1)$ tel que, les valeurs réelles des paramètres sont $\phi - true = 0.8$ et $\theta - true = -0.7$ ensuite, après l'initialisation des paramètres $\phi = 0.75$ et $\theta = -0.65$ nous utilisons l'algorithme de Gibbs pour générer 1000 itérations afin d'estimer les paramètres du modèle.

```

> # Install necessary librairie
> library(MASS)
> # Simulation des données pour un modèle BARMA(1,1)
> set.seed(123)
> n <- 200 # Taille de l'échantillon
> phi_true <- 0.8
> theta_true <- -0.7
>
> # Générer la série temporelle (y_t)
> epsilon <- rnorm(n) # Bruit blanc
> y <- numeric(n)
> y[1] <- rnorm(1) # Condition initiale
>
> for (t in 2:n) {
+   y[t] <- phi_true * y[t-1] + epsilon[t] + theta_true * epsilon[t-1]

```

```
+ }
>
> # Initialisation des paramètres
> phi <- 0.75
> theta <- -0.65
> n_iter <- 1000 # Nombre d'itérations MCMC
> phi_samples <- numeric(n_iter)
> theta_samples <- numeric(n_iter)
>
> # Fonction de vraisemblance conditionnelle pour phi et theta
> log_likelihood <- fonction(phi, theta, y, epsilon) {
+   n <- length(y)
+   ll <- 0
+   for (t in 2:n) {
+     ll <- ll + dnorm(y[t], mean = phi * y[t-1]
+       +theta * epsilon[t-1], sd = 1, log = TRUE)
+   }
+   return(ll)
+ }
>
> # Algorithme de Gibbs
> for (i in 1:n_iter) {
+   # Mise à jour de phi
+   phi_prop <- rnorm(1, phi, sigma_phi)
+   if (phi_prop > -1 && phi_prop < 1) {
+     log_ratio_phi <- log_likelihood(phi_prop, theta, y, epsilon)
+     - log_likelihood(phi, theta, y, epsilon)
+     if (log(runif(1)) < log_ratio_phi) {
+       phi <- phi_prop
+     }
+   }
+
+   # Mise à jour de theta
+   theta_prop <- rnorm(1, theta, sigma_theta)
+   if (theta_prop > -1 && theta_prop < 1) {
+     log_ratio_theta <- log_likelihood(phi, theta_prop, y, epsilon)
+     - log_likelihood(phi, theta, y, epsilon)
+     if (log(runif(1)) < log_ratio_theta) {
+       theta <- theta_prop
+     }
+   }
+
+   # Stocker les échantillons
+   phi_samples[i] <- phi
+   theta_samples[i] <- theta
+ }
>
> # Résultats : valeurs estimées pour phi et theta
```

```

> phi_est <- mean(phi_samples)
> theta_est <- mean(theta_samples)
>
> cat("Estimation de phi :", phi_est, "\n")
Estimation de phi : 0.8292103
> cat("Estimation de theta :", theta_est, "\n")
Estimation de theta : -0.7895644

```

• Interprétation des résultats

Après l'exécution du programme, les résultats montrent les estimateurs $\hat{\phi} = 0.8292103$ et $\hat{\theta} = -0.7895644$ associés à ϕ et θ respectivement, et qui sont proches des valeurs réelles des paramètres $\phi - true = 0.8$ et $\theta - true = -0.7$, ce qui indique que l'algorithme de Gibbs a bien convergé vers les vraies valeurs. D'où, l'algorithme de Gibbs fonctionne bien pour l'estimation des paramètres d'un modèle $\beta ARMA(1, 1)$.

3.3 Tests d'adéquation

3.3.1 Définition

Les tests d'adéquation consistent à déterminer et vérifier la compatibilité des données observées. Le principe de ces tests est de poser deux hypothèses, l'hypothèse nulle notée (H_0) qu'on désire contrôler et l'hypothèse alternative notée (H_1) qui représente généralement la négation de (H_0).

Nous distinguons divers tests d'adéquation, tels que le test de Kolmogorov-Smirnov, le test de Shapiro-Wilk et le test de Jarque-Bera, qui visent à vérifier la normalité des données, en posant l'hypothèse nulle selon laquelle les données proviennent d'une distribution normale. Cependant, dans ce qui suit, nous nous intéressons aux tests de porte-manteau.

Tests de porte-manteau

Les tests de porte-manteau sont fréquemment utilisés en analyse des séries temporelles, dans le cadre de l'évaluation de l'adéquation d'un modèle ajusté aux données. Ces tests vérifient si les résidus d'un modèle ajusté présentent une autocorrélation significative. En effet, l'absence d'autocorrélation révèle que le modèle est bien spécifié. Ces hypothèses sont données comme suit,

$$\begin{cases} H_0 : \rho_1 = \rho_2 = \dots = \rho_m = 0 \\ H_1 : \text{au moins une } \rho_i \neq 0 \end{cases}$$

où, ρ_i représente les autocorrélations d'un modèle.

L'hypothèse (H_0) signifie l'absence d'autocorrélations, tandis que l'hypothèse (H_1) indique qu'il existe au moins une autocorrélation entre les résidus d'un modèle.

Relativement aux modèles $\beta ARMA$, les tests de porte-manteau doivent prendre en compte les caractéristiques particulières de la série et les transformations des résidus. Les tests de porte-manteau les plus largement adoptés sont les suivants,

1. Test de Box-Pierce ou de Ljung-Box

Le test de Ljung-Box (1978) sert à vérifier l'absence d'autocorrélation dans des séries chronologiques multivariées.

La statistique du test de Ljung-Box est la suivante,

$$Q_{LB} = n(n+2) \sum_{k=1}^m \frac{(\rho_k)^2}{n-k}$$

où, ρ_k est l'autocorrélation des résidus au décalage k .

n est la taille de l'échantillon.

m est le nombre de décalages sur lesquels le test est effectué.

2. Test de Monti

Ce test est une variante du test de Ljung-Box, qui vérifie l'indépendance des résidus dans un modèle de séries chronologiques, en se basant sur les carrés des autocorrélations.

La statistique du test de Monti est définie par,

$$Q_M = n \sum_{k=1}^m (\rho_k)^2$$

où, $(\rho_k)^2$ est le carré de l'autocorrélation des résidus au décalage k .

3. Test de Dufour et Roy

Ce test est utilisé pour détecter l'autocorrélation dans les résidus d'un modèle.

La statistique du test de Dufour et Roy est donnée comme suit,

$$Q_{DR} = n \sum_{k=1}^m \left(\frac{\rho_k}{\sqrt{n-k}} \right)^2$$

4. Test de Kwan et Sim

Lors de l'existence d'autocorrélations entre les résidus d'un modèle, ce test prend en compte des aspects spécifiques liés aux erreurs de modélisation.

La statistique du test de Kwan et Sim est définie par la formule suivante,

$$Q_{KW} = \frac{T}{2} \sum_{j=1}^p \left(\frac{\hat{\rho}_j^2}{1 - \hat{\rho}_j^2} \right)$$

où,

$\triangleleft T$ est la taille de l'échantillon.

$\triangleleft \hat{\rho}_j^2$ représente l'estimation du coefficient d'autocorrélation d'ordre j .

$\triangleleft p$ est l'ordre maximal de l'autocorrélation que l'on souhaite tester.

5. Test de Lin et McLeod

Ce test est également utilisé pour tester l'adéquation, en se concentrant sur les autocorrélations des résidus.

La statistique du test de Lin et McLeod est la suivante,

$$Q_{LM} = T \sum_{j=1}^p \frac{\hat{\rho}_j^2}{1 - \hat{\rho}_j^2}$$

- où,
 $\triangleleft T$ est la taille de l'échantillon,
 $\triangleleft \hat{\rho}_j$ est l'estimation du coefficient d'autocorrélation d'ordre j ,
 $\triangleleft p$ est l'ordre maximal de l'autocorrélation testé.

• **Nouveaux tests de Porte-manteau.**

Nous proposons deux nouveaux tests de Porte-manteau qui sont basés sur la corrélation partielle des résidus. En effet, les statistiques du test associées à ces derniers sont les suivantes,

1. La statistique du premier test notée ' Q_1 ' est donnée par la formule suivante,

$$Q_1 = \sum_{k=1}^m (n - k - 3) \hat{Z}_{1k}^2$$

où, $\hat{Z}_{1k}^2 = \frac{1}{2} \log \frac{1+\hat{\pi}_k}{1-\hat{\pi}_k}$ $k = 1, \dots, m$ représente la transformation introduite par Fisher (1921).

2. La statistique de second test notée ' Q_4 ' est définie comme suit,

$$Q_4 = \sum_{k=1}^m \frac{(n - k)^2}{n - k - 1} \hat{Z}_{4k}$$

où, $\hat{Z}_{4k} = \sin^{-1}(\hat{\pi}_k)$ est la transformation introduite par Jenkins (1954).

3.4 Application sur données réelles

Considérons la proportion d'énergie hydroélectrique stockée dans le sud du Brésil comme variable d'intérêt où, les données sont des moyennes mensuelles allant de janvier 2001 à octobre 2016, les six observations utilisées pour évaluer la précision de quatre modèles sont de novembre 2016 à avril 2017.

Nous utilisons la fonction `read.table()` pour lire les données d'un fichier text nommé "hydro.txt" puis, les stockées dans la variable (data).

- On transforme les données en séries temporelles, puis on réalise une analyse statistique et graphique afin d'examiner leurs propriétés avant de prévoir leur évolution pour les six prochains mois.
- La table 3.1 ci-dessous présente quelques statistiques descriptives sur les données, en tenant compte de l'asymétrie négative et d'applatissage négatif.

min	max	median	mean	variance	asymmetry	excess kurtosis
0.2977	0.9862	0.7323	0.7069	0.0403	-0.3297	-1.1449

TABLE 3.1 – Statistiques descriptives sur les taux moyens d'énergie stockée dans le sud du Brésil.

• Interprétation

Nous observons une moyenne des données de 0.7069 et une médiane de 0.7323 qui est proche de la moyenne, ce qui suggère que la distribution des données n'est pas fortement asymétrique, et nous remarquons une faible variance de 0.0403, ce qui signifie que les valeurs sont relativement concentrées autour de la moyenne avec peu de variabilité, indiquant ainsi une certaine stabilité dans les données.

Une faible asymétrie de -0.3297 est un bon signe pour la stabilité du modèle. De plus, un kurtosis de -1.1449 inférieur à 3 montre que la distribution est plutôt plate avec moins de valeurs extrêmes, cela signifie que les prévisions ne sont pas influencées par des valeurs aberrantes³, ce qui est idéal pour obtenir un modèle stable et précis.

3. Représentent des points de données isolés qui se trouvent au-dessous ou bien au-dessus du reste de données. Ces valeurs peuvent indiquer des événements inhabituels ou des erreurs dans la collecte de données.

• Le graphique de l'ACF

Le graphique qui suit, sert à examiner les autocorrélations entre les valeurs de la série chronologique y .

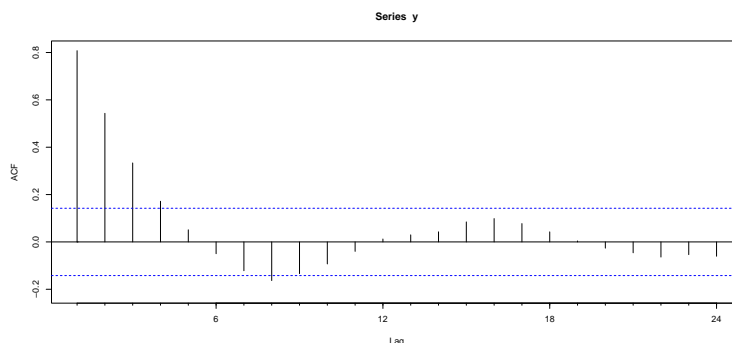


FIGURE 3.1 – Autocorrélations de la série temporelle y

Interprétation

Nous constatons une forte autocorrélation au premier lag, ce qui indique une forte dépendance entre les valeurs successives de la série (ie : les valeurs de la série sont fortement influencées par les valeurs précédentes). Ensuite, nous observons une décroissance progressive⁴ indiquant ainsi, une stationnarité faible de la série.

• Le graphique de la PACF

Le graphique ci-dessous représente les autocorrélations partielles de la série chronologique y .

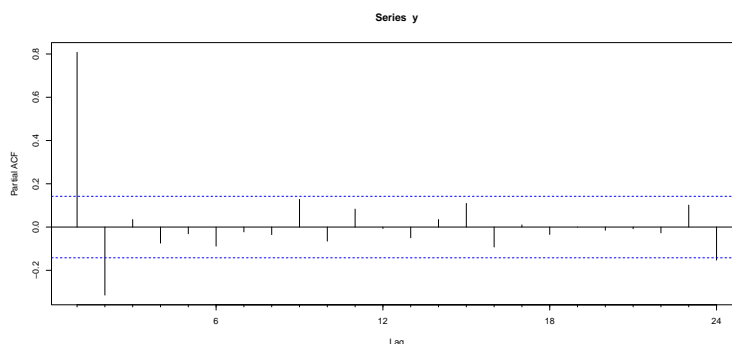


FIGURE 3.2 – Autocorrélations partielles de la série temporelle y

Interprétation

Nous observons une valeur de PACF très élevée autour de 0,8 juste au premier décalage, ce qui montre une forte corrélation entre les valeurs de la série et les valeurs précédentes. Cependant, au delà du premier lag nous remarquons une absence de corrélations significatives (ie : une décroissance rapide de la PACF après le premier lag), cela suggère une certaine stabilité dans la structure des dépendances⁵ qui est une caractéristique d'une série stationnaire.

4. Les valeurs d'autocorrélations diminuent lentement et de manière régulière à mesure qu'on s'éloigne du premier lag.

5. Stabilité des relations entre les valeurs de la série à différents moments dans le temps.

• Le modèle $\beta ARMA(1, 1)$ a été sélectionné en utilisant le critère d'Akaike⁶ (AIC, Akaike, 1974) dont l'AIC est égal à -307.9635. En maximisant numériquement la fonction log-vraisemblance conditionnelle par l'usage de l'algorithme quasi-Newton, l'estimation des paramètres a été réalisée. Il est noté que tous les paramètres sont significativement différents de zéro au seuil de 1%.

Après l'ajustement du modèle $\beta ARMA(1, 1)$, on aura la figure 3.3 qui illustre la série chronologique observée ainsi que les valeurs prédites par ce modèle.

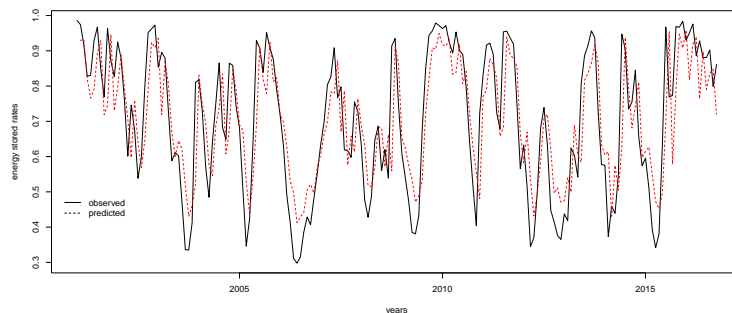


FIGURE 3.3 – Taux d'énergie stockée et valeurs prédites à partir du modèle $\beta ARMA(1, 1)$ ajusté.

• Interprétation graphique

La figure 3.3 montre les taux d'énergie stockée (lignes continues) et les valeurs prédites (lignes rouges en pointillés) obtenues à partir du modèle $\beta ARMA(1, 1)$. Cette figure permet de visualiser la performance prédictive du modèle, en le comparant aux valeurs observées. ce qui montre que le processus ajusté est capable de capturer efficacement la dynamique des données.

Remarque

Dire qu'un modèle est précis en utilisant les tests de portemanteau avec différentes valeurs de m^a si, les p-values associées aux tests sont supérieures au seuil α (typiquement 0.05) pour chaque valeur de m . Cela signifie l'acceptation de l'hypothèse nulle où les résidus ne sont pas autocorrélés, ce qui indique que le modèle capture de manière satisfaisante la dynamique des données.

a. Représente le nombre de retards (lag) d'autocorrélations prisent en compte pour évaluer la qualité d'ajustement d'un modèle.

6. Indique une meilleure qualité d'ajustement en prenant en compte le nombre des paramètres ajustés, tel que, le modèle le plus approprié est celui qui prend la plus faible valeur d'AIC.

• En appliquant les tests de portemanteau au modèle $\beta ARMA(1,1)$ pour $m \in \{5, 10, 15, 20, 25, 30\}$, nous avons obtenu les résultats illustrés dans les graphiques suivants,

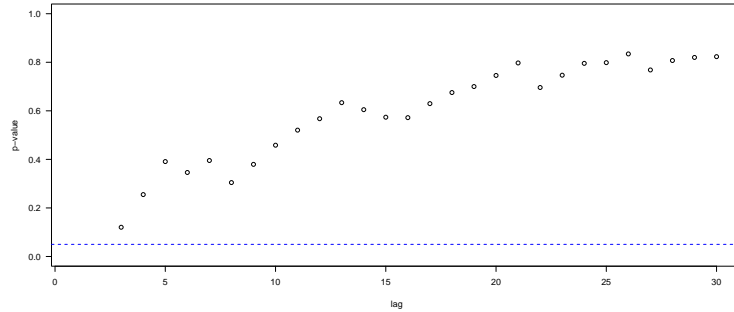


FIGURE 3.4 – P-values de test de Ljung-Box.

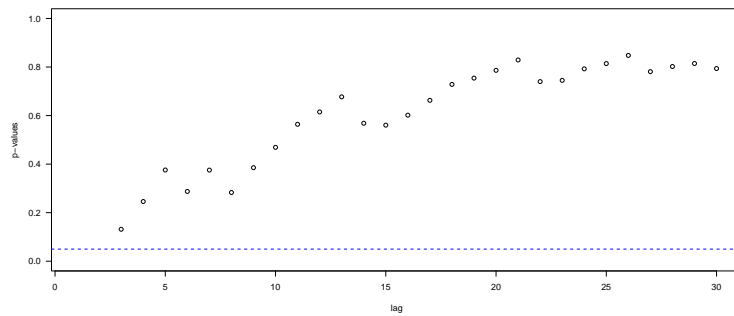


FIGURE 3.5 – P-values de test de Monti.

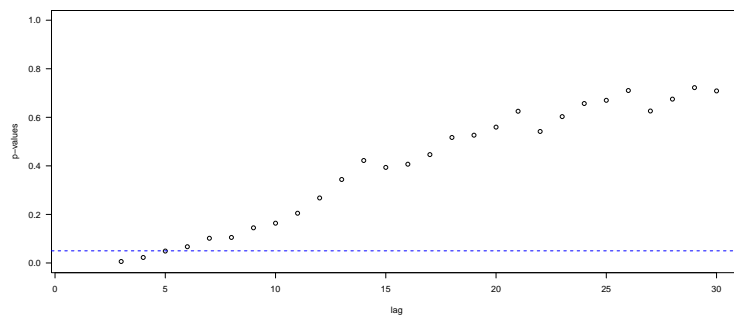


FIGURE 3.6 – P-values de test de Dufour et Roy.

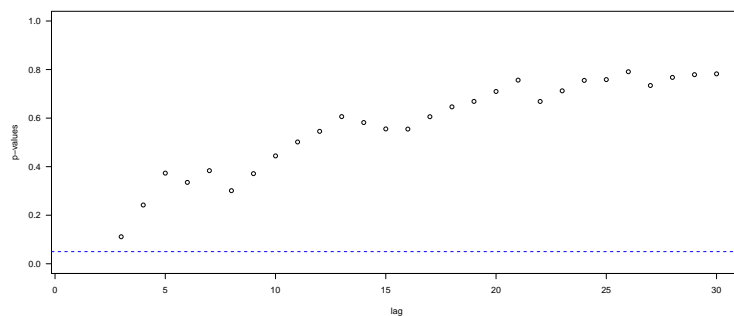


FIGURE 3.7 – P-values de test de Kwan et Sim (1).

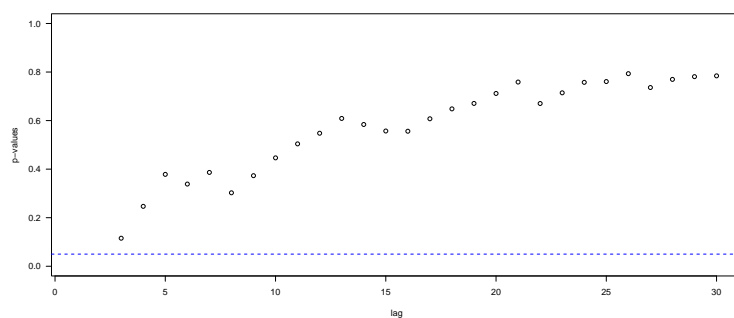
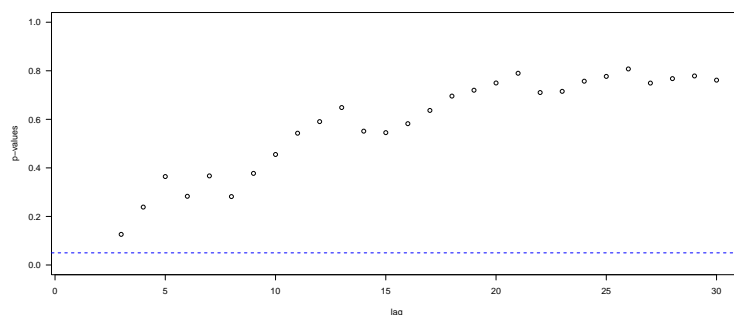


FIGURE 3.8 – P-values de test de Kwan et Sim (4).

FIGURE 3.9 – P-values de test Q_4 .

•Interprétation graphique

Nous remarquons que la majorité des p-values des différents tests de portemanteau se situent au-dessus de la ligne pointillée à 0.05 indiquant ainsi que le modèle $\beta ARMA(1, 1)$ semble bien ajusté. Cependant, pour des petites valeurs de m , le test de Dufour et Roy présente des p-values proches de 0.05, ce qui peut indiquer une légère inadéquation mais, globalement l'ensemble des tests montre que le modèle $\beta ARMA(1, 1)$ parvient à modéliser efficacement la dynamique des données

• **Application des tests de portemanteau sur un modèle $\beta AR(1)$**

Pour savoir si un modèle $\beta AR(1)$ représente correctement les données, nous allons appliquer les tests de portemanteau sur ce modèle simplifié. Les p-values de ces tests sont illustrées dans la table qui suit,

	m=5	m=10	m=15	m=20	m=25	m=30
Q_{LB}	0.006	0.006	0.026	0.058	0.128	0.182
Q_M	0.001	0.002	0.008	0.042	0.054	0.065
Q_{DR}	0.001	0.002	0.011	0.022	0.056	0.078
Q_{KW1}	0.005	0.005	0.021	0.043	0.087	0.120
Q_{KW4}	0.005	0.006	0.021	0.044	0.090	0.125
Q_4	0.001	0.002	0.006	0.029	0.038	0.045

• **Interprétation des résultats**

Les résultats suggèrent que tous les tests montrent des p-values qui dépassent souvent 0.05 pour les différentes valeurs de m. Néanmoins, seul le test Q_4 présente des p-values systématiquement inférieures à 0.05 pour toutes les valeurs de m, cela suggère que le modèle $\beta AR(1)$ n'est pas adéquat selon ce test.

• Nous souhaitons maintenant prévoir la série temporelle sur les six prochains mois, où nous adoptons l'erreur absolue moyenne (MAE)⁷ comme mesure de la précision des prévisions.

Posons un horizon $h = 1, \dots, 6$ et considérons quatre modèles de séries chronologiques différents ($\beta ARMA(1, 1)$, $ARMA(1, 1)$, $AR(2)$, $KARMA(1, 1)$ ⁸) et un algorithme de lissage exponentiel de Holt.⁹

• La table ci-dessous affiche les résultats des prévisions,

	h=1	h=2	h=3	h=4	h=5	h=6
$\beta ARMA(1, 1)$	0.1244	0.1444	0.1364	0.1484	0.1694	0.1839
$ARMA(1, 1)$	0.1518	0.1828	0.1820	0.1982	0.2211	0.2364
$KARMA(1, 1)$	0.1474	0.1788	0.1783	0.1950	0.2189	0.2352
$AR(2)$	0.1345	0.1690	0.1680	0.1830	0.2050	0.2198
Holt	0.1554	0.2110	0.2303	0.2629	0.2996	0.3264

• **Interprétation** Nous constatons que le modèle $\beta ARMA(1, 1)$ présente les erreurs de prévision les plus faibles pour tous les horizons, ce qui en fait le modèle le plus précis. Par exemple, pour $h = 1$ la valeur de la MAE est de 0.1244 qui est plus faible que les MAE des autres modèles.

Les modèles $ARMA(1, 1)$, $KARMA(1, 1)$ et $AR(2)$ ont des MAE plus élevées par rapport au modèle $\beta ARMA(1, 1)$ d'où ils sont moins précis. Cependant, le modèle (lissage exponentiel de Holt) donne les erreurs les plus élevées pour tous les horizons, signifiant qu'il s'adapte moins bien aux données.

Les résultats de ces prévisions confirment que le modèle $\beta ARMA(1, 1)$ est le plus compatible et le mieux approprié pour capturer la dynamique des données.

7. Désigne la moyenne des valeurs absolues des écarts entre les prévisions et les observations réelles. Telle que, une MAE plus faible indique un modèle plus précis.

8. Des modèles similaires aux modèles *ARMA*, mais avec plus de flexibilité grâce à l'utilisation de plusieurs lags, cela permet de mieux analyser et prédire des données des séries chronologiques complexes.

9. Une méthode de prévision pour les séries temporelles, qui permet de prédire des valeurs futures en prenant en compte la moyenne de la série et sa tendance.

3.5 Conclusion

Finalement, pour ce type de données, les modèles $\beta ARMA$ sont les plus efficaces pour la modélisation et la prévision des séries temporelles. De plus, nous concluons que le test de Kwan-Sim et le nouveau test (Q_4) sont les plus puissants pour examiner l'efficacité des modèles $\beta ARMA$ tandis que, le test de Dufour-Roy n'est pas efficace pour ce type de modèle. Cette étude ouvre la voie à des applications plus larges de ces tests de portemanteau dans d'autres modèles et pour différents types de données.

Conclusion générale

Cette étude a exploré les modèles bêta *ARMA* comme une avancée significative dans l'analyse des séries temporelles, en particulier, celles ayant des valeurs contraintes entre 0 et 1, dont nous avons d'abord établi les fondements des modèles *ARMA* classiques, souligné leurs propriétés et méthodes d'estimation, et mentionné certains tests d'hypothèses les plus couramment utilisés.

Ensuite, nous avons mis en avant l'importance des méthodes MMC, notamment les techniques MCMC pour l'estimation des modèles $\beta ARMA$. En outre, nous avons déterminé les différents tests de porte-manteau qui constituent des outils nécessaires pour l'évaluation de la pertinence des modèles $\beta ARMA$. Pour certaines séries temporelles, l'application de ces tests a révélé que les modèles bêta *ARMA* offrent une meilleure adéquation.

Enfin, cette humble recherche contribue à élargir et enrichir le cadre de la modélisation des séries chronologiques. Les résultats obtenus ouvrent la voie à de futures recherches, particulièrement sur l'application des modèles $\beta ARMA$ dans divers domaines et pour plusieurs types de données, renforçant ainsi leur pertinence, notamment en statistique.

Bibliographie

- [1] Akaike, H. A (1974), *new look at the statistical model identification*. IEEE Transactions on Automatic Control. 19(6) : 716-723.
- [2] Belkacem, C. (2023), *Chapitre 1 : Méthodes de Monte-Carlo, Master 2 (PS)*. Département de mathématiques, Faculté des sciences, Université Mouloud Mammeri de Tizi-Ouzou.
- [3] Benjamin, M.A, Rigby, R.A and Stasinopoulos, M.D. (2023), *Generalized autoregressive moving average models*. Journal of the American statistical Association, 98(461), 214-223.
- [4] Box, G.E.P and Jenkins, G.M (1976), *Time Series Analysis forecasting and Control*, 2nd ed., Holden-Day, San Francisco.
- [5] Christain P.R et George.C. (2010). *Introducing Monte Carlo Methods with R*. Springer Science & Business Media.
- [6] El Merouani, M. (2015), *Chapitre 2 : Modèles linéaires (de Box-Jenkins)*. Département de statistique et informatique, Faculté polydisciplinaire de Tétouan, Université Abdelmalek Essaâdi.
- [7] Fellag, H. (2023), *Séries Chronologiques 1, Master 1 (PS)*. Département de mathématiques, Faculté des sciences, Université Mouloud Mammeri de Tizi-Ouzou.
- [8] Ferrari, S.L.P and Cribari-Neto,F. (2004), *Beta regression for modelling retes and proportions*. Journal of Applied Statistics, 31(7), 799-815.
- [9] Fisher, R.A. (1921), *On the "probable error" of a coefficient of correlation deduced from a small sample*. Metron,1, 3-32.
- [10] Fortier, S. (2013), *Les modèles MA, AR et ARMA multidimensionnels : estimation et causalité*. CaMUS, Département de mathématiques, Université de Sherbrooke, Volume 4, (112-136).
- [11] Hameed, L.M.A(2022), *A proposed conditional method for estimating ARMA(1,1) model*. Int.J.Nonlinear Anal.Appl. 13 (1), 3011-3020.
- [12] Jenkins, G. (1954), *An angular transformation for the serial correlation coefficient*. Biometrika, 41(1/2), 261-265.
- [13] Joulin, A. (2019-2020), *Méthodes de Monte Carlo*. INSA Toulouse, Département GMM.
- [14] Ljung, G.M and Box, G.E. (1978), *On a measure of lack of fit in time series models*. Biometrika, 65(2), 297-303.
- [15] Millet, A. *Mémoire de fin d'études*. Spécialité Modélisation aléatoire, Universités Paris 7 et Paris 1.

-
- [16] Prenat, M. (2010), *Séries Chronologiques*. Master Ingénierie Mathématique, Université Paris-Sud.
- [17] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P (1992) *Numerical Recipes in C : The Art of Scientific Computing*, volume 2. Cambridge University Press, New York.
- [18] Rocha,A.V et Cribari-Neto,F (2009), *Beta autoregressive moving average models*. TEST, 26(2), 451-545.
- [19] Rocha,A.V et Cribari-Neto,F (2017), *Erratum to : Beta autoregressive moving average models* . TEST, 26(2), 451-459.
- [20] Sher,V.T, Cribari-Neto,F et Bayer, F.M (2017), *Portmanteau testing inference in beta autoregressive moving average models*.
- [21] Sher,V.T, Cribari-Neto,F et Bayer, F.M (2019), *Goodness-of-fit tests for β ARMA hydrological time series modeling*. e 2607.

Résumé

Ce mémoire se concentre sur les modèles beta ARMA qui sont adaptés pour les séries temporelles, qui prennent des valeurs entre 0 et 1. Nous présentons d'abord les modèles ARMA et les méthodes de simulation de Monte-Carlo. Ensuite, nous détaillons les principes des modèles $\beta ARMA$ ainsi que les méthodes d'estimations telles que la méthode numérique de Newton et les méthodes MCMC. Puis, nous déterminons les tests d'adéquation basés sur les différents tests de porte-manteau dont l'application sur les modèles beta ARMA se concentre sur ces derniers pour évaluer leur adéquation. L'objectif est d'améliorer les outils de modélisation pour des séries chronologiques contraintes.

Mots clés : Modèles $\beta ARMA$, Monte-Carlo, Tests porte-manteau, Modèles ARMA.

Abstract

This thesis focuses on beta ARMA models, which are suitable for time series that take values between 0 and 1. We first present ARMA models and Monte-Carlo simulation methods such as the Newton numerical method and MCMC techniques. Then we determine the goodness-of-fit tests based on various portmanteau tests, with an application on beta ARMA models focusing on these tests to evaluate their adequacy. The objective is to improve modeling tools for constrained time series.

Keywords : $\beta ARMA$ models, Monte-Carlo, Portmanteau tests, ARMA models.