

Script

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 :

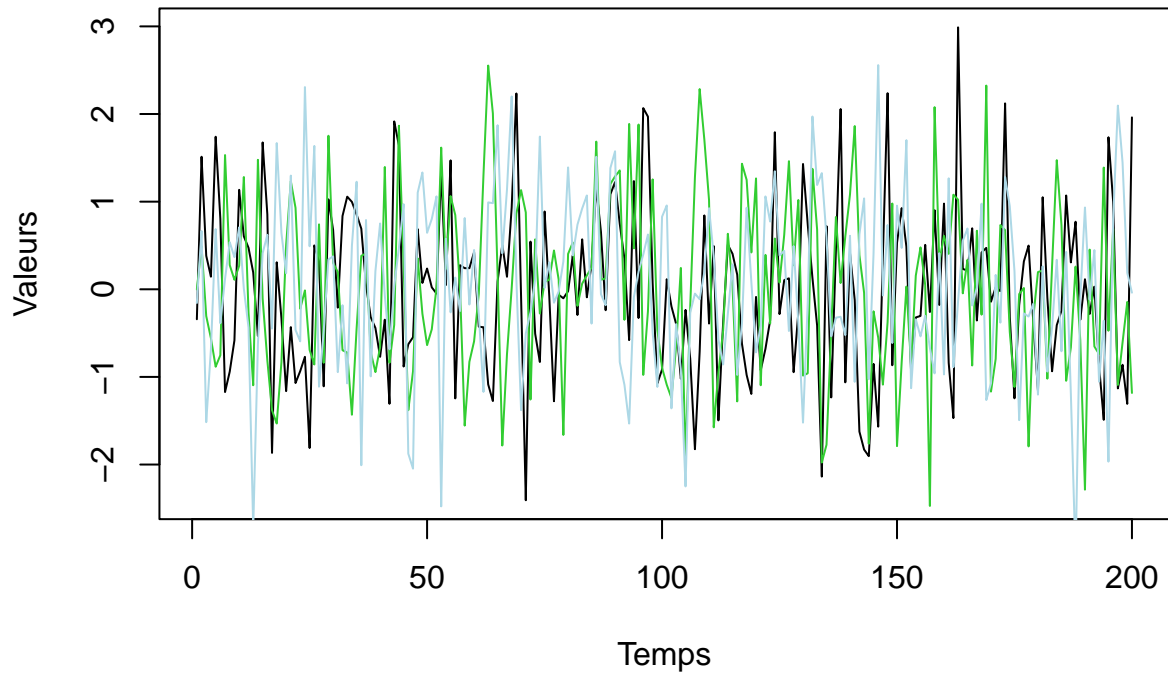
$$Y_t = X_t - \sin(t) = \varepsilon_t - 0.2 \times \varepsilon_{t-1}.$$

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 :

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

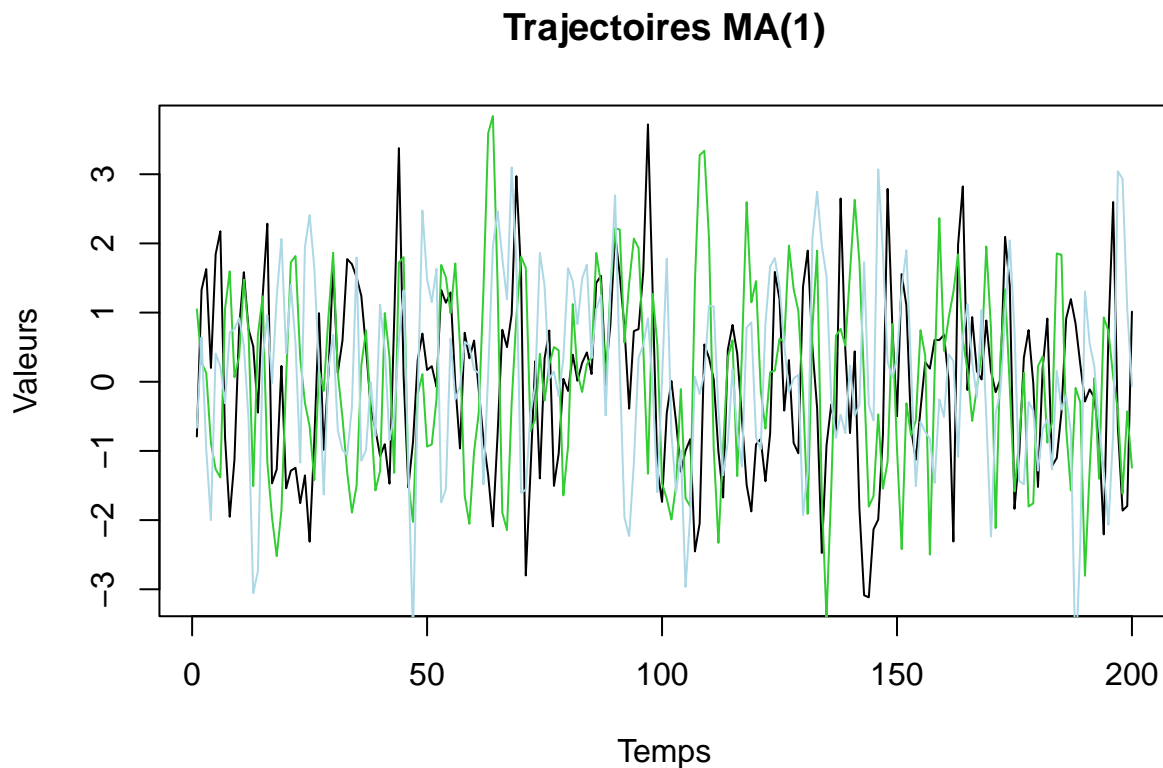
Le processus $X_t = \varepsilon_t - \varepsilon_{t-1}$ est stationnaire au second ordre car son espérance est constante et sa fonction d'autocovariance ne dépend que de h et non de t .

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 indépendantes, 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 .

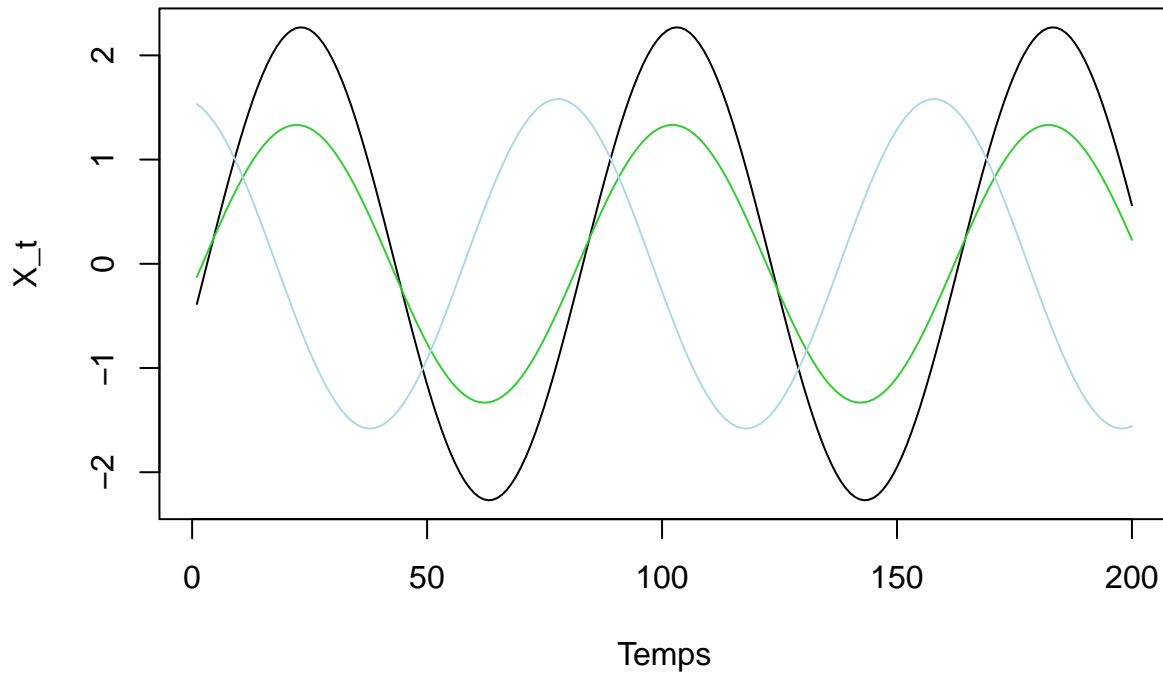
```
set.seed(123)
n <- 200
omega <- pi/40
A <- rnorm(n)
B <- rnorm(n)

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 je prends un ω proche de 0 mais $\neq 0$, la trajectoire de X_t s'apparentera à celle d'un processus non-stationnaire.

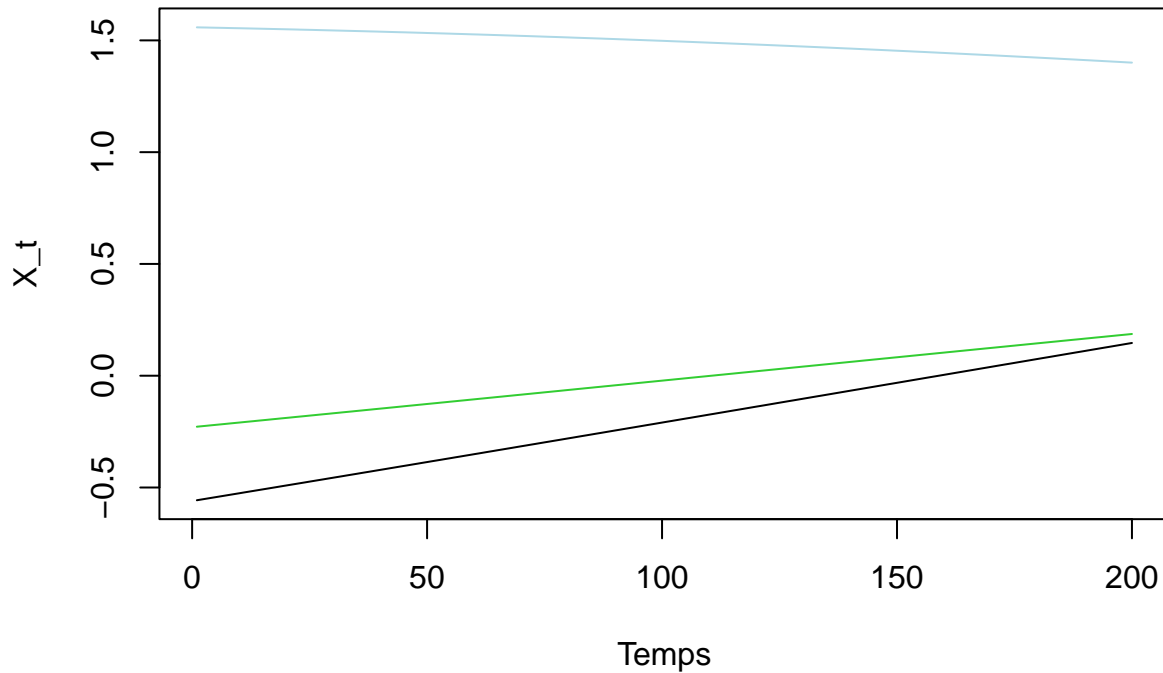
```
set.seed(123)
n <- 200
omega <- pi/2000
A <- rnorm(n)
B <- rnorm(n)

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 $r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

Le processus n'est donc pas stationnaire puisque les racines sont 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)

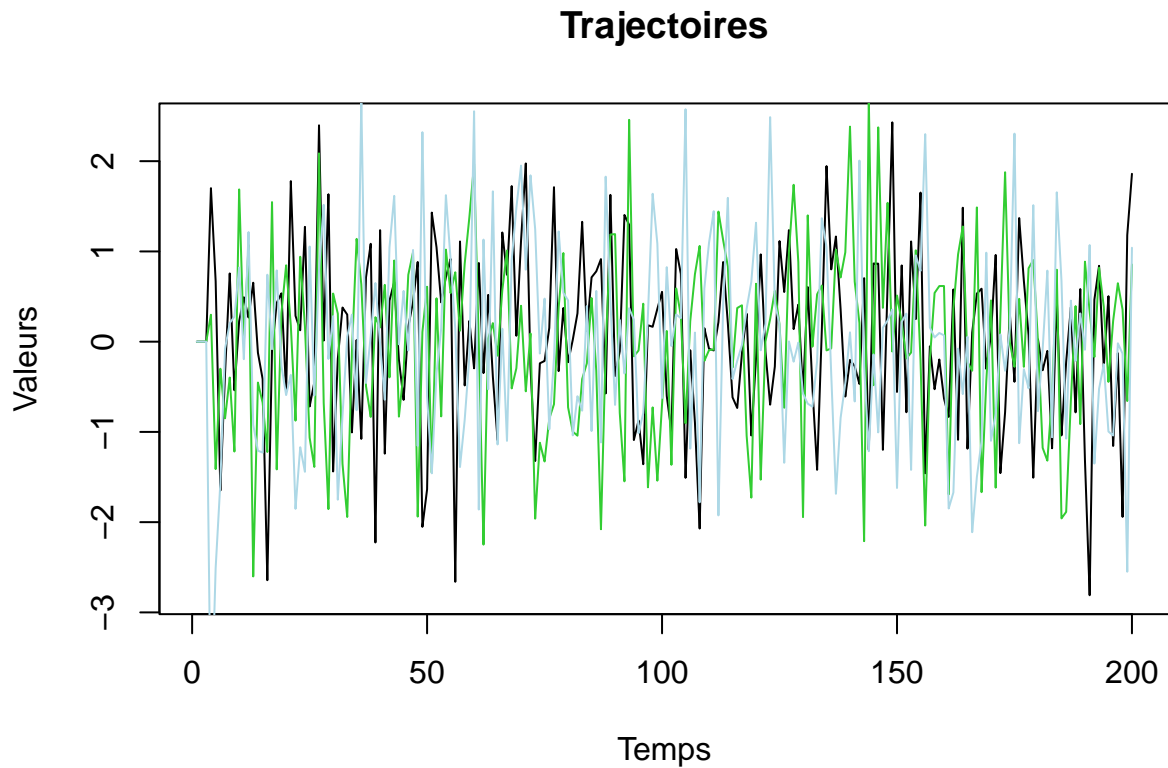
  X[1] <- rnorm(1)
  X[2] <- rnorm(1)
  X[3] <- rnorm(1)

  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 :

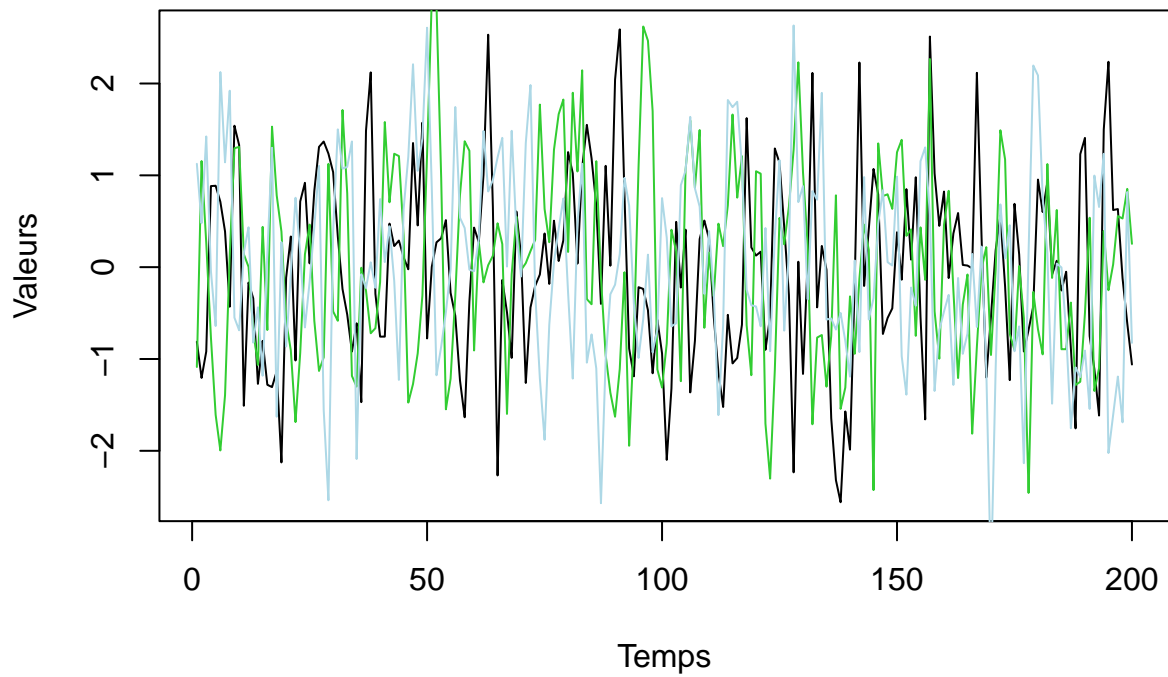
$$\begin{aligned} 1 - 0.25r &= 0 \\ r &= 4 \end{aligned}$$

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)

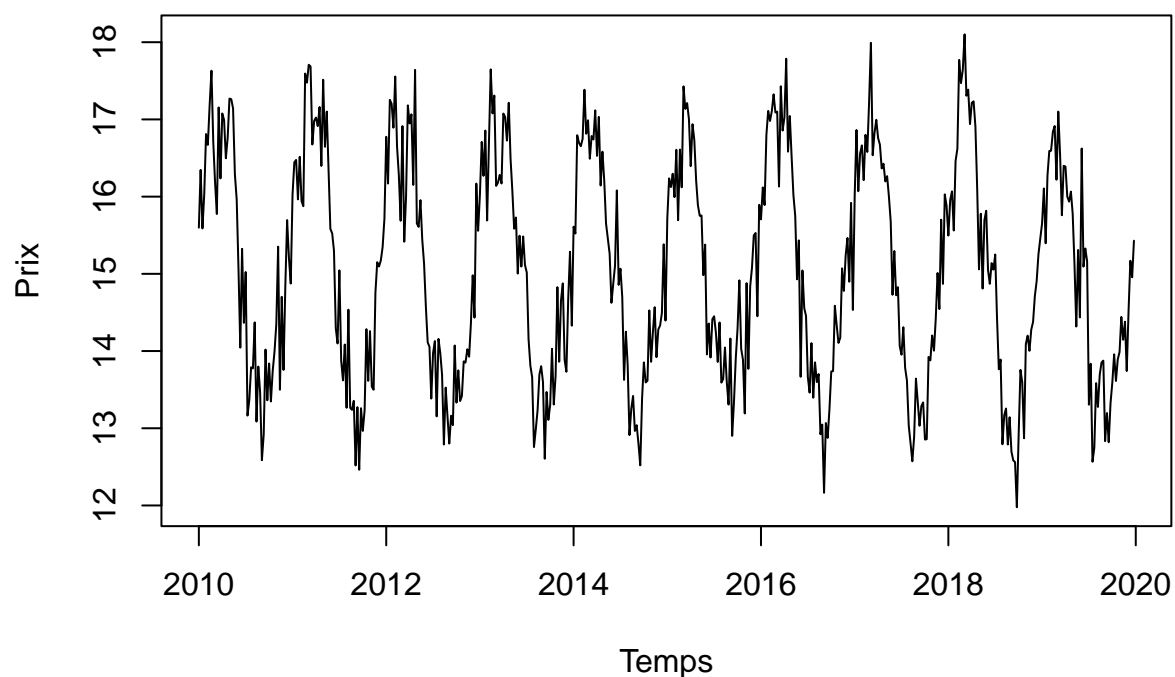


Exercice 2

a) Représentation graphique de la série p1

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

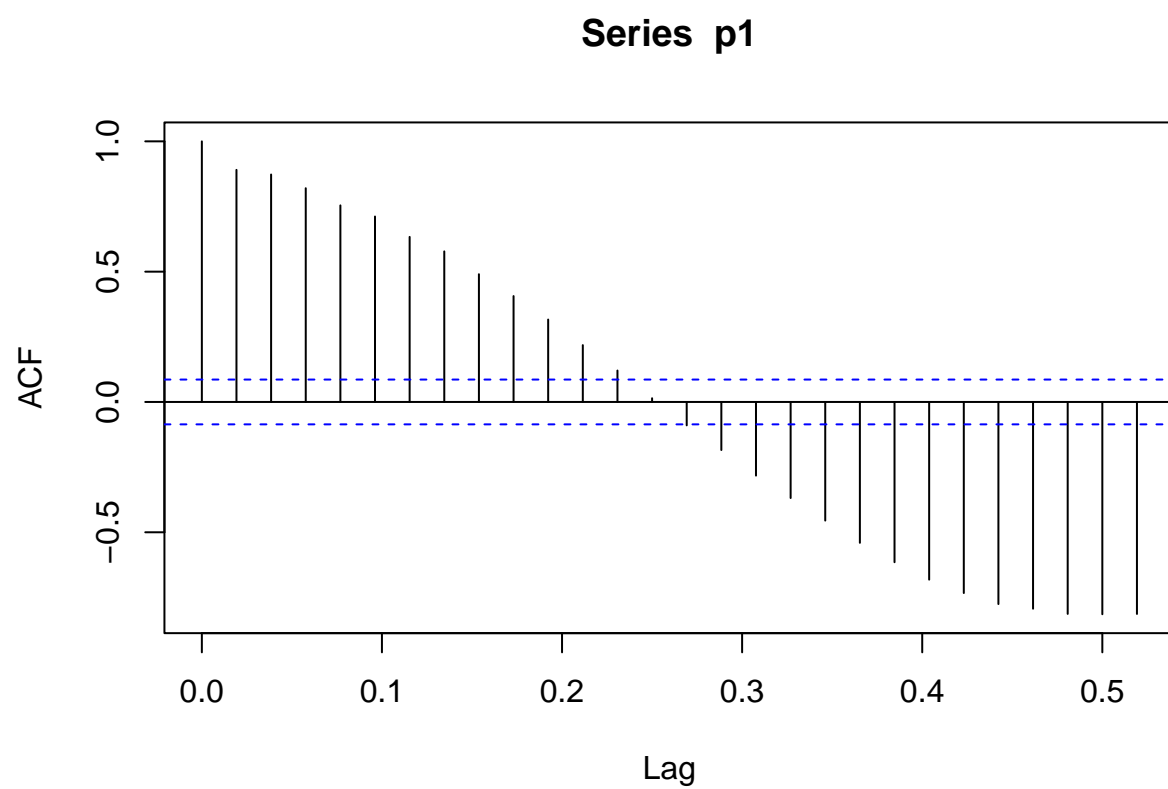
Série Temporelle des Prix



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.

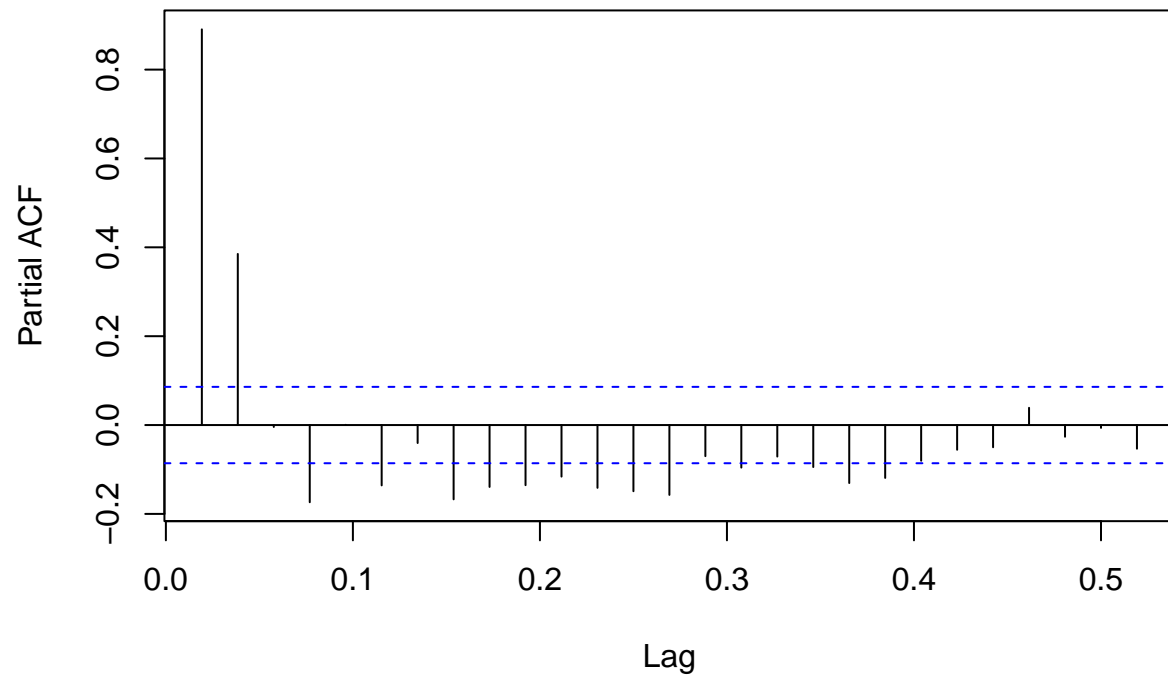
b) Fonctions d'autocorrélation acf et pacf

```
acf(p1)
```



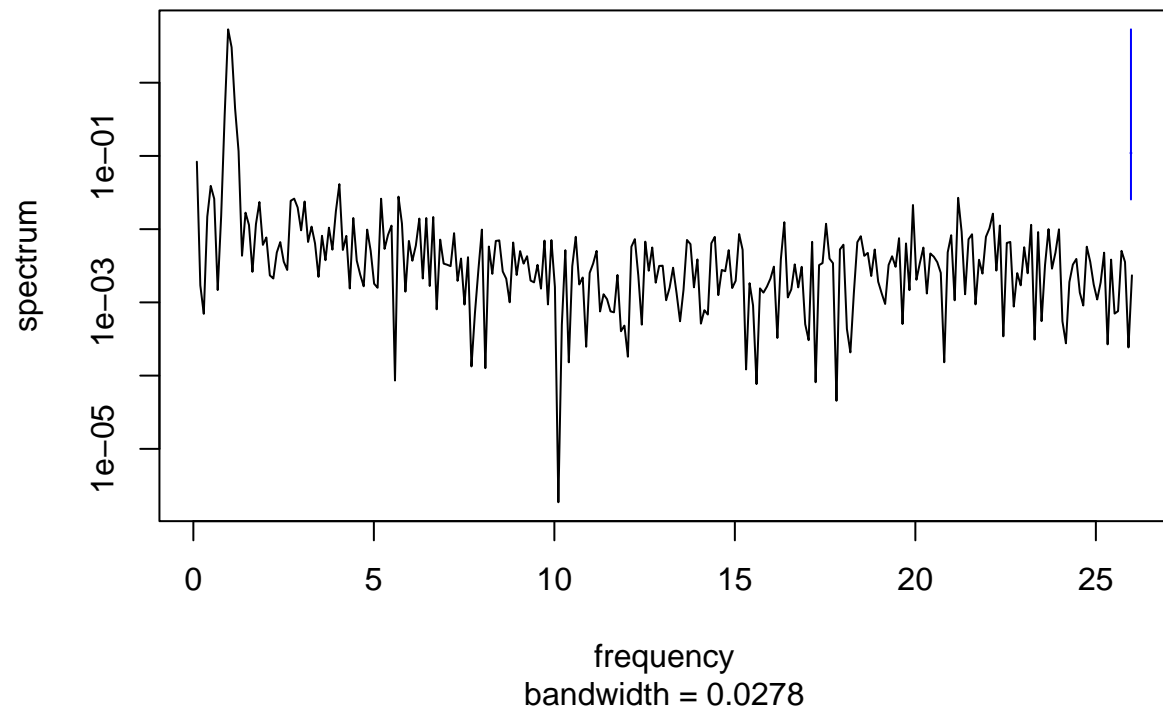
```
pacf(p1)
```

Series p1



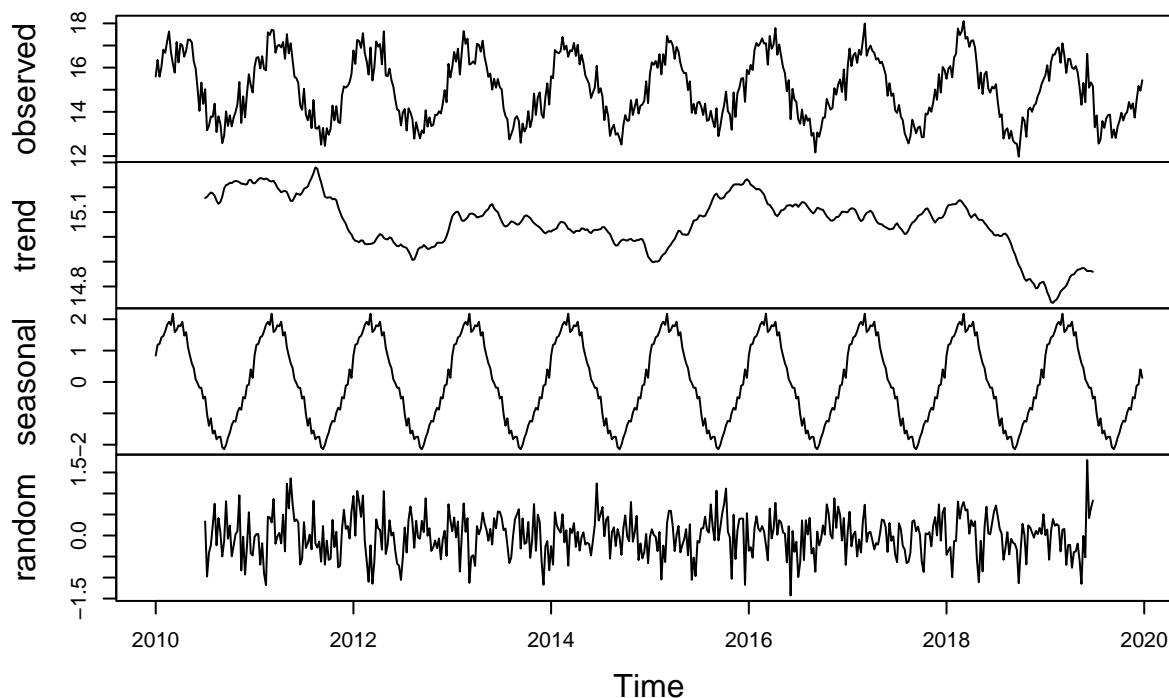
```
spectrum(p1)
```

Series: x
Raw Periodogram



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

Decomposition of additive time series



On remarque que la saisonnalité est très marquée et régulière, tandis que la tendance est plus compliquée à discerner, même s'il semble y avoir une tendance à la baisse.

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

- X_t représente la valeur de la série temporelle au temps t .
- m est la moyenne de la série.
- A et B sont les amplitudes des composantes sinusoïdales et cosinusoidales, respectivement.
- T est la période de la composante saisonnière. Pour des données hebdomadaires sur plusieurs années, T pourrait être 52, représentant le nombre de semaines par an.
- Y_t est le terme d'erreur ou la composante non-périodique.

On choisit donc un T de 52.

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

sin_term <- sin(2 * pi * t / T)
cos_term <- cos(2 * pi * t / T)

reg_model <- lm(p1 ~ sin_term + cos_term)

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

Zt <- reg_model$residuals - mean(reg_model$residuals)
```

```
adf_test <- adf.test(Zt)
```

```
## Warning in adf.test(Zt): p-value smaller than printed p-value
```

```
summary(reg_model)
```

```
##
## Call:
## lm(formula = p1 ~ sin_term + cos_term)
##
## 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_term     1.79372    0.03304    54.29  <2e-16 ***
## cos_term     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
```

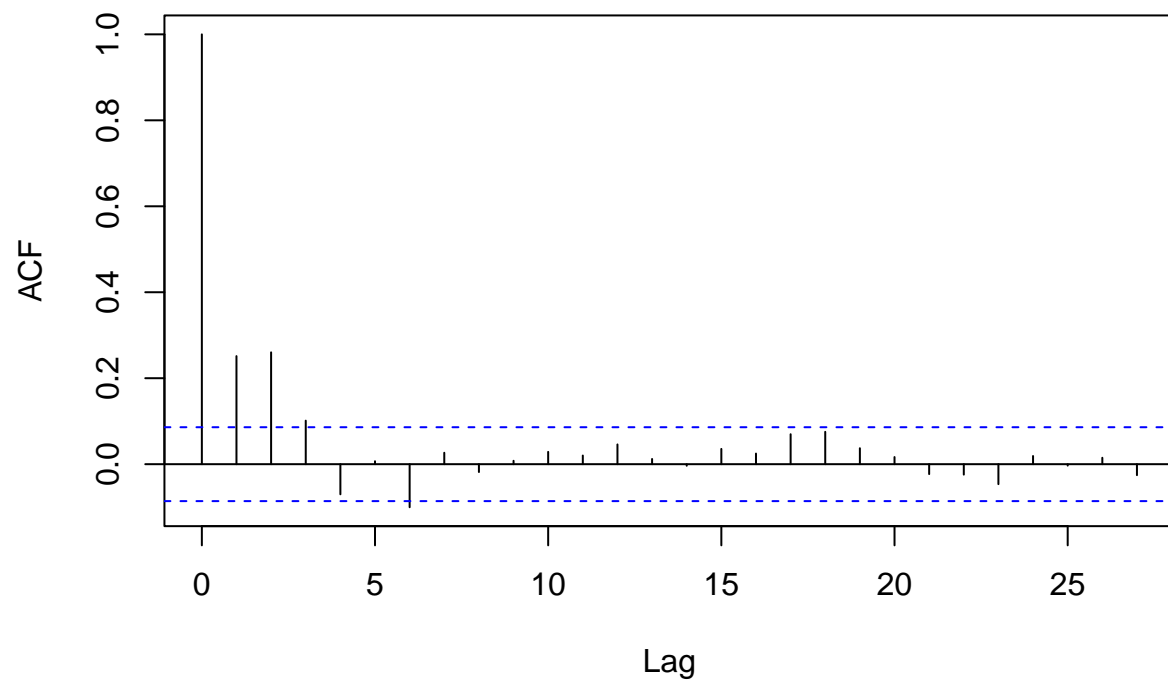
On remarque que tous les coefficients sont significatifs. La composante saisonnière sinusoïdale est donc une caractéristique importante de la série $p1$. L'écart type résiduel est faible (0.5328) et le coefficient de détermination ajusté est plutôt élevé (0.862), le modèle semble donc expliqué une grande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

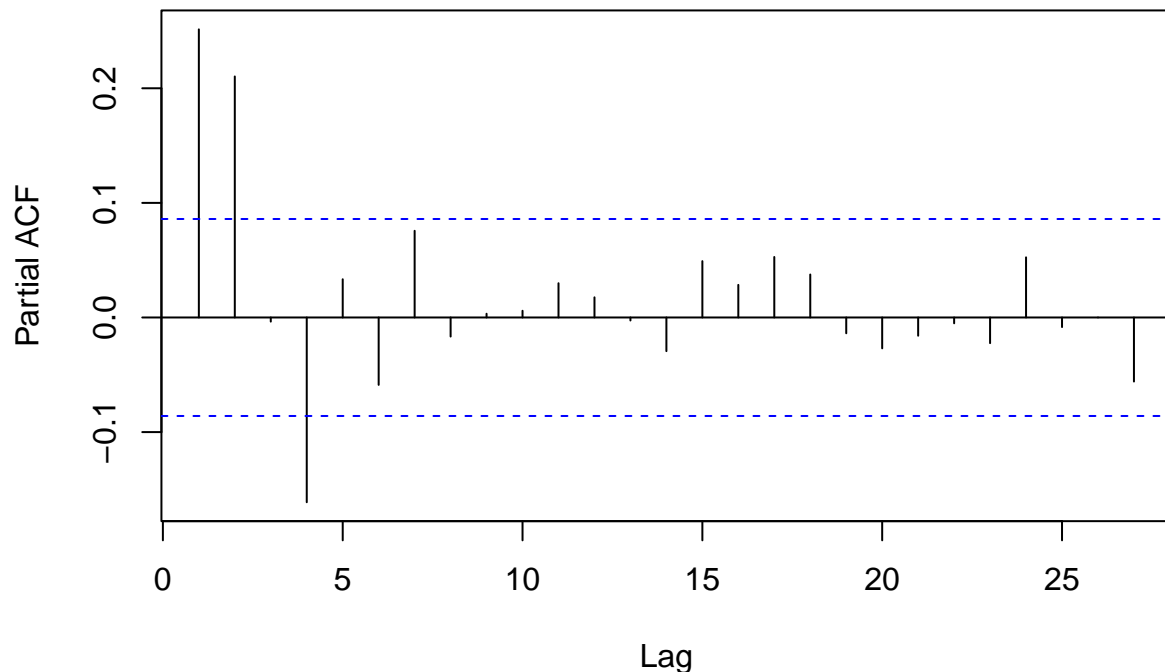
```
acf(Zt)
```

Series Z_t



```
pacf(Zt)
```


Series Z_t



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. 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.valeurs associées à chaque coefficients *coef/s.e.* en considérant que si sa valeur est inférieure à 1.96, considère la t.valeur comme non significative.

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

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

results <- data.frame(AR = rep(NA, nrow(orders)),
                      MA = rep(NA, nrow(orders)),
                      AIC = rep(NA, nrow(orders)),
                      stringsAsFactors = FALSE)
for (i in seq(nrow(orders))) {
  model <- models[[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

  results[i, "AR"] <- orders[i, "ar"]
  results[i, "MA"] <- orders[i, "ma"]
}
```

```

results[i, "AIC"] <- AIC(model)
results[i, "t.value non significatives"] <- compteur_t.values
}

kable(results, 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 petit peu plus faible que pour l'ARMA(4,1), bien qu'on puisse considérer les deux modèles comme équivalents.

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
```

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)
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 -0.0222536 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

h) Modèle le plus pertinent