

Projet de Séries temporelles, M1 IES-D3S

Thomas Fernandes, Vanessa Kenniche, Yassine Ouerghi

2024-01-16

Exercice 1

a) $X_t = \sin(t) + \varepsilon_t - 0.2 \times \varepsilon_{t-1}$

1. Espérance

$$\mathbb{E}[X_t] = \mathbb{E}[\sin(t)] + \mathbb{E}[\varepsilon_t] - 0.2\mathbb{E}[\varepsilon_{t-1}]$$

$$\mathbb{E}[X_t] = \mathbb{E}[\sin(t)] + 0 - 0.2 \times 0$$

$$\mathbb{E}[X_t] = \mathbb{E}[\sin(t)]$$

Par conséquent, l'espérance de X_t n'est pas constante car elle dépend de t à travers le terme $\sin(t)$.

2. Autocovariance

$$\gamma_X(h) = \text{Cov}(\sin(t) + \varepsilon_t - 0.2 \times \varepsilon_{t-1}, \sin(t+h) + \varepsilon_{t+h} - 0.2 \times \varepsilon_{t+h-1})$$

La covariance des sinus s'annule donc :

$$\gamma_X(0) = \text{Var}(\varepsilon_t - 0.2 \times \varepsilon_{t-1})$$

$$\gamma_X(1) = \text{Cov}(\varepsilon_t - 0.2 \times \varepsilon_{t-1}, \varepsilon_{t+1} - 0.2 \times \varepsilon_t)$$

$$\gamma_X(-1) = \text{Cov}(\varepsilon_t - 0.2 \times \varepsilon_{t-1}, \varepsilon_{t-1} - 0.2 \times \varepsilon_{t-2})$$

- $h = 0$: $\gamma_X(0) = \text{Var}(\varepsilon_t - 0.2 \times \varepsilon_{t-1}) = \sigma^2 + 0.04\sigma^2 = 1.04$
- $h = 1$ ou $h = -1$: $\gamma_X(1) = \gamma_X(-1) = -0.2\sigma^2$.

Le processus n'est ainsi pas stationnaire puisque son espérance n'est pas constante.

3. Décomposition saisonnière

Pour rendre ce processus stationnaire, nous allons enlever la composante de saisonnalité. Ainsi :

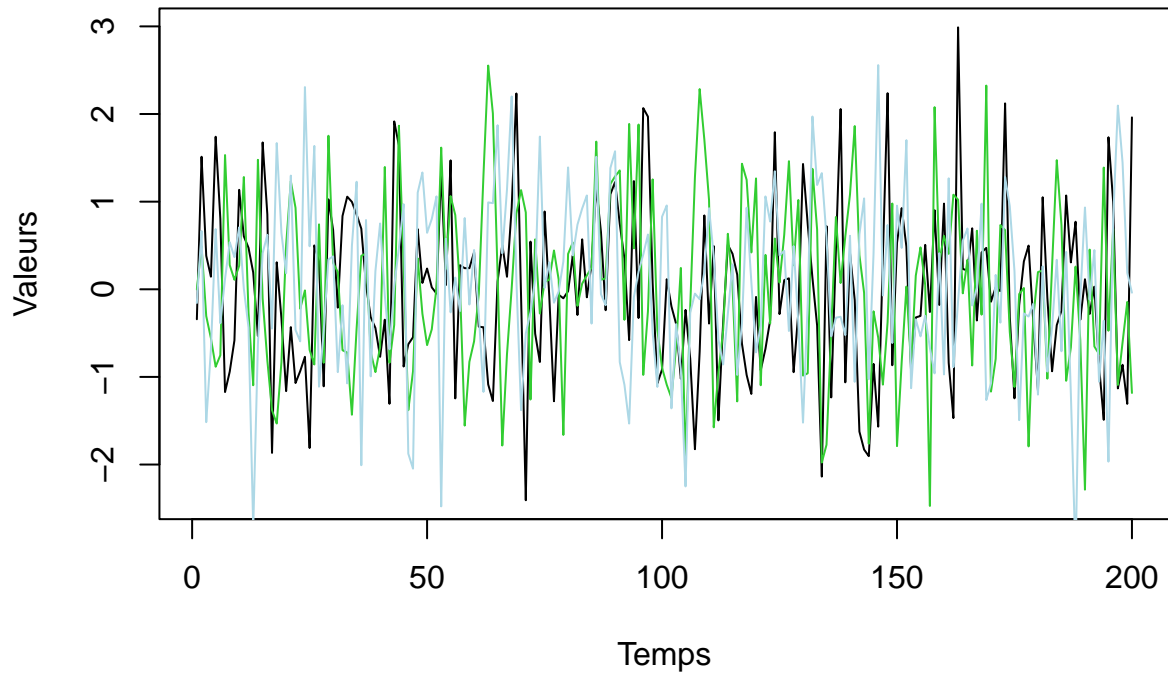
$$Y_t = X_t - \sin(t) = \varepsilon_t - 0.2 \times \varepsilon_{t-1}.$$

Le processus peut ainsi être réécrit sous la forme $X_t = (1 - 0.2B)\varepsilon_t$, correspondant à un MA(1).

```
set.seed(123)
a1 <- arima.sim(model = list(ma = 0.2), n = 200)
a2 <- arima.sim(model = list(ma = 0.2), n = 200)
a3 <- arima.sim(model = list(ma = 0.2), n = 200)

plot(a1, type="l", col="black", main="Trajectoires MA(1)",
     xlab = "Temps", ylab = "Valeurs")
lines(a2, col="limegreen")
lines(a3, col="lightblue")
```

Trajectoires MA(1)



b) $X_t = \varepsilon_t - \varepsilon_{t-1}$

1. Espérance

$$E(X_t) = E(\varepsilon_t - \varepsilon_{t-1}) = E(\varepsilon_t) - E(\varepsilon_{t-1}) = 0 - 0 = 0$$

2. Autocovariance

- Pour $h = 0$:

$$\begin{aligned}\gamma_X(0) &= \text{Cov}(X_t, X_{t+0}) \\ &= \text{Var}(X_t) \\ &= \text{Var}(\varepsilon_t) + \text{Var}(\varepsilon_{t-1}) \\ &= \sigma^2 + \sigma^2 \\ &= 2\end{aligned}$$

- Pour $h = 1$:

$$\begin{aligned}\gamma_X(1) &= \text{Cov}(X_t, X_{t+1}) \\ &= \text{Cov}(\varepsilon_t - \varepsilon_{t-1}, \varepsilon_{t+1} - \varepsilon_t) \\ &= \text{Cov}(\varepsilon_t, \varepsilon_{t+1}) - \text{Cov}(\varepsilon_t, \varepsilon_t) - \text{Cov}(\varepsilon_{t-1}, \varepsilon_{t+1}) + \text{Cov}(\varepsilon_{t-1}, \varepsilon_t)\end{aligned}$$

D'après la définition du bruit blanc, $\text{Cov}(\varepsilon_t, \varepsilon_s) = 0$, $t \neq s$, alors :

$$\begin{aligned}\gamma_X(1) &= 0 - \text{Var}(\varepsilon_t) - 0 + 0 \\ &= -\text{Var}(\varepsilon_t) \\ &= -1\end{aligned}$$

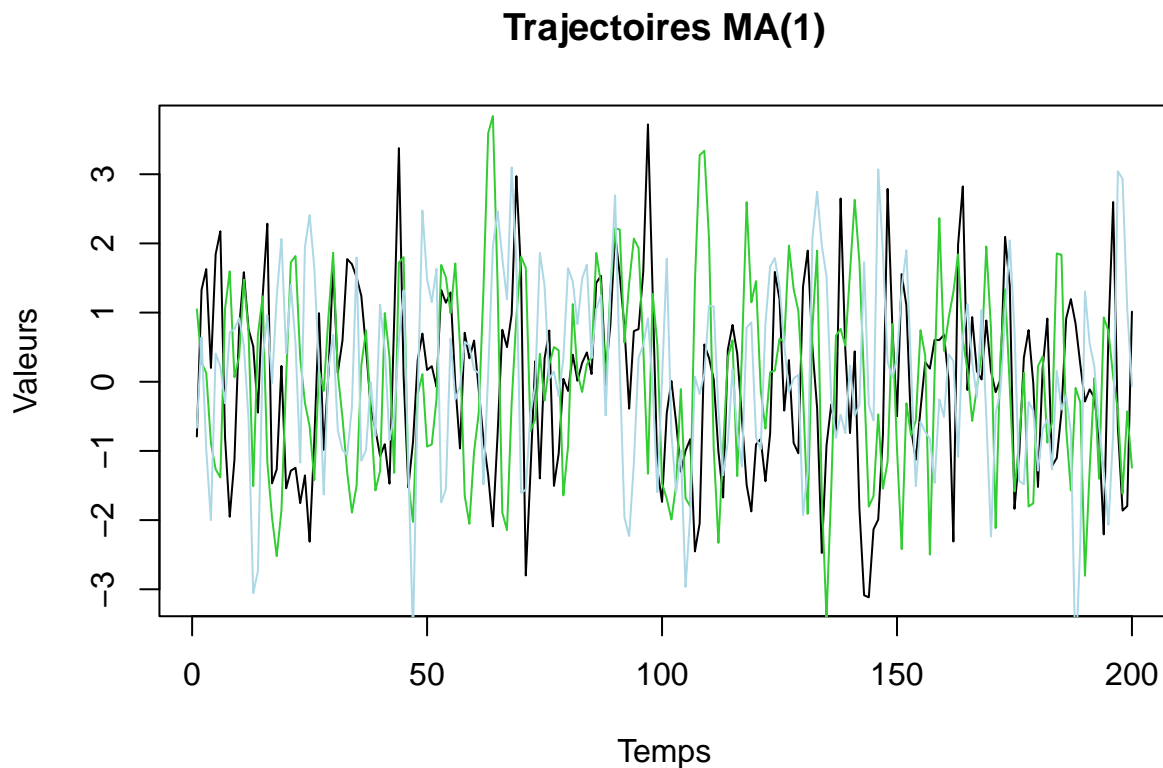
Le processus $X_t = \varepsilon_t - \varepsilon_{t-1}$ est stationnaire au second ordre car son espérance est constante et sa fonction d'autocovariance ne dépend que de h et non de t .

3. Forme canonique

Le processus, sous sa forme canonique, s'écrit comme : $X_t = (1 - B)\varepsilon_t$ correspondant à un un MA(1).

```
set.seed(123)
b1 <- arima.sim(model = list(ma = 1), n = 200)
b2 <- arima.sim(model = list(ma = 1), n = 200)
b3 <- arima.sim(model = list(ma = 1), n = 200)

plot(b1, type="l", col="black", main="Trajectoires MA(1)", xlab = "Temps", ylab = "Valeurs")
lines(b2, col="limegreen")
lines(b3, col="lightblue")
```



c) $X_t = A \cos(\omega t) + B \sin(\omega t)$

1. Espérance

$$E[X_t] = E[A] \cos(\omega t) + E[B] \sin(\omega t)$$

$$E[X_t] = E[A] \times E[\cos(\omega t)] + E[B] \times E[\sin(\omega t)] = 0$$

2. Autocovariance

$$\gamma(s, t) = E[(A \cos(\omega s) + B \sin(\omega s))(A \cos(\omega t) + B \sin(\omega t))]$$

$$= E[A^2 \cos(\omega s) \cos(\omega t) + AB \cos(\omega s) \sin(\omega t) + AB \sin(\omega s) \cos(\omega t) + B^2 \sin(\omega s) \sin(\omega t)]$$

Étant donné que A et B sont indépendants, on peut simplifier cette expression en utilisant $E[A^2]$, $E[B^2]$ et le fait que $E[AB] = E[A]E[B] = 0$:

$$\gamma(s, t) = E[A^2] \cos(\omega s) \cos(\omega t) + E[B^2] \sin(\omega s) \sin(\omega t)$$

Puisque A et B ont une variance finie on peut les remplacer par leurs variances :

$$\begin{aligned} \gamma(s, t) &= \sigma_A^2 \cos(\omega s) \cos(\omega t) + \sigma_B^2 \sin(\omega s) \sin(\omega t) \\ &= \sigma_A^2 \cdot \frac{1}{2} [\cos(\omega s - \omega t) + \cos(\omega s + \omega t)] + \sigma_B^2 \cdot \frac{1}{2} [\cos(\omega s - \omega t) - \cos(\omega s + \omega t)] \\ &= \frac{1}{2} (\sigma_A^2 + \sigma_B^2) \cos(\omega(t - s)) \end{aligned}$$

Le processus $X_t = A \cos(\omega t) + B \sin(\omega t)$ est stationnaire au second ordre, car son espérance est constante et sa fonction d'autocovariance dépend uniquement de la différence entre les temps s et t .

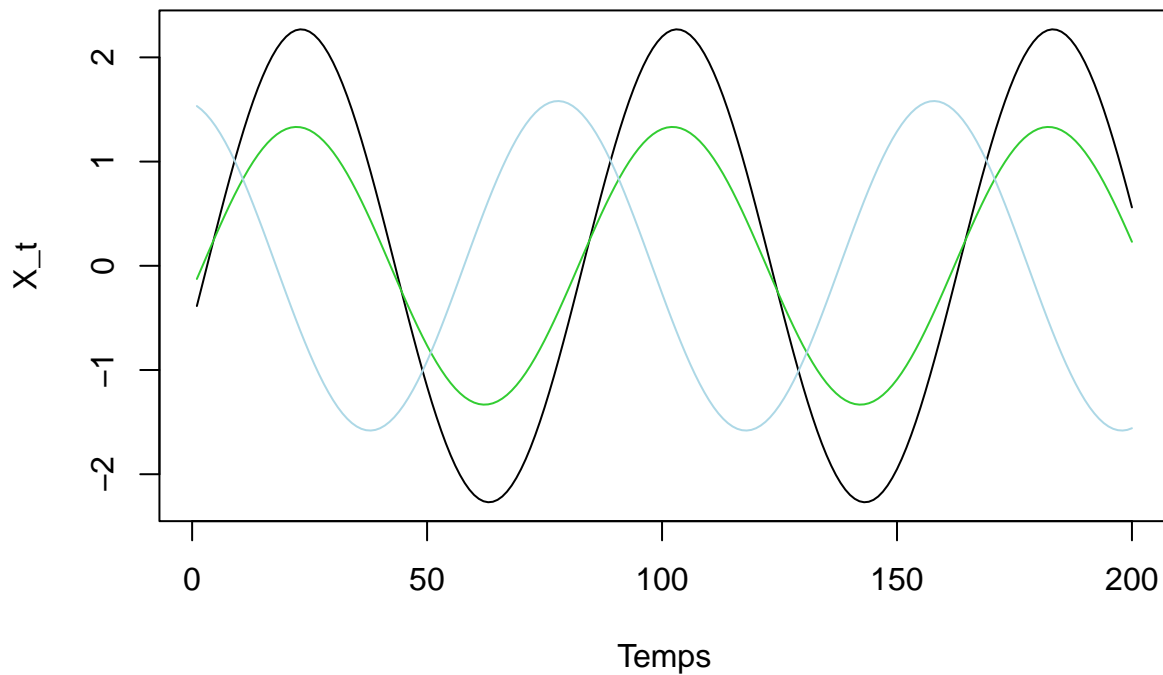
```
set.seed(123)
n <- 200
omega <- pi/40
A <- rnorm(n)
B <- rnorm(n)

t <- 1:n
trajectoires <- matrix(0, nrow = n, ncol = 3)

for (i in 1:3) {
  trajectoires[, i] <- A[i] * cos(omega * t) + B[i] * sin(omega * t)
}

plot(t, trajectoires[, 1], type = 'l', col = 'black', ylim = range(trajectoires),
      xlab = 'Temps', ylab = 'X_t', main = 'Trajectoires pour omega = pi/40')
lines(t, trajectoires[, 2], col = 'limegreen')
lines(t, trajectoires[, 3], col = 'lightblue')
```

Trajectoires pour $\omega = \pi/40$



Ce qui est intéressant avec ce processus, c'est qu'il est déterministe. En connaissant le A , B et ω de départ, on peut déterminer toute la trajectoire de X_t . Cependant, notre processus est défini pour ω fixé dans $[0, 2\pi]$ et simulé sur $n = 200$. Or, si on prend un ω proche de 0 mais $\neq 0$, la trajectoire de X_t s'apparentera à celle d'un processus non-stationnaire.

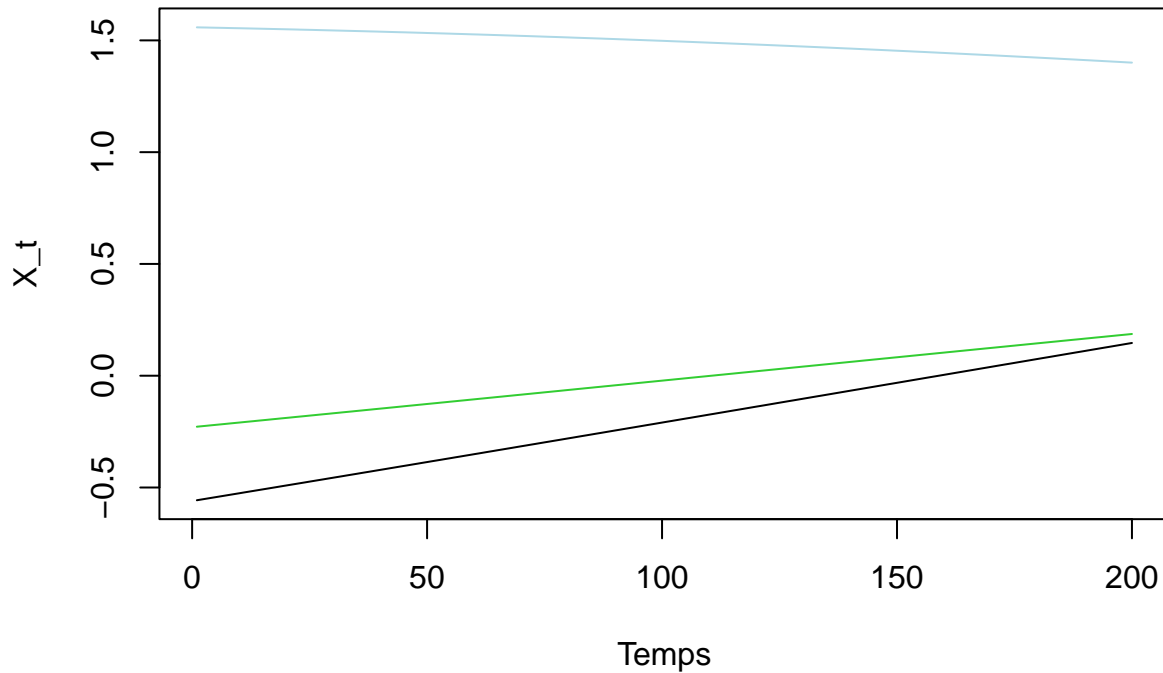
```
set.seed(123)
n <- 200
omega <- pi/2000
A <- rnorm(n)
B <- rnorm(n)

t <- 1:n
trajectoires <- matrix(0, nrow = n, ncol = 3)

for (i in 1:3) {
  trajectoires[, i] <- A[i] * cos(omega * t) + B[i] * sin(omega * t)
}

plot(t, trajectoires[, 1], type = 'l', col = 'black', ylim = range(trajectoires),
      xlab = 'Temps', ylab = 'X_t', main = 'Trajectoires pour omega = pi/2000')
lines(t, trajectoires[, 2], col = 'limegreen')
lines(t, trajectoires[, 3], col = 'lightblue')
```

Trajectoires pour omega = pi/2000



d) $X_t = 2X_{t-1} - X_{t-2} + \varepsilon_t$

1. Espérance :

$$\mathbb{E}[X_t] = 2\mathbb{E}[X_{t-1}] - \mathbb{E}[X_{t-2}] + \mathbb{E}[\varepsilon_t]$$

$$\mathbb{E}[X_t] = \mathbb{E}[X_t] + 0$$

Le processus est égale à lui-même et donc constant dans le temps.

3. Forme canonique

Le modèle peut se réécrire sous sa forme canonique comme : $(1 - 2B + B^2)X_t = \varepsilon_t$, correspondant à un AR(2).

1 est racine évidente du polynôme caractéristique $(1 - 2B + B^2)$.

Le processus n'est donc pas stationnaire puisqu'il a une racine sur le cercle unité.

4. Différenciation

Pour le rendre stationnaire, on va différencier le processus à l'ordre 2.

$$\Delta X_t = (2X_{t-1} - X_{t-2} + \varepsilon_t) - X_{t-1}$$

$$\Delta X_t = X_{t-1} - X_{t-2} + \varepsilon_t$$

$$\Delta^2 X_t = \Delta X_t - \Delta X_{t-1}$$

$$\Delta^2 X_t = (X_{t-1} - X_{t-2} + \varepsilon_t) - (X_{t-2} - X_{t-3} + \varepsilon_{t-1})$$

$$\Delta^2 X_t = X_{t-1} - 2X_{t-2} + X_{t-3} + \varepsilon_t - \varepsilon_{t-1}$$

Le processus correspond donc à un ARIMA(2,2,0).

```
processus_d <- function(n) {
  epsilon <- rnorm(n)
  X <- rep(0, n)
  delta2_X <- rep(0, n)

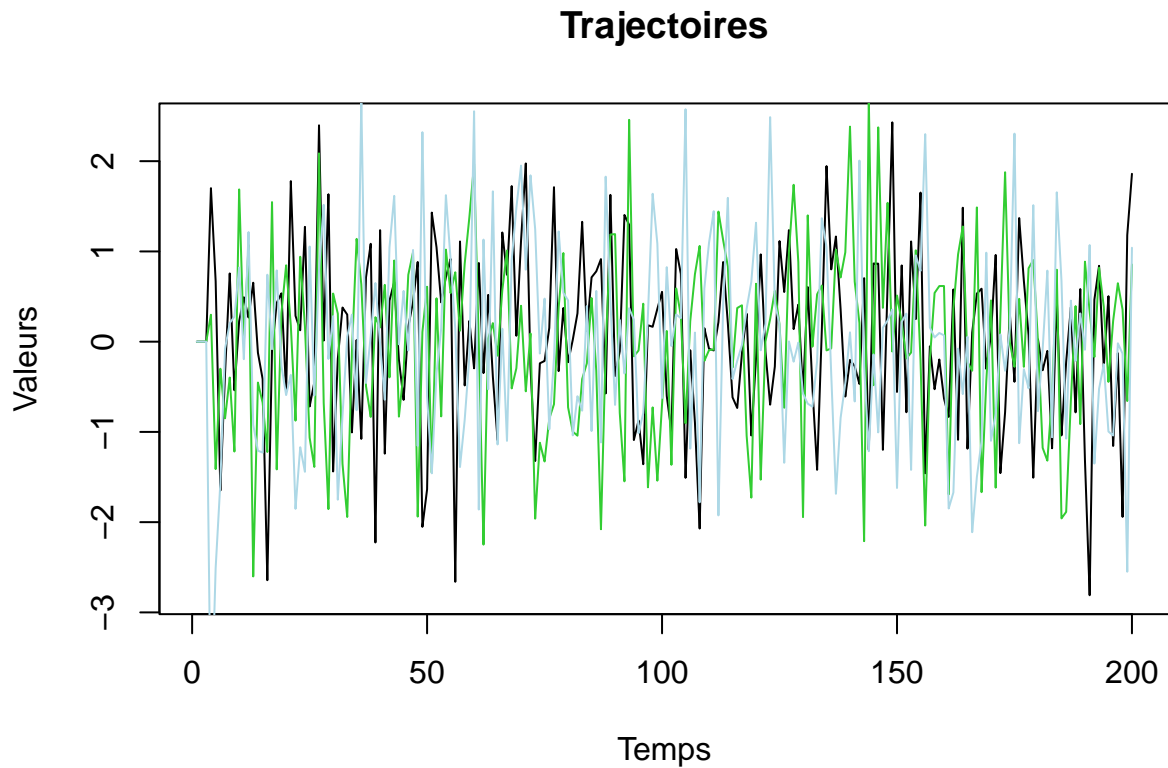
  X[1] <- rnorm(1)
  X[2] <- rnorm(1)
  X[3] <- rnorm(1)

  for (t in 4:n) {
    X[t] <- 2*X[t-1] - X[t-2] + epsilon[t]
    delta2_X[t] <- X[t-1] - 2*X[t-2] + X[t-3] + epsilon[t] - epsilon[t-1]
  }

  return(delta2_X)
}

n = 200
d1 <- processus_d(n)
d2 <- processus_d(n)
d3 <- processus_d(n)

plot(d1, type="l", col="black", main="Trajectoires", xlab="Temps", ylab="Valeurs")
lines(d2, col="limegreen")
lines(d3, col="lightblue")
```



e) $X_t - 4X_{t-1} = \varepsilon_t - 0.25\varepsilon_{t-1}$

Nous pouvons réécrire le processus comme $(1 - 4B)X_t = (1 - 0.25B)\varepsilon_t$.

Solution de l'équation caractéristique pour la partie AR :

$$\begin{aligned} 1 - 4r &= 0 \\ r &= 0.25 \end{aligned}$$

Solution de l'équation caractéristique pour la partie MA :

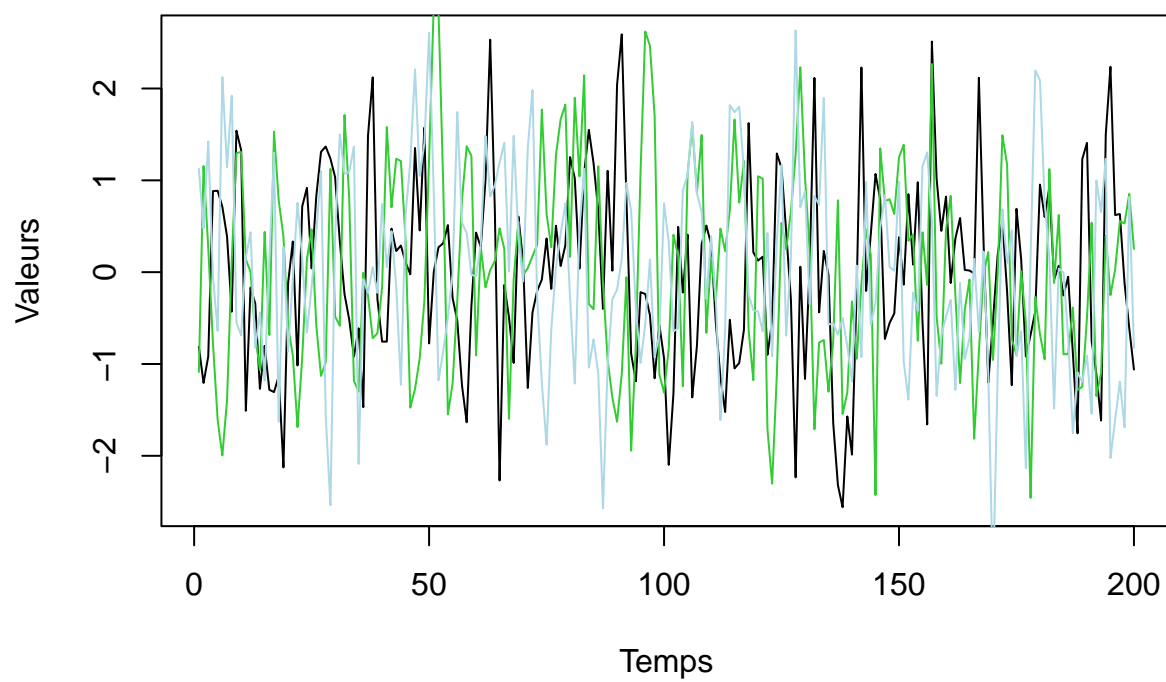
$$\begin{aligned} 1 - 0.25r &= 0 \\ r &= 4 \end{aligned}$$

La racine est l'extérieur du cercle unité, on inverse et on trouve $r = 0.25$. Notre processus est donc un ARMA(0.25, 0.25).

```
set.seed(123)
e1 <- arima.sim(model = list(ma = 0.25, ar = 0.25), n = 200)
e2 <- arima.sim(model = list(ma = 0.25, ar = 0.25), n = 200)
e3 <- arima.sim(model = list(ma = 0.25, ar = 0.25), n = 200)

plot(e1, type="l", col="black", main="Trajectoires MA(1)", xlab = "Temps", ylab = "Valeurs")
lines(e2, col="limegreen")
lines(e3, col="lightblue")
```

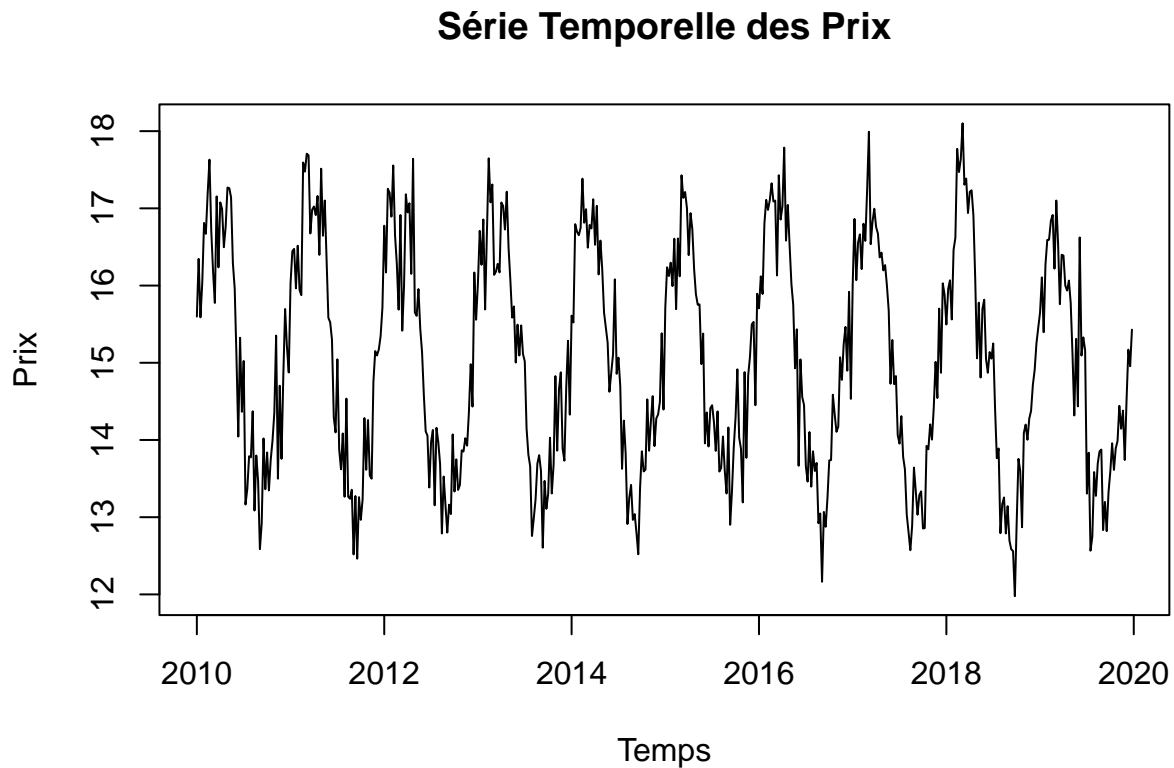

Trajectoires MA(1)



Exercice 2

a) Représentation graphique de la série p1

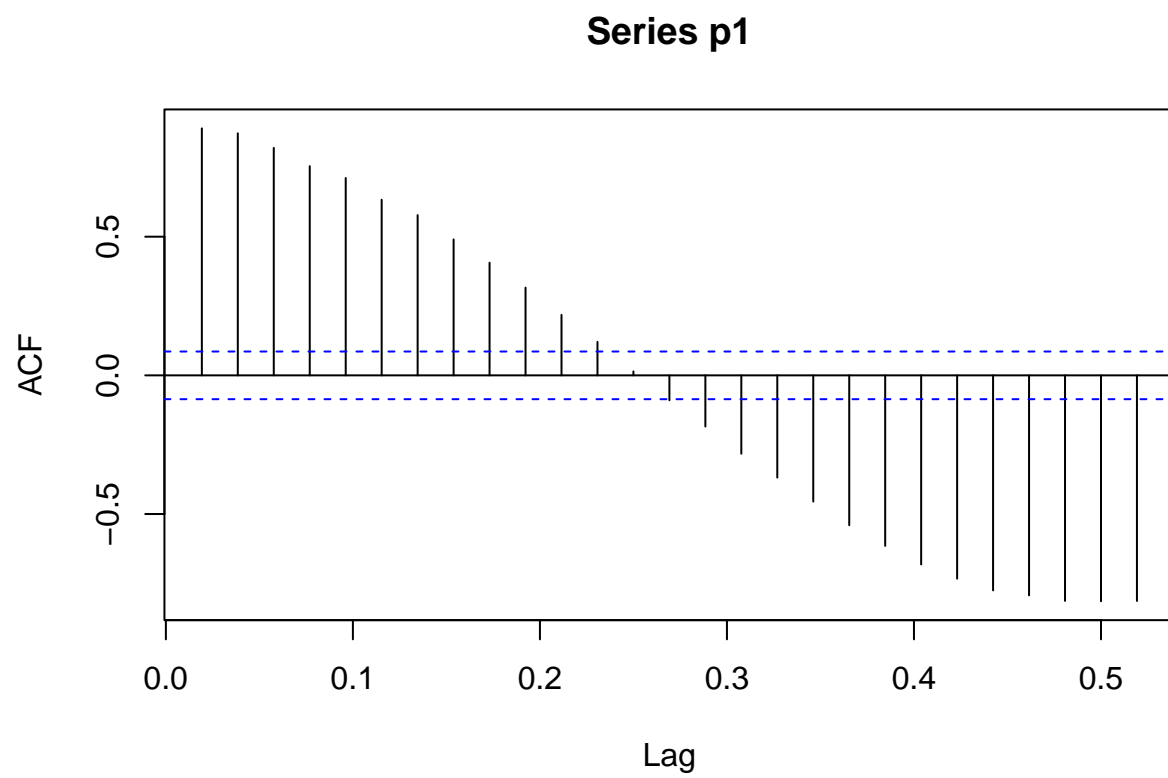
```
source("prix.R")  
plot(p1, type = "l", main = "Série Temporelle des Prix", xlab = "Temps", ylab = "Prix")
```



D'après le graphique, on peut émettre l'hypothèse que la série est stationnaire en moyenne et en variance puisque la moyenne est constante et la variance homoscédastique. De plus, la covariance entre 2 points semble constante, l'écart entre chaque fluctuation est le même et laisse penser à une saisonnalité. On peut cependant constater une légère tendance à la baisse à partir de 2019.

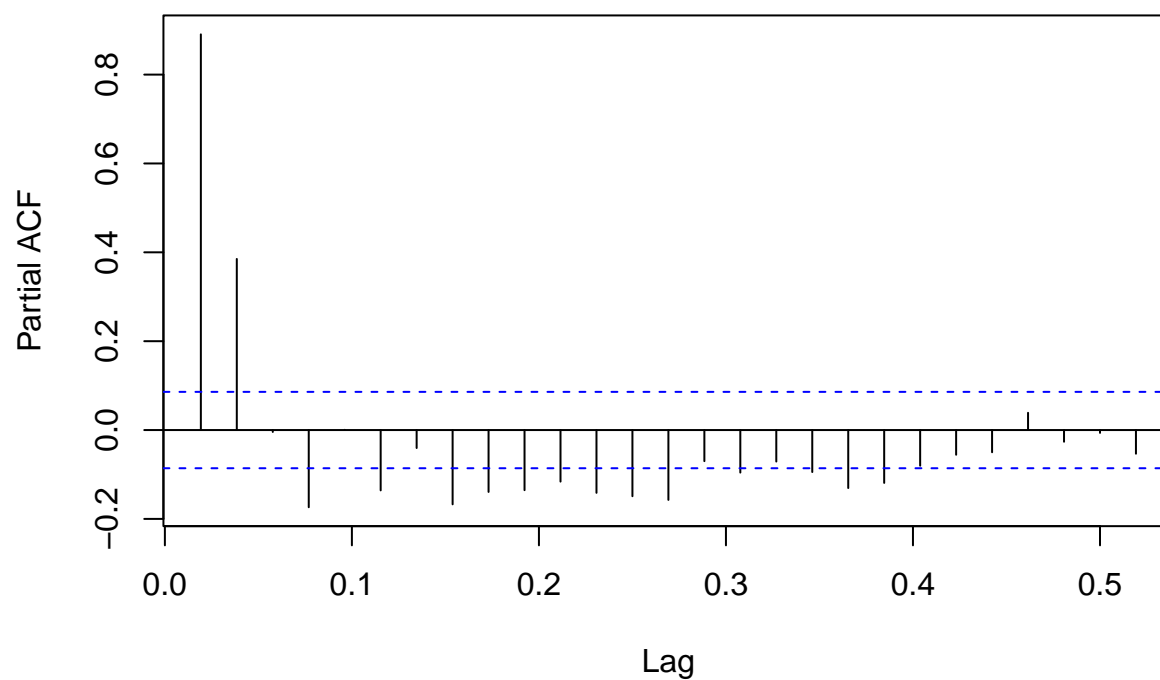
b) Fonctions d'autocorrélation acf et pacf

```
acf(p1)
```



```
pacf(p1)
```

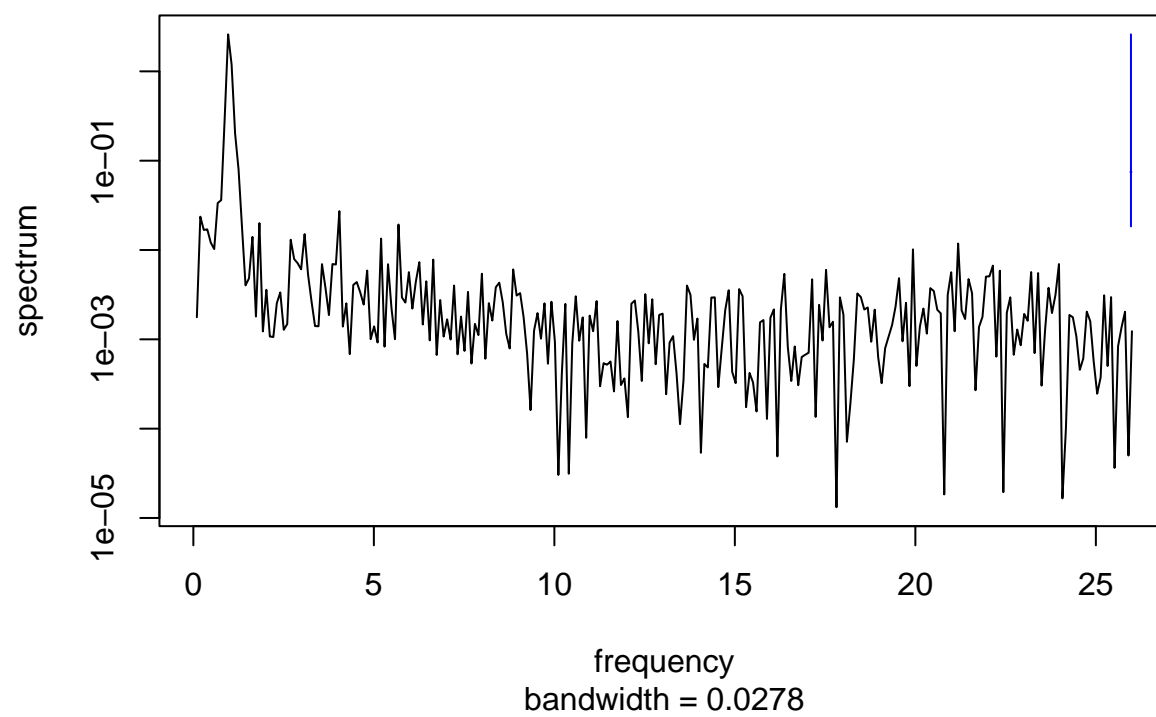
Series p1



```
spectrum(p1)
```

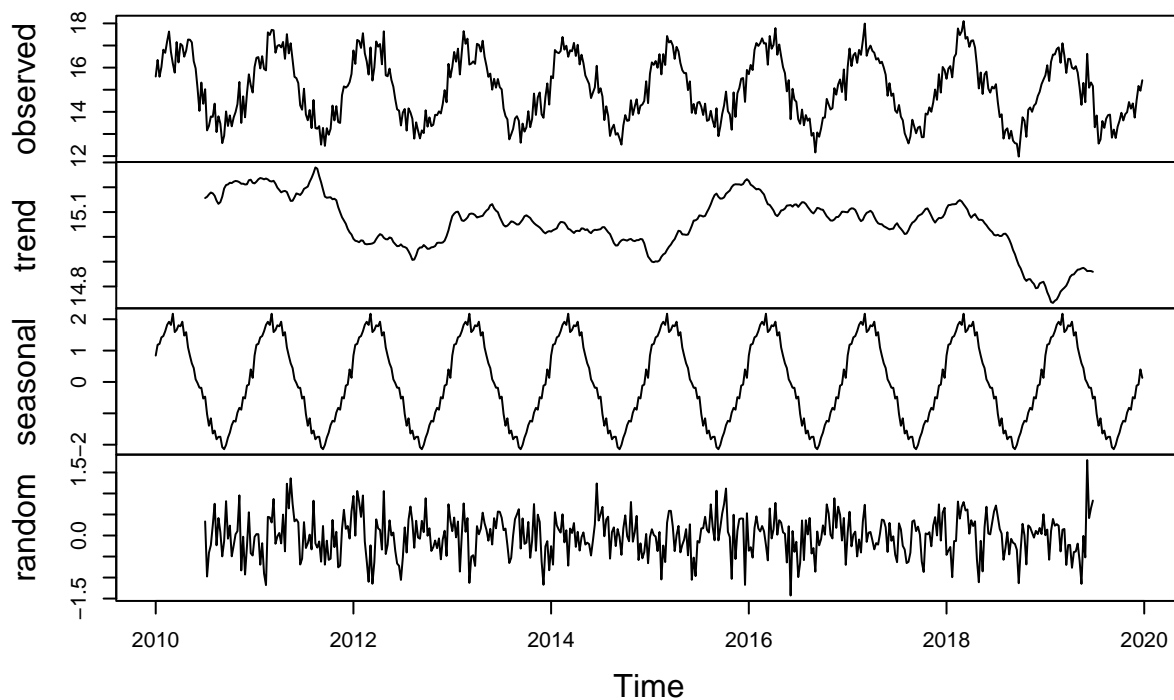
```
## Estimated spectral density
##   series: p1
##   method: Raw Periodogram
##   nseasons: 52
##   frequency range: (-26,26]
##
##   sort method for the table: decreasing magnitudes
##
##           freq      spec    % Total   Cum. %
## [1,] 0.9629630 2.59600166 0.500874477 0.5008745
## [2,] 1.0592593 1.19245040 0.230072260 0.7309467
## [3,] 0.8666667 0.29395235 0.056715384 0.7876621
## [4,] 1.1555556 0.20169648 0.038915467 0.8265776
## [5,] 1.2518519 0.07964283 0.015366346 0.8419439
## [6,] 0.7703704 0.03614575 0.006973988 0.8489179
```

Series: p1
Raw Periodogram



```
p1_decomp <- decompose(p1)
plot(p1_decomp)
```

Decomposition of additive time series



- ACF : On observe une autocorrelation qui s'apparente à un sinusóide, cela indique souvent la présence d'une racine unitaire ou d'une tendance dans la série temporelle, ce qui suggère que la série n'est pas stationnaire.
- PACF : Comme le deuxième lag ne se situe pas dans la borne de confiance, on peut dire que notre processus n'est pas un bruit blanc.
- Périodogramme : Il nous indique par la présence des pics que la série temporelle a une variance significative, pouvant correspondre à des cycles ou saisonnalités.

- Décompose :

- Composante observée (en haut) : C'est la série temporelle originale qui combine les effets de la tendance, de la saisonnalité et des composantes aléatoires.
- Composante de tendance (deuxième graphique) : Cette ligne lisse montre comment la série évolue sur le long terme, sans tenir compte des fluctuations saisonnières. On peut voir que la tendance est relativement stable mais présente une légère baisse et hausse périodique.
- Composante saisonnière (troisième graphique) : Cela montre un motif clair et répétitif qui semble se reproduire chaque année. La régularité et l'amplitude des pics et des creux indiquent que la série temporelle est influencée par des facteurs saisonniers, confirmant l'hypothèse faite en a).
- Composante aléatoire (en bas) : Ce qui reste après avoir retiré la tendance et la saisonnalité de la série observée. Dans une série temporelle bien modélisée, cette composante devrait ressembler à un bruit blanc.

$$c) X_t = m + A \sin\left(\frac{2\pi t}{T}\right) + B \cos\left(\frac{2\pi t}{T}\right) + Y_t$$

On choisit un T de 52 correspondant au nombre de semaines.

```
T <- 52
t <- 1:length(p1)

sin_terme <- sin(2 * pi * t / T)
```

```

cos_terme <- cos(2 * pi * t / T)

reg_modele <- lm(p1 ~ sin_terme + cos_terme)

m <- coef(reg_modele)[1]
A <- coef(reg_modele)[2]
B <- coef(reg_modele)[3]

Zt <- reg_modele$residuals - mean(reg_modele$residuals)

adf_test <- adf.test(Zt)
summary(reg_modele)

##
## Call:
## lm(formula = p1 ~ sin_terme + cos_terme)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.51534 -0.32911  0.01565  0.33530  1.46808
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.05321    0.02336  644.30  <2e-16 ***
## sin_terme    1.79372    0.03304   54.29  <2e-16 ***
## cos_terme    0.56812    0.03304   17.19  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5328 on 517 degrees of freedom
## Multiple R-squared:  0.8625, Adjusted R-squared:  0.862
## F-statistic: 1621 on 2 and 517 DF,  p-value: < 2.2e-16

```

```
adf_test
```

```

##
## Augmented Dickey-Fuller Test
##
## data:  Zt
## Dickey-Fuller = -7.3263, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary

```

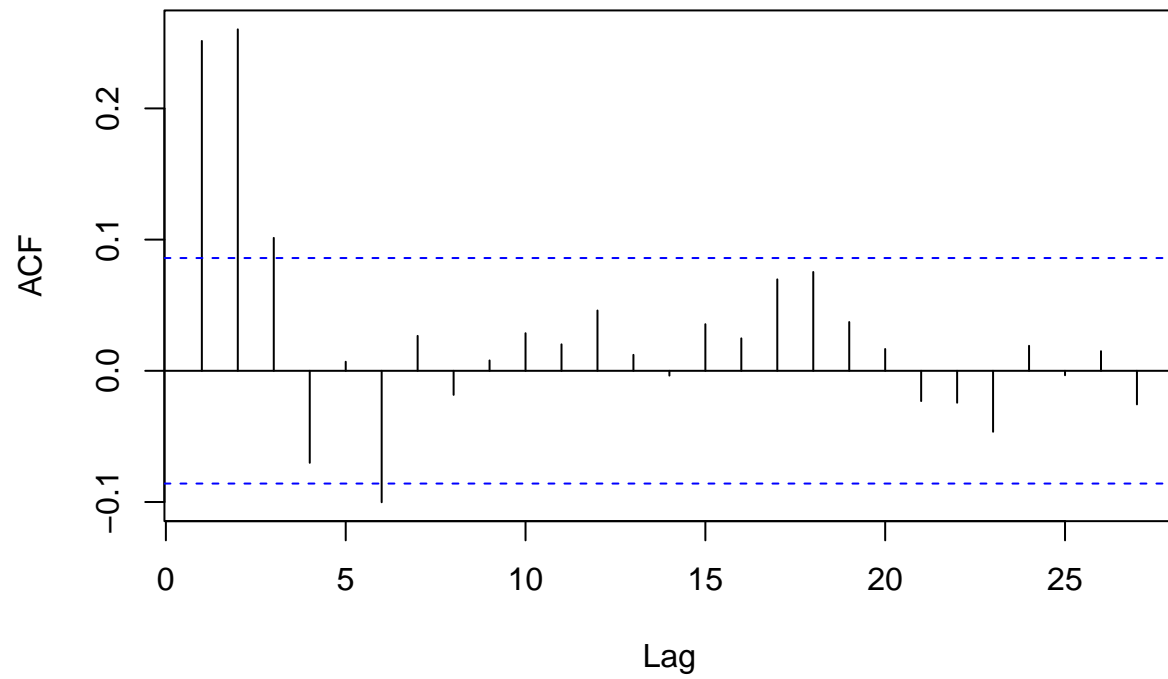
On remarque que tous les coefficients sont significatifs. La composante saisonnière sinusoïdale est donc une caractéristique importante de la série $p1$. L'écart type résiduel est faible (0.5328) et le coefficient de détermination ajusté est plutôt élevé (0.862), le modèle semble donc expliqué une grande partie de la variance des données.

Le p.value du test de Dickey-Fuller étant inférieur au seuil de 5%, on rejette l'hypothèse nulle d'une racine unitaire. Les résidus Z_t du modèle sont stationnaires.

d) Choix du modèle

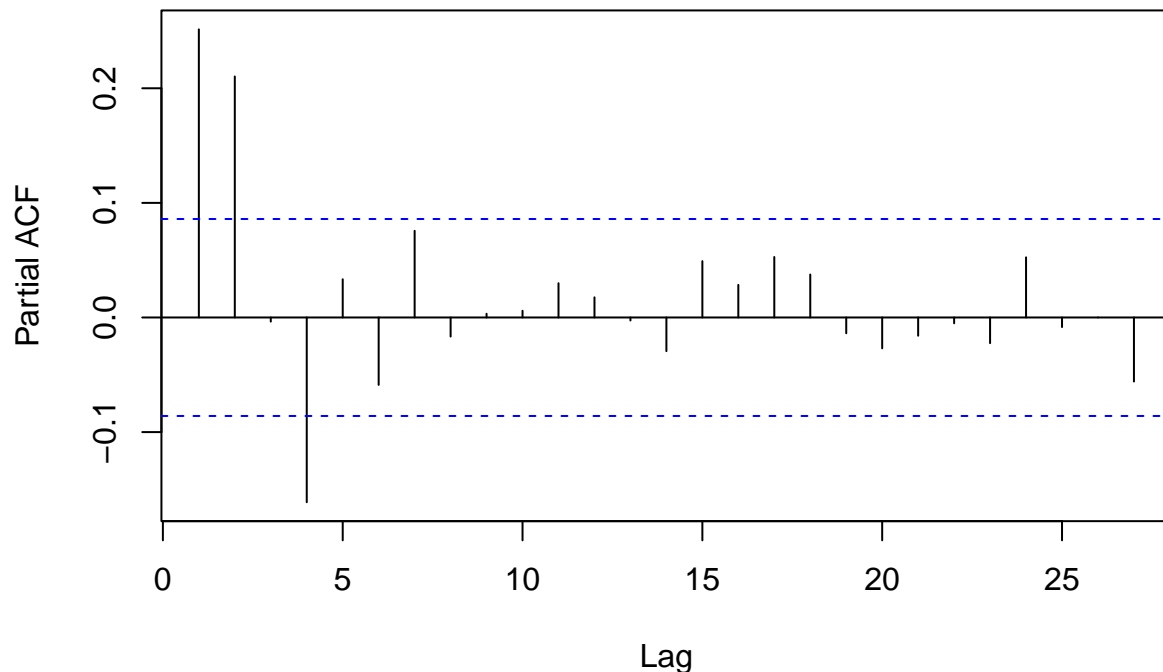
```
acf(Zt)
```

Series Zt



`pacf(Zt)`

Series Zt



```
Box.test(Zt, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: Zt
## X-squared = 33.062, df = 1, p-value = 8.928e-09
```

Z_t n'est pas un bruit blanc puisqu'il y a des autocorrelations significatives sur l'acf comme sur le pacf, suggérant que notre modèle contiendra une partie AR et une partie MA. Sur la partie AR, le dernier retard est atteint en lag = 4 et pour la partie MA, le dernier retard est atteint en lag = 6. En complément, nous effectuons un test de Ljung-Box afin de confirmer cette hypothèse. Effectivement, sa p.value supérieure à 0.05 confirme l'hypothèse que ce n'est pas un bruit blanc.

Nous allons ainsi tester par le moyen d'une boucle tous les modèles en faisant varier la partie AR de 0 à 4 et la partie MA de 0 à 6. Pour chaque modèle, on vérifiera les t.values associées à chaque coefficients *coef/s.e.* en considérant que si sa valeur est inférieure à 1.96, on considérera la t.value comme non significative.

```
ordres <- expand.grid(ar = 0:4, ma = 0:6)[-1, ]
modeles <- list()

for (i in seq(nrow(ordres))) {
  modeles[[i]] <- Arima(Zt, order = c(ordres[i, "ar"], 0, ordres[i, "ma"]))
}

resultats <- data.frame(AR = rep(NA, nrow(ordres)),
                        MA = rep(NA, nrow(ordres)),
                        AIC = rep(NA, nrow(ordres)),
```

```

                                stringsAsFactors = FALSE)
for (i in seq(nrow(ordres))) {
  model <- modeles[[i]]
  coefs <- model$coef
  se_coefs <- sqrt(diag(vcov(model)))
  t.values <- abs(coefs / se_coefs)
  compteur_t.values <- sum(t.values < 1.96) - 1

  resultats[i, "AR"] <- ordres[i, "ar"]
  resultats[i, "MA"] <- ordres[i, "ma"]
  resultats[i, "AIC"] <- AIC(model)
  resultats[i, "t.value non significatives"] <- compteur_t.values
}

kable(resultats, booktabs = TRUE) %>%
  kable_styling(latex_options = c("HOLD_position"))

```

| AR | MA | AIC | t.value non significatives |
|----|----|----------|----------------------------|
| 1 | 0 | 789.9238 | 0 |
| 2 | 0 | 768.4178 | 0 |
| 3 | 0 | 770.4101 | 1 |
| 4 | 0 | 758.7975 | 1 |
| 0 | 1 | 800.4477 | 0 |
| 1 | 1 | 778.3851 | 0 |
| 2 | 1 | 770.4147 | 2 |
| 3 | 1 | 763.2294 | 0 |
| 4 | 1 | 757.8453 | 0 |
| 0 | 2 | 767.7087 | 0 |
| 1 | 2 | 763.7444 | 1 |
| 2 | 2 | 763.1060 | 1 |
| 3 | 2 | 765.1027 | 3 |
| 4 | 2 | 758.4314 | 4 |
| 0 | 3 | 760.8459 | 0 |
| 1 | 3 | 757.8281 | 0 |
| 2 | 3 | 758.9232 | 2 |
| 3 | 3 | 760.7775 | 4 |
| 4 | 3 | 760.2288 | 5 |
| 0 | 4 | 761.8718 | 1 |
| 1 | 4 | 759.1082 | 1 |
| 2 | 4 | 760.9157 | 6 |
| 3 | 4 | 762.2010 | 6 |
| 4 | 4 | 761.7376 | 8 |
| 0 | 5 | 762.9353 | 2 |
| 1 | 5 | 760.2818 | 2 |
| 2 | 5 | 763.0230 | 7 |
| 3 | 5 | 759.6398 | 2 |
| 4 | 5 | 763.2266 | 8 |
| 0 | 6 | 758.3870 | 2 |
| 1 | 6 | 759.3247 | 5 |
| 2 | 6 | 761.3083 | 8 |
| 3 | 6 | 763.1383 | 9 |
| 4 | 6 | 765.0393 | 10 |

Il y a deux modèles qui se distinguent clairement des autres : ARMA(1,3) et l'ARMA(4,1). Ils ont toutes leurs t.values significatives et ont un AIC plus faible que les autres. On gardera l'ARMA(1,3) car son AIC est un peu plus faible que celui de l'ARMA(4,1), bien qu'on puisse considérer les deux modèles comme équivalents du point de vue de ce critère.

e) Prévision pour le mois de janvier

Pour la fonction forecast, on choisit un $h = 4$, correspondant aux 4 premières semaines de janvier.

```
arma_xt <- Arima(Zt, order=c(1,0,3))
Zt_forecast <- forecast(arma_xt, h=4)
```

On doit ensuite reconstruire les prédictions pour X_t en ajoutant la trend et la saisonnalité.

```

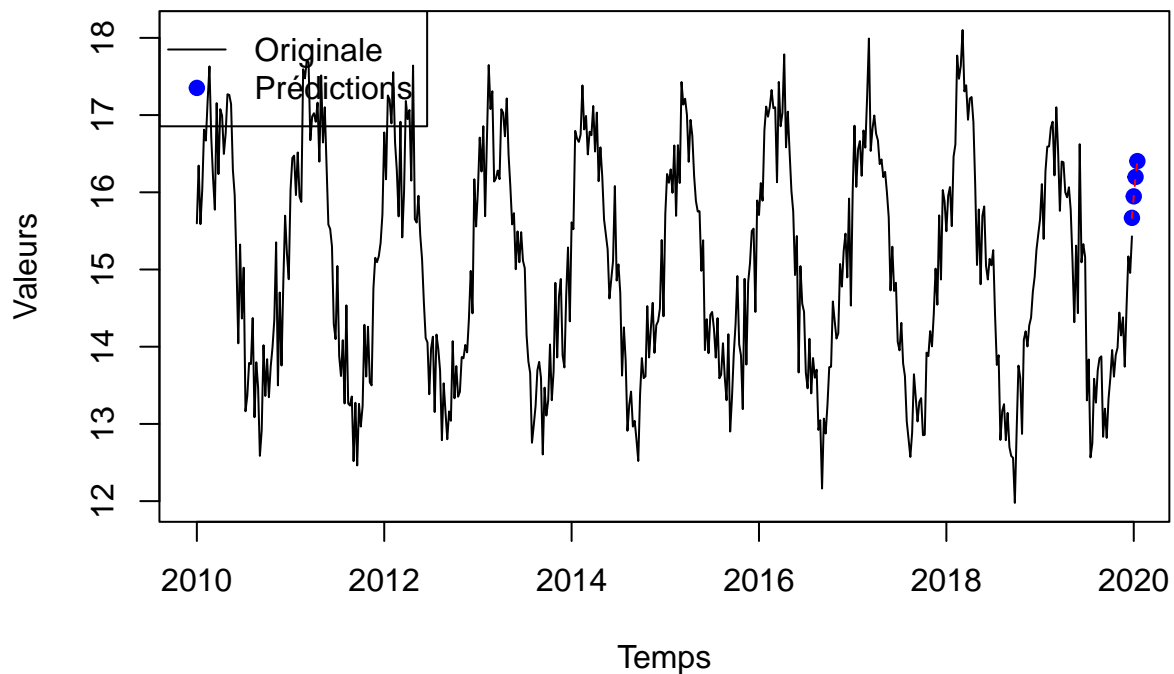
t_future <- length(p1) + 1:length(Zt_forecast$mean)
sin_term_future <- sin(2 * pi * t_future / 52)
cos_term_future <- cos(2 * pi * t_future / 52)
Xt_future <- m + A * sin_term_future + B * cos_term_future + Zt_forecast$mean
Xt_future

## Time Series:
## Start = 521
## End = 524
## Frequency = 1
## [1] 15.66913 15.94741 16.19854 16.40389

time_future <- seq(max(time(p1)), by = 1/52, length.out = 4)
plot(p1, main = "Données originales avec prédictions", xlab = "Temps", ylab = "Valeurs", type = "l")
points(time_future, Xt_future, col = "blue", pch = 19)
lines(time_future, Xt_future, col = "red", lty = 2)
legend("topleft", legend = c("Originale", "Prédictions"), col = c("black", "blue"), pch = c(NA, 19), lty = c(1, 2))

```

Données originales avec prédictions



Les prédictions semblent suivre la tendance, continuant la hausse en cours au mois de décembre.

f) SARIMA

```

sarima_Xt <- auto.arima(p1)
summary(sarima_Xt)

## Series: p1
## ARIMA(0,0,3)(2,1,0)[52] with drift

```

```
##
## Coefficients:
##      ma1      ma2      ma3      sar1      sar2      drift
##      0.1950 0.2721 0.1673 -0.6107 -0.3304 -5e-04
## s.e. 0.0464 0.0451 0.0523 0.0474 0.0496 5e-04
##
## sigma^2 = 0.3332: log likelihood = -416.16
## AIC=846.33 AICc=846.57 BIC=875.37
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.0003056615 0.5441267 0.4104175 -0.08892772 2.749445 0.7177345
##              ACF1
## Training set -0.003545234
```

La fonction `auto.arima()` choisi le modèle $SARIMA_{52}[(0,0,3)(2,1,0)]$. Ainsi, il choisi 3 composantes MA pour la partie non saisonnière et 2 composantes AR ainsi qu'une différenciation pour la partie saisonnière.

```
january_forecast <- forecast(sarima_Xt, h=4)
january_forecast
```

```
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2020.000      15.67229 14.93248 16.41209 14.54085 16.80372
## 2020.019      16.22468 15.47095 16.97842 15.07194 17.37742
## 2020.038      15.79242 15.01225 16.57258 14.59926 16.98557
## 2020.058      16.11875 15.32883 16.90867 14.91067 17.32683
```

g) Application de l'opérateur de différenciation $(1 - B^{52})$

```
p1_diff <- diff(p1, lag = 52)
eacf(p1_diff)
```

```
## AR/MA
##  0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x o o o o o o o o o o o
## 1 x x x o o o o o o o o o o o
## 2 x x x x o o o o o o o o o o
## 3 x x x o o o o o o o o o o o
## 4 x o x o x o o o o o o o o o
## 5 x o x o o o o o o o o o o o
## 6 x x x x o x o o o o o o o o
## 7 x x x x o o o o o o o o o o
```

```
arma_model_diff <- auto.arima(p1_diff, seasonal = FALSE)
summary(arma_model_diff)
```

```
## Series: p1_diff
## ARIMA(1,0,3) with zero mean
##
## Coefficients:
##      ar1      ma1      ma2      ma3
##      -0.5820 0.7645 0.3188 0.2628
## s.e. 0.1876 0.1859 0.0593 0.0446
##
## sigma^2 = 0.4662: log likelihood = -483.57
## AIC=977.13 AICc=977.26 BIC=997.88
```

```
##
## Training set error measures:
##           ME           RMSE           MAE           MPE           MAPE           MASE
## Training set -0.02222536 0.6798283 0.5490841 569.9611 690.0975 0.5611167
##           ACF1
## Training set 0.005028545
```

```
january_forecast_diff <- forecast(arma_model_diff, h=4)
kbl(january_forecast_diff, booktabs = TRUE) %>%
  kable_styling(latex_options = c("hold_position"))
```

| | Point Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
|----------|----------------|------------|-----------|-----------|----------|
| 2020.000 | -0.0662999 | -0.9412823 | 0.8086824 | -1.404470 | 1.271870 |
| 2020.019 | -0.0632689 | -0.9527012 | 0.8261635 | -1.423538 | 1.297001 |
| 2020.038 | -0.0019736 | -0.9106402 | 0.9066930 | -1.391659 | 1.387712 |
| 2020.058 | 0.0011487 | -0.9156305 | 0.9179279 | -1.400944 | 1.403242 |

```
AIC(arma_model_diff)
```

```
## [1] 977.133
```

Avec une différenciation de 52 sur un modèle de type ARMA, on revient au même résultat que pour Z_t . La méthode du coin et la fonction `auto.arima` considère $ARMA(1,3)$ comme le meilleur modèle.

h) Modèle le plus pertinent

Effectuer une comparaison d'AIC est pertinent pour choisir le meilleur modèle.

```
df <- data.frame(
  Modèle = character(0),
  AIC = numeric(0),
  MAE = numeric(0)
)

df <- rbind(df, data.frame(Modèle = "Modèle C", AIC = AIC(reg_modele), MAE = mean(abs(reg_modele$residuals))))
df <- rbind(df, data.frame(Modèle = "Modèle E", AIC = arma_xt$aic, MAE = mean(abs(arma_xt$residuals))))
df <- rbind(df, data.frame(Modèle = "Modèle F", AIC = sarima_Xt$aic, MAE = mean(abs(sarima_Xt$residuals))))
df <- rbind(df, data.frame(Modèle = "Modèle G", AIC = AIC(arma_model_diff), MAE = mean(abs(arma_model_diff$residuals))))

kable(df, booktabs = TRUE) %>%
  kable_styling(latex_options = c("HOLD_position"))
```

| Modèle | AIC | MAE |
|----------|----------|-----------|
| Modèle C | 825.8374 | 0.4154645 |
| Modèle E | 757.8281 | 0.3900134 |
| Modèle F | 846.3282 | 0.4104175 |
| Modèle G | 977.1330 | 0.5490841 |

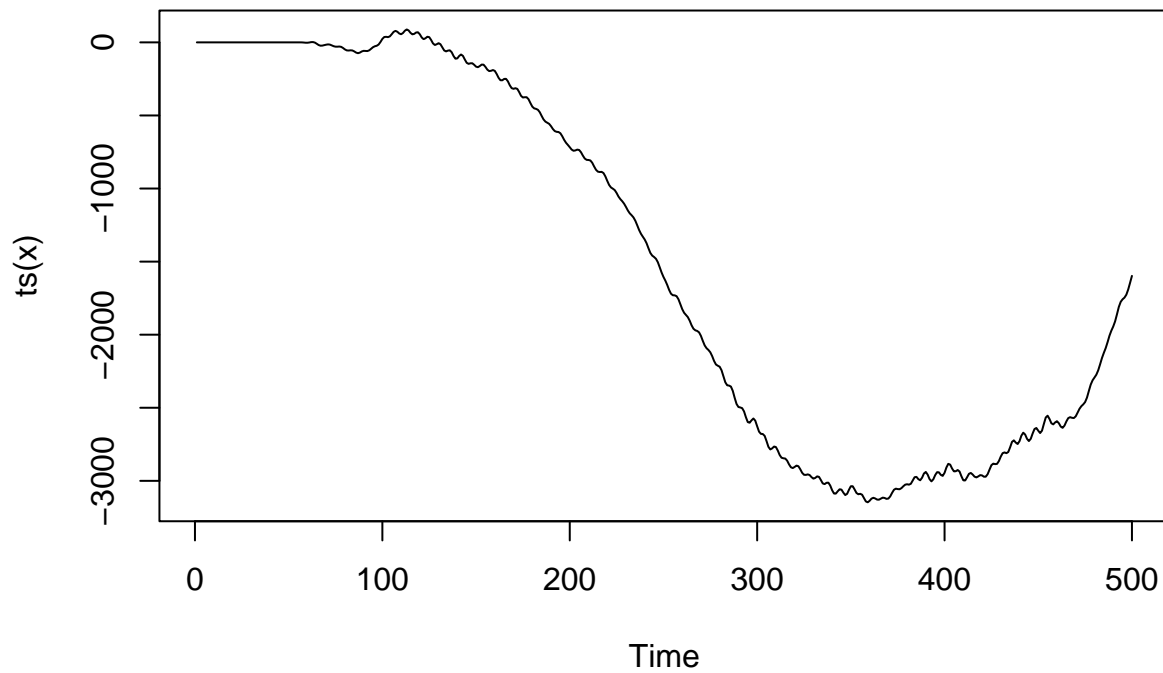
On choisit donc le modèle estimé en e, c'est celui qui a l'AIC le plus faible. De plus, c'est celui qui semble faire le moins d'erreurs de prédiction obtenant un $MAE = 0.4$. Cela signifie qu'en moyenne, les prédictions du modèle e s'écartent de 0.40€ du prix réel de la série p1.

i)

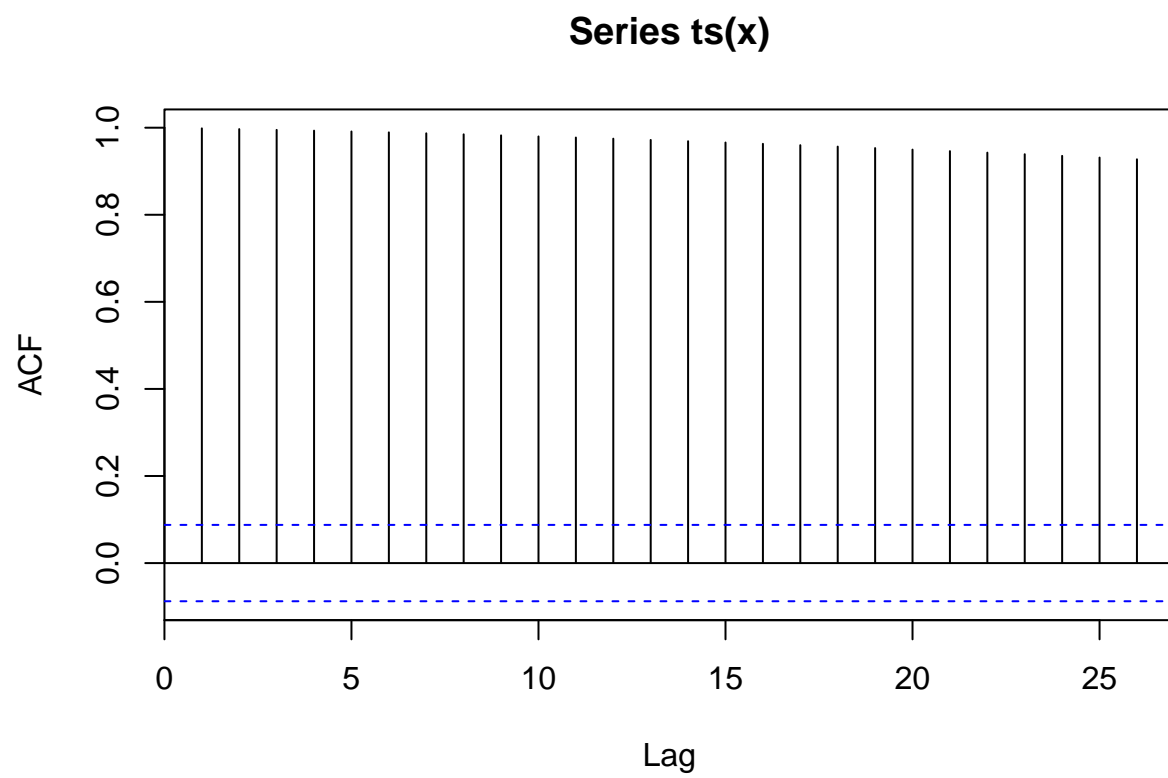
Forme réduite : $(1 - B^{52})(1 - 2B + B^2)(1 - 2B)X_t = (1 - B)\varepsilon_t$

```
set.seed(123)
x <- sim_sarima(n = 500,
               model = list(ar = c(2, -2, 1),
                           ma = 1,
                           iorder = 1,
                           siorder = 1,
                           nseasons = 52,
                           sigma2 = 1))
```

```
plot(ts(x))
```

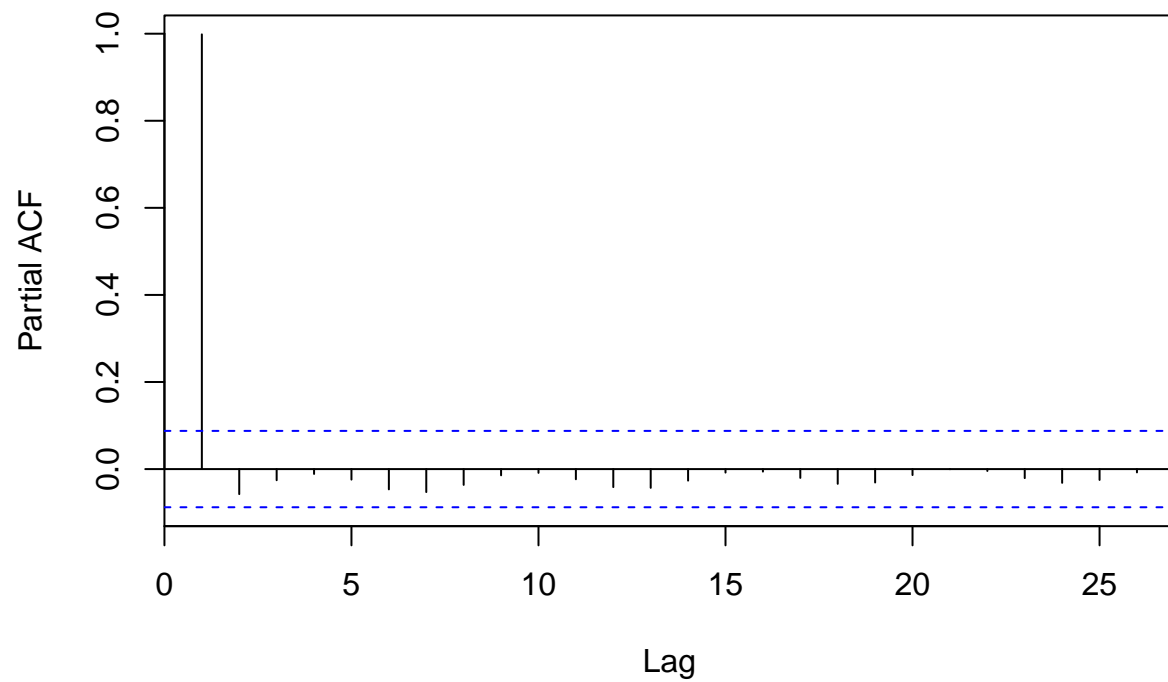


```
# Test de périodicité et stationnarité de  $X_t$ 
acf(ts(x))
```



```
pacf(ts(x))
```


Series ts(x)



```
adf.test(x)
```

```
## Warning in adf.test(x): p-value greater than printed p-value
```

```
##
```

```
## Augmented Dickey-Fuller Test
```

```
##
```

```
## data: x
```

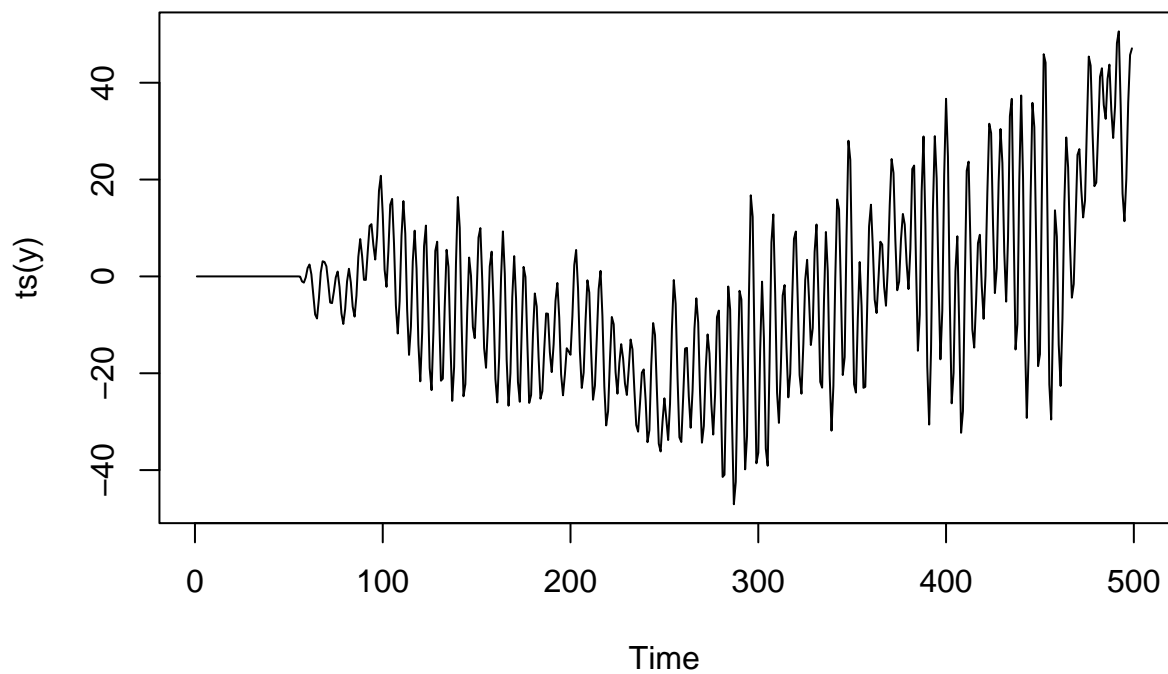
```
## Dickey-Fuller = 0.35621, Lag order = 7, p-value = 0.99
```

```
## alternative hypothesis: stationary
```

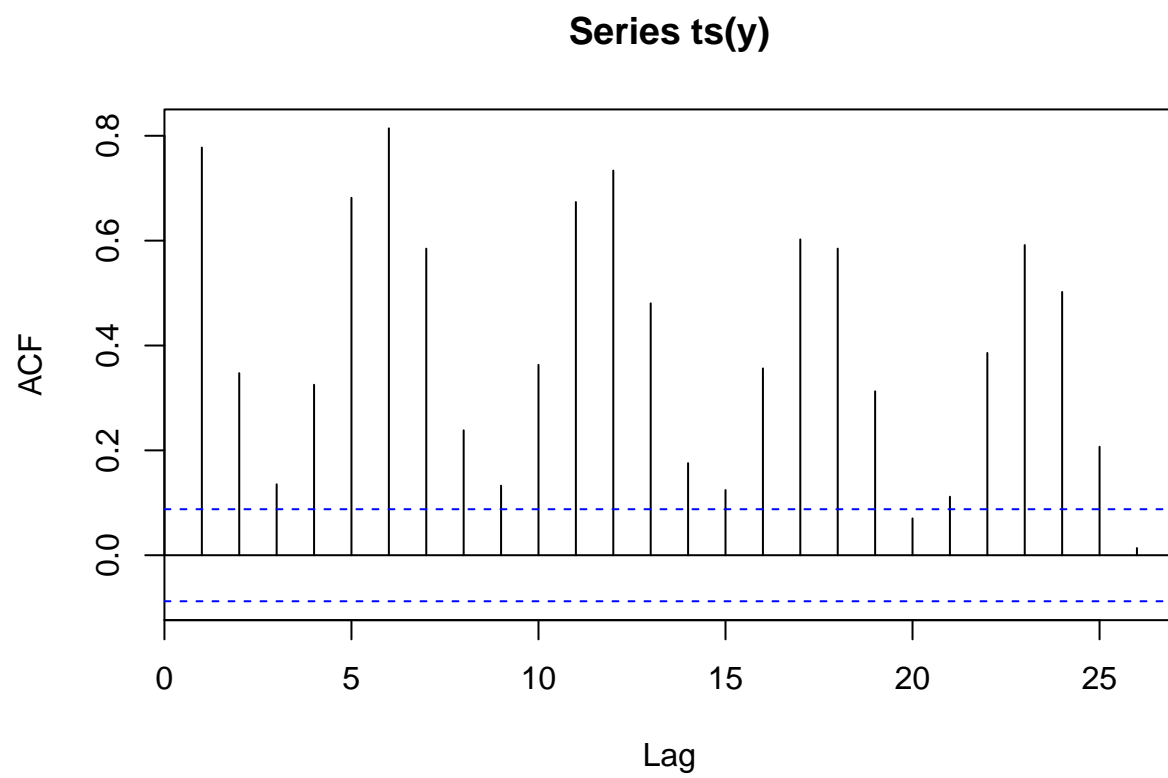
```
# Série différenciée de  $X_t$ ,  $X_t - X_{t-1}$ 
```

```
y <- diff(x,1)
```

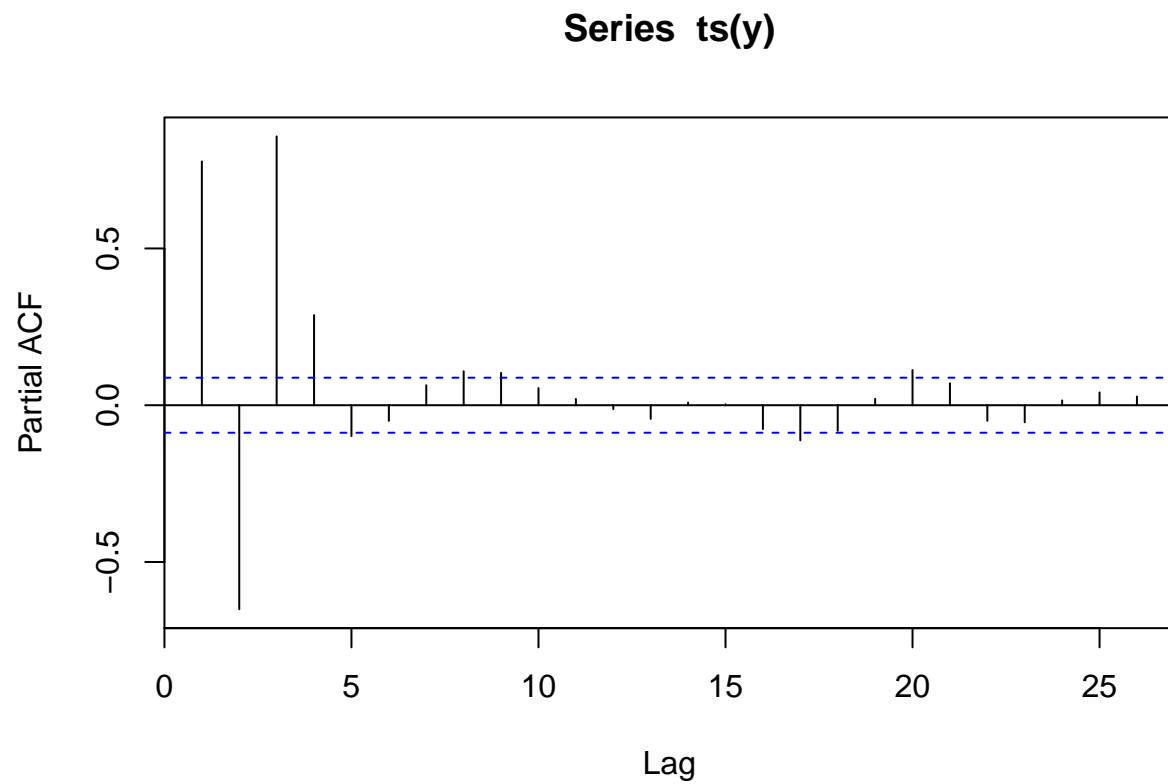
```
plot(ts(y))
```



```
# Test de périodicité et stationnarité de  $X_t - X_{t-1}$   
acf(ts(y))
```



```
pacf(ts(y))
```



```
adf.test(y)
```

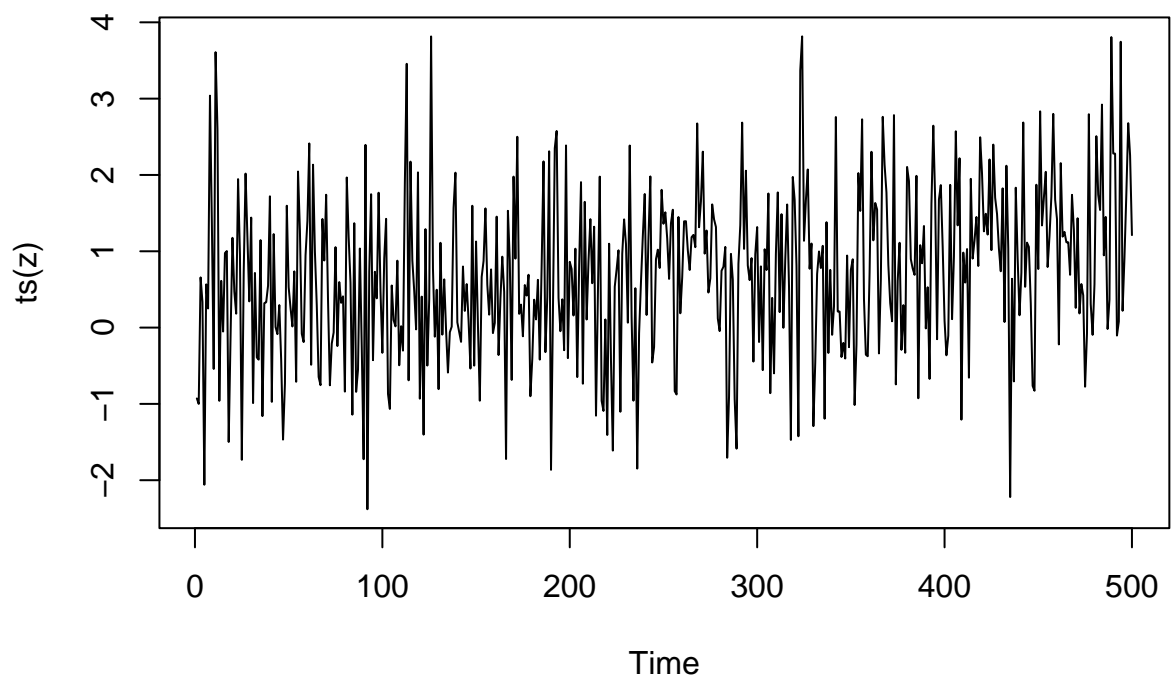
```
##  
## Augmented Dickey-Fuller Test  
##  
## data: y  
## Dickey-Fuller = -1.7088, Lag order = 7, p-value = 0.7016  
## alternative hypothesis: stationary
```

```
# Série Yt
```

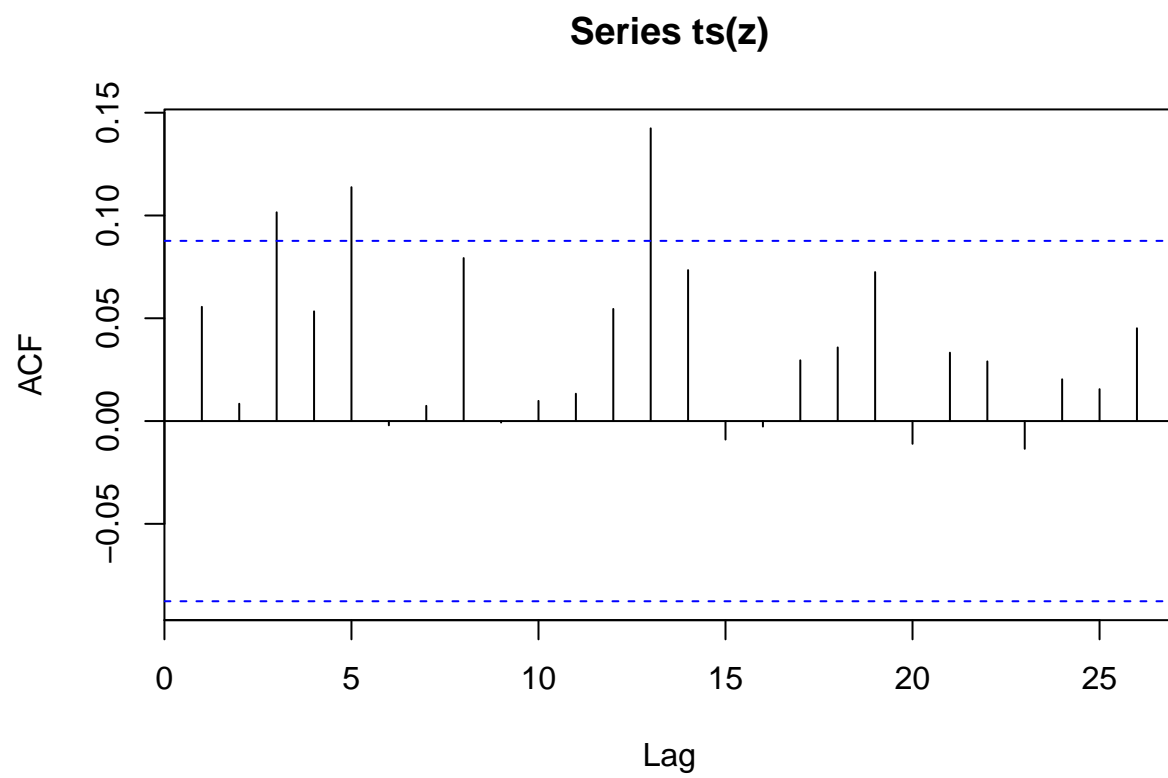
```
model_z <- Arima(ts(rnorm(500), frequency = 52), order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1)))
```

```
z <- simulate(model_z, nsim = 500)
```

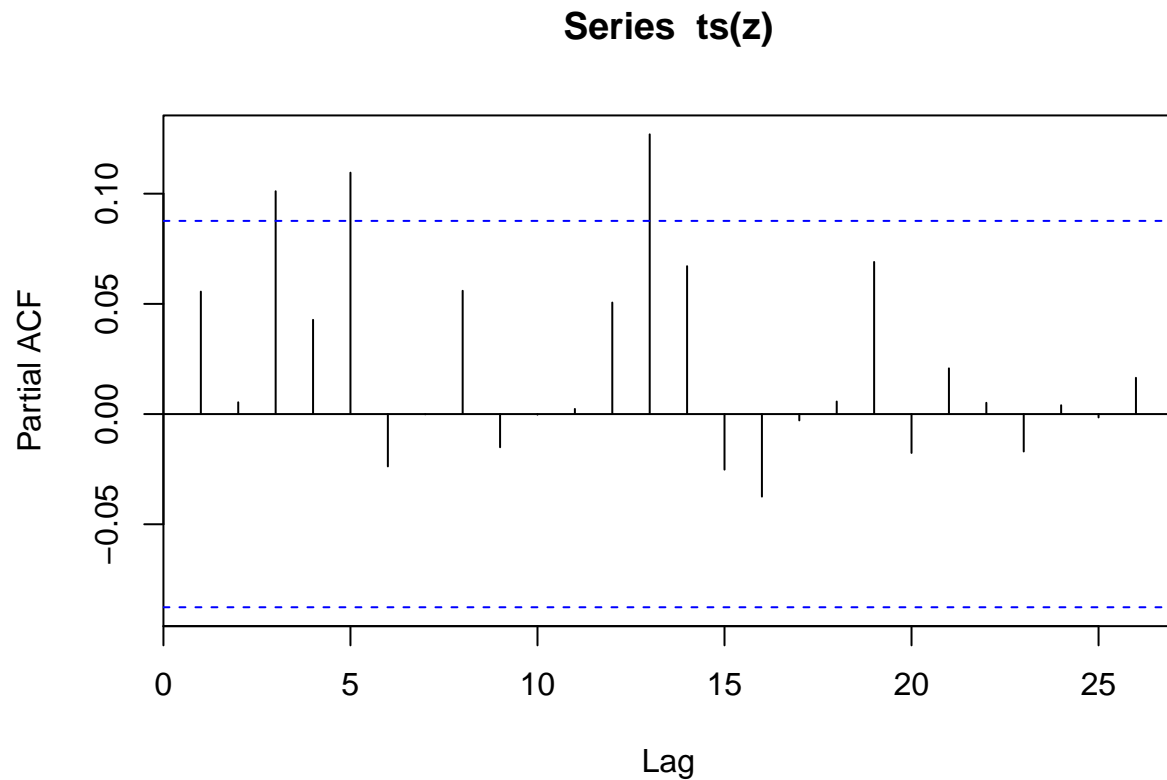
```
plot(ts(z))
```



```
acf(ts(z))
```



```
pacf(ts(z))
```



```
adf.test(z)
```

```
## Warning in adf.test(z): p-value smaller than printed p-value
##
## Augmented Dickey-Fuller Test
##
## data: z
## Dickey-Fuller = -7.7097, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

La série temporelle X_t n'est pas stationnaire, et cela reste vrai même après différenciation. Cependant, la série différenciée révèle une saisonnalité, comme on peut le voir dans le graphique de la fonction d'autocorrélation (ACF), où des cycles de significativité apparaissent à certains décalages (lags). En revanche, la série Y_t semble être stationnaire. Ceci est indiqué par le fait que l'autocorrélation de ses décalages, observée à la fois dans l'ACF et dans la fonction d'autocorrélation partielle (PACF), est très faible. De plus, le test de Dickey-Fuller vient confirmer cette hypothèse en rejetant H_0 qui suppose que Y_t est non-stationnaire.

j) k)

```
eacf(z)
```

```
## AR/MA
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o x o x o o o o o o o x o
## 1 x o o o x o o o o o o o x o
## 2 o x o o x o o o o o o o o o
## 3 x x x o o o o o o o o o o o
```

```
## 4 x x x o x o o o o o o o o o
## 5 x o x x o o o o o o o o o o
## 6 x o x o o o o o o o o o o o
## 7 o x x o o o o o o o o o o o

modele_ar <- Arima(z, order = c(0, 0, 0))
modele_ma <- Arima(z, order = c(0, 0, 0))
modele_arma <- Arima(z, order = c(0, 0, 0))

forecast_1 <- forecast(modele_arma, h = 1)
forecast_2 <- forecast(modele_arma, h = 2)

print(forecast_1)
```

| ## | Point Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
|-------------|----------------|------------|----------|-----------|----------|
| ## 20.23077 | 0.686162 | -0.7168053 | 2.089129 | -1.459491 | 2.831815 |

```
print(forecast_2)
```

| ## | Point Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
|-------------|----------------|------------|----------|-----------|----------|
| ## 20.23077 | 0.686162 | -0.7168053 | 2.089129 | -1.459491 | 2.831815 |
| ## 20.25000 | 0.686162 | -0.7168053 | 2.089129 | -1.459491 | 2.831815 |

On choisit le modèle AR(0), MA(0) et donc ARMA(0,0). On en conclut que c'est un bruit blanc. Notre hypothèse est que Concernant la question j), nous avons déterminé que le modèle est AR(0) et MA(0), alors les formes $AR(\infty)$ et $MA(\infty)$ sont simplement le terme constant, car il n'y a pas de termes dépendant des valeurs passées ou des erreurs passées.

l)

```
diff_ordre <- 1
diff_order_saison <- 1
periode <- 52
Y_t <- diff(diff(x, differences = diff_ordre),
            differences = diff_order_saison,
            lag = periode)

model_Y <- Arima(Y_t, order = c(1, 0, 0))

phi <- model_Y$coef[1]

n_simulations <- 999
n <- 500

forecasts_501 <- numeric(n_simulations)
forecasts_502 <- numeric(n_simulations)

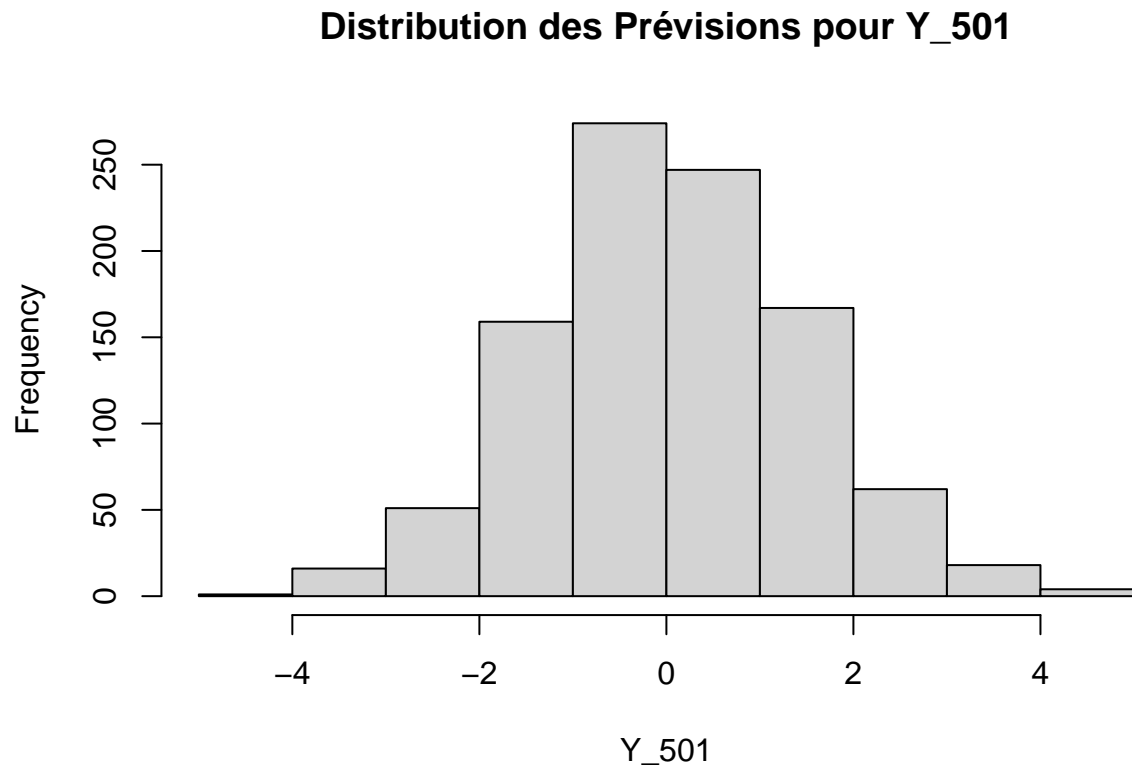
for (i in 1:n_simulations) {
  series <- arima.sim(model = list(ar = c(phi)), n = n)
  fit <- Arima(series, order = c(1,0,0))
  forecast <- forecast(fit, h = 2)

  forecasts_501[i] <- forecast$mean[1]
  forecasts_502[i] <- forecast$mean[2]
}
```



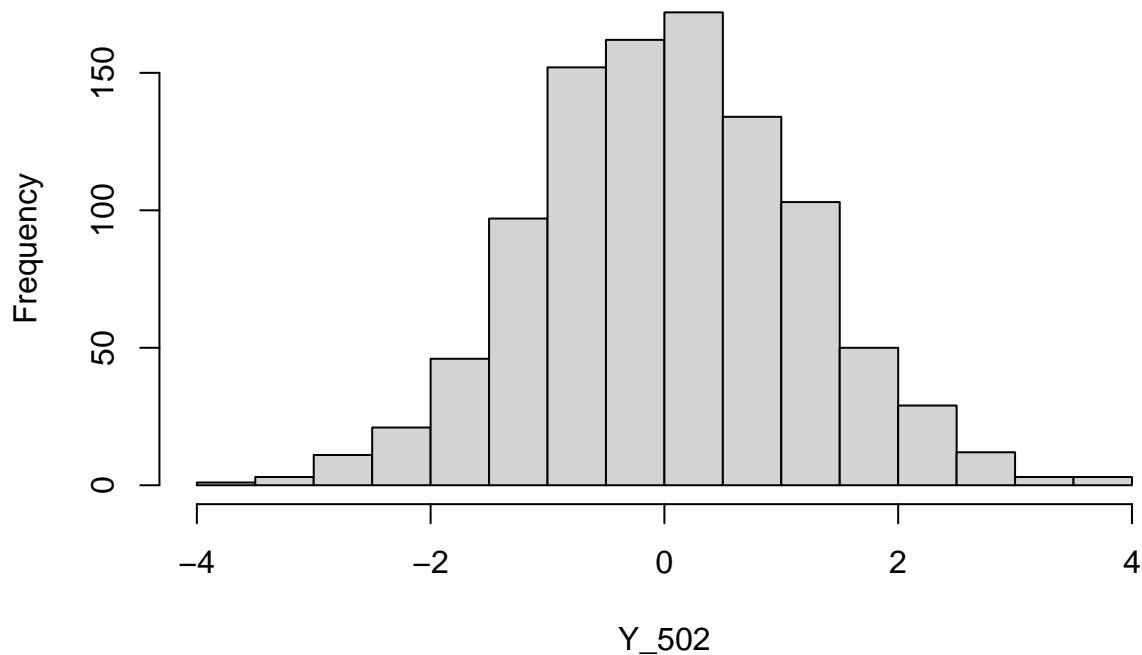
```
mean_501 <- mean(forecasts_501)
mean_502 <- mean(forecasts_502)
sd_501 <- sd(forecasts_501)
sd_502 <- sd(forecasts_502)

hist(forecasts_501, main = "Distribution des Prévisions pour Y_501", xlab = "Y_501")
```



```
hist(forecasts_502, main = "Distribution des Prévisions pour Y_502", xlab = "Y_502")
```

Distribution des Prévisions pour Y_502



```
print(paste("Moyenne des prévisions pour Y_501:", mean_501, "et écart-type:", sd_501))
```

```
## [1] "Moyenne des prévisions pour Y_501: 0.045192579853745 et écart-type: 1.4105194857378"
```

```
print(paste("Moyenne des prévisions pour Y_502:", mean_502, "et écart-type:", sd_502))
```

```
## [1] "Moyenne des prévisions pour Y_502: 0.0356324848956562 et écart-type: 1.14379607318929"
```

Les distributions des deux prévisions sont normales et centrées à zéro, notre modèle AR(1) prédit donc en moyenne une valeur proche de 0 pour les pas futurs. Les valeurs des écarts-types montrent que les prévisions se situent relativement proche de la moyenne. De plus, ces écarts types sont assez faibles, justifiant que les prédictions sont assez précises.