



Mo LI

ENSAE Paris - la deuxième année
2019-2020

Séries temporelles linéaires
rapport

Projet final - Séries temporelles linéaires

Table des matières

1	Partie 1 : Les Données	3
1.1	Question 1	3
1.2	Question 2	3
1.3	Question 3	4
2	Partie 2 : Modèles ARMA	4
3	Partie 3 : Prévision	4
3.1	Question 6	4
3.2	Question 7	5
3.3	Question 8	6
3.4	Question 9	6
A	Graphiques	6
B	code	12

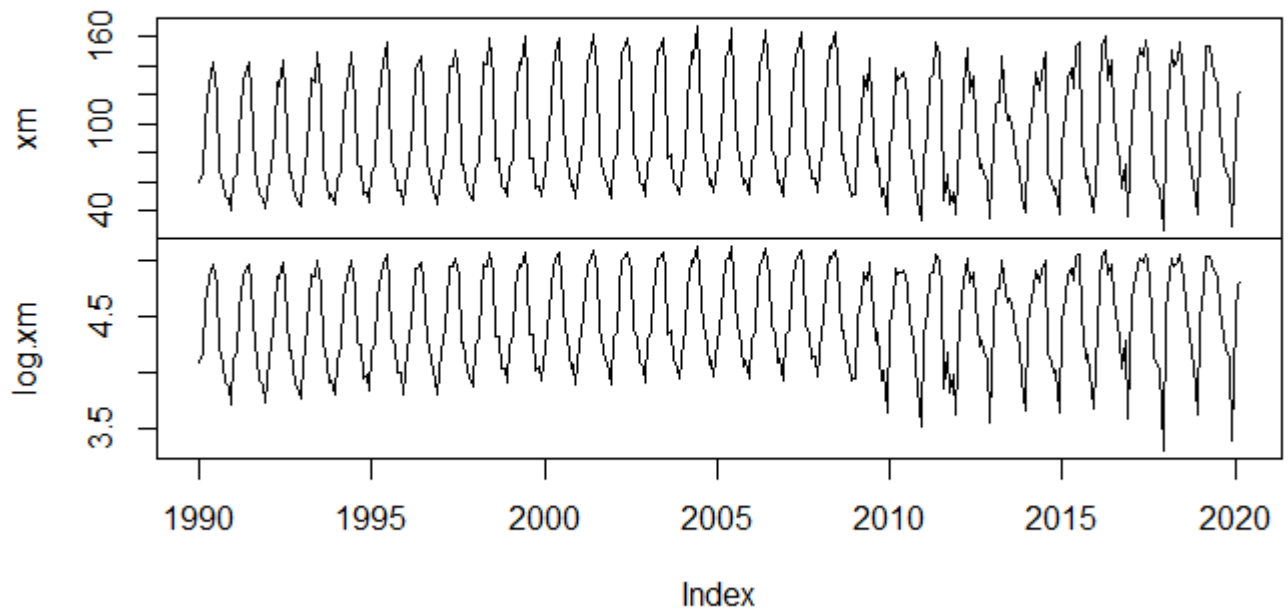


FIGURE 1: Les données avant et après la transformation

1 Partie 1 : Les Données

1.1 Question 1

Les données sont "Fabrication de glaces et sorbets" de Janvier 1990 jusqu'à Mars 2020, qui peut être téléchargé sur ce lien : <https://www.insee.fr/fr/statistiques/serie/010537265>.

Pour commencer, nous avons fait une transformation logarithmique de notre série, qui rendre la série plus stationnaire, donc plus favorable pour les tests de racine unitaire. Nos données avant et après la transformation sont présentées dans le graphique 1.

1.2 Question 2

En regardant la série transformée, elle semble très saisonnière mais sans tendance. En utilisant la commande *decompose()* de R qui permet de décomposer la série initiale en une tendance, une saisonnalité (événements récurrents aux mêmes périodes) et d'un bruit blanc, la saisonnalité est très claire dans notre série. Le résultat de notre décomposition se trouve dans le graphique 5, et on trouve que l'effet de saisonnalité est fort. Egalement, la saisonnalité est aussi identifiée sur l'autocorrélogramme (6) et l'autocorrélogramme partiel (7) car il y a des fortes autocorrélations et autocorrélations partielles empiriques entre les variables décalées d'un an.

Donc, afin d'enlever la saisonnalité, on a d'abord différencié la série à l'ordre 12.

Pour simplifier les calculs suivants, nous avons centralisé la série différenciée. En suite, nous avons passé un test de Phillips-Perron ainsi que un test de ADF sans constante et sans tendance, et on revient au résultat que la série a bien rejeté la racine unitaire et elle est donc stationnaire, comme la p-valeur est très petite.

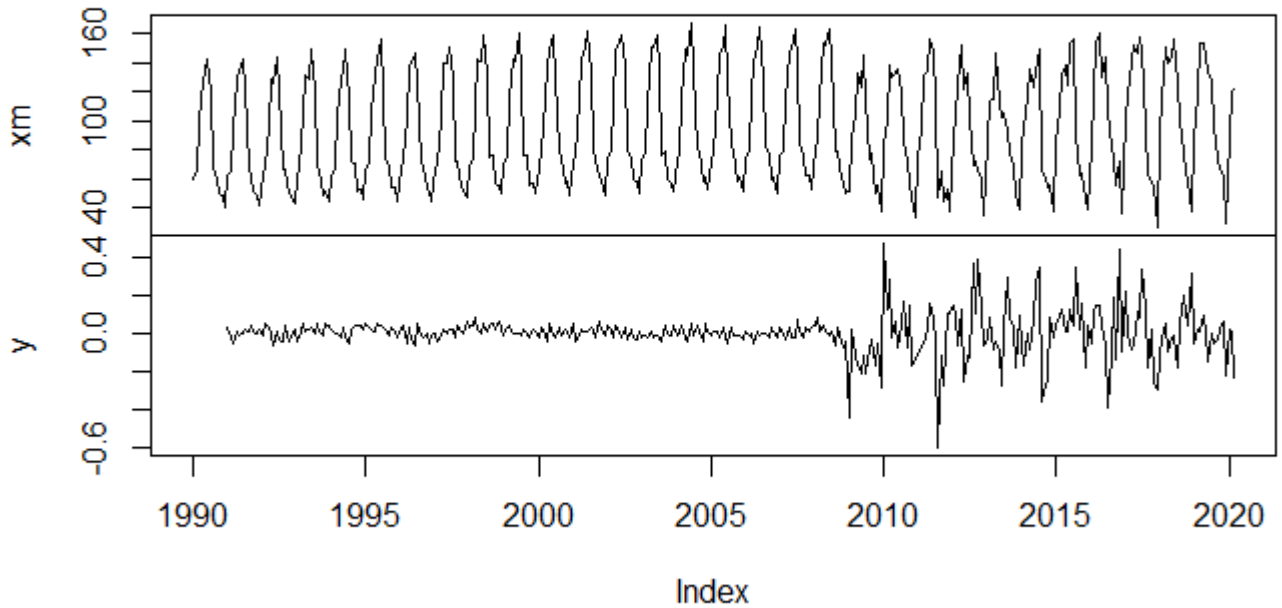


FIGURE 2: la série avant et après la transformation

1.3 Question 3

Le graphique 2 représentent respectivement la série avant et après la transformation.

2 Partie 2 : Modèles ARMA

Nous cherchons un modèle $ARIMA(p, d, q)$ pour estimer et présenter notre série choisie. Comme la série est stationnaire, on a bien $d = 0$. Et puis, pour déterminer p_{max} et q_{max} de notre série, nous avons examiné l'autocorrélogramme et l'autocorrélogramme partiel ci-dessous. Pour simplifier le modèle (et pour l'intérêt des temps de calculs), on prende $p_{max} = 24$ et $q_{max} = 12$.

Comme on a déjà fait dans TD4, on estime tout les arima possibles et en vérifie l'ajustement et la validité. Après 25×13 fois d'itération, nous avons trouvé 34 modèles validés (le coefficient du dernier paramètre de MA et AR sont significatifs et les résidus sont non-corrélés). (Voir le tableau 1).

En suite, pour choisir le meilleur modèle, nous avons calculé AIC et BIC de chaque modèle. On trouve que le modèle $ARMA(24, 2)$ minimise BIC (-582.7782) et $ARMA(19, 11)$ minimise AIC (-694.2743). Les résidus simulés des deux modèles sont présentés dans le graphique 8 et 9, qui nous dit que les résidus sont non-corrélés.

3 Partie 3 : Prévision

3.1 Question 6

On va désormais prendre le modèle $ARMA(24, 2)$ par exemple.

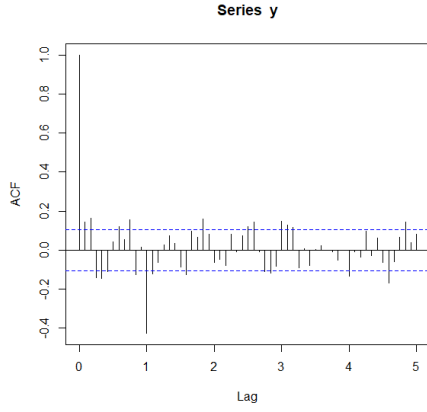


FIGURE 3: Autocorrélogramme

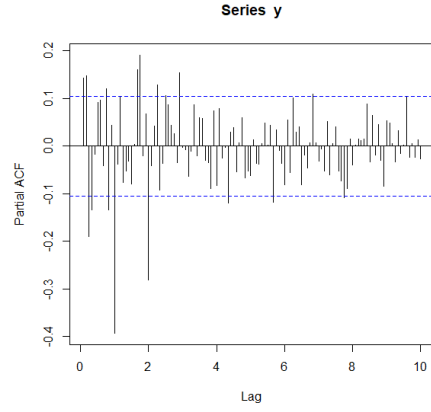


FIGURE 4: Autocorrélogramme partiel

On note T la longueur de la série et on suppose que les résidus de notre modèle suivent une loi Gaussienne, alors $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. Le modèle ARMA(24, 2) s'écrit comme :

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_{24} X_{t-24} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$$

Donc on a les équations suivantes :

$$\begin{aligned} X_{T+1} &= \phi_1 X_T + \phi_2 X_{T-1} + \cdots + \phi_{24} X_{T-23} + \epsilon_{T+1} + \theta_1 \epsilon_T + \theta_2 \epsilon_{T-1} \\ X_{T+2} &= \phi_1 X_{T+1} + \phi_2 X_T + \cdots + \phi_{24} X_{T-22} + \epsilon_{T+2} + \theta_1 \epsilon_{T+1} + \theta_2 \epsilon_T \\ \hat{X}_{T+1|T} &= \phi_1 X_T + \phi_2 X_{T-1} + \cdots + \phi_{24} X_{T-23} + \theta_1 \epsilon_T + \theta_2 \epsilon_{T-1} \\ \hat{X}_{T+2|T} &= \phi_1 \hat{X}_{T+1|T} + \phi_2 X_T + \cdots + \phi_{24} X_{T-22} + \theta_2 \epsilon_T \end{aligned}$$

Alors,

$$\begin{pmatrix} X_{T+1} - \hat{X}_{T+1|T} \\ X_{T+2} - \hat{X}_{T+2|T} \end{pmatrix} = \begin{pmatrix} \epsilon_{T+1} \\ (\phi_1 + \theta_1)\epsilon_{T+1} + \epsilon_{T+2} \end{pmatrix}$$

où la matrice de covariance :

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \phi_1 + \theta_1 \\ \phi_1 + \theta_1 & 1 + (\phi_1 + \theta_1)^2 \end{bmatrix}$$

Donc par l'hypothèse de la normalité, on a $(X - \hat{X})^T \Sigma^{-1} (X - \hat{X}) \sim \chi(2)$. On peut alors construire notre intervalle de confiance au niveau α par :

$$\left\{ X \in \mathbf{R}^2 \mid (X - \hat{X})^T \Sigma^{-1} (X - \hat{X}) \leq q_2^{1-\alpha} \right\}$$

, où $q_2^{1-\alpha}$ est le quantile de la loi $\chi(2)$.

3.2 Question 7

Afin d'obtenir la région de confiance, nous avons fait l'hypothèse suivantes :

Les résidus sont indépendants et identiquement distribués de loi $\mathcal{N}(0, \sigma^2)$; notre modèle suit bien un ARMA(24,2); les coefficients estimés de ce modèle dans la partie 2 sont les vrais coefficients; et la matrice Σ est inversible.

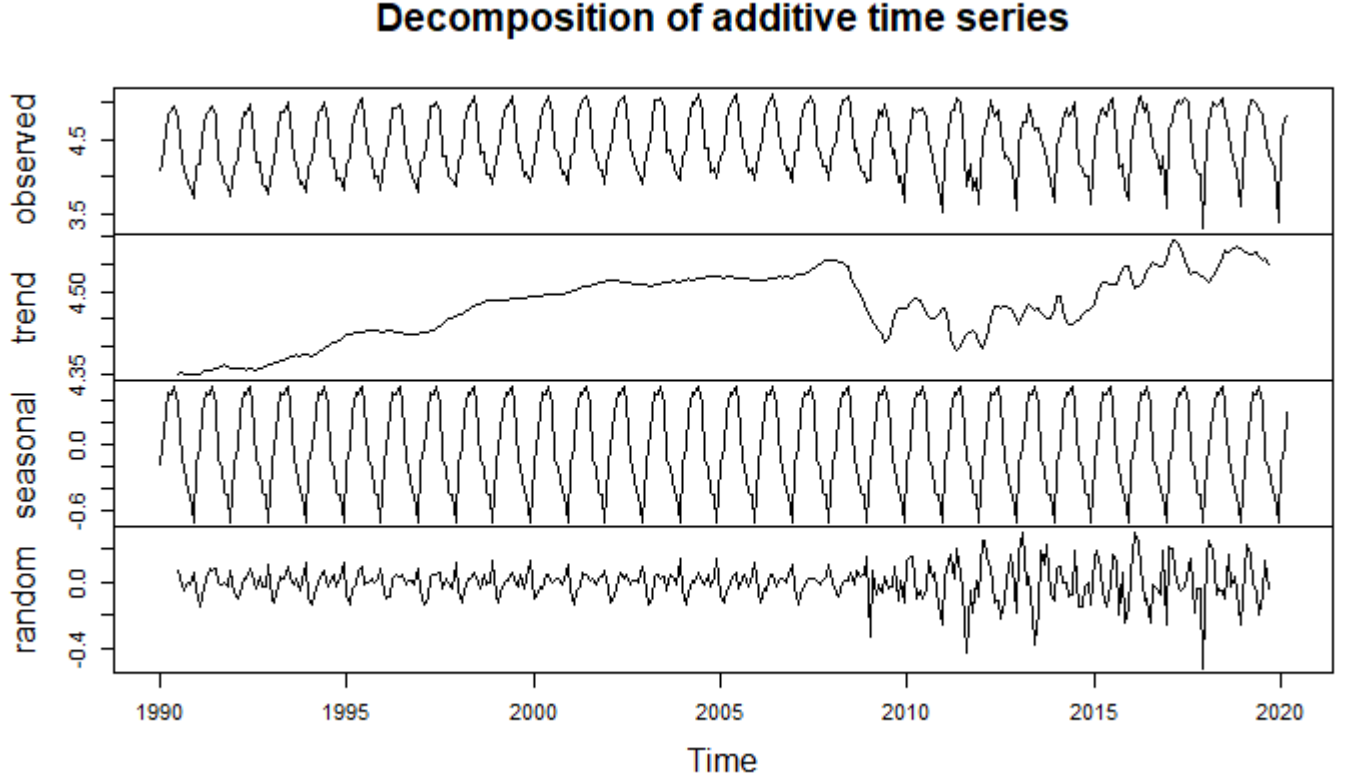


FIGURE 5

3.3 Question 8

D'après la Question 6, nous avons trouvé les prévisions de X_{T+1} et X_{T+2} (10) ainsi que leur intervalle de confiance (11).

3.4 Question 9

L'information de Y_{T+1} permet d'améliorer la prévision de $\hat{X}_{T+1|T}$ si la variable Y_{T+1} cause instantanément la variable X_T au sens de Granger. Ainsi, on peut prédire mieux X_T en exploitant les nouvelles valeur de Y_T . Plus précisément, il faut que

$$\hat{X}_{T+1|\{X_u, Y_u, u \leq t\}} \neq \hat{X}_{T+1|\{X_u, Y_u, u \leq t\} \cup \{Y_{T+1}\}}$$

.

A Graphiques

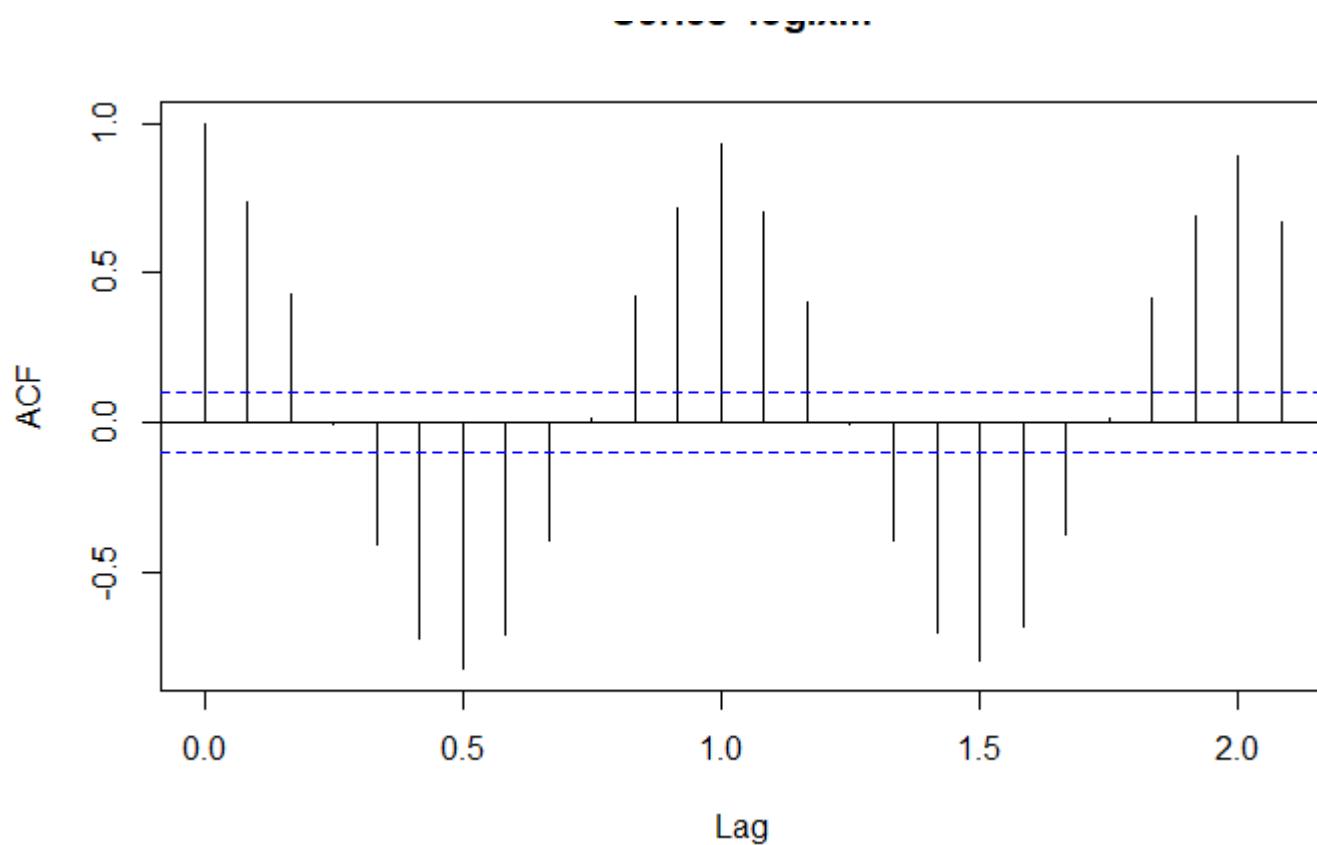


FIGURE 6: Autocorrélogramme

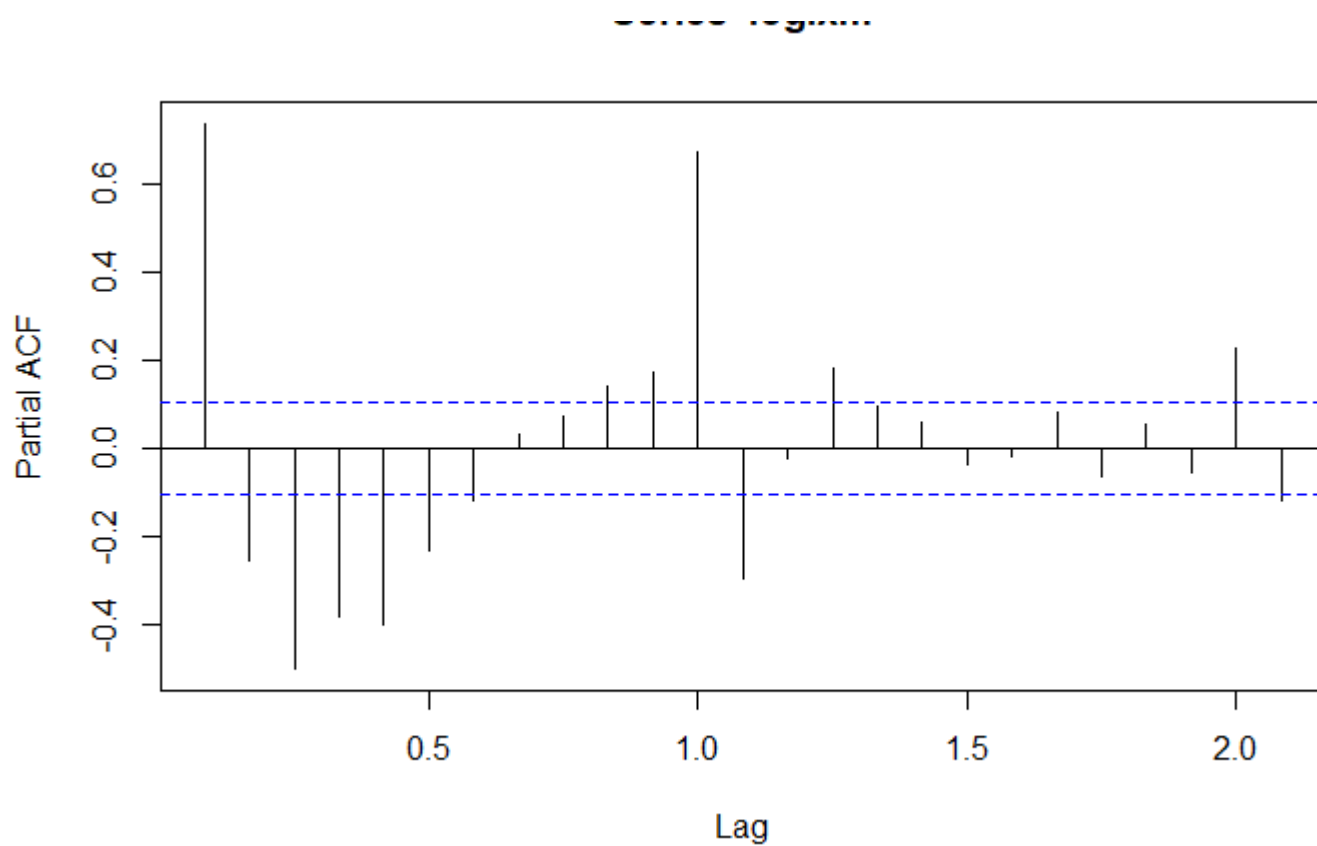


FIGURE 7: Autocorrélogramme partiel

TABLE 1: Les modèles possibles

p	q	arsignif	masignif	resnocorr	ok
24	0	1		1	1
24	2	1	1	1	1
21	3	1	1	1	1
23	3	1	1	1	1
21	4	1	1	1	1
24	4	1	1	1	1
22	5	1	1	1	1
23	5	1	1	1	1
20	6	1	1	1	1
21	6	1	1	1	1
22	6	1	1	1	1
21	7	1	1	1	1
23	7	1	1	1	1
24	7	1	1	1	1
18	8	1	1	1	1
19	8	1	1	1	1
21	8	1	1	1	1
15	9	1	1	1	1
16	9	1	1	1	1
21	9	1	1	1	1
14	10	1	1	1	1
16	10	1	1	1	1
18	10	1	1	1	1
22	10	1	1	1	1
13	11	1	1	1	1
15	11	1	1	1	1
18	11	1	1	1	1
19	11	1	1	1	1
20	11	1	1	1	1
22	11		1	1	1
14	12	1	1	1	1
15	12	1	1	1	1
17	12			1	1
22	12	1	1	1	1

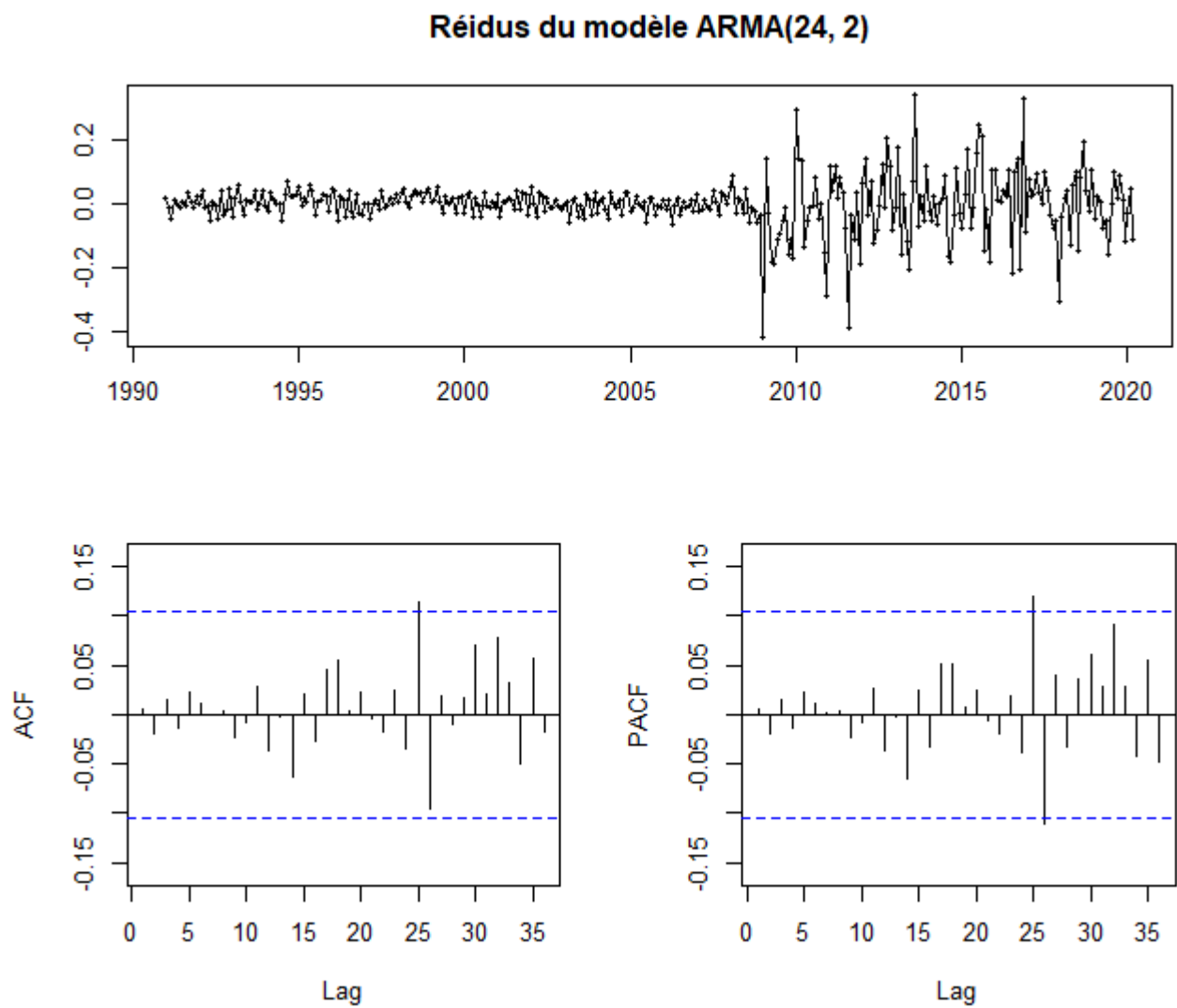


FIGURE 8: Réidus du modèle ARMA(24, 2)

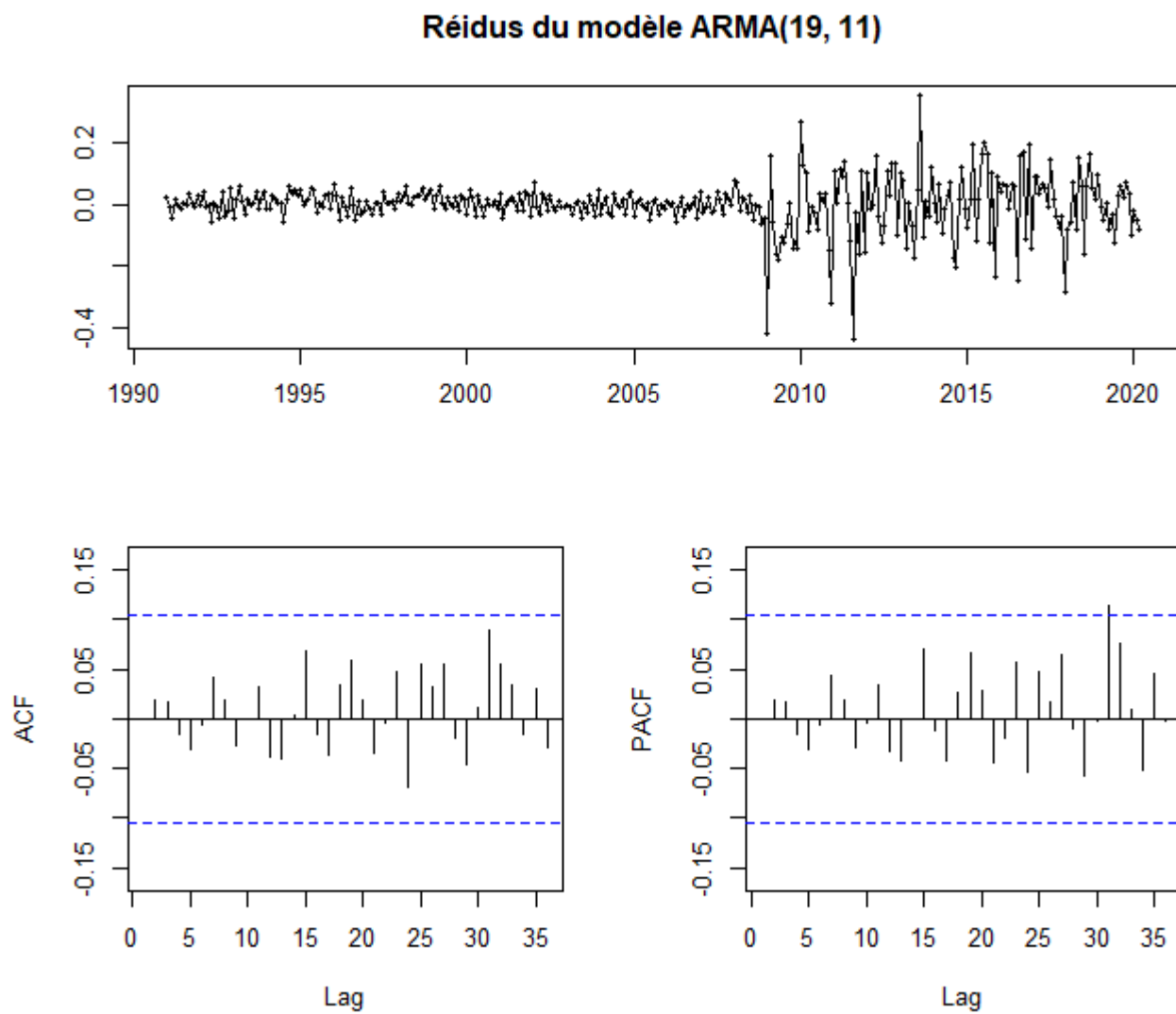


FIGURE 9: Réidus du modèle ARMA(19, 11)

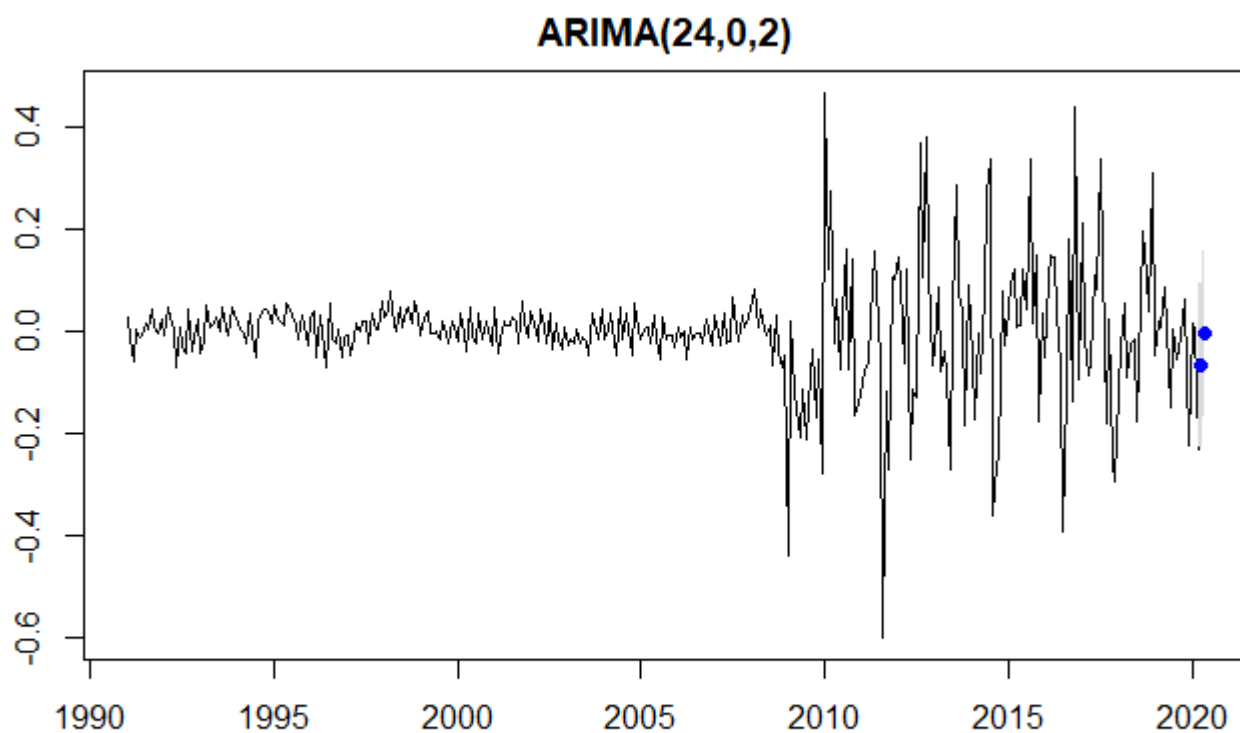
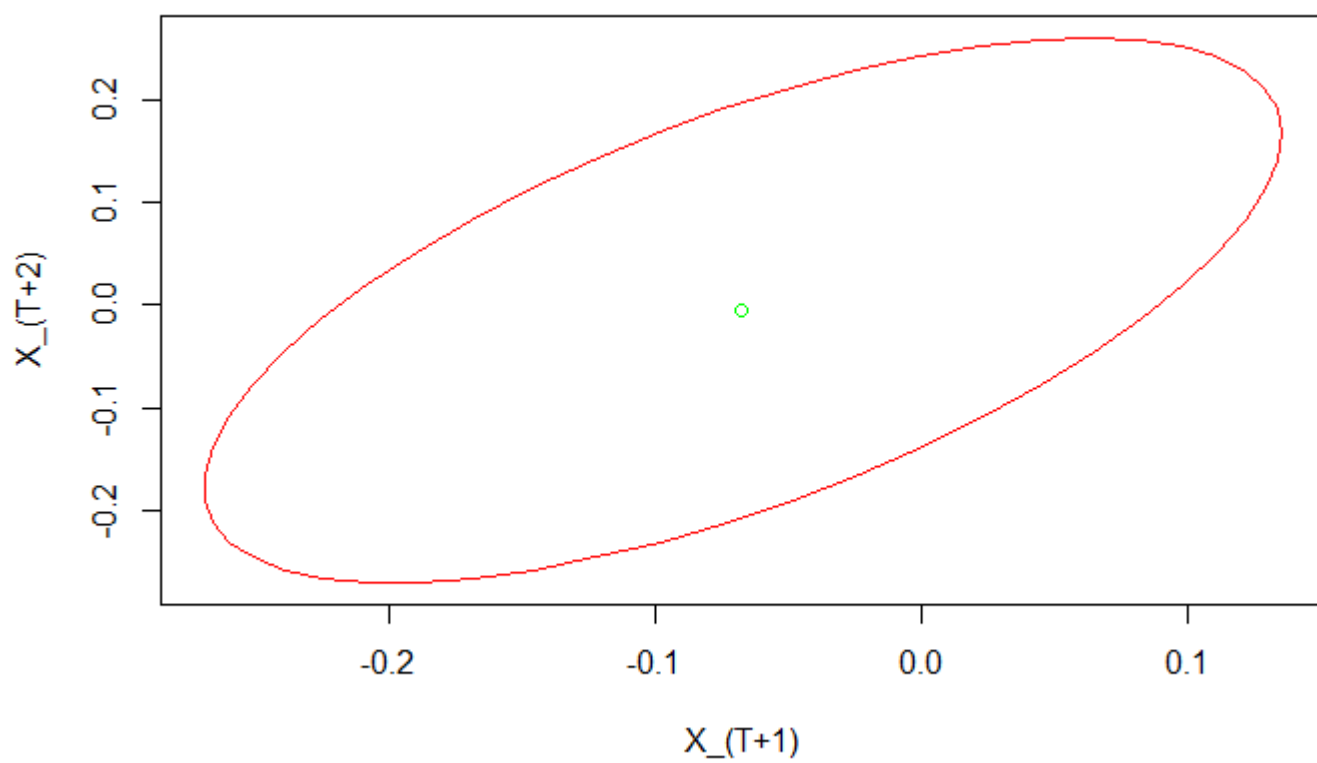
FIGURE 10: l'estimation de X_{T+1} et X_{T+2} 

FIGURE 11: Intervalle de confiance

B code

```

setwd("D:/2019-2020 ENSAE Paris/2019-2020 2eme semestre/STL/Projet final")

library('tseries')
library('lmtest')
library('FitAR')
library('forecast')
library('zoo')

data <- read.csv("valeurs_mensuelles.csv", sep = ";")
xm <- rev(zoo(data[[2]])[-c(1,2)])
dates <- as.yearmon(seq(from = 1990,to = 2020 + 2/12, by = 1/12))
xm <- zoo(xm, order.by = dates)
log.xm <- log(xm)
n <- length(xm)

plot(cbind(xm, log.xm), main = '')

acf(log.xm)
pacf(log.xm)

plot(decompose(ts(log.xm, frequency = 12, start = c(1990,1))))

desaison.xm <- diff(log.xm, 12)
y <- desaison.xm - mean(desaison.xm)
plot(y)

y.ts <- ts(y, frequency = 12, start = c(1990,1))
de <- decompose(y.ts)
plot(de$seasonal)

pp.test(y)
adfTest(y, type = "nc")

plot(cbind(xm, y), main = '')

par(mfrow = c(1, 2))
acf(y, 60)
pacf(y, 120)

modelchoice <- function(p, q, data = y, k = 24) {
  estim <- try(arima(data, c(p,0,q), optim.control=list(maxit=20000)))
  if (class(estim)=="try-error") return(c("p"=p,"q"=q,"arsignif"=NA,
    "masignif"=NA,"resnocorr"=NA, "ok"=NA))
  arsignif <- if (p==0) NA else signif(estim)[3,p]<=0.05
  masignif <- if (q==0) NA else signif(estim)[3,p+q]<=0.05
  resnocorr <- sum(Qtests(estim$residuals,24,
    length(estim$coef)-1)[,2]<=0.05,na.rm=T)==0

```

```

checks <- c(arsignif, masignif, resnocorr)
ok <- as.numeric(sum(checks, na.rm=T)==(3-sum(is.na(checks))))

return(c("p"=p, "q"=q, "arsignif"=arsignif, "masignif"=masignif,
"resnocorr"=resnocorr, "ok"=ok)) }

armamodelchoice <- function(pmax,qmax){
  pqs <- expand.grid(0:pmax,0:qmax)
  t(apply(matrix(1:dim(pqs)[1]),1,function(row) { p <- pqs[row,1];
  q <- pqs[row,2]; cat(paste0("Computing ARMA(",p," ",q," ") \n));
  modelchoice(p,q) }))) }

pmax <- 24
qmax <- 12
armamodels <- armamodelchoice(pmax,qmax)

selec <- armamodels[armamodels[, "ok"] == 1 & !is.na(armamodels[, "ok"]),]
selec

pqs <- apply(selec,1,function(row) list("p" = as.numeric(row[1]),
"q"=as.numeric(row[2])))
names(pqs) <- paste0("arma(",selec[,1],",",selec[,2],")")
models <- lapply(pqs, function(pq) arima(y, c(pq[["p"]],0,pq[["q"]]))))
aicbic <- vapply(models, FUN.VALUE=numeric(2), function(m) c("AIC"=AIC(m),
"BIC"=BIC(m)))

SerieForcast2 <- forecast(models$`arma(24,2)`, h=2, level=c(95))
plot(SerieForcast2, main = "ARIMA(24,0,2)")

library('car')
Variance_residus = var(residuals(models$`arma(24,2)`))
Matrix_cov = matrix(,nrow = 2,ncol = 2)
Matrix_cov[1,1] <- Variance_residus
Matrix_cov[2,1] <- Variance_residus * (1 + models$`arma(24,2)`$coef[1] +
models$`arma(24,2)`$coef[2])
Matrix_cov[1,2] <- Variance_residus*(1 + models$`arma(24,2)`$coef[1] +
models$`arma(24,2)`$coef[2])
Matrix_cov[2,2] <- Variance_residus*(1 + (1 + models$`arma(24,2)`$coef[1] +
models$`arma(24,2)`$coef[2])^2)

alpha <- 0.05
Quantile <- qchisq( 1 - alpha , df = 2)

X1 <- -0.067123960
X2 <- -0.004853908

Center = c(X1,X2)
plot(Center[1], Center[2])
x <- ellipse(center = Center, shape = Matrix_cov, radius = sqrt(Quantile))

```

```
plot(x, type = 'l', col = "red" , xlab = "X_(T+1)", ylab = "X_(T+2)")  
points(X1, X2, col = "green")
```