Package vars

Paul GUILLOTTE & Jules CORBEL

12/02/2019

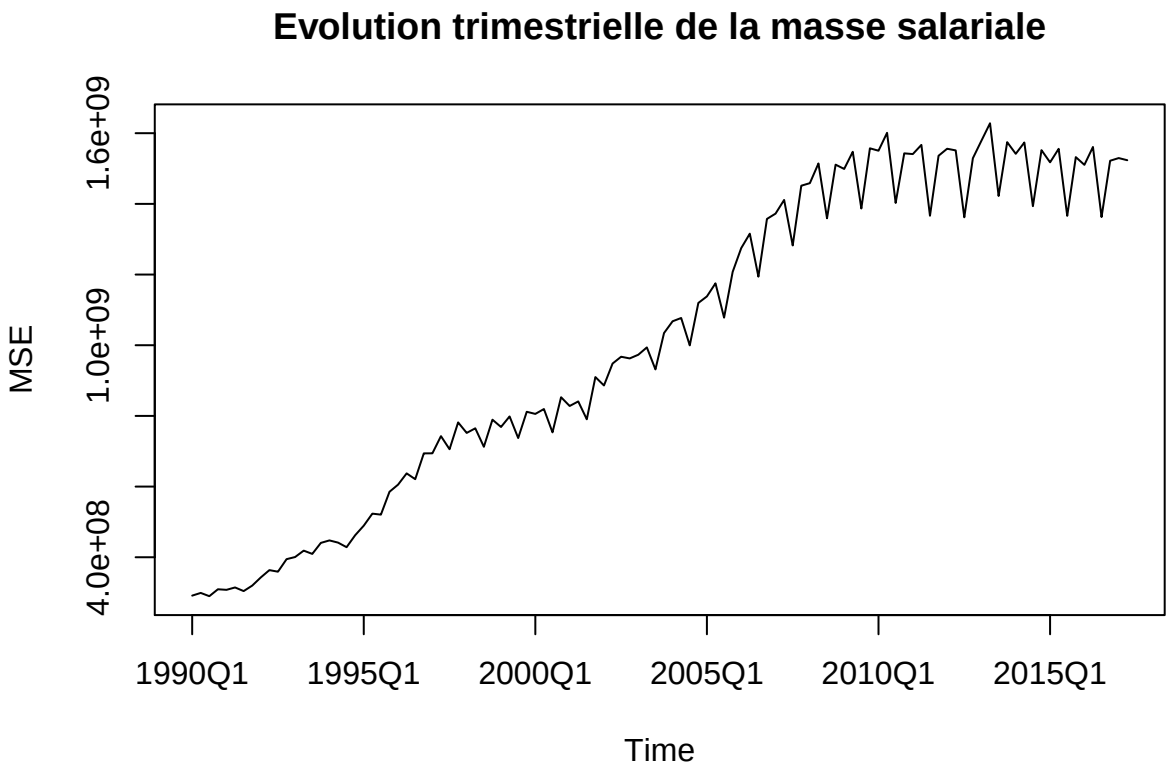
Nous nous intéresserons dans ce document à la mise en place de modèles VAR afin de prédire la masse salariale trimestrielle. Un modèle VAR, pour Vecteur AutoRégressif, a pour objectif de capturer les interdépendances entre les différentes séries temporelles à notre disposition. Ainsi, chaque variable est expliquée par ses propres valeurs passées ainsi que par les valeurs passées des autres variables du modèle.

Visualisation des séries

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.

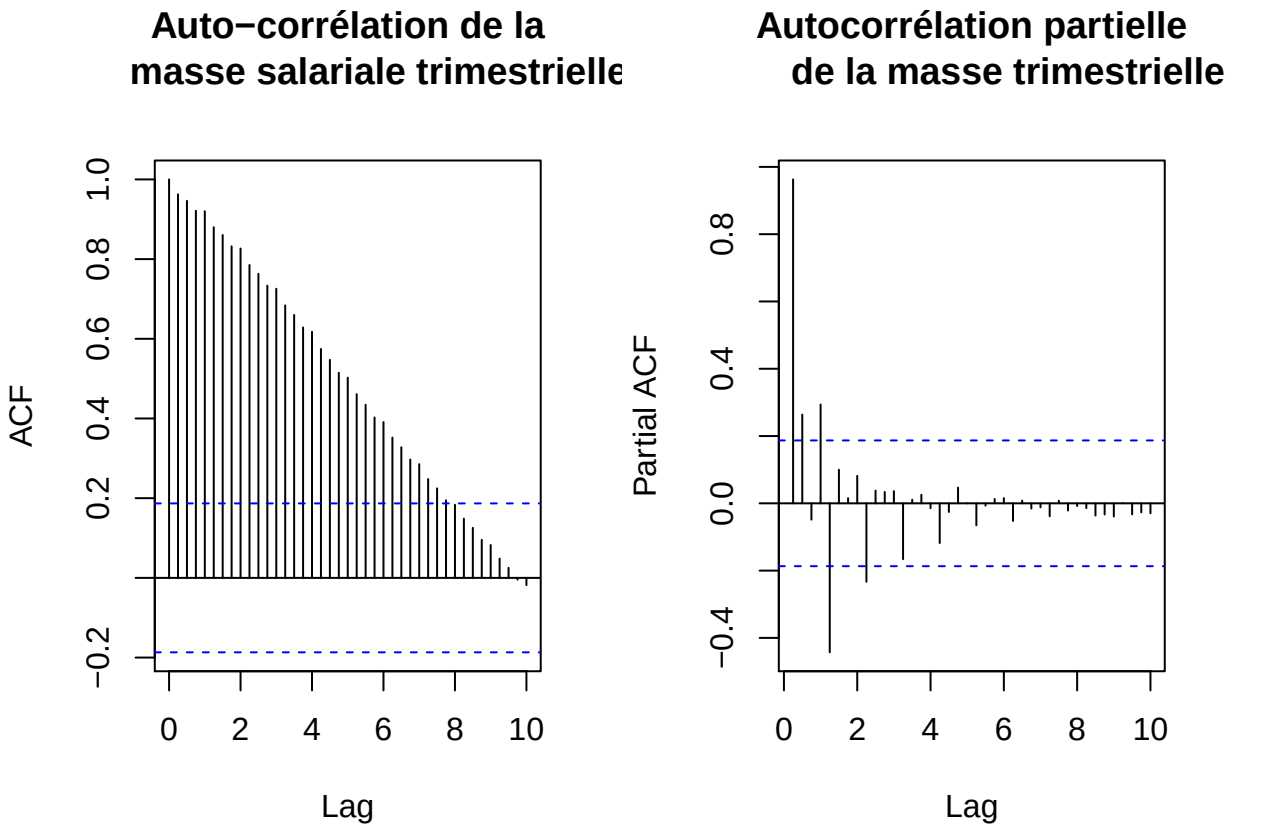
Masse salariale

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




```
par(mfrow=c(1,2))
acf(MSE, main="Auto-corrélation de la
    masse salariale trimestrielle", lag.max=40)
```

```
pacf(MSE, main="Autocorrélation partielle
     de la masse trimestrielle", lag.max=40)
```




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

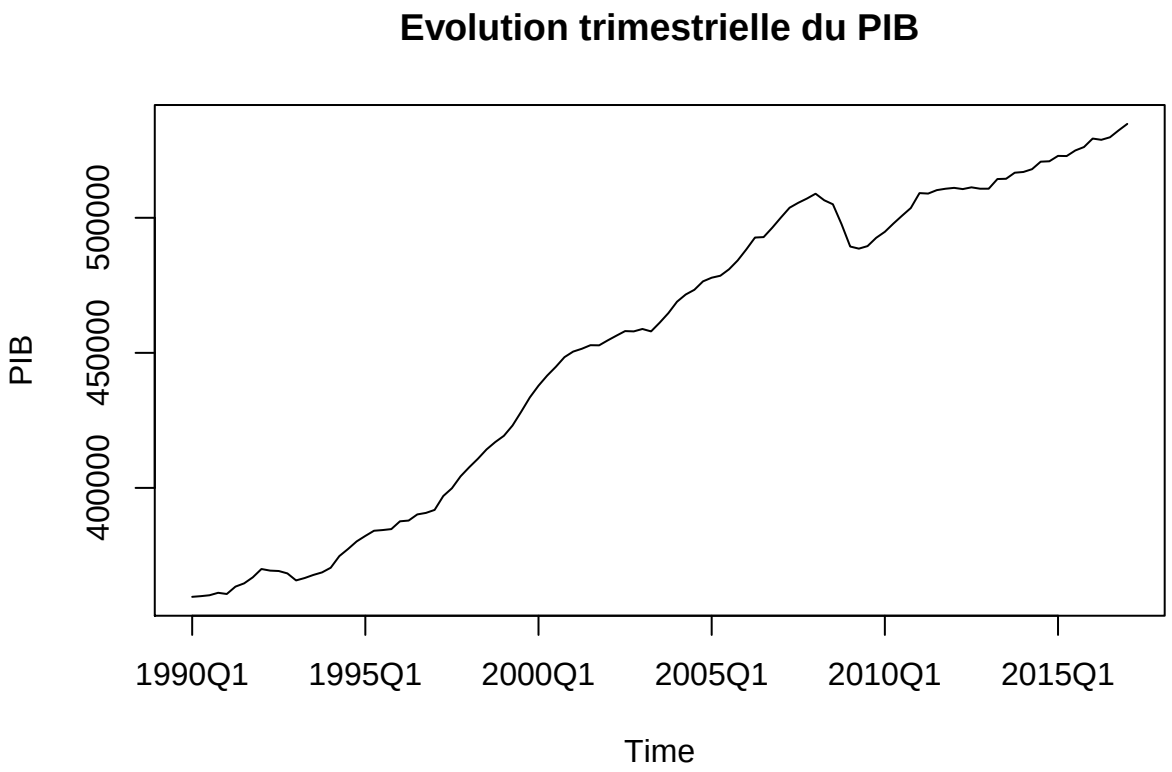
La masse salariale trimestrielle possède une composante de tendance de 1990 à 2010. La série tend par la suite à stagner. Nous remarquons également une saisonnalité sur cette série, qui est de plus en plus marquée à mesure que le temps passe.

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 fonction ACF qui décroît régulièrement. Nous effectuons également un

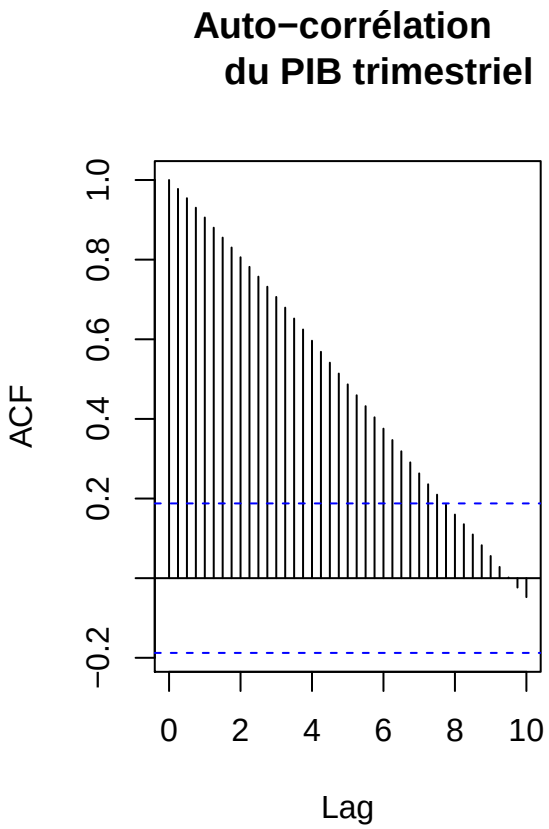
test de KPSS (test de stationnarité) servant à vérifier si la série est stationnaire ou non (sous l'hypothèse H_0 la série est stationnaire, et sous l'hypothèse H_1 elle ne l'est pas). La série est dite stationnaire si ses propriétés statistiques (espérance, variance et auto-corrélation) sont fixes au 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.

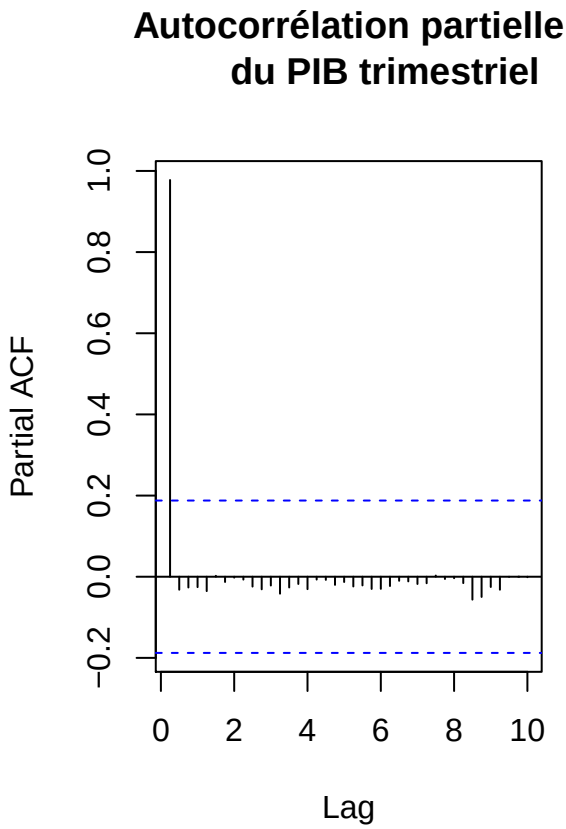
PIB

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




```
par(mfrow=c(1,2))
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
```

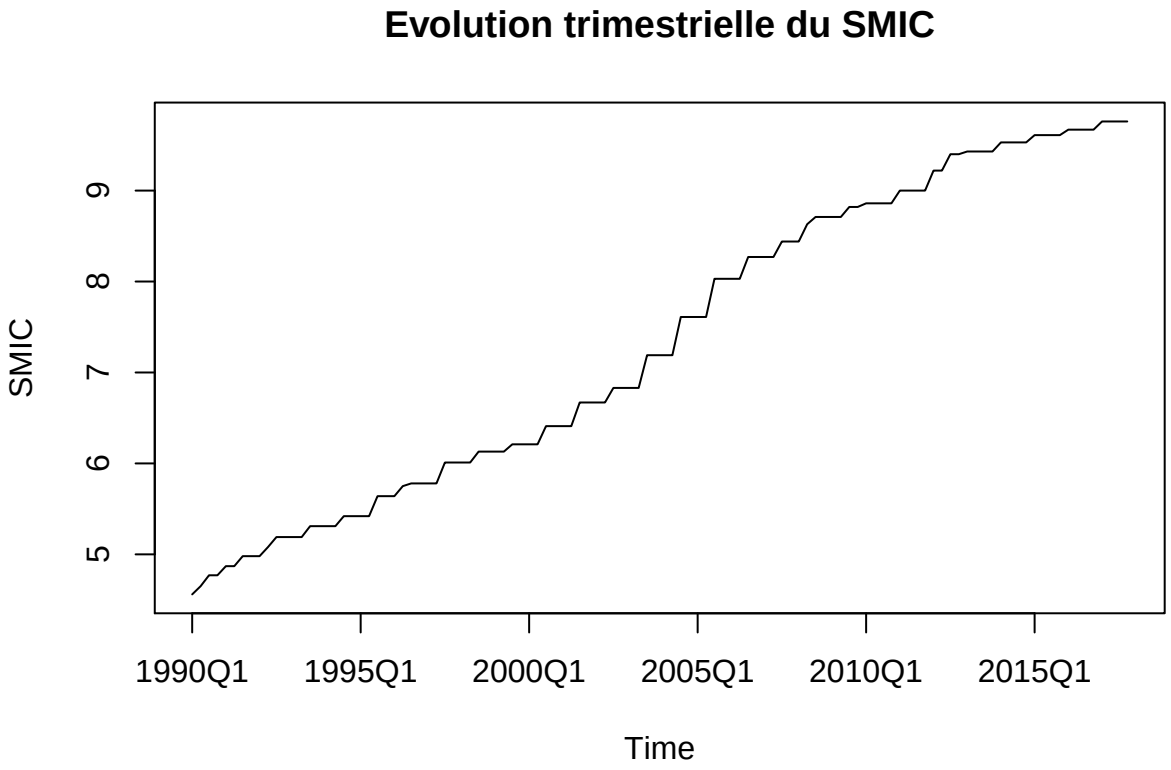
```
  adf.test(PIB)
```

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

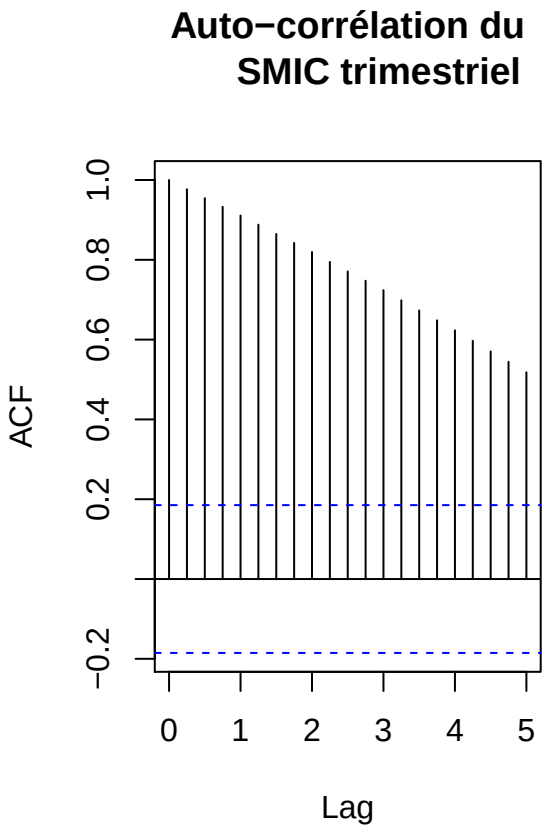
Comme pour la masse salariale, le PIB annuel possède une tendance. Cependant, il ne semble pas posséder de saisonnalité. Cette série ne semble donc pas non plus stationnaire. Nous effectuons à nouveau un test de KPSS. La p-value est de 0.01 ce qui nous confirme que la série n'est pas stationnaire avec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

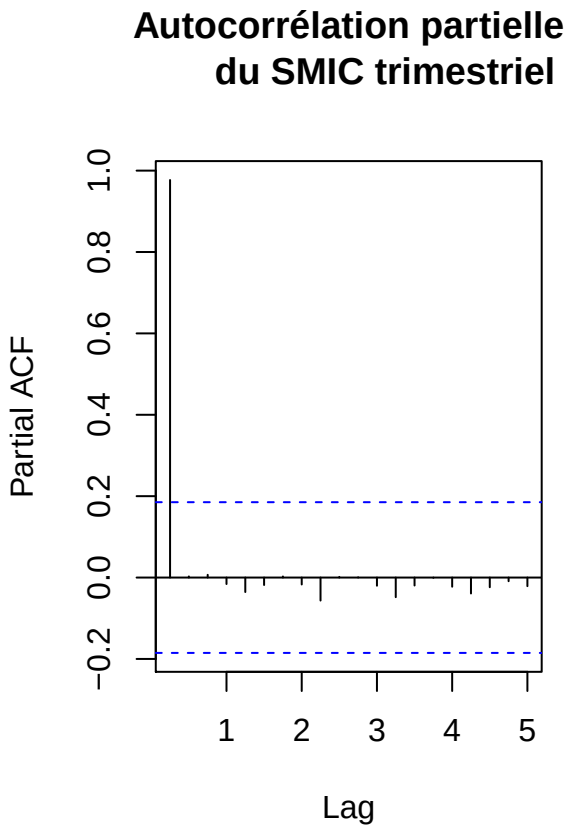
SMIC

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




```r
par(mfrow=c(1,2))
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
```

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

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

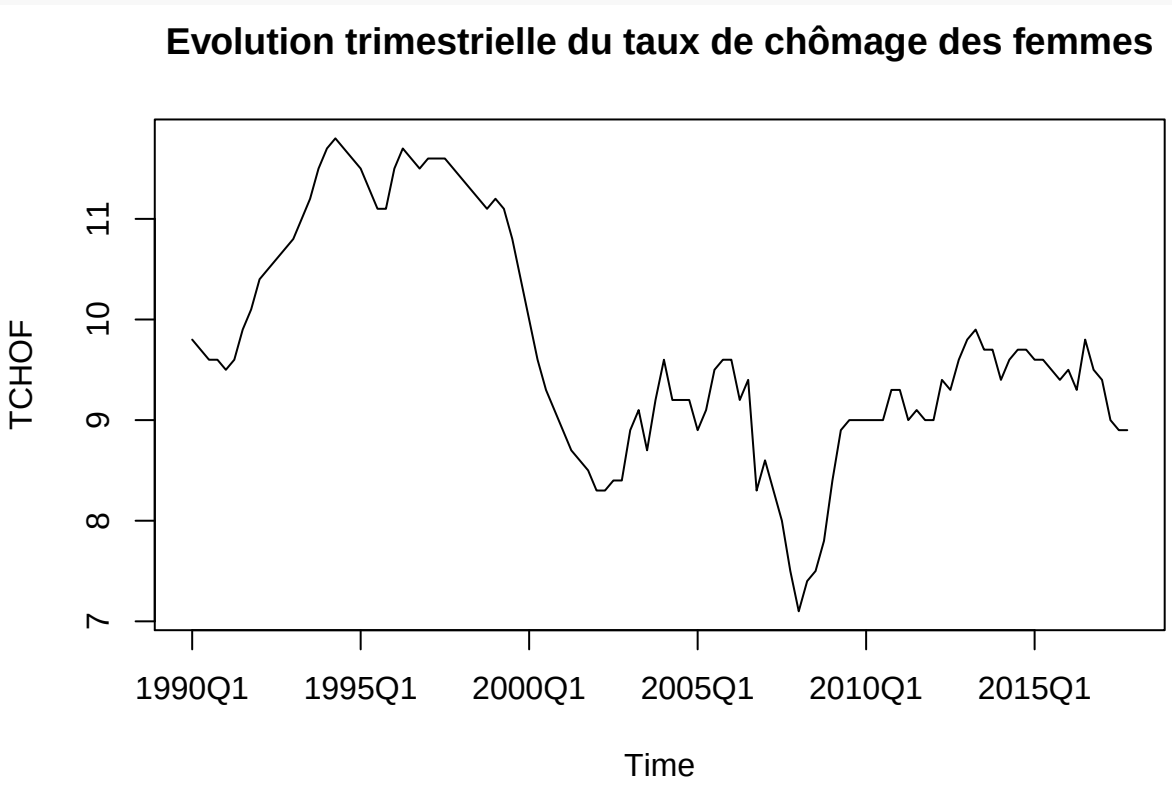
```
##
##  Augmented Dickey-Fuller Test
##
## data:  SMIC
## Dickey-Fuller = -1.4174, Lag order = 4, p-value = 0.8184
## alternative hypothesis: stationary
```

Au regard de la représentation graphique, 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.

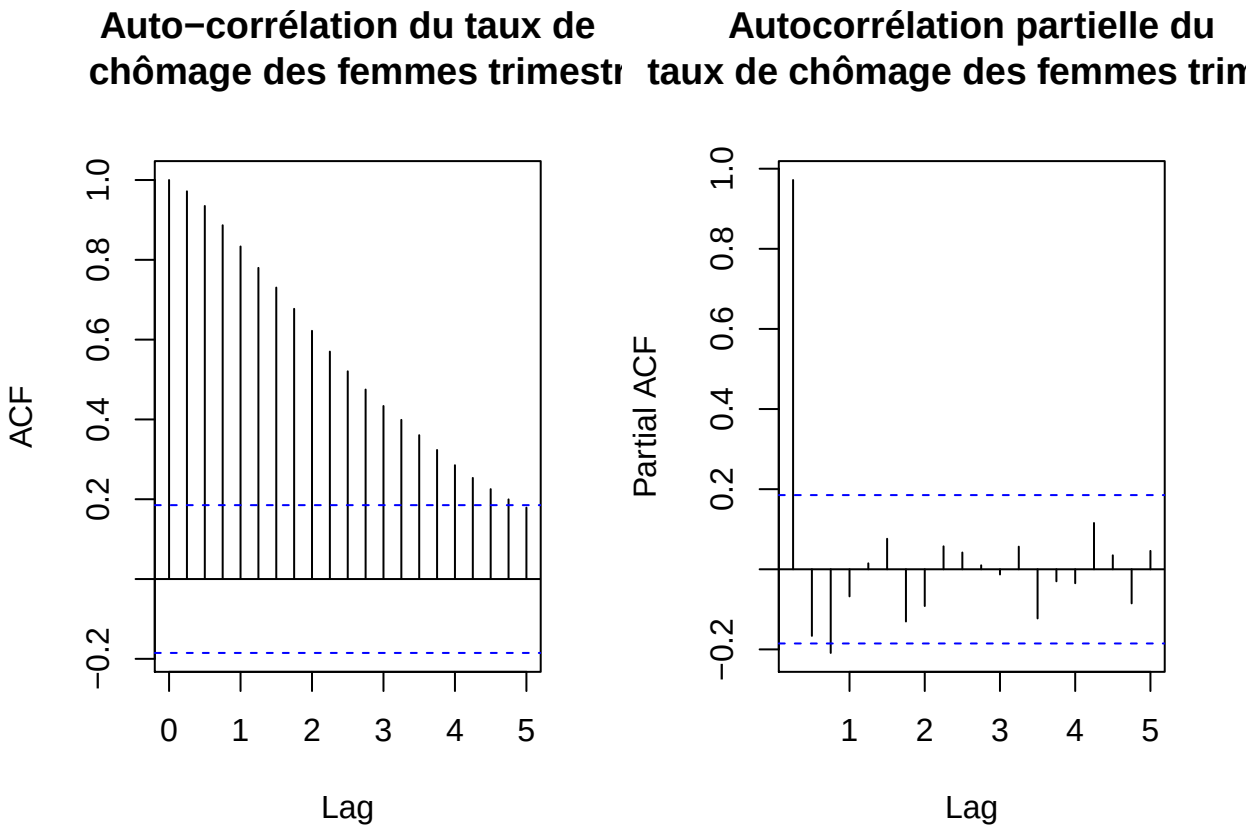
Taux de chômage des femmes

```
  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")
```

```
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
```




```
par(mfrow=c(1,2))
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)
```




```
  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
```

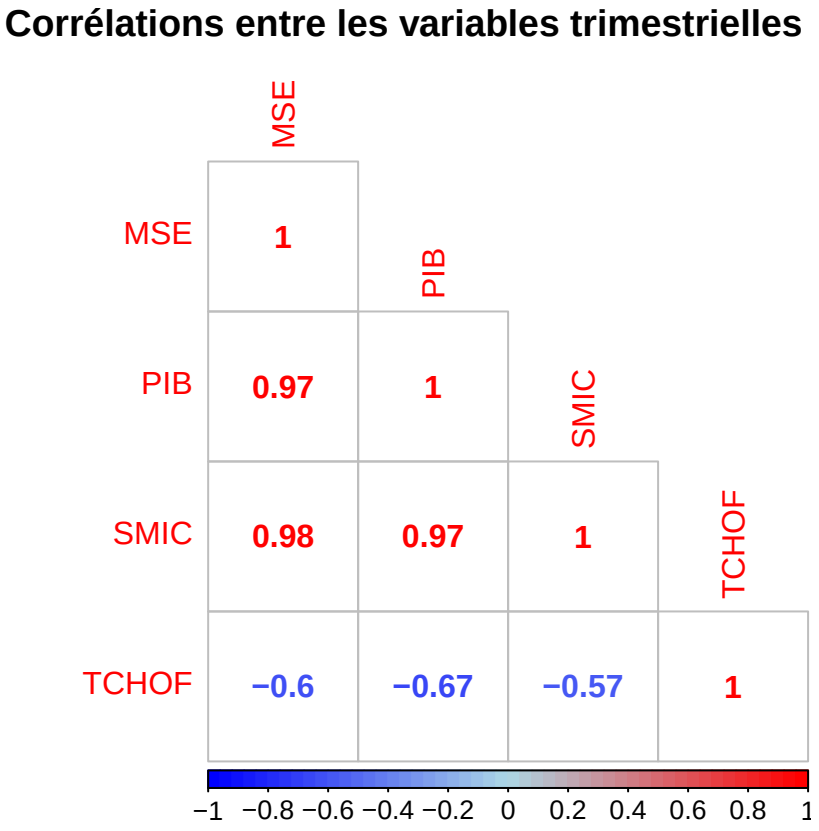
```
  adf.test(TCHOF)
```

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

Pour cette dernière série 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 fonction d'auto-corrélation. 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 risque de première espèce de 5%.

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
```

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.

Transformation des séries

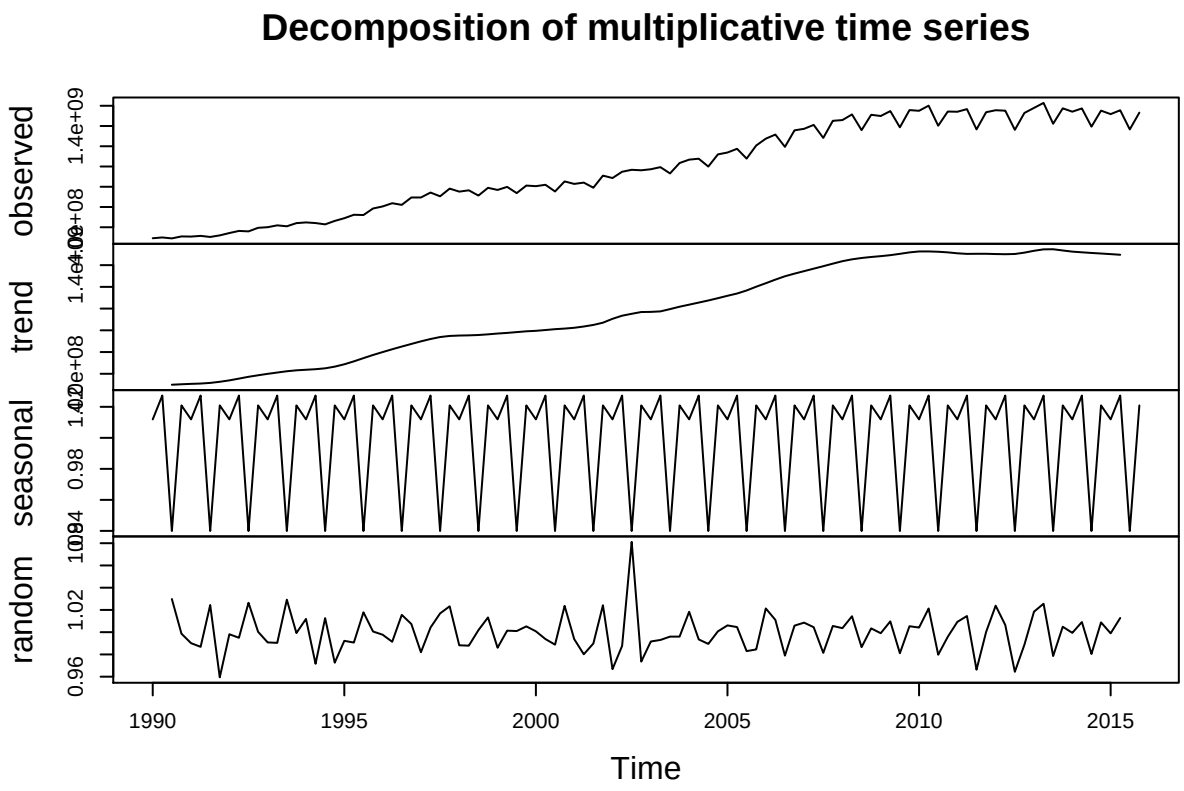
Pour chacune des séries, nous allons créé un échantillon d'apprentissage, qui nous permettra de construire les différents modèles, ainsi qu'un échantillon de test, qui nous permettra de comparer les prédictions des

modèles construits avec des vraies valeurs. L'échantillon d'apprentissage sera composé de toutes les valeurs jusqu'au 4e trimestre 2015, et celui de test de toutes les valeurs à partir du 1er trimestre 2016. **Utiliser des (S)AR(I)MA pour estimer et stationnariser les modèles**

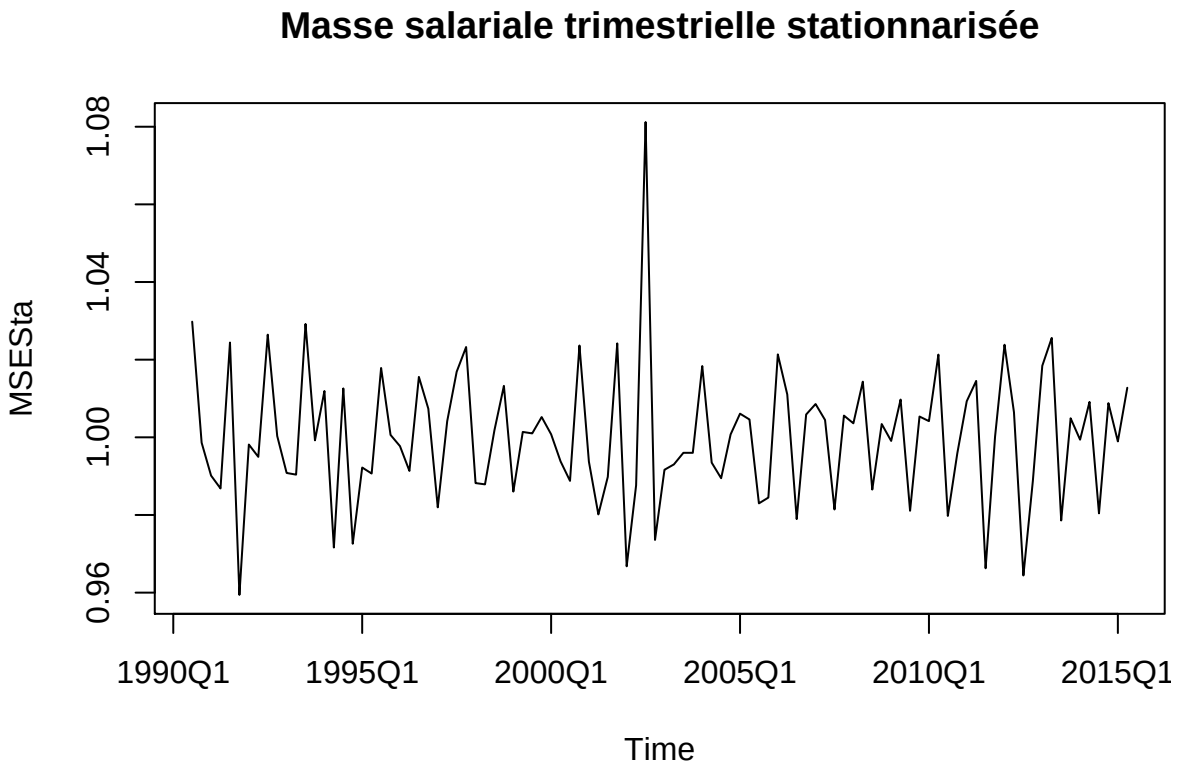
Nous allons maintenant transformer les séries pour les rendre stationnaires, afin de pouvoir appliquer les différents modèles ensuite. Afin de stationnariser les séries, nous utiliserons la fonction decompose qui permet de découper la série en trois : la tendance, la saisonnalité et les résidus, afin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

Masse salariale

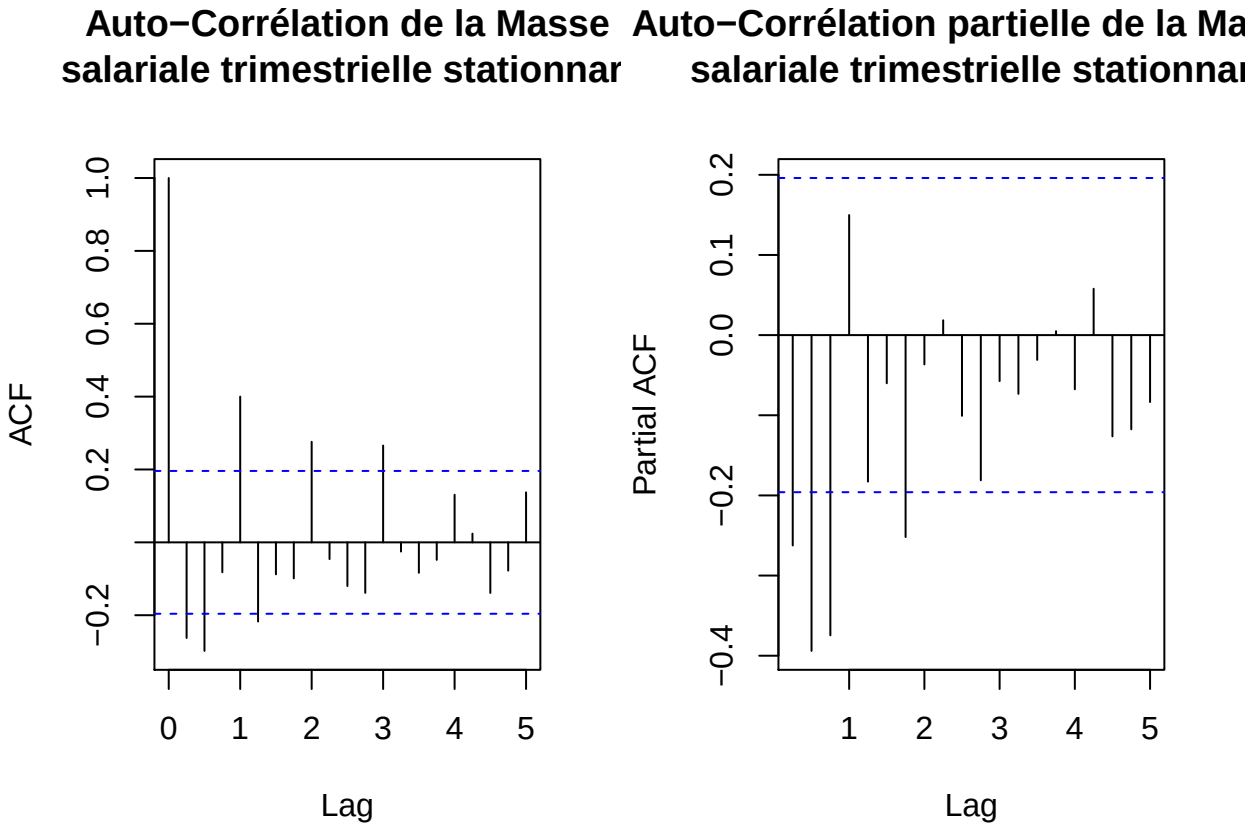
```
MSETrain <- window(MSE, end=c(2015,4))
MSETest <- window(MSE, start=2016)
plot(decompose(MSETrain, "multiplicative"))
```




```
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", "2015Q1"
```




```
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)
```

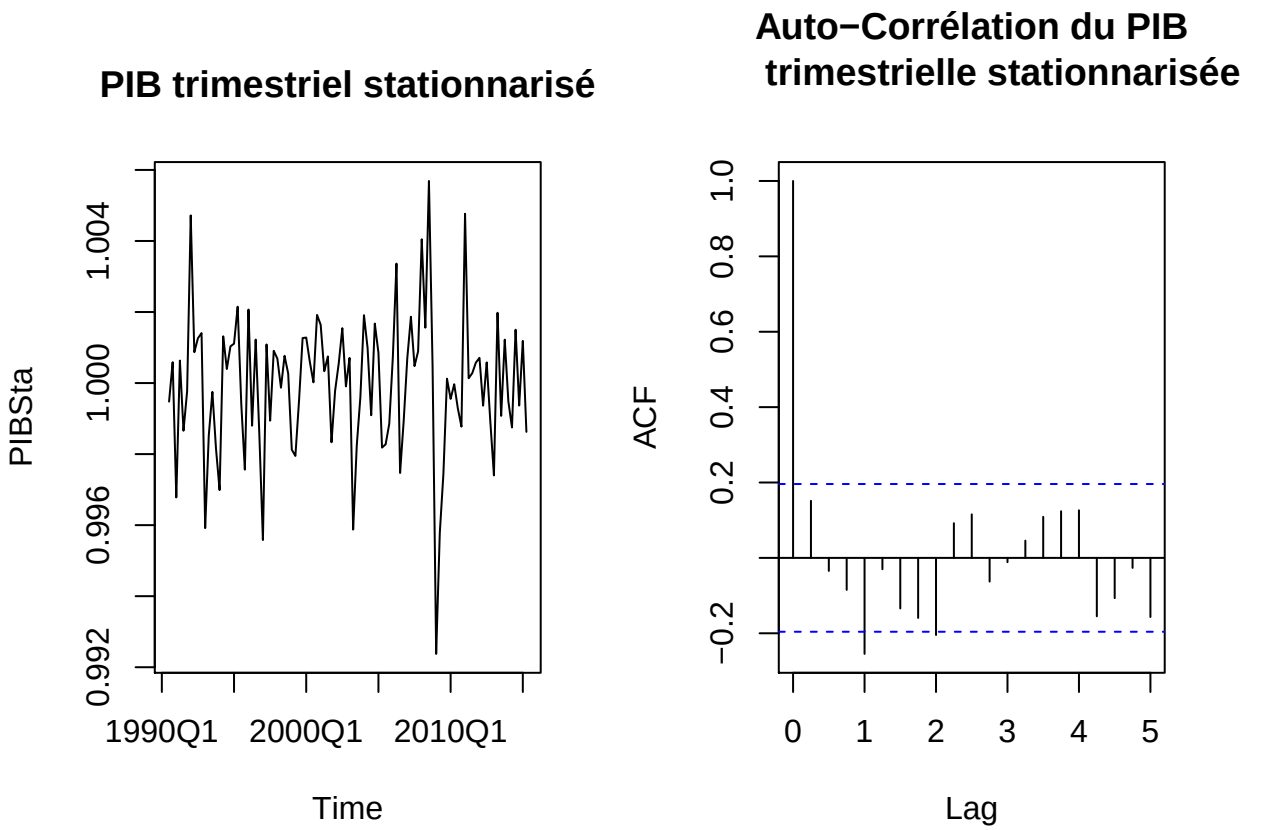
```
## Warning in adf.test(MSESta): p-value smaller than printed p-value
```

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

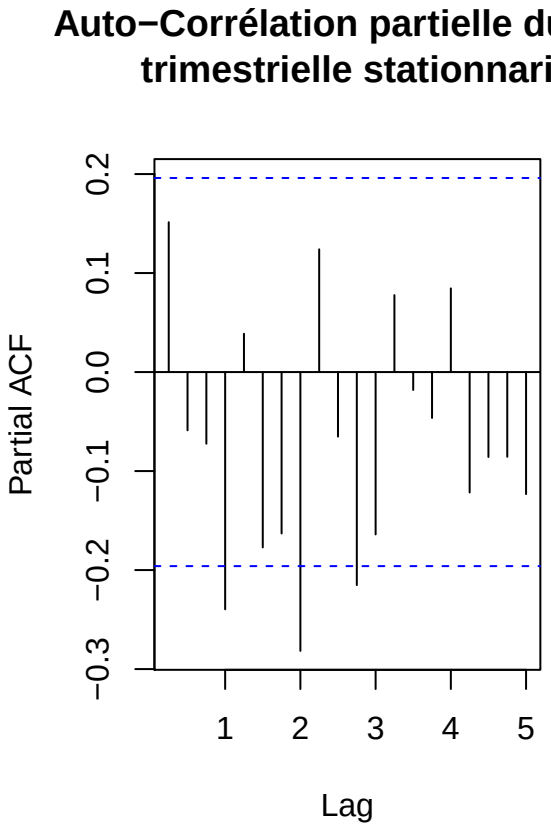
Nous nous intéressons aux ACF, PACF et test de KPSS afin de vérifier si les résidus obtenus à l'aide de la fonction decompose sont stationnaires. Bien que l'ACF et la PACF nous mettent en garde d'une possible non stationnarité de la série, la p-value des tests de KPSS et Dickey Fuller augmenté nous amène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

PIB

```
PIBTrain <- window(PIB, end=c(2015,4))
PIBTest <- window(PIB, start=2016)
PIBSta <- na.omit(decompose(PIBTrain, "multiplicative")$random)
par(mfrow=c(1,2))
plot(PIBSta, main="PIB trimestriel stationnarisé", xaxt="n")
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
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
```

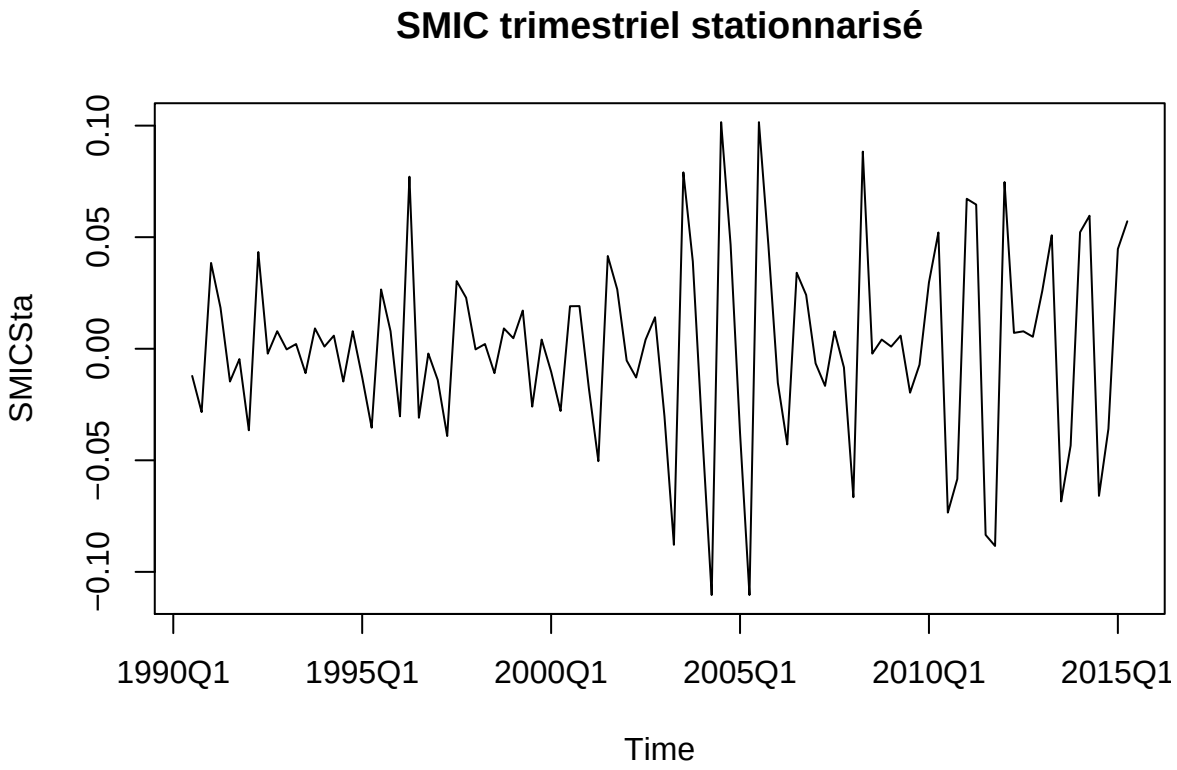
```
##
##  Augmented Dickey-Fuller Test
##
## data:  PIBSta
## Dickey-Fuller = -5.0084, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

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.

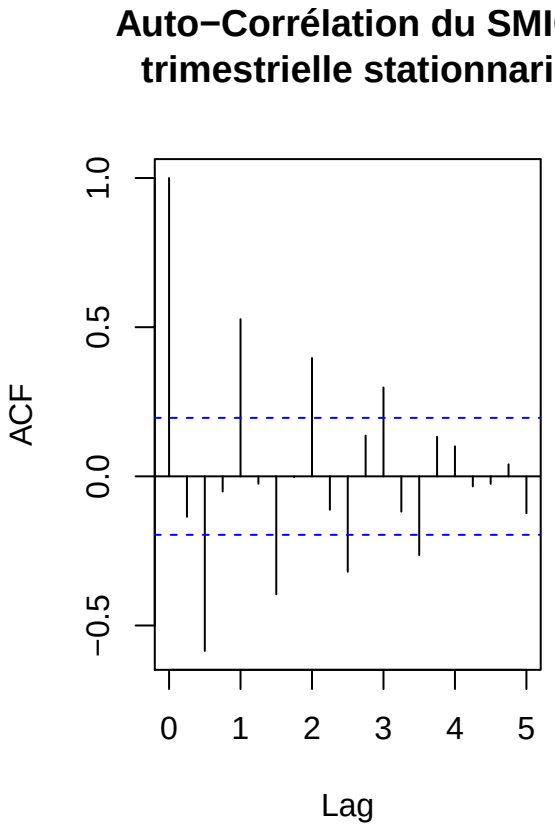
SMIC

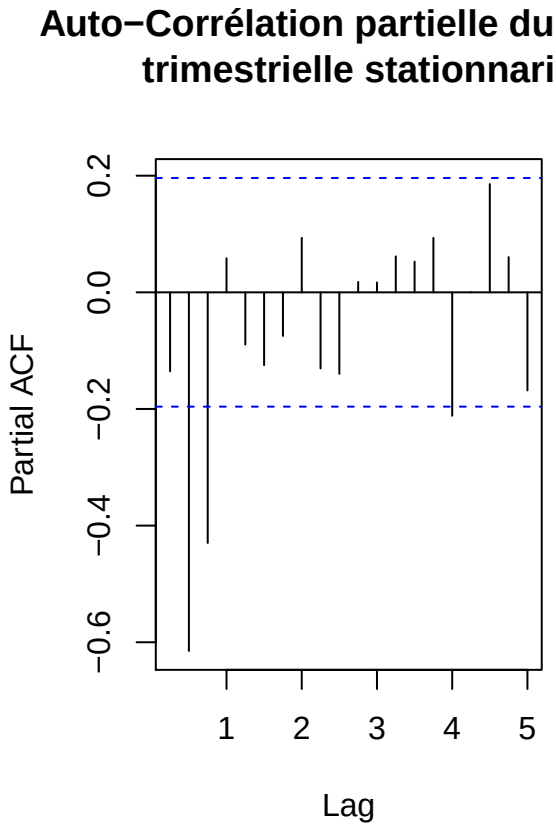
```
SMICTrain <- window(SMIC, end=c(2015,4))
SMICTest <- window(SMIC, start=2016)
```

```r
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", "2015Q1"))
```




```r
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)
```

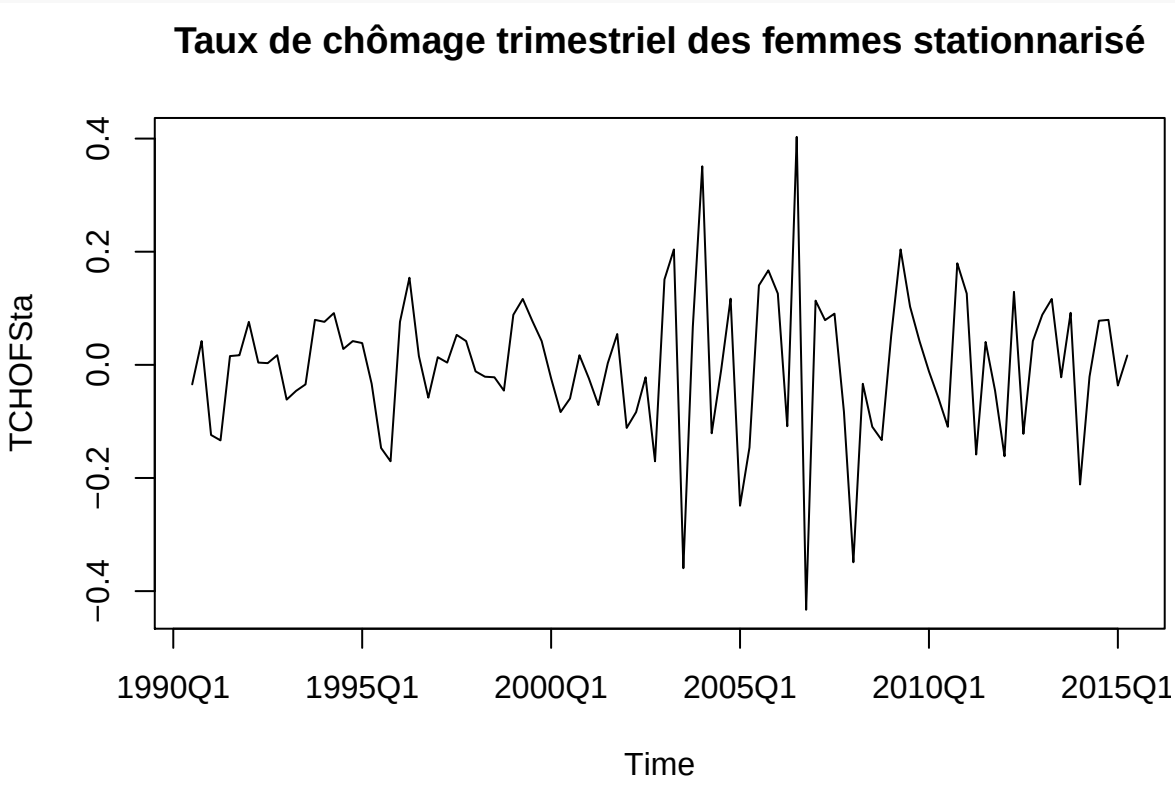
```
## Warning in adf.test(SMICSta): p-value smaller than printed p-value
```

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

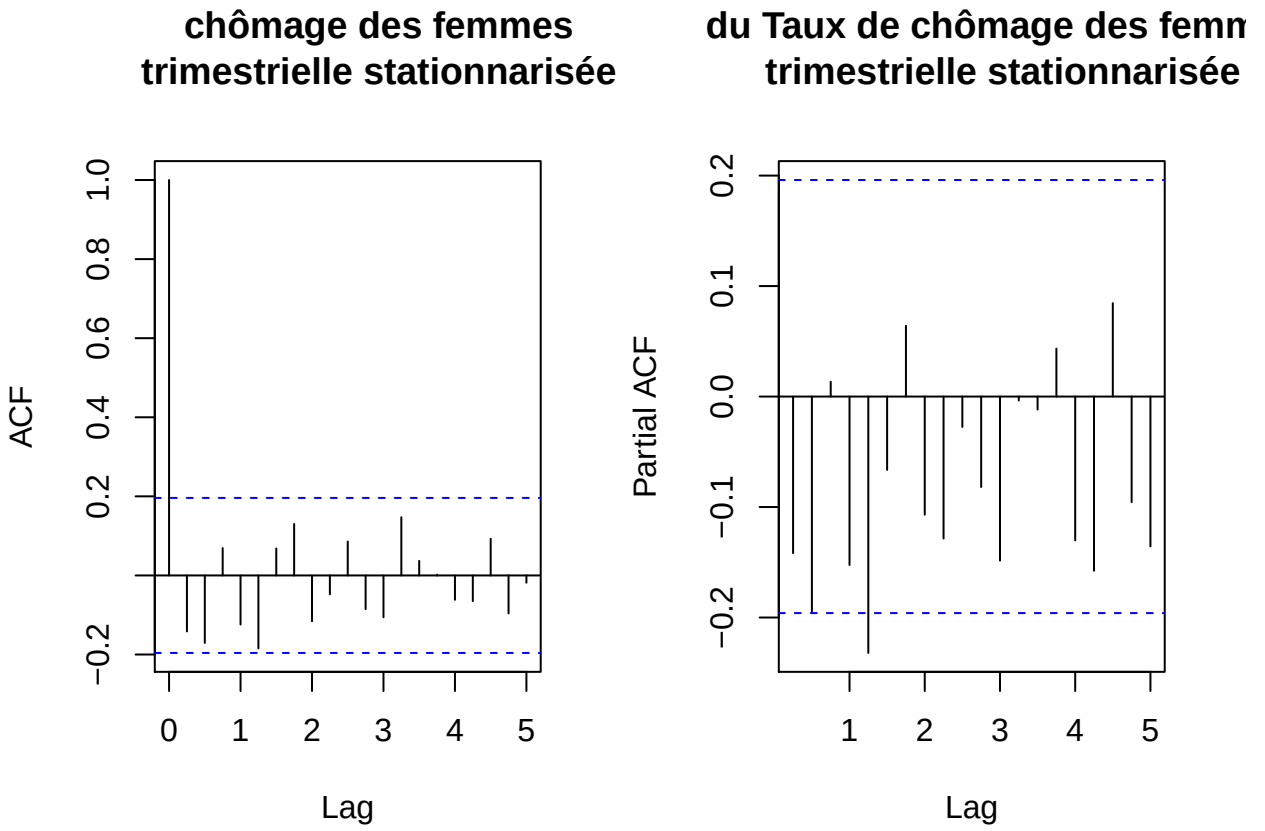
Comme pour la masse salariale, les ACF et PACF semblent montrer que la série résiduelle pourrait ne pas être stationnaire. Cependant le test de KPSS ainsi que le test de Dickey Fuller augmenté nous permettent de conclure à la stationnarité des résidus.

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", "2015Q1"
```




```
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
```

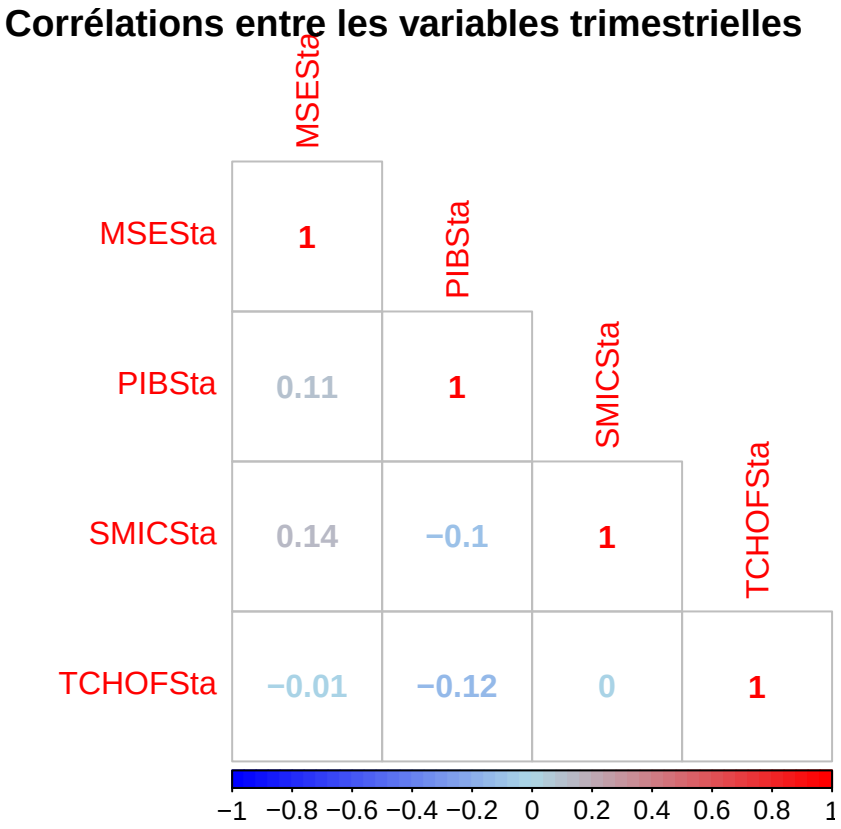
```
##
##  Augmented Dickey-Fuller Test
##
## data:  TCHOFSta
## Dickey-Fuller = -6.6221, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

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.

Corrélation entre les variables stationnarisées

```
corrplot(cor(cbind(MSESta, PIBSta, SMICSta, TCHOFSta)), method = "number", type="lower",
         p.mat=cor.mtest(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), 0.95)[[1]], insig="n",
         col=colorRampPalette(c("blue", "light blue", "red"))(50), title = "
         Corrélations entre les variables trimestrielles")
```




```
corr <- cor.mtest(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), 0.95)[[1]]
rownames(corr) <- c("MSE","PIB","SMIC","TCHOF")
colnames(corr) <- c("MSE","PIB","SMIC","TCHOF")
corr
```

```
##             MSE       PIB      SMIC     TCHOF
## MSE   0.0000000 0.2808591 0.1511096 0.9397487
## PIB   0.2808591 0.0000000 0.3027994 0.2329023
## SMIC  0.1511096 0.3027994 0.0000000 0.9786726
## TCHOF 0.9397487 0.2329023 0.9786726 0.0000000
```

On s'aperçoit que la transformation de nos séries a permis de supprimer les corrélations entre elles. En effet, en regardant le tableau des p-value, l'hypothèse nulle d'absence de corrélations n'est rejeté pour aucun couple de varaiables (avec un seuil de 5%).

Calcul de l'ordre p

Afin de mettre en place une modélisation VAR, nous devons dans un premier temps nous intéresser à l'ordre p du modèle VAR. L'ordre p correspond à l'ordre de l'opérateur de retard, c'est-à-dire le nombre de valeurs du

passé qui ont un impact sur la valeur à un instant défini. Dans le package **vars**, la fonction VARselect permet de déterminer les l'ordre des modèles VAR à selectionner en fonction de 4 critères (AIC, HQ, SC et FPE).

Pour les critères suivants, p correspond à l'ordre du modèle VAR, T le nombre d'observations utilisées pour la phase d'apprentissage, K le nombre de variables et $\tilde{\Sigma}_u(p) = \frac{1}{T}\Sigma_{t=1}^{T}\hat{u}_t\hat{u}_t'$ (la matrice de covariance des résidus du modèle).

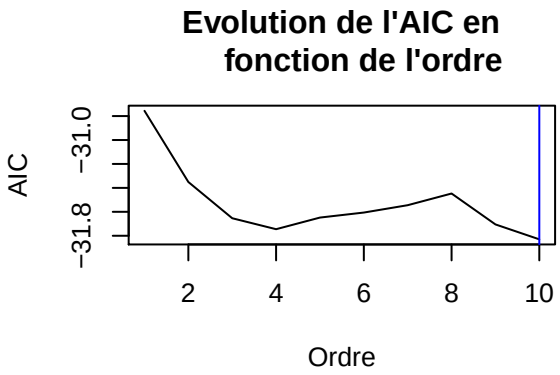
Le critère AIC (Aikaike information criterion) se calcule, dans ce package, de la manière suivante : $AIC(p) = \ln\det(\tilde{\Sigma}_u(p)) + \frac{2}{T}pK^2$. L'objectif est de minimiser ce critère. Cela suppose donc que le déterminant de la matrice $\tilde{\Sigma}_u(p)$ soit strictement positif. Ce critère est asymptotiquement effiace : si le nombre d'observations tend vers l'infini, sa variance est aussi faible que possible.

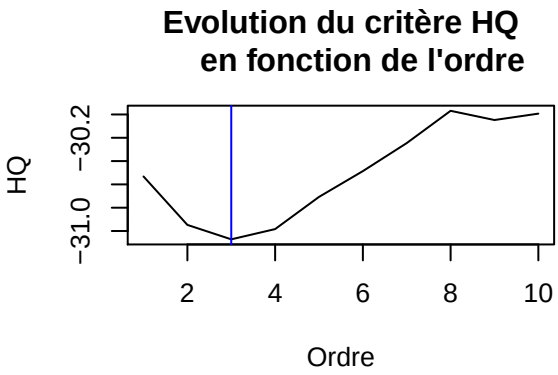
Le critère HQ (Hannan-Quinn criterion) se calcule, dans ce package, de la manière suivante : $HQ(p) = \ln\det(\tilde{\Sigma}_u(p)) + \frac{2\ln(\ln(T))}{T}pK^2$. L'objectif est de minimiser ce critère. Encore une fois, cela suppose que le déterminant de la matrice $\tilde{\Sigma}_u(p)$ soit strictement positif.

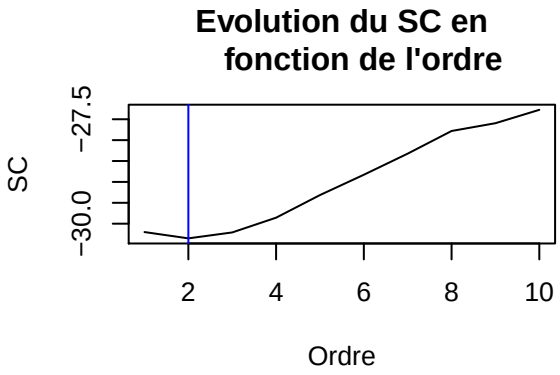
Le critère SC (Schwarz criterion) se calcule dans ce package de la manière suivante : $SC(p) = \ln\det(\tilde{\Sigma}_u(p)) + \frac{\ln(T)}{T}pK^2$. L'objectif est de minimiser ce critère. Ce critère est équivalent au BIC.‘

Dans cette partie, nous développerons le fonctionement de la méthodologie en l'appliquant uniquement au modèle complet, soit celui prenant en compte les variables PIB, SMIC et taux de chômage des femmes.

```
selec <- VARselect(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), lag.max=10)
par(mfrow=c(2,2))
plot(seq(1:10),t(selec$criteria[1,]), type="l", main="Evolution de l'AIC en
     fonction de l'ordre",
     xlab="Ordre", ylab="AIC")
abline(v=which.min(selec$criteria[1,]), col="blue")
plot(seq(1:10),t(selec$criteria[2,]), type="l", main="Evolution du critère HQ
     en fonction de l'ordre",
     xlab="Ordre", ylab="HQ")
abline(v=which.min(selec$criteria[2,]), col="blue")
plot(seq(1:10),t(selec$criteria[3,]), type="l", main="Evolution du SC en
     fonction de l'ordre",
     xlab="Ordre", ylab="SC")
abline(v=which.min(selec$criteria[3,]), col="blue")
```







On s'aperçoit que les différents critères à notre disposition nous donnent des ordres à choisir différents. Ainsi, le meilleur AIC correspond à un modèle d'ordre 10, le meilleur HQ à un modèle d'ordre 3 et le meilleur SC à un modèle d'ordre 2. L'ordre de l'AIC étant trop grand (car trop de coefficients à estimer par rapport au nombre d'observations à notre disposition), nous ne souhaitons pas conserver cet ordre. De plus, on se rend compte que l'AIC du modèle avec un ordre 10 est similaire à celle d'un modèle avec un ordre 4. Les modèles HQ et SC sont meilleurs avec respectivement un ordre 3 et 2. Nous allons donc, dans la suite de l'analyse, essayer les trois modèles définit par les différents critères : ici, nous allons donc nous intéresser aux modèles d'ordre 2, 3 et 4. ** Tester tous les ordres définit par les 3 critères : Ici, on devra tester les modèles d'ordre 2, 3 et 4 **

Estimation du modèle VAR(p)

Dans la partie précédente, nous avons sélectionné le meilleur ordre pour notre modèle VAR. Il s'agit maintenant d'estimer différents modèles afin de pouvoir prédire la MSE. L'exemple que nous avons pris est pour le modèle complet, avec les trois ordres déterminés précédemment (2, 3 et 4).

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, t le décalage de la série et u_t une matrice K-dimensionnelle composée des résidus du modèle (indépendants et identiquement distribués).

Ordre 4

Dans le package **vars**, la fonction utilisée pour construire des modèles VAR est VAR, qui prend en entrée la série temporelle multivariée, l'ordre du processus et le type de régresseurs à inclure. Dans notre cas, *type* vaut *const* car la série est stationnarisée et donc centrée en une constante μ. Ci-dessous, le modèle d'ordre 4.

```
modele<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=4, type="const")
```

Les coefficients du modèle associés à un retard de 1 sont les suivants :

```
A1<-cbind(modele$varresult$MSESta$coefficients[1:4],
          modele$varresult$PIBSta$coefficients[1:4],
          modele$varresult$SMICSta$coefficients[1:4],
          modele$varresult$TCHOFSta$coefficients[1:4])
colnames(A1)<-rownames(A1)<-c("MSE", "PIB", "SMIC", "TCHOF")
A1
```

```
##               MSE           PIB         SMIC      TCHOF
## MSE   -0.49571073 -0.0313494733  0.059314123 -0.8827534
## PIB    0.89614537  0.1217021125  1.181337380 -5.5627534
## SMIC   0.06696359 -0.0051254681 -0.496926461  0.6600874
## TCHOF -0.01279913  0.0005912438 -0.009601216 -0.3220845
```

Dans cette fonction, nous ne disposons pas des erreurs standards associées aux coefficients. Les indicateurs de qualité du modèle sont présents ci-dessous.

```
selection<-VARselect(cbind(MSESta, PIBSta, SMICSta, TCHOFSta))
selection$criteria[,4]
```

```
##        AIC(n)         HQ(n)         SC(n)        FPE(n)
## -3.174520e+01 -3.098355e+01 -2.985646e+01  1.664221e-14
```

L'erreur quadratique moyenne de ce modèle pour les données prédites est la suivante :

```
#ordre4 <- forecast(VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=4, type="const"))
#EQM(ordre4$forecast$MSESta$mean)
```

Ordre 3

On s'intéresse ensuite au modèle d'ordre 3 :

```
modele2<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=3, type="const")
```

Les coefficients du modèle associés à un retard de 1 sont les suivants :

```
A1modele2<-cbind(modele2$varresult$MSESta$coefficients[1:4],
          modele2$varresult$PIBSta$coefficients[1:4],
          modele2$varresult$SMICSta$coefficients[1:4],
          modele2$varresult$TCHOFSta$coefficients[1:4])
colnames(A1modele2)<-rownames(A1modele2)<-c("MSE", "PIB", "SMIC", "TCHOF")
A1modele2
```

```
##                MSE           PIB         SMIC      TCHOF
## MSE   -0.555340616 -0.0171271696 -0.01904054 -0.4049235
## PIB    0.587560629  0.1810606347  0.43555589 -6.7105659
## SMIC   0.007893159 -0.0001595774 -0.53423113  0.1543612
## TCHOF -0.007157107  0.0011494898 -0.01568059 -0.2784882
```

Les indicateurs de qualité du modèle sont présents ci-dessous.

```
selection<-VARselect(cbind(MSESta, PIBSta, SMICSta, TCHOFSta))
selection$criteria[,3]
```

```
##        AIC(n)         HQ(n)         SC(n)        FPE(n)
## -3.165371e+01 -3.107127e+01 -3.020938e+01  1.805099e-14
```

L'erreur quadratique moyenne de ce modèle pour les données prédites est la suivante :

```
#ordre4 <- forecast(VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=3, type="const"))
#EQM(ordre4$forecast$MSESta$mean)
```

Ordre 2

On s'intéresse ensuite au modèle d'ordre 3 :

```
modele3<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=2, type="const")
```

Les coefficients du modèle associés à un retard de 1 sont les suivants :

```
A1modele3<-cbind(modele3$varresult$MSESta$coefficients[1:4],
          modele3$varresult$PIBSta$coefficients[1:4],
          modele3$varresult$SMICSta$coefficients[1:4],
          modele3$varresult$TCHOFSta$coefficients[1:4])
colnames(A1modele3)<-rownames(A1modele3)<-c("MSE", "PIB", "SMIC", "TCHOF")
A1modele3
```

```
##                MSE          PIB        SMIC      TCHOF
## MSE   -0.395727547 -0.012959090  0.12910889 -0.5160394
## PIB    0.829559295  0.187330124  0.94138632 -6.5905696
## SMIC   0.030196701  0.006086945 -0.23992603  0.1033734
## TCHOF -0.009163581  0.001036053 -0.02214107 -0.2172584
```

Les indicateurs de qualité du modèle sont présents ci-dessous.

```
selection<-VARselect(cbind(MSESta, PIBSta, SMICSta, TCHOFSta))
selection$criteria[,2]
```

```
##        AIC(n)         HQ(n)         SC(n)        FPE(n)
## -3.135082e+01 -3.094759e+01 -3.035090e+01  2.430391e-14
```

L'erreur quadratique moyenne de ce modèle pour les données prédites est la suivante :

```
#ordre4 <- forecast(VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=2, type="const"))
#EQM(ordre4$forecast$MSESta$mean)
```

Vérification des hypothèses

Afin que le modèle soit valide, il faut vérifier les trois hypothèses suivantes : homoscédasticité des résidus (test multivarié de ARCH-LM), normalité des résidus (test Jarque-Bera) et absence de corrélations entre les résidus (test du Portmanteau multivarié). Nous allons donc appliquer ces trois tests.

Verification de la stabilité

Pour vérifier si le processus VAR est stable, c'est-à-dire qu'il génère des séries stationnaires, nous devons calculer les valeurs propres de la matrice des coefficients :

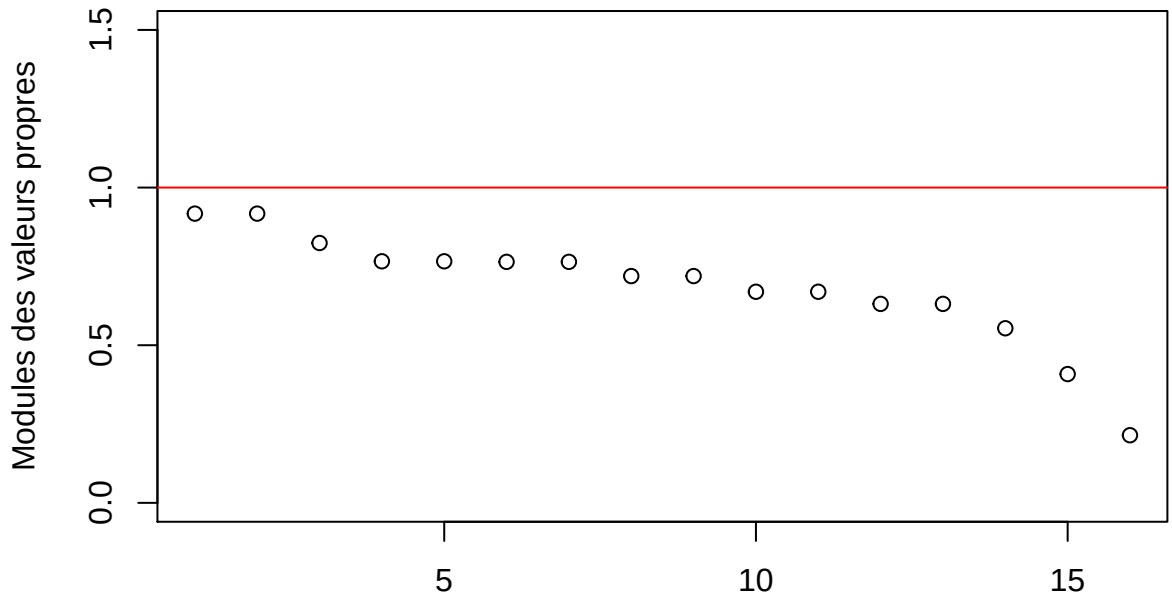
$$A = \begin{bmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix}$$

Si les modules des valeurs propres de A sont inférieures à 1, alors le processus VAR est stable. Nous allons donc vérifier que le processus VAR(4) créé précédemment avec toutes les variables à notre disposition est bien stable.

```
#Construction de la matrice A
A <- matrix(0,nrow=16, ncol=16)
A[1:4,1:4] = A1
A[1:4,5:8] = A2
A[1:4,9:12] = A3
A[1:4,13:16] = A4
A[5:8,1:4] = diag(1,4,4)
A[9:12,5:8] = diag(1,4,4)
A[13:16,9:12] = diag(1,4,4)
#Calcul des valeurs propres
vp <- eigen(A)
Mod(vp$values)
```

```
##  [1] 0.9174340 0.9174340 0.8241534 0.7661190 0.7661190 0.7644083 0.7644083
##  [8] 0.7192911 0.7192911 0.6695288 0.6695288 0.6310697 0.6310697 0.5537349
## [15] 0.4084401 0.2145192
```

```
plot(seq(1,16), Mod(vp$values), xlab="",
     ylab="Modules des valeurs propres", ylim=c(0,1.5))
abline(h=1, col="red")
```



On s'aperçoit que tous les modules sont inférieurs à 1, le processus VAR(4) est donc stable.

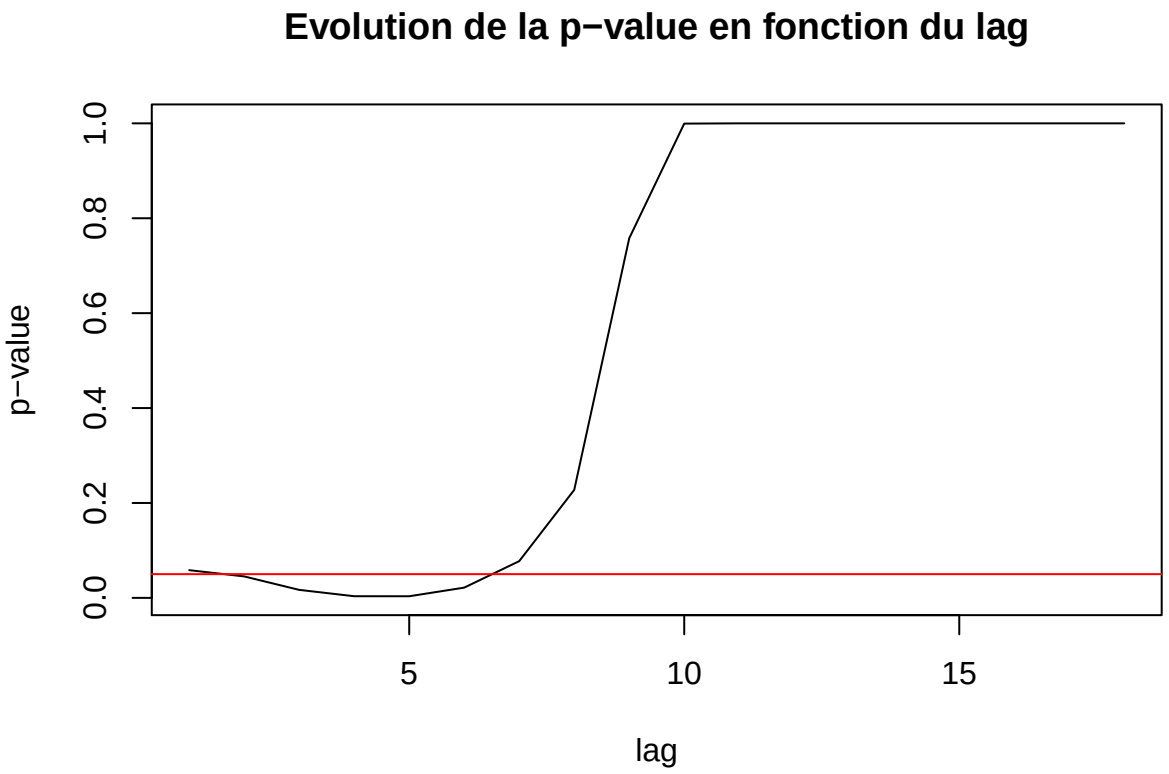


Figure 1:

Test ARCH (homoscédasticité des résidus)

Le test multivarié de ARCH-LM permet de tester l'homoscédasticité des résidus. La statistique de test est la suivante : $VARCH_{LM}(q) = \frac{1}{2}TK(K+1)R_m^2$, où $R_m^2 = 1 - \frac{2}{K(K+1)}tr(\hat{\Omega}\hat{\Omega}_O^{-1})$, et $\hat{\Omega}$ est la matrice de covariance de la régression suivante : $vech(\hat{u}_t\hat{u}_t{}^T) = \beta_0 + B_1 vech(\hat{u}_{t-1}\hat{u}_{t-1}^T) + ... + B_q vech(\hat{u}_{t-q}\hat{u}_{t-q}^T) + v_t$. La dimension de β_O est $\frac{1}{2}K(K+1)$ et celle des matrices des coefficients B_i est $\frac{1}{2}K(K+1)$Ö$\frac{1}{2}K(K+1)$. La statistique de test suit une loi de $\chi^2(qK^2(K+1)^2/4)$, donc dans notre cas $\chi^2(16q * 25/4)$. L'hypothèse nulle de ce test est $H_0 : B_0 = B_1 = ... = B_q = 0$ (homoscédasticité).

```
a1 <- c()
for(i in 1:18){
a1[i] <- arch.test(modele, lags.multi = i)$arch.mul$p.value
}
plot(a1, type="l", main="Evolution de la p-value en fonction du lag", xlab="lag", ylab="p-value")
abline(h=0.05, col="red")
```

On s'aperçoit au regard de la figure 1, avec un seuil de confiance de 5% (ligne rouge), qu'on rejette l'hypothèse nulle d'homoscédasticité pour un retard faible (inférieur à 7). Cependant, en augmentant le nombre de valeurs prises en compte pour calculer la nouvelle, on se rend compte qu'on conserve l'hypothèse d'homoscédasticité. On observe également que la valeur de la p-value converge vers 1 au fur et à mesure qu'on augmente le retard. Ainsi, en prenant l'ensemble des résidus, nous conservons l'hypothèse d'homoscédasticité.

Test normalité (normalité des résidus)

Le test de Jarque-Bera pour séries multivariées permet de tester la normalité des résidus. Il utilise les résidus standardisés à l'aide d'une décomposition de Cholesky de la matrice de variance-covariance des résidus centrés.

Il est important noter que l'ordre dans lequel les variables sont stockées dans la matrice a une importance sur les résultats. La statistique de test est la suivante : $JB_{mv} = s_3^2 + s_4^2$, où s_3^2 et s_4^2 se calculent de la sorte : $s_3^2 = Tb_1^T b_1/6$ et $s_4^2 = T(b_2 - 3_K)^T(b_2 - 3_K)/24$, avec b_1 et b_2 qui sont respectivement les vecteurs des moments non-centrés d'ordre trois et quatre des résidus standardisés. La statistique de test suit une loi de $\chi^2(2K)$. Ce test compare en fait le coefficient kurtosis K (l'aplatissement de la fonction de densité) et le coefficient skewness S (asymétrie de la fonction de densité) d'une loi normale à ceux des résidus testés. L'hypothèse nulle est donc $H0 : S = 0$ et $K = 3$.

```
normality.test(modele)$jb.mul$JB
```

```
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object modele
## Chi-squared = 71.388, df = 8, p-value = 2.599e-12
```

Ici, on rejette l'hypothèse H0, avec un seuil de confiance de 5%. Les résidus obtenus ne suivent pas une loi normale.

Test Portmanteau (corrélations des résidus)

Le test de Portmanteau multivarié permet de tester l'auto-corrélation (au sein d'une même série) et la corrélation croisée (entre les différentes séries) des résidus.

Verification de la Matrice C_0

La statistique de Portmanteau est $Q_h = T\sum_{j=1}^{h} tr(\hat{C}_j^T \hat{C}_0^{-1} \hat{C}_j \hat{C}_0^{-1})$, et elle suit une loi de $\chi^2(K^2h - n^*)$, n^* étant le nombre de coefficients à estimer. Pour qu'elle existe, il faut donc vérifier que la matrice $\hat{C}_0$ est inversible pour que la statistique puisse être définie. Les matrices $\hat{C}_i$ s'écrivent $\hat{C}_i = \frac{1}{T}\sum_{t=i+1}^{T} \hat{u}_t \hat{u}_{t-i}^T$, donc $\hat{C}_0$ s'écrit $\hat{C}_0 = \frac{1}{T}\sum_{t=1}^{T} \hat{u}_t \hat{u}_t^T$. Nous allons donc vérifier qu'elle est inversible pour le modèle complet que nous avons mis en place.

```
C0 <- matrix(nrow = 4, ncol=4, 0)
for(i in 1:nrow(residuals(modele))){
  C0 <- C0 + residuals(modele)[i,]%*%t(residuals(modele)[i,])
}
C0 <- (1/nrow(residuals(modele))) * C0
C0
```

```
##              MSESta        PIBSta       SMICSta      TCHOFSta
## [1,]  1.777427e-04  1.302790e-06  7.666911e-06 -9.660844e-05
## [2,]  1.302790e-06  3.036813e-06 -7.311009e-06 -2.784179e-05
## [3,]  7.666911e-06 -7.311009e-06  7.893080e-04  2.705167e-04
## [4,] -9.660844e-05 -2.784179e-05  2.705167e-04  1.041230e-02
```

```
d <- det(C0)
names(d) <- "Déterminant de la matrice"
d
```

```
## Déterminant de la matrice
##              4.173756e-15
```

Le déterminant de la matrice n'étant pas nul, la matrice $\hat{C}_0$ est donc inversible.

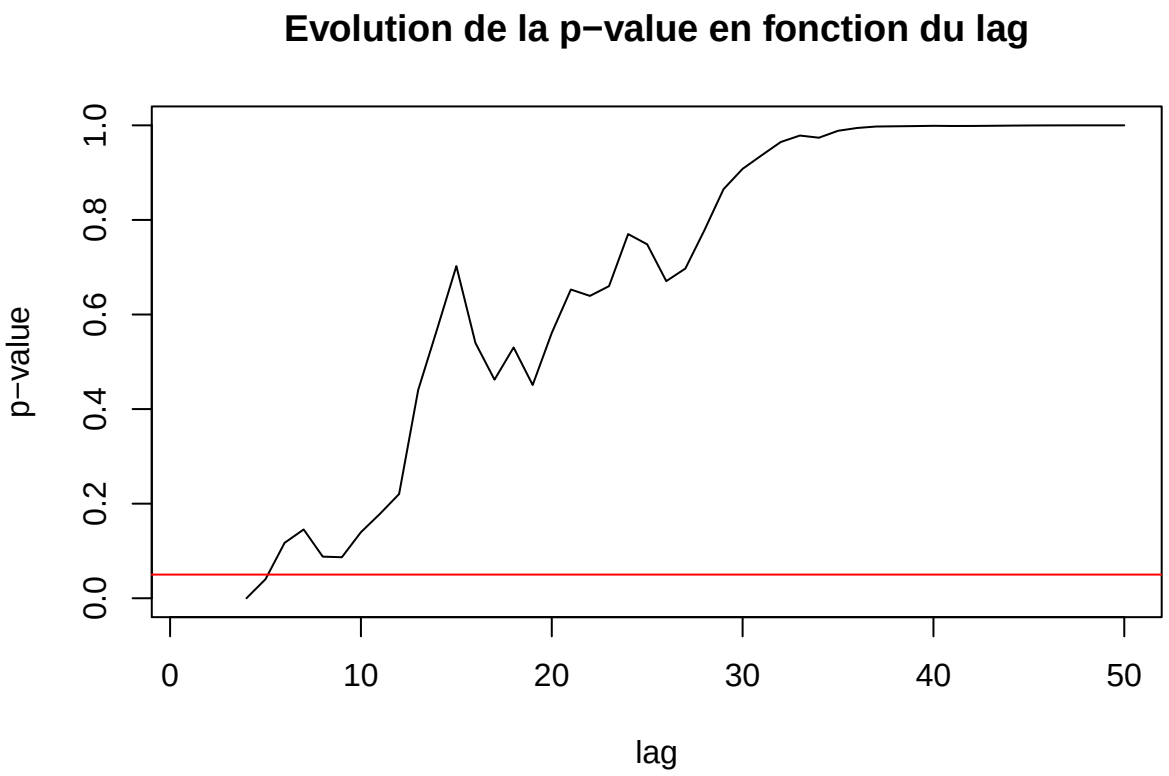


Figure 2:

Application du test

Les 3 premières p-values ne peuvent être calculées à cause de la valeur des degrés de liberté. En effet, comme nous l'avons expliqué plus haut, la statistique de test suit une loi de $\chi^2(K^2h - n^*)$. Or, avec un retard compris entre 1 et 3, les degrés de liberté sont négatifs et il n'est donc pas possible d'appliquer le test. L'hypothèse nulle de ce test est l'absence de corrélations croisées et d'auto-corrélations.

```
a1 <- c()
for(i in 1:3){
  a1[i] <- NA
}
for(i in 4:50){
  a1[i] <- serial.test(modele, lags.pt=i, type="PT.asymptotic")$serial$p.value
}
plot(a1, type="l", main="Evolution de la p-value en fonction du lag", xlab="lag", ylab="p-value")
abline(h=0.05, col="red")
```

Au regarde la figure 2 Comme pour le test ARCH, on rejette l'hypothèse nulle, avec un seuil de confiance à 5% (ligne rouge) pour un retard faible (5 ou moins). Cependant, pour un retard grand (supérieur à 5), on conserve l'hypothèse nulle d'absence d'auto-corrélations et de corrélations croisées. On observe également que la p-value converge vers 1 à mesure qu'on augmente le retard. Ainsi, en prenant en compte l'ensemble des résidus, on conserve l'hypothèse d'absence d'auto-corrélations et de corrélations croisées.

Prévisions

Maintenant que nous avons estimé l'ordre des différents modèle VAR, et que nous avons explicité l'estimation des modèles, nous cherchons désormais à trouver celui dont les prédictions sont les plus proches de la réalité.

Après avoir comparé tous les modèles possibles (7 : 3 modèles avec deux variables, 3 modèles avec trois variables et un modèle avec les quatre variables), nous nous apercevons que le meilleur en terme de prédictions est le modèle prenant en compte le SMIC (en plus de la masse salariale).

```
#SMIC
VARselect(cbind(MESSta, SMICSta), lag.max=10)
```

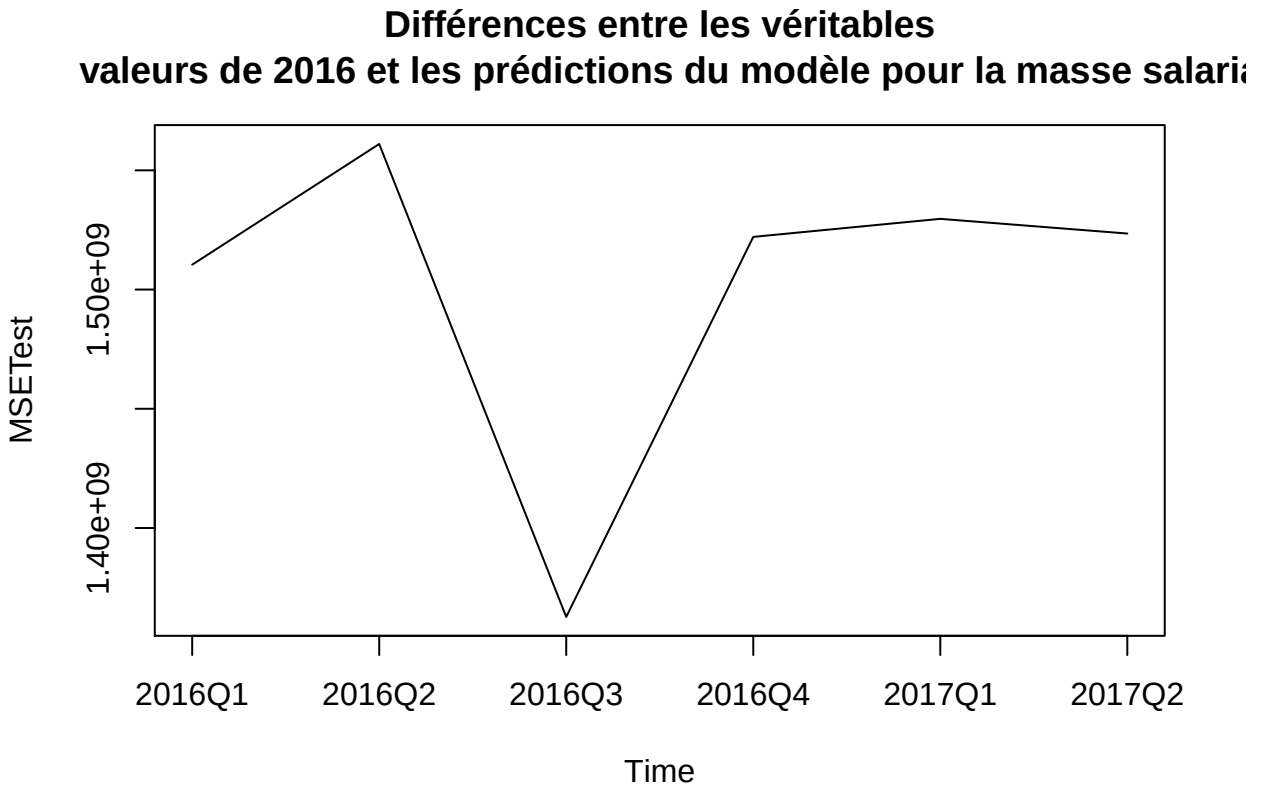
```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      3      3      3      3
##
## $criteria
##                   1             2             3             4
## AIC(n) -1.437017e+01 -1.501927e+01 -1.537233e+01 -1.534062e+01
## HQ(n)  -1.430297e+01 -1.490726e+01 -1.521552e+01 -1.513901e+01
## SC(n)  -1.420352e+01 -1.474151e+01 -1.498347e+01 -1.484066e+01
## FPE(n)  5.742953e-07  3.001341e-07  2.109387e-07  2.178884e-07
##                   5             6             7             8
## AIC(n) -1.530318e+01 -1.525968e+01 -1.530535e+01 -1.524475e+01
## HQ(n)  -1.505677e+01 -1.496846e+01 -1.496933e+01 -1.486393e+01
## SC(n)  -1.469212e+01 -1.453752e+01 -1.447208e+01 -1.430038e+01
## FPE(n)  2.264531e-07  2.369032e-07  2.268275e-07  2.416993e-07
##                   9            10
## AIC(n) -1.520513e+01 -1.520784e+01
## HQ(n)  -1.477950e+01 -1.473741e+01
## SC(n)  -1.414965e+01 -1.404126e+01
## FPE(n)  2.524033e-07  2.528867e-07
```

```
VAR(cbind(MESSta, SMICSta), p=3, type="const")
```

```
##
## VAR Estimation Results:
## ========================
##
## Estimated coefficients for equation MESSta:
## ===========================================
## Call:
## MESSta = MESSta.l1 + SMICSta.l1 + MESSta.l2 + SMICSta.l2 + MESSta.l3 + SMICSta.l3 + const
##
##    MESSta.l1   SMICSta.l1    MESSta.l2   SMICSta.l2    MESSta.l3
## -0.546437348  0.007337897 -0.548543337 -0.010463905 -0.386580207
##   SMICSta.l3        const
## -0.012418003  2.480944065
##
##
## Estimated coefficients for equation SMICSta:
## ============================================
## Call:
## SMICSta = MESSta.l1 + SMICSta.l1 + MESSta.l2 + SMICSta.l2 + MESSta.l3 + SMICSta.l3 + const
##
##    MESSta.l1   SMICSta.l1    MESSta.l2   SMICSta.l2    MESSta.l3
```

```
## -0.009893467 -0.539604002 -0.226391235 -0.735711730 -0.349005460
##   SMICSta.l3        const
## -0.460486731  0.586091963
```

```
VARSMICSta <- forecast(VAR(cbind(MSESta, SMICSta), p=3, type="const"))
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
     valeurs de 2016 et les prédictions du modèle pour la masse salariale", xaxt="n")
axis(side=1, at=seq(2016,2017.25,0.25), labels=c("2016Q1", "2016Q2", "2016Q3", "2016Q4", "2017Q1", "201
```




```
#Reconstruction de la variable stationnaire
#recon <- VARSMICSta$forecast$MSESta$mean * MSETrendTest * MSESeasonalTest
#lines(recon, col = "red")
#legend('bottomleft', legend = c('Vraies valeurs', 'Prévisions du modèle'),
#       col=c('black', 'red'), lty=1, cex=0.8)
```

Nous nous intéressons donc à l'erreur quadratique moyenne de cette prévision.

```
#EQM(MSETest, recon)
```