Package vars

Paul GUILLOTTE & Jules CORBEL

01/02/2019

Abstract

bbbbb

Contents

Introduction

1 Analyse descriptive des séries

1.1 Rappel sur la stationnarité du second ordre

Avant de commencer à analyser les séries, nous rappelons des bases sur des notions dont nous aurons besoin par la suite.

Dans de nombreux modèles de séries temporelles, la série en entrée doit satisfaire une hypothèse de stationnarité. Les conditions de la stationnarité du second ordre.

$E[y_t] = \mu \forall t = 1...T$

$Var[y_t] = \sigma^2 \neq \infty \forall t = 1...tVar[y_t] = \sigma^2 \neq \infty \forall i = 1...T$

$Cov[y_i, Z_{i-k}] = f(k) \forall i = 1...t, \forall k = 1...t$

Nous nous intéressons dans cette partie aux différentes séries trimestrielles à notre disposition. Dans un premier temps, nous nous intéressons aux corrélations entre les variables deux à deux afin de nous faire une première idée du lien qu'il existe entre les variables.

1.2 Masse salariale

```
MSE <- ts(trim$MSE, start = 1990, end = c(2017, 2), frequency=4)
plot(MSE, main="Evolution trimestrielle de la masse salariale", xaxt="n", cex.main=0.9)
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1",
                                           "2010Q1", "2015Q1"))
```

```
par(mfrow=c(1,2), cex.main=0.8)
acf(MSE, main="Auto-corrélation de la
    masse salariale trimestrielle", lag.max=20)
pacf(MSE, main="Autocorrélation partielle
     de la masse salariale trimestrielle", lag.max=20)
```

```
kpss.test(MSE)
```

```
## Warning in kpss.test(MSE): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  MSE
## KPSS Level = 3.6772, Truncation lag parameter = 2, p-value = 0.01
```

```
adf.test(MSE)
```

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

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

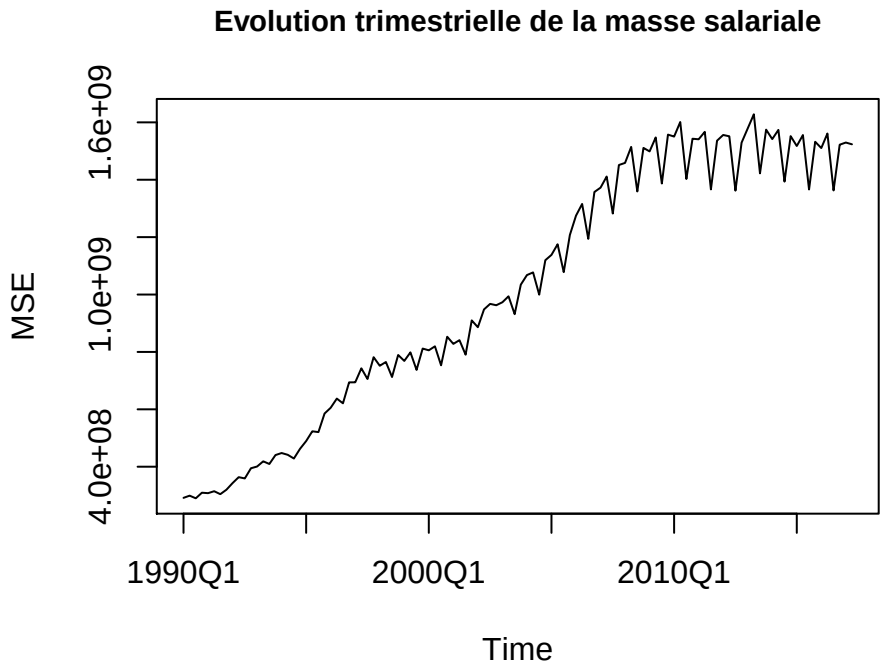


Figure 1:

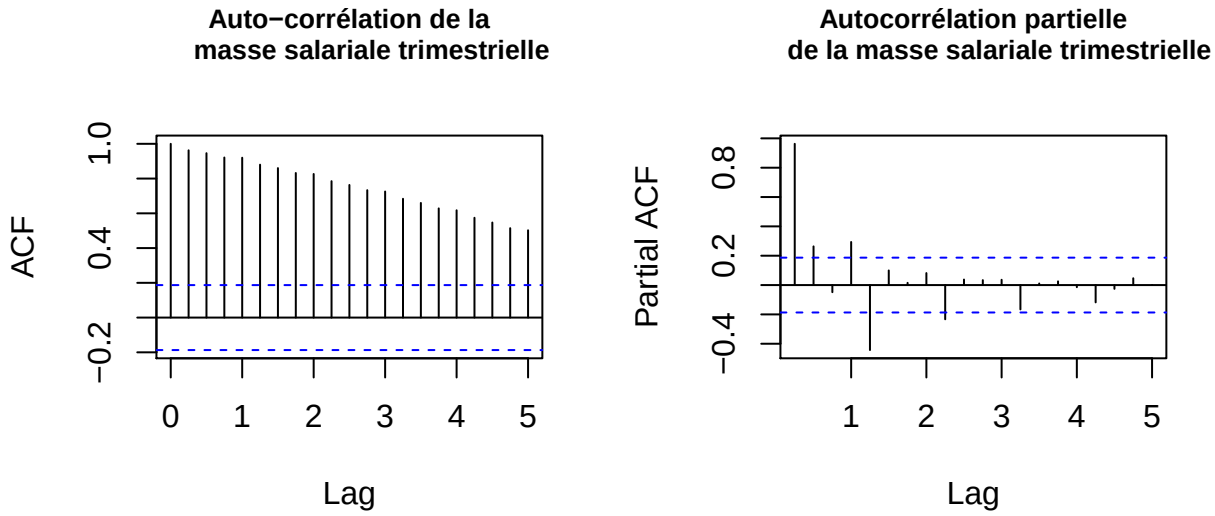


Figure 2:

```
##
## data:  MSE
## Dickey-Fuller = -0.20821, Lag order = 4, p-value = 0.99
## alternative hypothesis: stationary
```

La masse salariale trimestrielle, représentée en Figure 1 possède une composante de tendance de 1990 à 2010. La série tend par la suite à stagner. Nous remarquons également une saisonnalité sur cette série, qui est de plus en plus marquée à mesure que le temps passe.

Comme la série comporte une tendance et une saisonnalité, elle ne correspond pas aux deux premières conditions de la stationnarité du second ordre, soit que la série possède une moyenne et un écart-type constants. Cela est confirmé par la Figure 2, qui nous montre fonction ACF qui décroît régulièrement. Nous effectuons également un test de KPSS (test de stationnarité) servant à vérifier si la série est stationnaire ou non (sous l'hypothèse H_0 la série est stationnaire, et sous l'hypothèse H_1 elle ne l'est pas). La série est dite stationnaire si ses propriétés statistiques (espérance, variance et auto-corrélation) sont fixes au cours du temps. La p-value est de 0.01 ce qui nous confirme que la série n'est pas stationnaire avec un risque de première espèce de 5%. Nous mettons également en place un test de racines unitaires, le test de Dickey Fuller augmenté. Son hypothèse nulle est que la série a été générée par un processus présentant une racine unitaire, et donc que la série n'est pas stationnaire. Ici, avec un risque de premier espèce à 5%, on conserve l'hypothèse nulle est on conclut, à l'aide des deux tests effectués, que la série n'est pas stationnaire.

1.3 PIB

```
PIB <- ts(trim$PIB, start = 1990, end = c(2017, 1), frequency=4)
plot(PIB, main="Evolution trimestrielle du PIB",xaxt="n", cex.main=0.9)
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1",
```

```
par(mfrow=c(1,2), cex.main=0.8)
acf(PIB, main="Auto-corrélation
    du PIB trimestriel", lag.max=40)
pacf(PIB, main="Autocorrélation partielle
     du PIB trimestriel", lag.max=40)
```

```
par(mfrow=c(1,1))
kpss.test(PIB)
```

```
## Warning in kpss.test(PIB): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  PIB
## KPSS Level = 3.6473, Truncation lag parameter = 2, p-value = 0.01
```

```
adf.test(PIB)
```

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

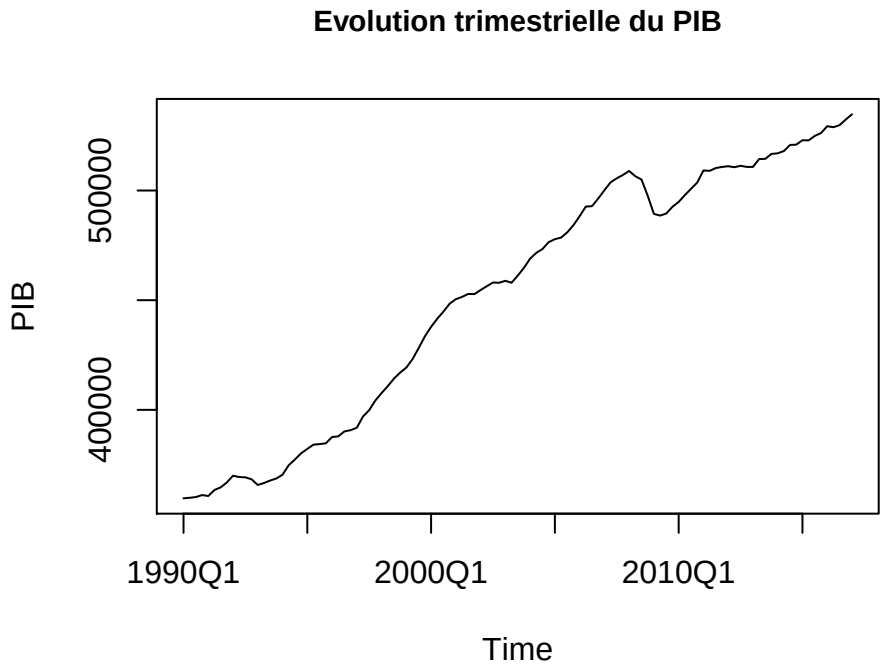


Figure 3:

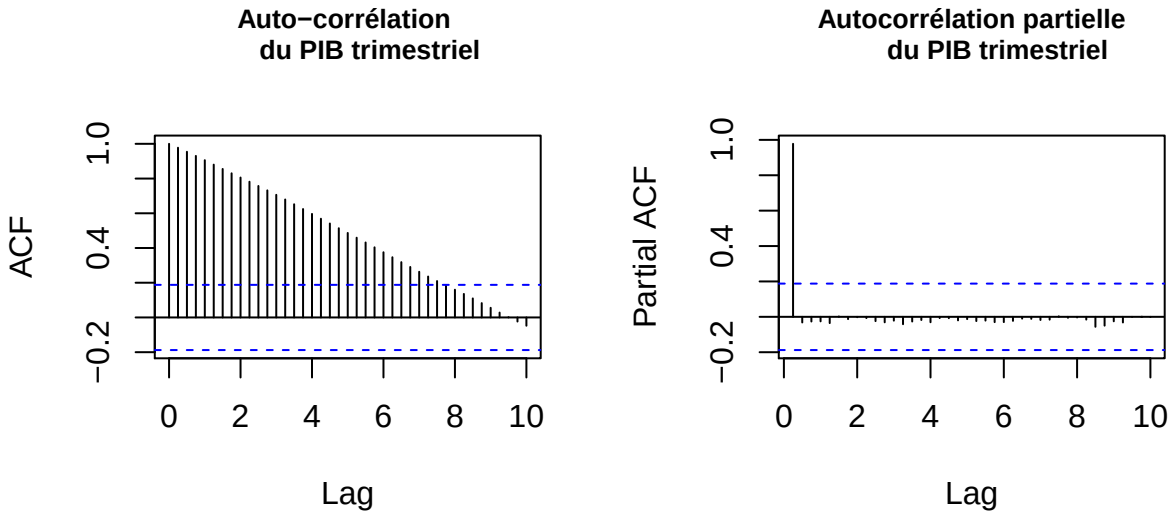


Figure 4:

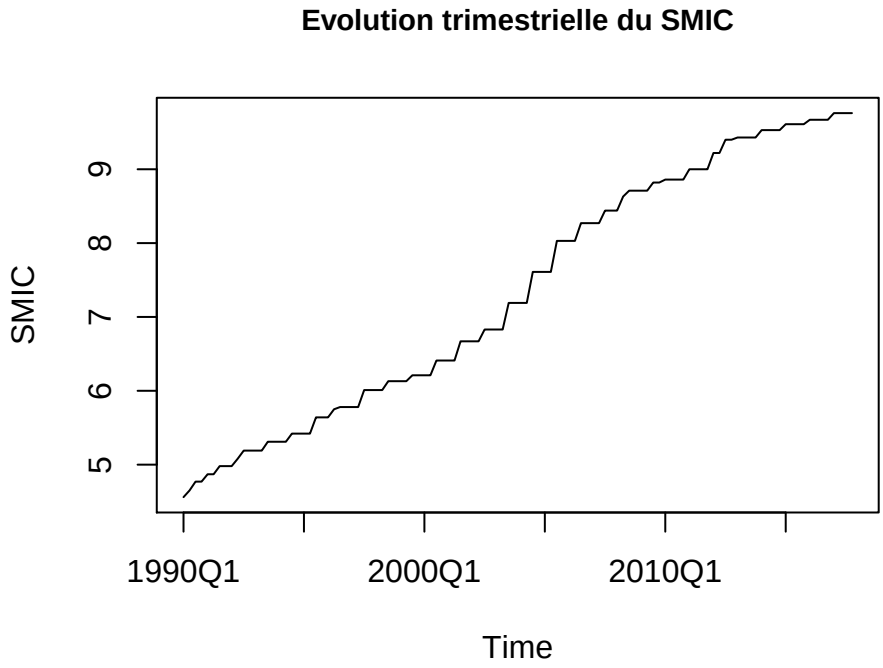


Figure 5:

```
## data:  PIB
## Dickey-Fuller = -1.3274, Lag order = 4, p-value = 0.8557
## alternative hypothesis: stationary
```

La Figure 3 nous montre le PIB trimestriel qui, comme pour la masse salarialepossède une tendance. Cependant, il ne semble pas posséder de saisonnalité. Cette série ne semble donc pas non plus stationnaire. Nous effectuons à nouveau un test de KPSS. La p-value est de 0.01 ce qui nous confirme que la série n'est pas stationnaire avec un risque de première espèce de 5%. Même conclusion au regard du test augmenté de Dickey Fuller.

1.4 SMIC

```
SMIC <- ts(trim$SMIC, start = c(1990,1), end = c(2017, 4), frequency = 4)
plot(SMIC, main="Evolution trimestrielle du SMIC", xaxt="n", cex.main=0.9)
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1",
```

```
par(mfrow=c(1,2), cex.main=0.8)
acf(SMIC, main="Auto-corrélation du
    SMIC trimestriel", lag.max=20)
pacf(SMIC, main="Autocorrélation partielle
     du SMIC trimestriel", lag.max=20)
```

```
par(mfrow=c(1,1))
kpss.test(SMIC)
```

```
## Warning in kpss.test(SMIC): p-value smaller than printed p-value
```

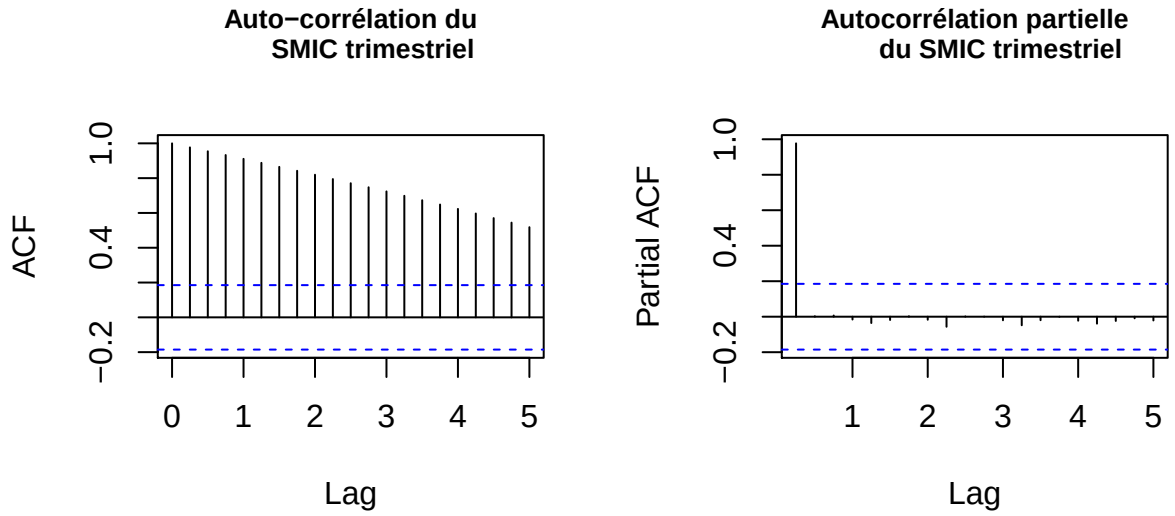


Figure 6:

```
##
##  KPSS Test for Level Stationarity
##
## data:  SMIC
## KPSS Level = 3.8382, Truncation lag parameter = 2, p-value = 0.01
```

```
adf.test(SMIC)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  SMIC
## Dickey-Fuller = -1.4174, Lag order = 4, p-value = 0.8184
## alternative hypothesis: stationary
```

Au regard de la Figure 5, on s'aperçoit qu'il y a bien une tendance. Pour la saisonnalité, il est plus difficile de savoir s'il en existe une ou pas, puisque la série semble augmenter seulement à certains temps. Les tests de KPSS et de Dickey Fuller augmenté nous confirment que la série n'est pas stationnaire.

1.5 Taux de chômage des femmes

```
TCHOF <- ts(trim$TCHOF, start = c(1990,1), end = c(2017, 4), frequency = 4)
plot(TCHOF, main="Evolution trimestrielle du taux de chômage des femmes", xaxt="n", cex.main=
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1",
```

```
par(mfrow=c(1,2), cex.main=0.8)
acf(TCHOF, main="Auto-corrélation du taux de
    chômage des femmes trimestriel", lag.max=20)
pacf(TCHOF, main="Autocorrélation partielle du
     taux de chômage des femmes trimestriel", lag.max=20)
```

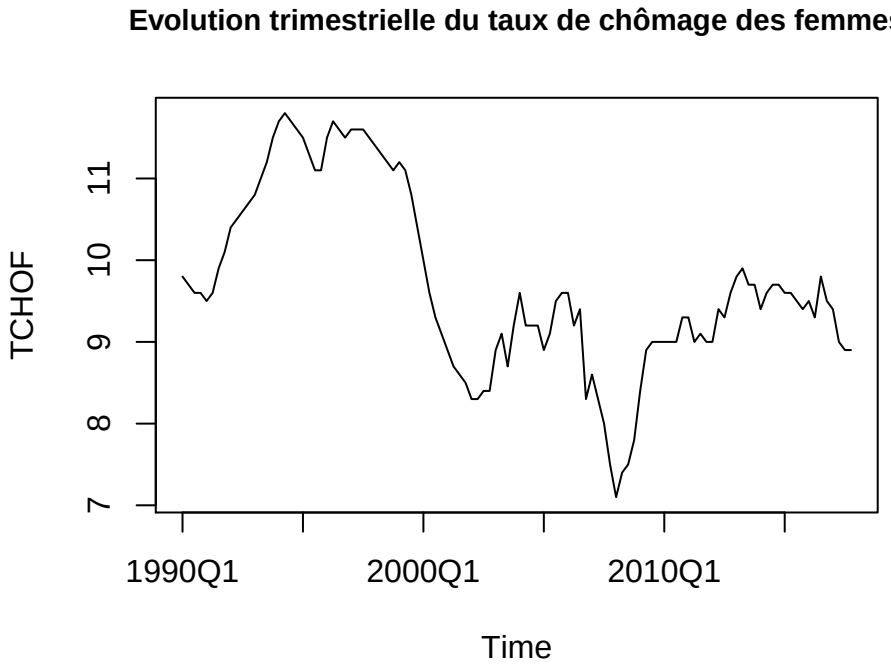


Figure 7:

```
  par(mfrow=c(1,1))
  kpss.test(TCHOF)
```

```
## Warning in kpss.test(TCHOF): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  TCHOF
## KPSS Level = 1.6407, Truncation lag parameter = 2, p-value = 0.01
```

```
  adf.test(TCHOF)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  TCHOF
## Dickey-Fuller = -2.5838, Lag order = 4, p-value = 0.3344
## alternative hypothesis: stationary
```

Pour cette dernière série (Figure 7) qui représente le taux de chômage trimestriel des femmes, il ne semble pas y avoir de saisonnalité. On remarque cependant qu'il y a bien une tendance, au regard de la Figure 8. En regardant la série de plus près, on s'aperçoit que la tendance semble être "par morceaux" : d'abord une hausse de 1990 à 1996, puis elle décroît jusqu'en 2002, avant d'augmenter à nouveau jusqu'en 2007, de chuter jusqu'en 2010. Si la série ne possède pas une tendance uniforme sur toute la durée étudiée, elle semble donc bien posséder une tendance par morceaux. Les tests KPSS et de Dickey Fuller augmenté nous confirment que la série n'est pas stationnaire, avec un

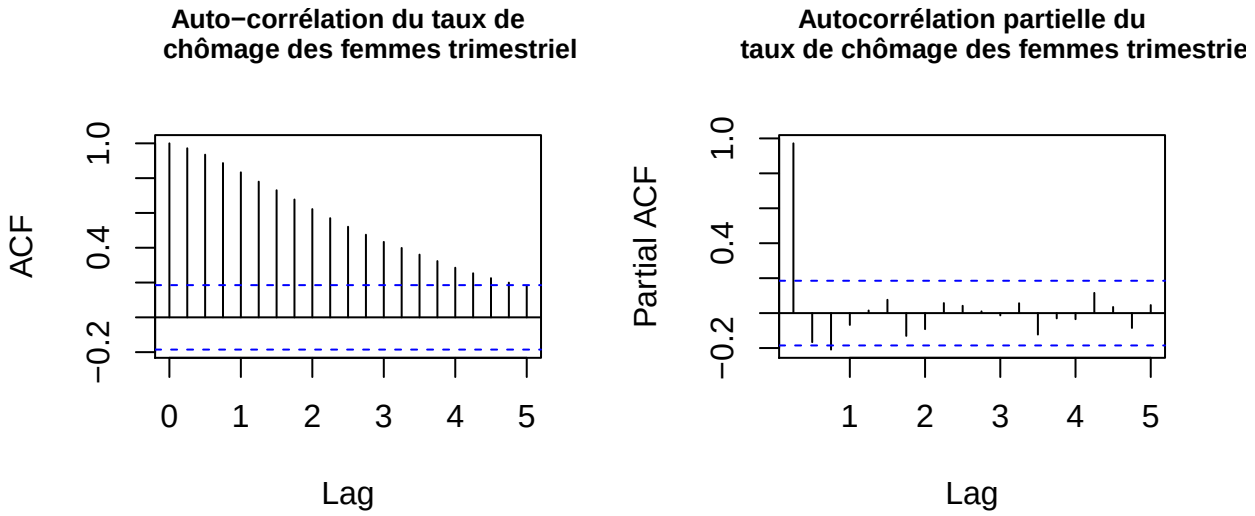


Figure 8:

risque de première espèce de 5%.

1.6 Calcul des corrélations

```
corrplot(cor(trim[1:109,-1]), method = "number", type="lower",
         p.mat=cor.mtest(trim[1:109,-1], 0.95)[[1]], insig="pch",
         col=colorRampPalette(c("blue", "light blue", "red"))(50), title = "
         Corrélations entre les variables trimestrielles")
```

```
corr <- cor.mtest(trim[1:109,-1], 0.95)[[1]]
rownames(corr) <- c("MSE","PIB","SMIC","TCHOF")
colnames(corr) <- c("MSE","PIB","SMIC","TCHOF")
corr
```

```
##                MSE          PIB         SMIC        TCHOF
## MSE   0.000000e+00 3.851955e-69 1.436967e-74 3.321841e-12
## PIB   3.851955e-69 0.000000e+00 1.898200e-71 2.387179e-15
## SMIC  1.436967e-74 1.898200e-71 0.000000e+00 1.377731e-10
## TCHOF 3.321841e-12 2.387179e-15 1.377731e-10 0.000000e+00
```

Nous affichons la matrice des corrélations des différentes variables en Figure 9. On se rend compte que le taux de chômage des femmes est corrélé négativement avec toutes les autres variables. Le trio de variables PIB, masse salariale et SMIC sont extrêmement liées entre elles. En regardant le tableau des p-values associées au test de Student (H0 : La corrélation entre les deux variables est nulle), on s'aperçoit que toutes les variables prises deux à deux présentes une corrélation.

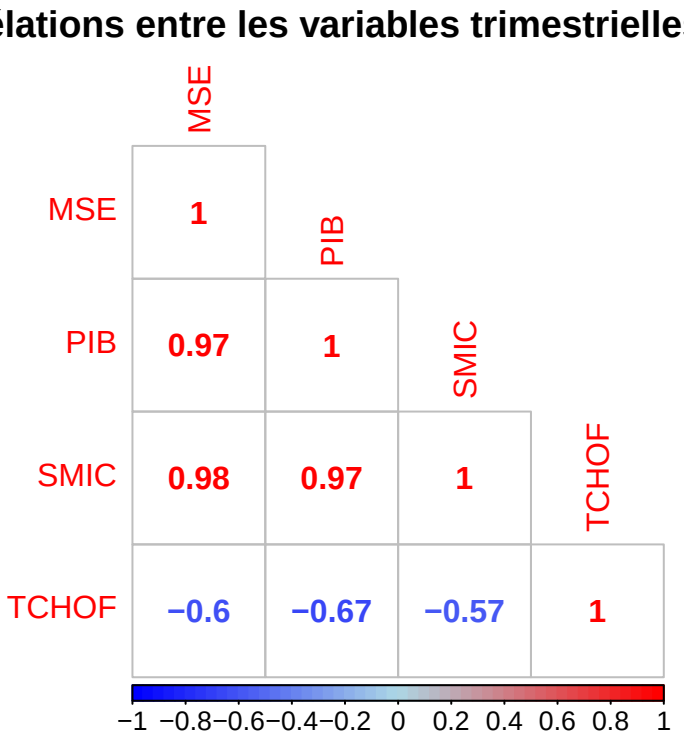


Figure 9:

2 Modélisation individuelle

Une fois que nous avons analysé le comportement des différentes séries temporelles à notre disposition, nous souhaitons les modéliser afin de prédire les valeurs futures de ces différentes séries. En effet, si nous voulons prédire la MSE pour des valeurs futures, nous aurons également besoin des valeurs associées pour les variables explicatives, qui ne seront peut-être pas à notre disposition. Nous avons utilisé à la fois des modèles basés sur un lissage exponentiel et des processus ARMA.

2.1 Découpage des séries

Pour chacune des séries, nous allons créer un échantillon d'apprentissage, qui nous permettra de construire les différents modèles, ainsi qu'un échantillon de test, qui nous permettra de comparer les prédictions des modèles construits avec des vraies valeurs. L'échantillon d'apprentissage sera composé de toutes les valeurs du premier trimestre 1990 jusqu'au 4e trimestre 2015, tandis que celui de test comprendra toutes les valeurs à partir du 1er trimestre 2016.

```
MSETrain <- window(MSE, start=1990, end=c(2015,4))
MSETest <- window(MSE, start=2016, end=c(2017,2))
PIBTrain <- window(PIB, start=1990, end=c(2015,4))
PIBTest <- window(PIB, start=2016, end=c(2017,1))
SMICTrain <- window(SMIC, start=1990, end=c(2015,4))
SMICTest <- window(SMIC, start=2016, end=c(2017,2))
TCHOFTrain <- window(TCHOF, start=1990, end=c(2015,4))
TCHOFTest <- window(TCHOF, start=2016, end=c(2017,2))
```

2.2 Comparaison des différents modèles

Afin de comparer les modèles construits pour chaque série avec les différentes méthodes, nous calculons l'erreur quadratique moyenne (EQM), soit les moyennes des différences au carré entre les valeurs de test et les valeurs prédites par le modèle.

2.3 Lissage exponentiel

2.3.1 Définition

Le lissage exponentiel permet de prédire les valeurs d'une série temporelle en lissant successivement les données à partir d'une valeur initiale. Plus les observations sont éloignées dans le passé, moins leur poids est important lors du calcul. Pour une série stationnaire, la formule de calcul d'une valeur est la suivante : $s_t = \alpha y_t + (1-\alpha)s_{t-1}$, le paramètre α étant le facteur de lissage. Le nom de cette méthode est un lissage exponentiel **simple**. Afin de modéliser les séries possédant une tendance, nous introduisons un paramètre β permettant de la prendre en compte, la méthode étant appelée lissage exponentiel **double**. Enfin, Holt et Winters ont également modifié la méthode pour qu'elle puisse modéliser les séries comportant une saisonnalité en introduisant un paramètre γ. Ils ont donné leur nom à cette méthode, qui est donc un lissage exponentiel de **Holt-Winters**.

Dans notre cas, nous ne calculons pas nous-mêmes α β et γ Ces paramètres sont déterminés automatiquement par la fonction *ets* du package **forecast** de façon à optimiser la qualité de la prédiction. Cette fonction permet également de choisir la méthode à utiliser, grâce à l'argument model. Afin de mesurer la qualité de notre modèle, nous avons choisi d'utiliser l'**AICc**(Akaike Information Criterion with correction). Le choix de l'AICc par rapport à l'AIC s'explique par le faible nombre de données que nous possédons par rapport au nombre de paramètres à estimer. C'est ce critère qui nous servira par la suite afin de comparer nos différents modèles.

Prenons l'exemple de la MSE. Nous avons vu dans la partie 1.2 que la série possédait une tendance linéaire ainsi qu'une saisonnalité multiplicative. L'argument model de la fonction *ets** prendra donc la valeur "ZAM", (erreur sélectionnée automatiquement, tendance linéaire, saisonnalité multiplicative). On peut également remarquer que lorsque tous les paramètres sont automatiquement sélectionnés (valeur "ZZZ"), les paramètres retenus sont les mêmes que ceux que nous avions rentré.

```
LEMSE<-ets(MSETrain, "ZAM")
print(LEMSE)
```

```
## ETS(M,A,M)
##
## Call:
##  ets(y = MSETrain, model = "ZAM")
##
##   Smoothing parameters:
##     alpha = 0.7675
##     beta  = 0.1111
##     gamma = 0.2325
##
##   Initial states:
##     l = 279219343.2211
```

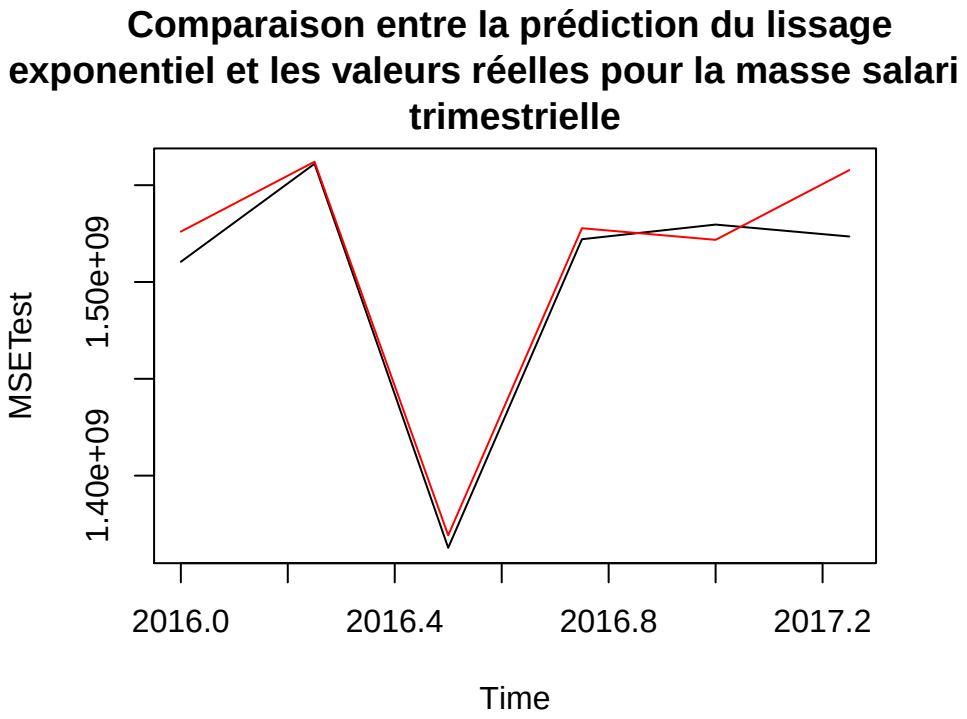


Figure 10:

```
##     b = 12053621.0848
##     s = 1.0092 0.9655 1.0159 1.0094
##
##   sigma:  0.0279
##
##      AIC     AICc      BIC
## 4031.536 4033.451 4055.336
```

```
PredLEMSE <- forecast(LEMSE, h = 6)
plot(MSETest, main="Comparaison entre la prédiction du lissage
exponentiel et les valeurs réelles pour la masse salariale
trimestrielle")
lines(PredLEMSE$mean, col="red")
```

```
EQM(MSETest, PredLEMSE$mean)
```

```
## [1] 2.592281e+14
```

```
ets(MSETrain, "ZZZ")
```

```
## ETS(M,A,M)
##
## Call:
##  ets(y = MSETrain, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 0.7675
```

```
##     beta  = 0.1111
##     gamma = 0.2325
##
##   Initial states:
##     l = 279219343.2211
##     b = 12053621.0848
##     s = 1.0092 0.9655 1.0159 1.0094
##
##   sigma:  0.0279
##
##      AIC     AICc      BIC
## 4031.536 4033.451 4055.336
```

On obtient donc un AICc de 4033.451 pour le modèle ainsi qu'une erreur quadratique moyenne de $2.6 * 10^{14}$. Le graphique obtenu en figure 10 nous montrent que le modèle obtenu nous donne des prédictions très proches de la réalité.

2.3.2 Résultats obtenus

```
par(mfrow=c(2,2))
plot(PIBTest, ylim=c(min(PIBTest,PredLEPIB$mean),max(PIBTest,PredLEPIB$mean)), main=
"Comparaison entre la prédiction du lissage
exponentiel et les valeurs réelles pour le PIB
trimestriel", cex.main=0.8)
lines(PredLEPIB$mean, col="red")

plot(SMICTest, ylim=c(min(SMICTest,PredLESMIC$mean),max(SMICTest,PredLESMIC$mean)),
main="Comparaison entre la prédiction du lissage
exponentiel et les valeurs réelles pour le SMIC
trimestriel", cex.main=0.8)
lines(PredLESMIC$mean, col="red")

plot(TCHOFTest, ylim=c(min(TCHOFTest,PredLETCHOF$mean),max(TCHOFTest,PredLETCHOF$mean)),
main="Comparaison entre la prédiction du lissage
exponentiel et les valeurs réelles pour le taux
de chômage trimestriel", cex.main=0.8)
lines(PredLETCHOF$mean, col="red")
```

Les graphiques de la figure 11 nous montrent des résultats mitigés. Pour le PIB et le SMIC, les prédictions suivent la forme de la série mais en sont éloignées. Pour le taux de chômage des femmes, la méthode de lissage utilisée est un lissage exponentiel simple, ce qui nous donne donc des prédictions constantes soit de mauvaise qualité.

Nous résumons dans le tableau suivant les résultats obtenus pour chaque série estimée par un lissage exponentiel.

Variable	Tendance	Saisonnalité	Argument model	AIC
MSE	linéaire	multiplicative	ZAM	4033.45

Variable	Tendance	Saisonnalité	Argument model	AIC
PIB	linéaire	absente	ZAN	2053.15
SMIC	linéaire	additive	ZAA	-84.96
TCHOF	absente	absente	ZNN	204.37

2.4 Modèles ARMA

2.4.1 Définition

Les modèles **ARMA(p,q)** sont une autre famille de modèles permettant d'estimer une série temporelle. Il est divisé en deux parties : une partie autorégressive **AR** auquel est associé un ordre p qui donne le nombre de valeurs passées qui vont être utiles dans la prédiction, et une partie moyennes mobiles **MA** qui permet de de prendre en compte les q innovations de la série dans le futur.

L'une des propriétés des processus ARMA est qu'ils sont utilisés pour modéliser des séries stationnaires, donc par extension des séries qui ne possèdent ni tendance ni saisonnalité. Afin de modéliser des séries non stationnaires, on généralise les processus ARMA en processus **ARIMA(p,d,q)**, d représentant l'ordre de différenciation de la série. Les séries saisonnières sont elles modélisées par des processus $SARIMA(p,d,q)(P,D,Q)_s$ qui modélisent des séries avec une saisonnalité de période s.

Comme pour le lissage exponentiel, nous ne calculons pas nous-mêmes les ordres des processus. Pour cela, la fonction *auto.arima* du package **forecast** nous a été très utile. Elle permet en effet de trouver les ordres du processus qui optimisent un critère défini à l'avance et de calculer un modèle avec ces coefficients. Nous avons choisi d'optimiser l'**AICc**(Akaike Information Criterion with correction), pour els raisons évoquées dans la partie 2.3.1

```
ARIMAMSE<-auto.arima(MSETrain, ic="aicc")
print(ARIMAMSE)
```

```
## Series: MSETrain
## ARIMA(0,1,1)(0,1,1)[4]
##
## Coefficients:
##           ma1     sma1
##       -0.1986  -0.3857
## s.e.   0.1014   0.0929
##
## sigma^2 estimated as 7.396e+14:  log likelihood=-1834.54
## AIC=3675.09   AICc=3675.34   BIC=3682.87
```

```
PredARIMAMSE<- forecast(ARIMAMSE, h=6)
plot(MSETest, main="Comparaison entre le modèle SARIMA et les données de
    validation pour la masse salariale trimestrielle")
lines(PredARIMAMSE$mean, col="red")
```

Pour la MSE, on obtient par exemple un modèle $SARIMA(0,1,1)(0,1,1)_4$ ainsi qu'un AICc de

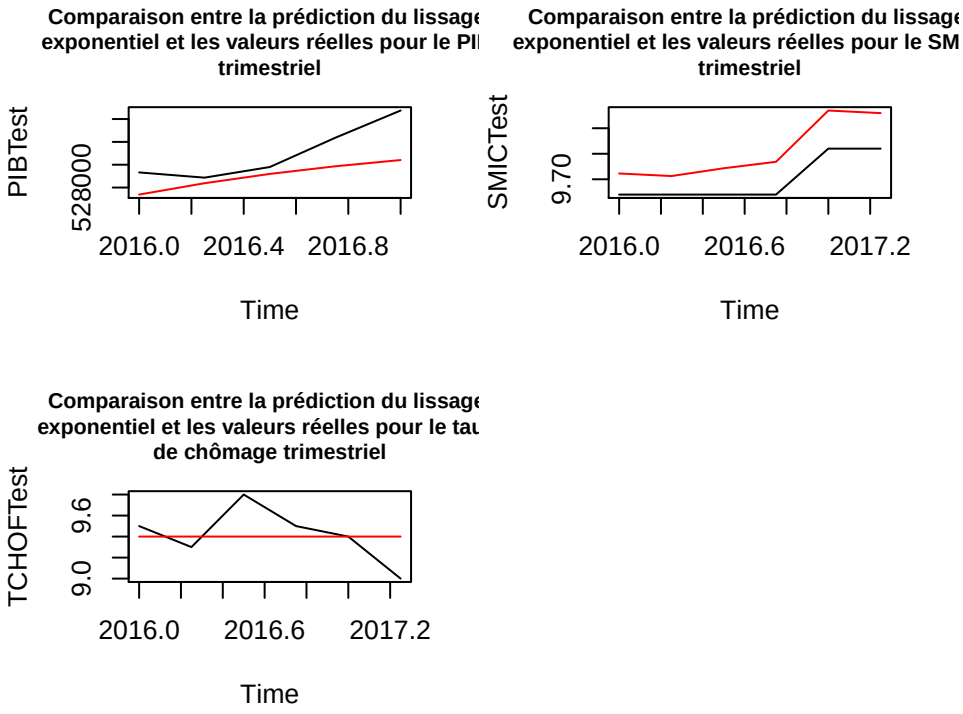


Figure 11:

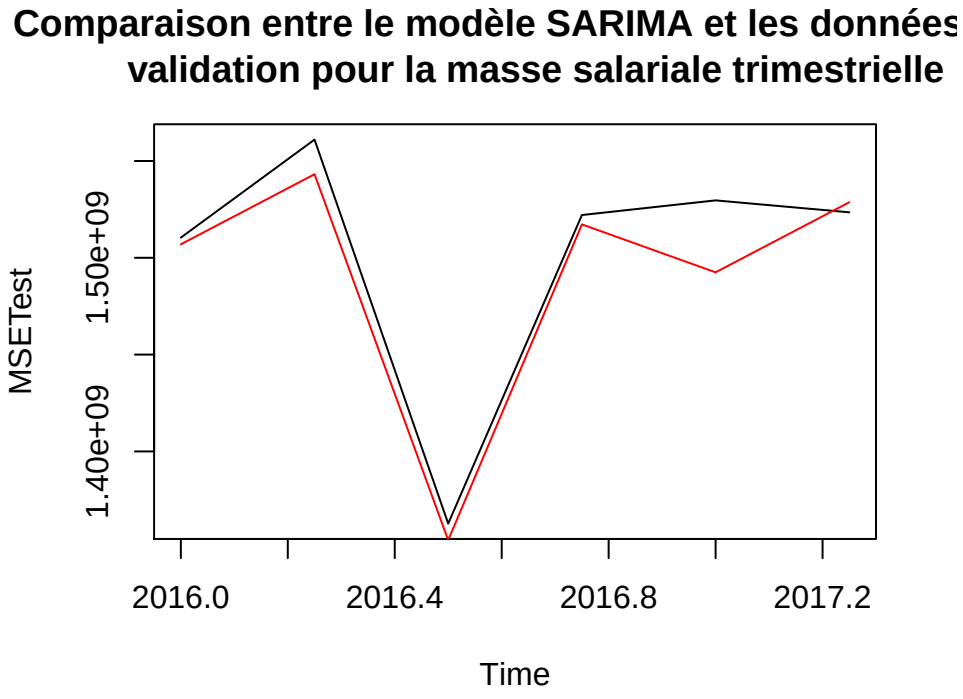


Figure 12:

3896.01. La figure 12 nous donne des prédictions d'assez bonne qualité mais qui semblent moins bonnes que celles obtenues par lissage exponentiel.

2.4.2 Résultats obtenus

```
par(mfrow=c(2,2))
ARIMAPIB<-auto.arima(PIBTrain, ic="aicc", seasonal=F)
print(ARIMAPIB)
```

```
## Series: PIBTrain
## ARIMA(2,1,0) with drift
##
## Coefficients:
##          ar1     ar2      drift
##       0.4701  0.1596  1585.8365
## s.e.  0.0966  0.0966   456.3009
##
## sigma^2 estimated as 3153980:  log likelihood=-915.48
## AIC=1838.95   AICc=1839.36   BIC=1849.49
```

```
PredARIMAPIB<- forecast(ARIMAPIB, h=5)
plot(PIBTest, ylim=c(min(PIBTest,PredARIMAPIB$mean),max(PIBTest,PredARIMAPIB$mean)),
main="Comparaison entre le modèle SARIMA et les données de
    validation pour le PIB trimestriel", cex.main=0.8)
lines(PredARIMAPIB$mean, col="red")

ARIMASMIC<-auto.arima(SMICTrain, ic="aicc")
print(ARIMASMIC)
```

```
## Series: SMICTrain
## ARIMA(1,0,0)(1,1,0)[4] with drift
##
## Coefficients:
##          ar1     sar1   drift
##       0.8645  -0.3643  0.0486
## s.e.  0.0514   0.0961  0.0080
##
## sigma^2 estimated as 0.003998:  log likelihood=134.95
## AIC=-261.89   AICc=-261.47   BIC=-251.47
```

```
PredARIMASMIC<- forecast(ARIMASMIC, h=6)
plot(SMICTest, ylim=c(min(SMICTest,PredARIMASMIC$mean),max(SMICTest,PredARIMASMIC$mean)),
main="Comparaison entre le modèle SARIMA et les données de
    validation pour le SMIC trimestriel", cex.main=0.8)
lines(PredARIMASMIC$mean, col="red")

ARIMATCHOF<-auto.arima(TCHOFTrain, ic="aicc", seasonal=F)
print(ARIMATCHOF)
```

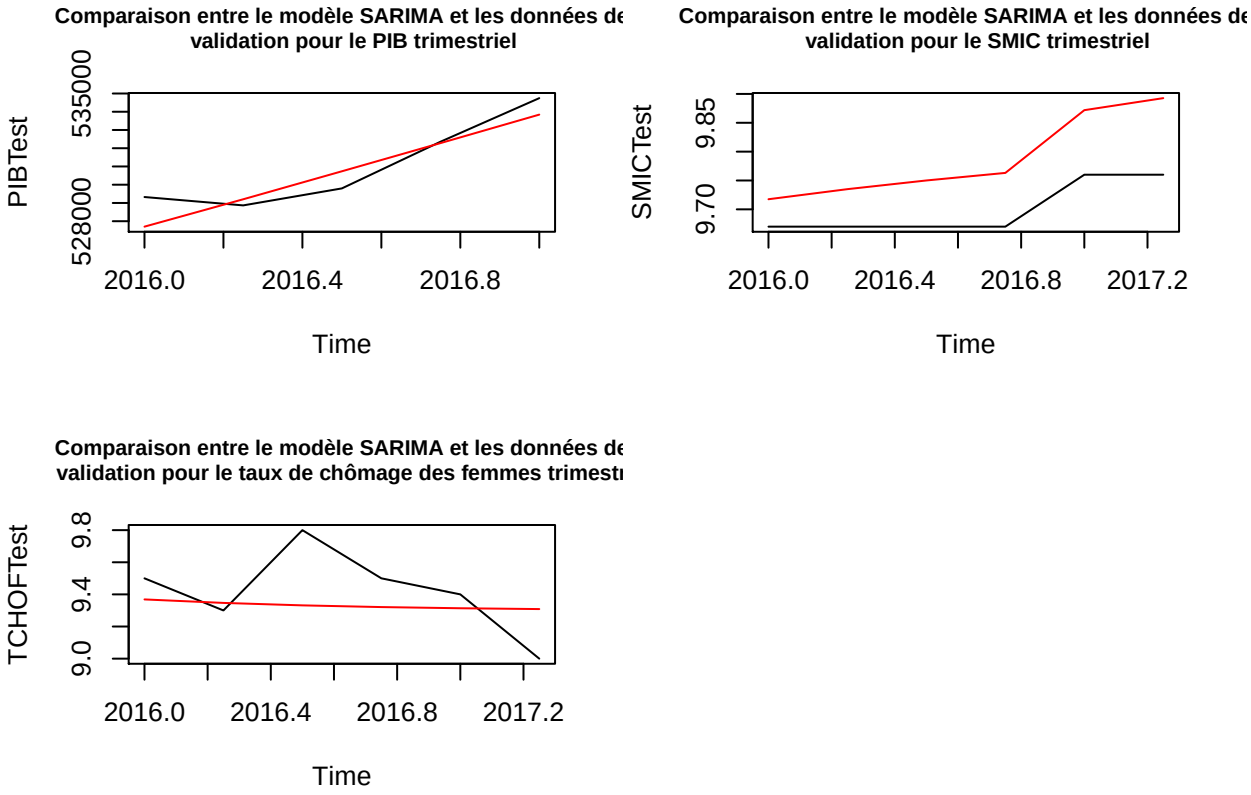


Figure 13:

```
## Series: TCHOFTrain
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.6974  -0.5005
## s.e.  0.1669   0.1935
##
## sigma^2 estimated as 0.06173:  log likelihood=-1.76
## AIC=9.52   AICc=9.76   BIC=17.42
```

```
PredARIMATCHOF<- forecast(ARIMATCHOF, h=6)
plot(TCHOFTest, main="Comparaison entre le modèle SARIMA et les données de
    validation pour le taux de chômage des femmes trimestriel", cex.main=0.8)
lines(PredARIMATCHOF$mean, col="red")
```

Comme pour le lissage exponentiel, nous résumons les résultats obtenus dans un tableau pour plus de lisibilité. Le PIB et le taux de chômage des femmes ne comportent pas de partie saisonnière car comme vu dans les parties 1.3 et 1.5 on ne constate pas de saisonnalité dans l'analyse descriptive de la série. On peut également voir sur la figure 13 que les prédictions du PIB semblent de bien meilleure qualité

Variable	Ordre du processus	AICc

Variable	Ordre du processus	AICc
MSE	(0,1,1)(0,1,1)	3675.34
PIB	(2,1,0)	1839.36
SMIC	(1,0,0)(1,1,0)	-261.47
TCHOF	(0,1,1)	9.76

2.5 Comparaison des différents modèles

Une fois que nous avons construit les deux types de modèles pour chacune des variables, nous souhaitons les comparer pour savoir quel modèle est le plus efficace pour prédire chacune des variables. Pour cela, les EQM, calculant l'erreur de prédiction, de chacun des modèles sont synthétisées dans le tableau suivant. L'AICc ne peut pas être utilsié ici car les méthodes à comparer sont différentes. Il n'est donc pas sûr que la méthode utilisée pour calculer la vraisemblance soit la même.

```
resultats<-matrix(nrow=4, ncol=2, dimnames = list(c("MSE", "PIB", "SMIC", "TCHOF"),
                                                  c("lissage", "ARMA")))
resultats[1,1] = EQM(MSETest, PredLEMSE$mean)
resultats[2,1] = EQM(PIBTest, PredLEPIB$mean)
resultats[3,1] = EQM(SMICTest, PredLESMIC$mean)
resultats[4,1] = EQM(TCHOFTest, PredLETCHOF$mean)
resultats[1,2] = EQM(MSETest, PredARIMAMSE$mean)
resultats[2,2] = EQM(PIBTest, PredARIMAPIB$mean)
resultats[3,2] = EQM(SMICTest, PredARIMASMIC$mean)
resultats[4,2] = EQM(TCHOFTest, PredARIMATCHOF$mean)
resultats
```

```
##            lissage         ARMA
## MSE   2.592281e+14 3.060118e+14
## PIB   5.885443e+06 8.892292e+05
## SMIC  3.371267e-03 8.609164e-03
## TCHOF 5.833300e-02 6.223076e-02
```

Nous nous rendons compte que le lissage a une EQM plus faible pour la masse salariale (notre variable d'intérêt), ainsi que pour le SMIC et le taux de chômage des femmes. En ce qui concerne le PIB, le modèle ARIMA est plus performant. Cette analyse va nous servir par la suite, comme expliqué au début de la partie 2

2.6 Estimation de la valeur manquante du PIB

Contrairement aux autres variables, nous n'avons à notre disposition pour le PIB que les valeurs jusqu'au premier trimestre de 2017. Ceci nous impose de négliger la dernière valeur de toutes les autres séries pour que toutes les variables soient étudiées sur la même période. Pour éviter ce problème, nous décidons d'estimer la variable du PIB pour le 2e trimestre de 2017. Afin de faire

cela, nous allons utiliser la valeur estimée par le modèle SARIMA correspondant à la variable PIB, étant donné que c'est celui qui donnait les meilleures prédictions.

```
PredARIMAPIB<- forecast(ARIMAPIB, h=6)
new.value <- PredARIMAPIB$mean[6]
PIBTest<-ts(c(PredARIMAPIB$mean, new.value), start = 2016, end = c(2017, 2), frequency=4)
```

3 Modélisation ARMA avec variables exogènes

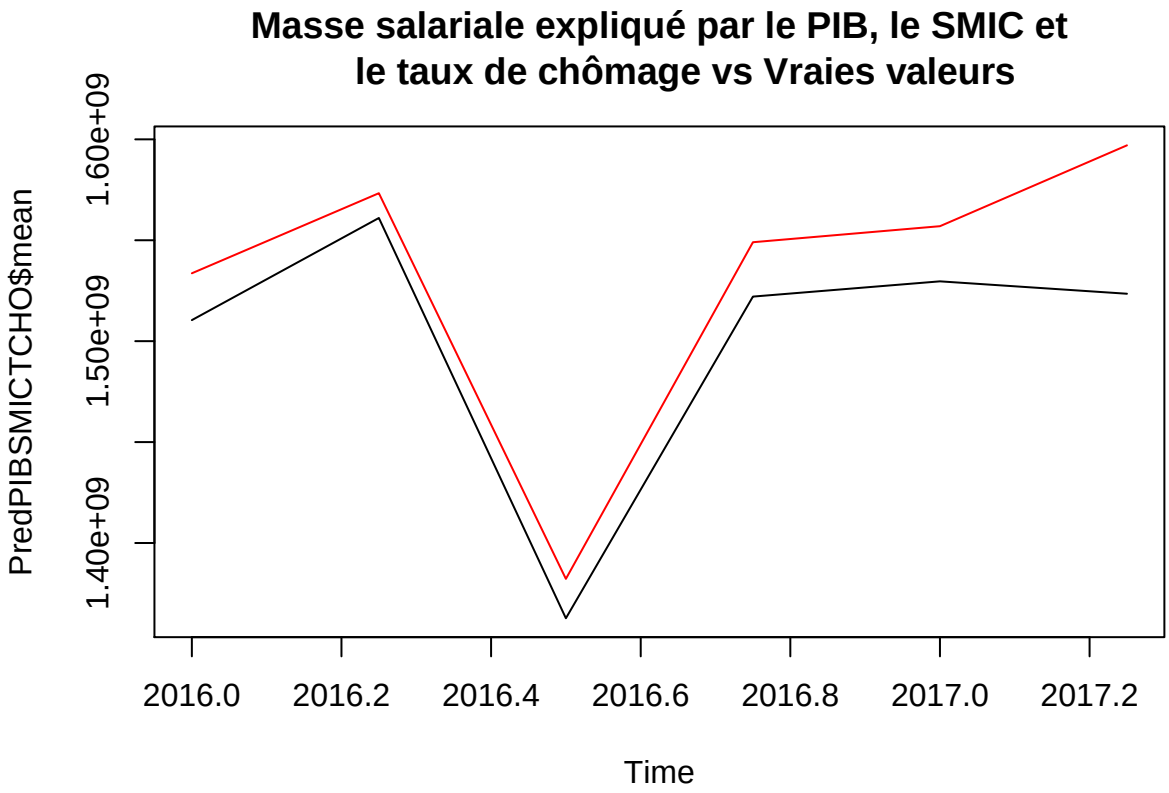
3.1 Définition

Maintenant que nous avons modélisé chaque série individuellement, nous souhaitons savoir s'il est possible d'améliorer la qualité de prédiction de la série MSE trimestrielle à l'aide des autres variables à notre disposition. Pour ce faire, nous allons construire des modèles SARIMA prenant en compte des variables exogènes.

$Y_t = \beta_0 + \beta_1 X_{1t} + ... + \beta_k X_{kt} + \epsilon_t$

Y_t est la variable à modéliser. X_i pour $i = 1, ..., k$ correspond à la $i$ème variable exogène. β_i pour i allant de $i = 0, ..., k$ correspond aux coefficients d'une régression linéaire. Enfin, le résidu ϵ_t suit un processus de type ARMA.

```
#PIB & SMIC & TCHO
SARIMAPIBSMICTCHO <- auto.arima(MSETrain, xreg = cbind(PIBTrain, SMICTrain, TCHOFTrain))
PredPIBSMICTCHO <- forecast(SARIMAPIBSMICTCHO, xreg = cbind(PIBTest, SMICTest, TCHOFTest))
plot(PredPIBSMICTCHO$mean, col="red",
     ylim=c(min(MSETest,PredPIBSMICTCHO$mean), max(MSETest,PredPIBSMICTCHO$mean)),
     main = "Masse salariale expliqué par le PIB, le SMIC et
     le taux de chômage vs Vraies valeurs")
lines(MSETest)
```




```
EQM(PredPIBSMICTCHO$mean, MSETest)
```

```
## [1] 1.324892e+15
```

Ici, les résidus suivent un SARIMA(0,0,0)(0,1,0)[4], et le modèle possède 3 variables exogènes : le coefficient correspondant à la variable PIB est $\beta_1 = 706.0045$, celui correspondant à la variable SMIC est $\beta_2 = 209266766$ et enfin celui correspondant au taux de chômage est $\beta_3 = -1644706$

3.2 Résultats obtenus

Nous allons désormais nous intéresser à la construction des différents modèles prenant en compte 1 variable exogène (3 modèles), 2 variables exogènes (3 modèles) et 3 variables exogènes (1 modèle). La qualité de ces 7 modèles est representée dans le tableau ci-dessous.

Variable	Ordre du processus	AICc
Aucune	(0,1,1)(0,1,1)	3675.34
PIB	(1,0,0)(2,1,0)	3715.99
SMIC	(0,1,0)(0,1,0)	3688.92
TCHOF	(0,0,0)(1,1,0)	3809.82
PIB & SMIC	(0,0,0)(0,1,0)	3807.37
PIB & TCHOF	(1,0,0)(0,1,0)	3721.36
SMIC & TCHOF	(0,1,0)(0,1,0)	3690.57
COMPLET	(0,0,0)(0,1,0)	3809.5

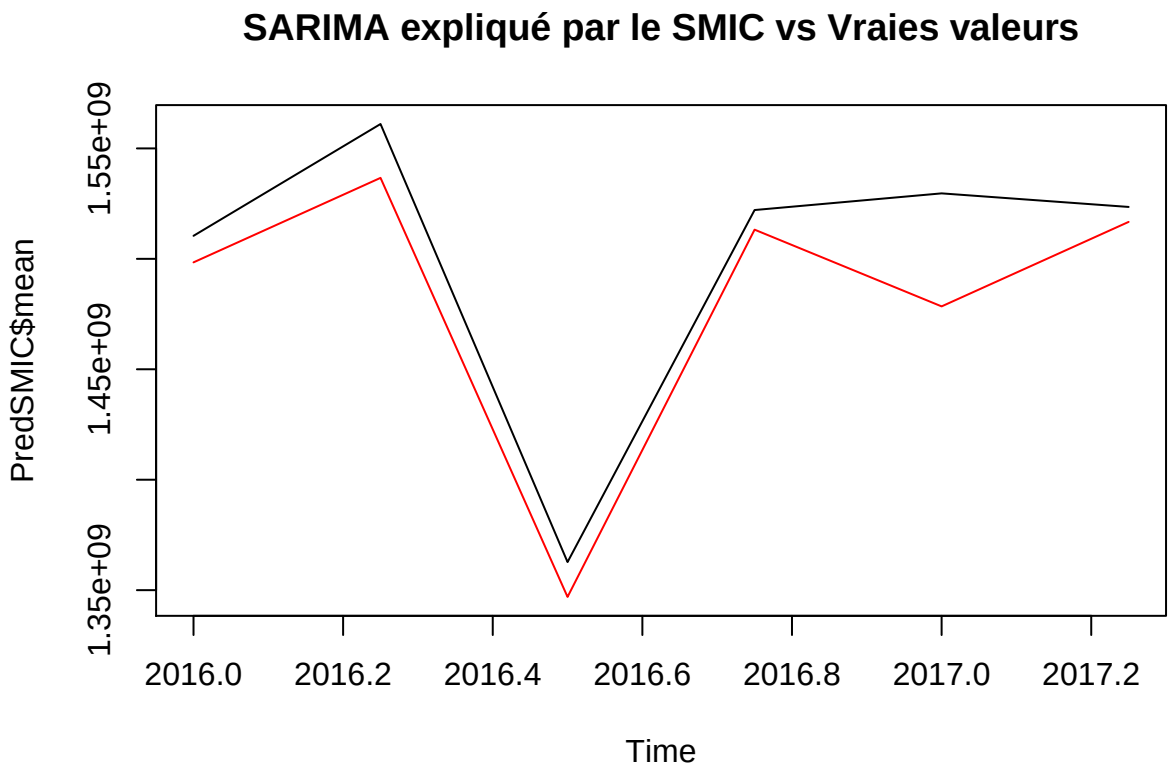


Figure 14:

On se rend compte qu'aucun modèle ARIMA ne prenant en compte des variables exogènes n'a une qualité meilleure que celui ne prenant en compte aucune variable exogène (en comparant les AIC corrigés). Si on omet le modèle sans variable exogène, le meilleur modèle prenant en compte au moins une variable exogène est celui prenant en compte le SMIC (dans la 14).

```
#SMIC
SARIMASMIC <- auto.arima(MSETrain, xreg = SMICTrain)
PredSMIC <- forecast(SARIMASMIC, xreg = SMICTest)
plot(PredSMIC$mean, col="red",
     ylim=c(min(MSETest,PredSMIC$mean), max(MSETest,PredSMIC$mean)),
     main = "SARIMA expliqué par le SMIC vs Vraies valeurs")
lines(MSETest)
```

```
EQM(PredSMIC$mean, MSETest)
```

```
## [1] 6.219316e+14
```

4 Modélisation VAR

4.1 Définition des modèles

4.1.1 Ecriture

Dans la partie précédente, nous avons mis en place des modèles permettant de modéliser et prédire la valeur de la masse salariale uniquement. Cependant, il pourrait être pertinent de tenter de modéliser toutes les variables en même temps. C'est ce que les modèles VAR nous permettent de faire.

Un modèle VAR s'écrit sous la forme suivante :

$y_t = \sum_{i=1}^{p} A_i y_{t-i} + u_t$

A_i représentent les matrices de coefficients du modèle pour un ordre i et u_t une matrice K-dimensionnelle composée des résidus du modèle (indépendants et identiquement distribués). Enfin, p correspond à l'ordre du modèle, qui est en fait le nombre de valeurs du passé prises en compte pour calculer la valeur présente.

4.1.2 Hypothèses

4.1.2.1 Stabilité du modèle

Pour que le modèle soit valide, nous devons vérifier que l'hypothèse de stabilité est bien respecté. Cette dernière permet d'assurer que les différentes séries générées par le modèle sont stationnaires. Pour vérifier si un processus VAR est stable, nous devons calculer les valeurs propres de la matrice des coefficients suivantes :

$$A = \begin{bmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix}$$

Si les modules des valeurs propres de A sont inférieures à 1, alors le processus VAR est stable.

4.1.2.2 Hypothèses sur les résidus

Pour que le modèle soit valide, certaines conditions sur les résidus doivent également être validées. Il s'agit des suivantes :

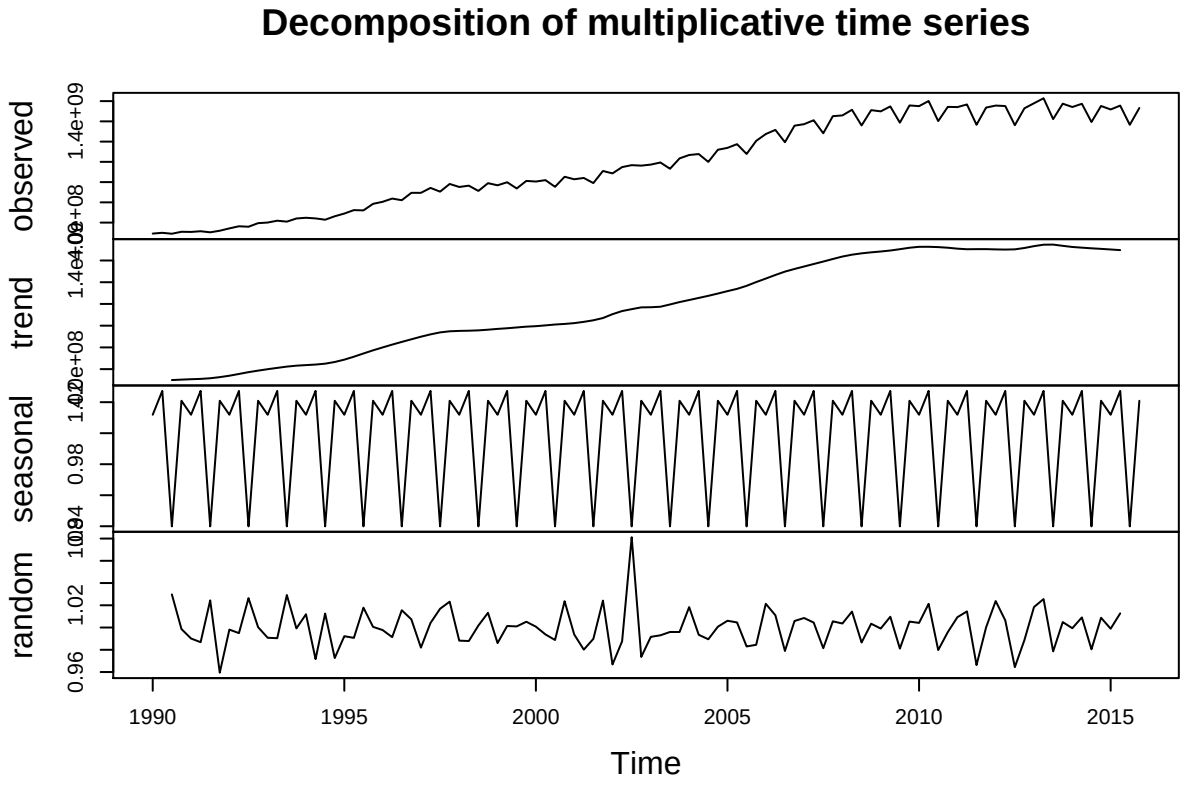
- Homoscédasticité
- Normalité
- Absence d'auto-corrélations et de corrélations croisées

4.2 Transformation des séries

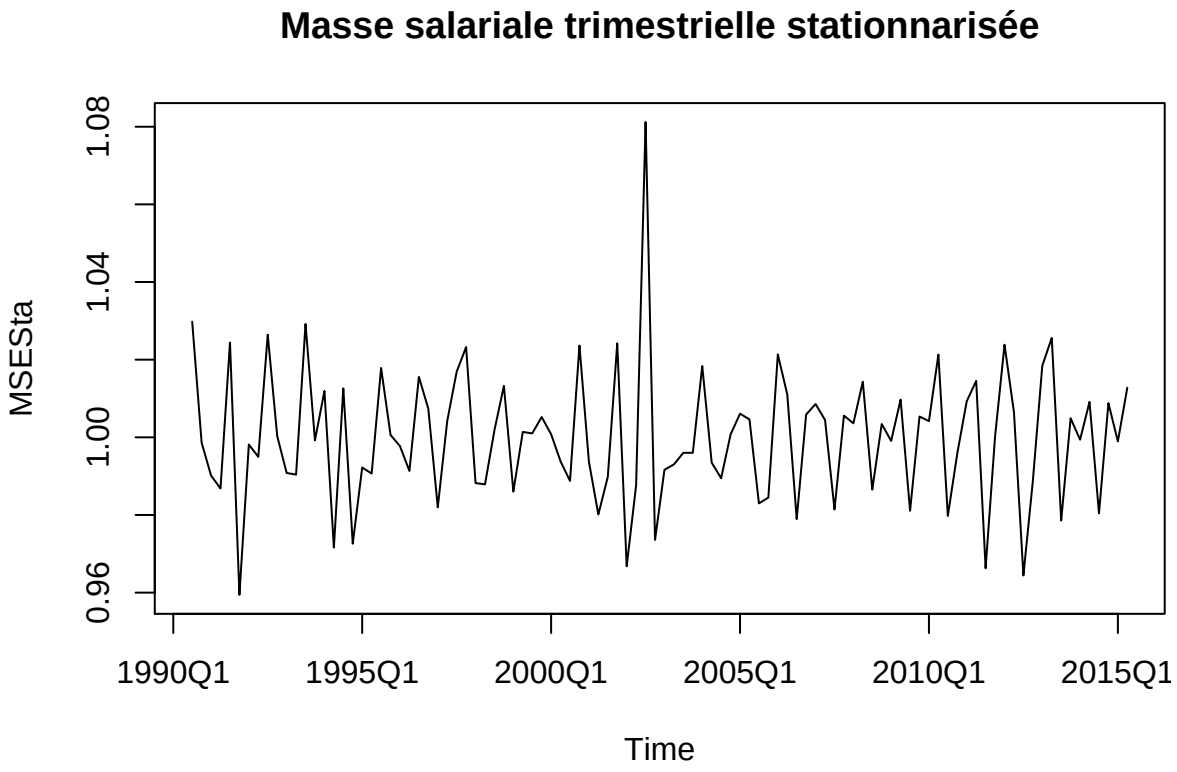
Nous allons maintenant transformer les séries pour les rendre stationnaires, afin de pouvoir appliquer les modèles VAR ensuite. Afin de stationnariser les séries, nous utiliserons la fonction decompose qui permet de découper la série en trois : la tendance, la saisonnalité et les résidus, afin de pouvoir ensuite travailler avec les résidus. Nous ne stationnariserons que les échantillons d'apprentissage.

4.2.1 Masse salariale

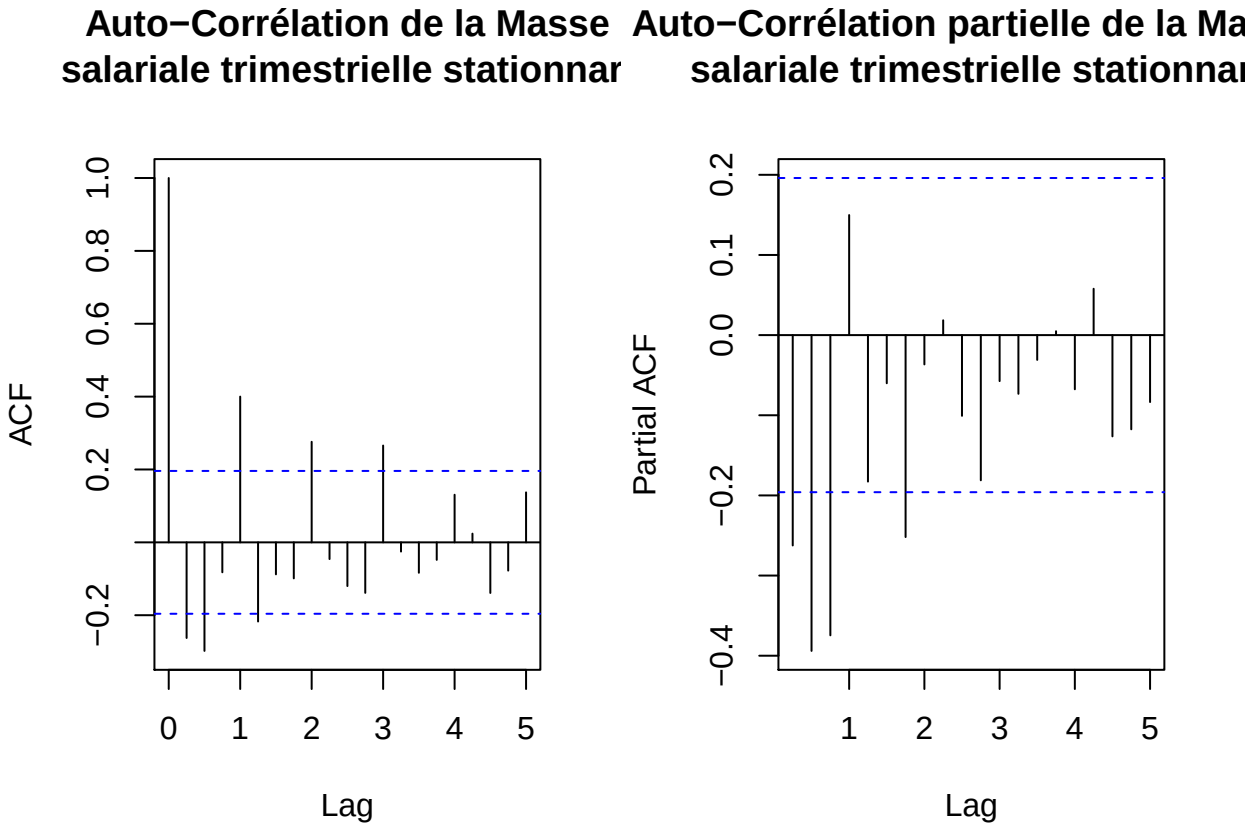
```
MSETrain <- window(MSE, end=c(2015,4))
MSETest <- window(MSE, start=2016)
plot(decompose(MSETrain, "multiplicative"))
```




```
MSESta <- na.omit(decompose(MSETrain, "multiplicative")$random)
MSETrendTest<-window(decompose(MSETrain, "multiplicative")$trend)
MSESeasonalTest<-window(decompose(MSETrain, "multiplicative")$seasonal)
plot(MSESta, main="Masse salariale trimestrielle stationnarisée", xaxt="n")
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1",
```




```
par(mfrow=c(1,2))
acf(MSESta, main="Auto-Corrélation de la Masse
    salariale trimestrielle stationnarisée")
pacf(MSESta, main="Auto-Corrélation partielle de la Masse
     salariale trimestrielle stationnarisée")
```




```
  par(mfrow=c(1,1))
  kpss.test(MSESta)
```

```
## Warning in kpss.test(MSESta): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  MSESta
## KPSS Level = 0.017376, Truncation lag parameter = 2, p-value = 0.1
```

```
  adf.test(MSESta)
```

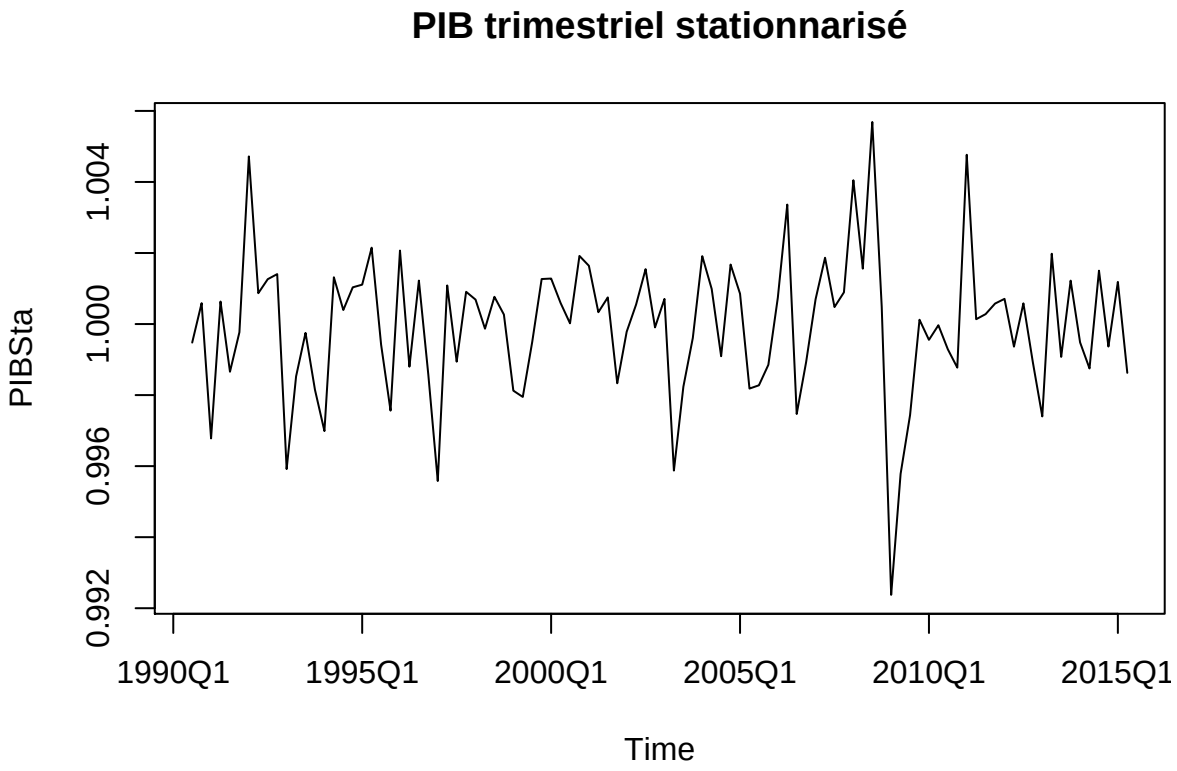
```
## Warning in adf.test(MSESta): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  MSESta
## Dickey-Fuller = -6.3219, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

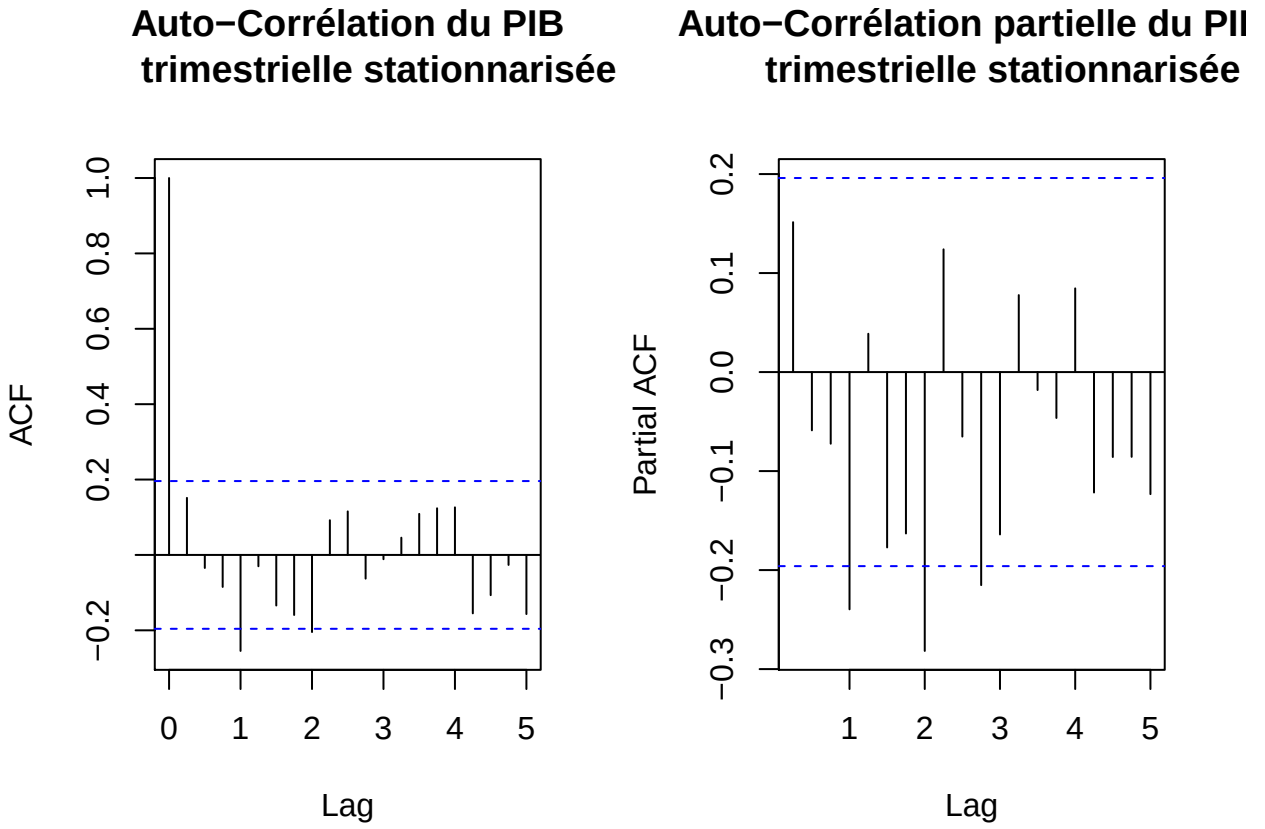
Nous nous intéressons aux ACF, PACF et test de KPSS afin de vérifier si les résidus obtenus à l'aide de la fonction decompose sont stationnaires. Bien que l'ACF et la PACF nous mettent en garde d'une possible non stationnarité de la série, la p-value des tests de KPSS et Dickey Fuller augmenté nous amène à confirmer que notre série est désormais stationnarisée (avec un seuil de confiance à 5% pour les deux tests effectués).

4.2.2 PIB

```
PIBTrain <- window(PIB, end=c(2015,4))
PIBTest <- window(PIB, start=2016)
PIBSta <- na.omit(decompose(PIBTrain, "multiplicative")$random)
plot(PIBSta, main="PIB trimestriel stationnarisé", xaxt="n")
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1",
```




```
par(mfrow=c(1,2))
acf(PIBSta, main="Auto-Corrélation du PIB
    trimestrielle stationnarisée")
pacf(PIBSta, main="Auto-Corrélation partielle du PIB
    trimestrielle stationnarisée")
```




```
  par(mfrow=c(1,1))
  kpss.test(PIBSta)
```

```
## Warning in kpss.test(PIBSta): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  PIBSta
## KPSS Level = 0.027524, Truncation lag parameter = 2, p-value = 0.1
```

```
  adf.test(PIBSta)
```

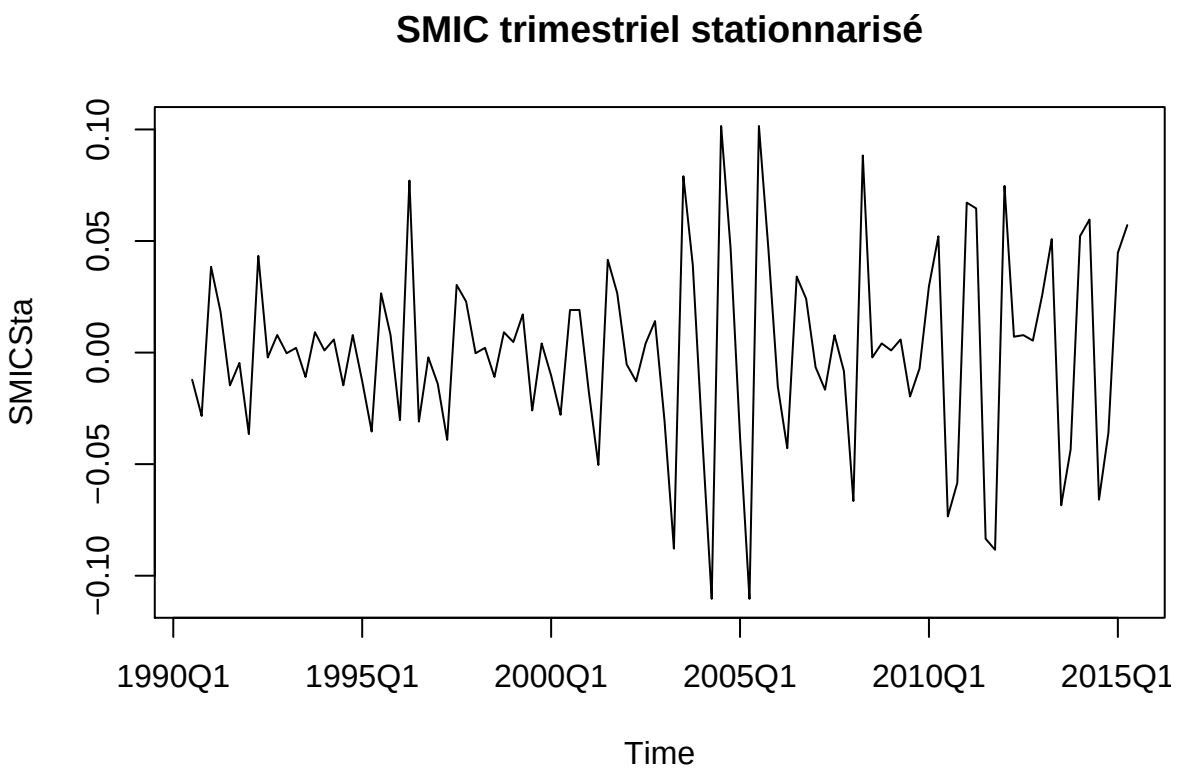
```
## Warning in adf.test(PIBSta): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  PIBSta
## Dickey-Fuller = -5.0084, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

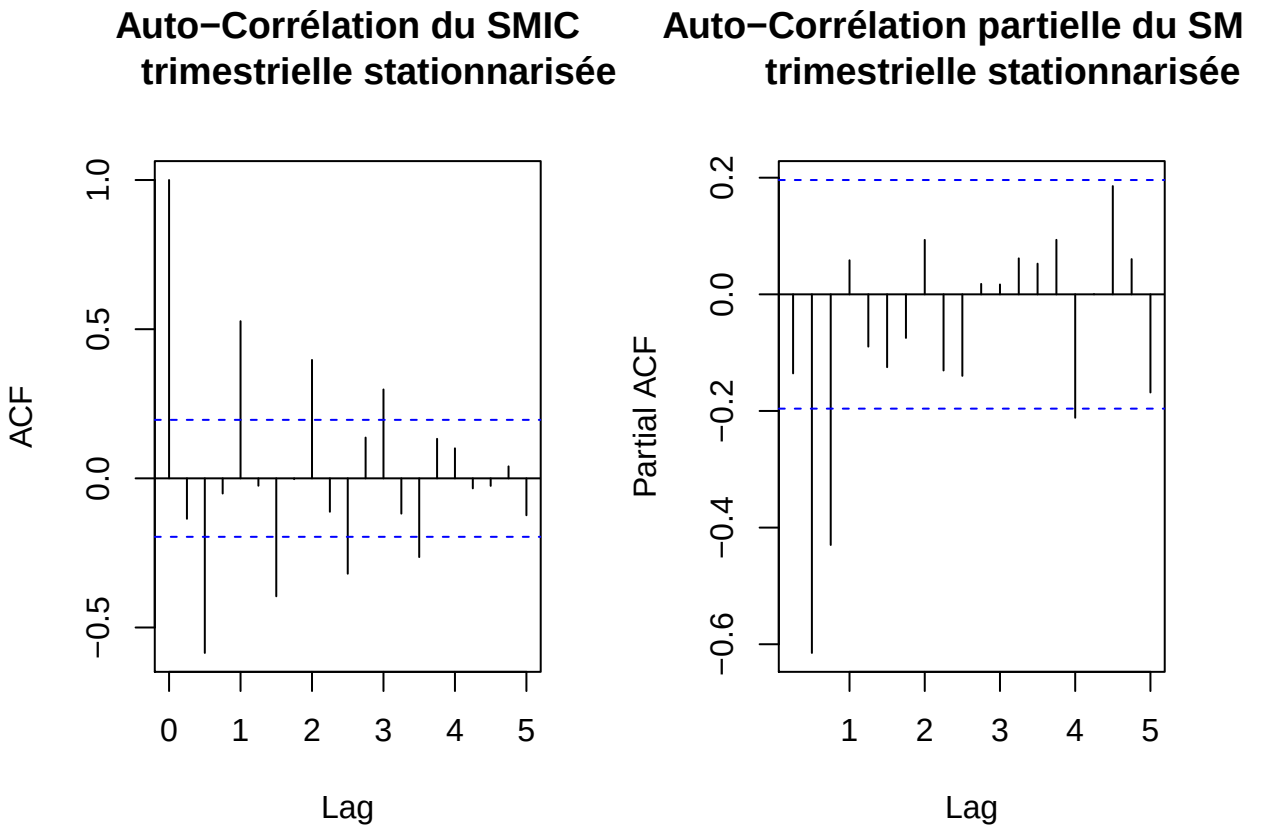
Nous nous intéressons aux ACF, PACF, test de KPSS et test de Dickey Fuller augmenté afin de vérifier si les résidus obtenus à l'aide de la fonction decompose sont stationnaires. Au regard de ces différentes informations, nous pouvons conclure à la stationnarité des résidus.

4.2.3 SMIC

```
SMICTrain <- window(SMIC, end=c(2015,4))
SMICTest <- window(SMIC, start=2016)
SMICSta <- na.omit(decompose(SMICTrain)$random)
plot(SMICSta, main="SMIC trimestriel stationnarisé", xaxt="n")
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1",
```




```
par(mfrow=c(1,2))
acf(SMICSta, main="Auto-Corrélation du SMIC
    trimestrielle stationnarisée")
pacf(SMICSta, main="Auto-Corrélation partielle du SMIC
    trimestrielle stationnarisée")
```




```
  par(mfrow=c(1,1))
  kpss.test(SMICSta)
```

```
## Warning in kpss.test(SMICSta): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  SMICSta
## KPSS Level = 0.043771, Truncation lag parameter = 2, p-value = 0.1
```

```
  adf.test(SMICSta)
```

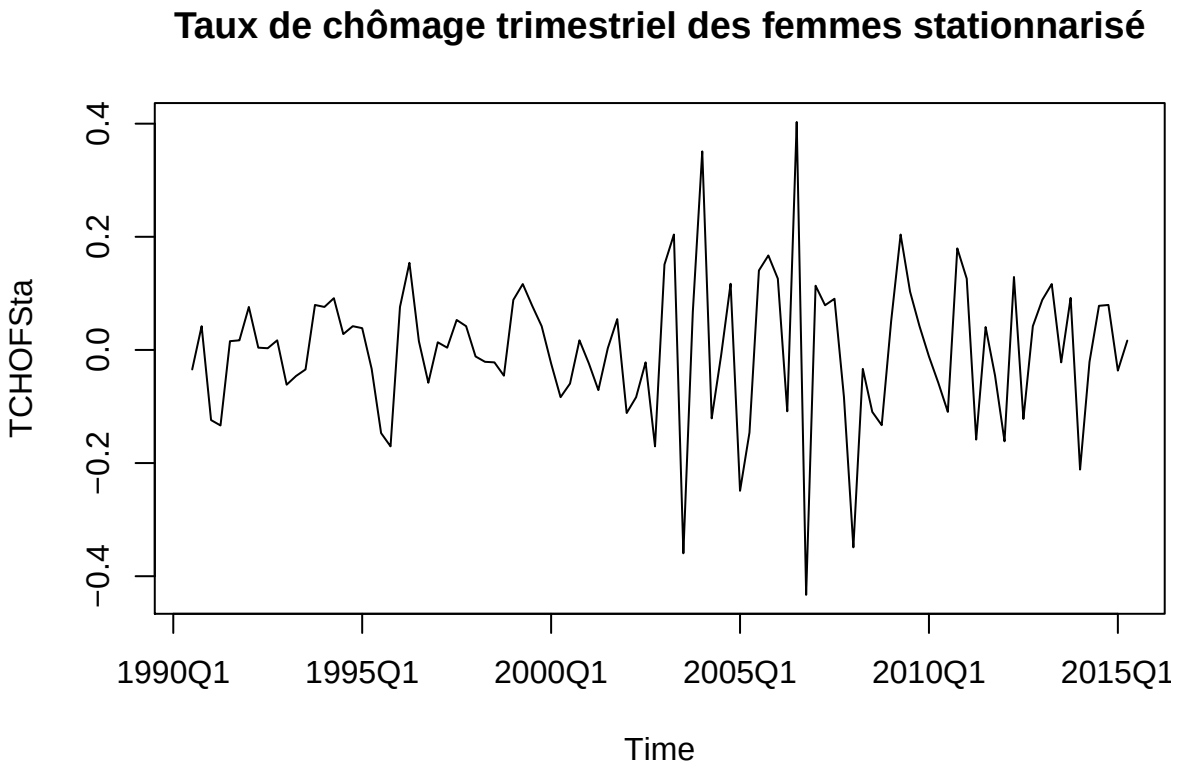
```
## Warning in adf.test(SMICSta): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  SMICSta
## Dickey-Fuller = -6.357, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

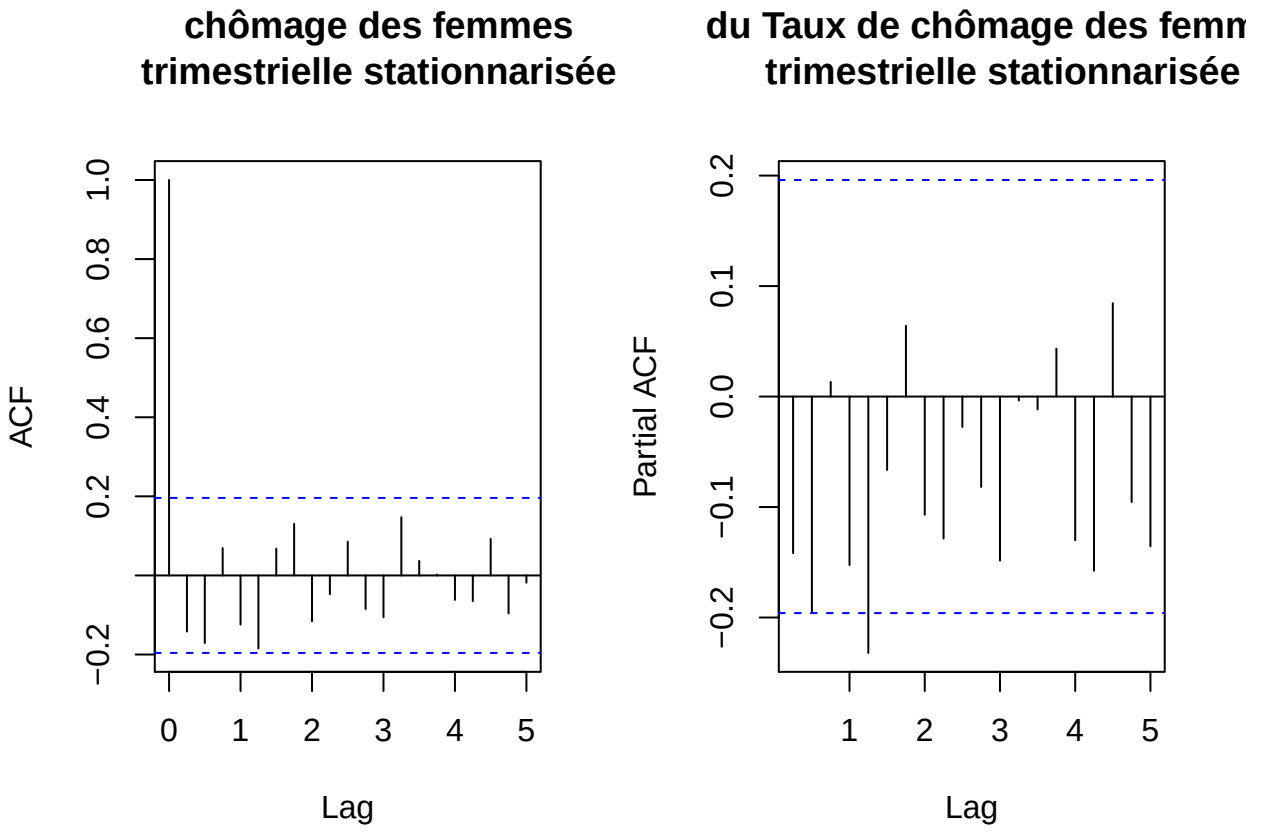
Comme pour la masse salariale, les ACF et PACF semblent montrer que la série résiduelle pourrait ne pas être stationnaire. Cependant le test de KPSS ainsi que le test de Dickey Fuller augmenté nous permettent de conclure à la stationnarité des résidus.

4.2.4 Taux de chômage des femmes

```
TCHOFTrain <- window(TCHOF, end=c(2015,4))
TCHOFTest <- window(TCHOF, start=2016)
TCHOFSta <- na.omit(decompose(TCHOFTrain)$random)
plot(TCHOFSta, main="Taux de chômage trimestriel des femmes stationnarisé", xaxt="n")
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1",
```




```
par(mfrow=c(1,2))
acf(TCHOFSta, main="Auto-Corrélation du Taux de
    chômage des femmes
    trimestrielle stationnarisée")
pacf(TCHOFSta, main="Auto-Corrélation partielle
    du Taux de chômage des femmes
    trimestrielle stationnarisée")
```




```
  par(mfrow=c(1,1))
  kpss.test(TCHOFSta)
```

```
## Warning in kpss.test(TCHOFSta): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  TCHOFSta
## KPSS Level = 0.022077, Truncation lag parameter = 2, p-value = 0.1
```

```
  adf.test(TCHOFSta)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  TCHOFSta
## Dickey-Fuller = -6.6221, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

En ce qui concerne le taux de chômage des femmes, en regardant l'ACF, PACF, le test de KPSS et le test de Dickey Fuller augmenté, on peut conclure que la série résiduelle est stationnaire.

Maintenant que toutes les séries ont été stationnarisées, nous allons pouvoir construire des modèles VAR.

4.3 Mise en place de modèles VAR avec le package vars