

# Package vars

*Paul GUILLOTTE & Jules CORBEL*

*01/02/2019*

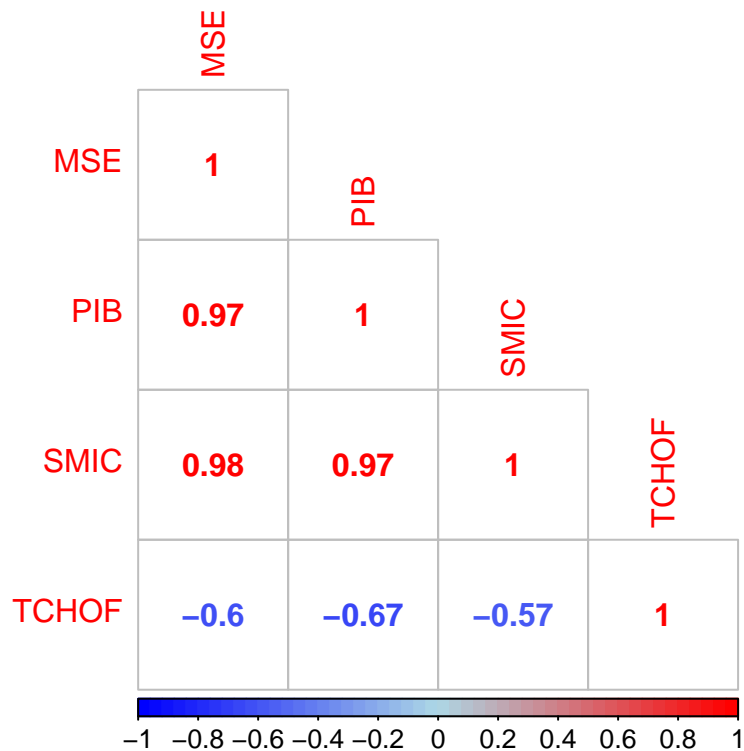
Nous nous intéresserons dans ce document à la mise en place de modèles VAR afin de prédire la masse salariale trimestrielle. Un modèle VAR, pour Vecteur AutoRégressif, a pour objectif de capturer les interdépendances entre les différentes séries temporelles à notre disposition. Ainsi, chaque variable 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.

```
trim <- read.csv("~/PFE_Time_Series/Data/Data_Trim.csv", sep=";", dec=",")
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")
```

### Corrélations entre les variables trimestrielles



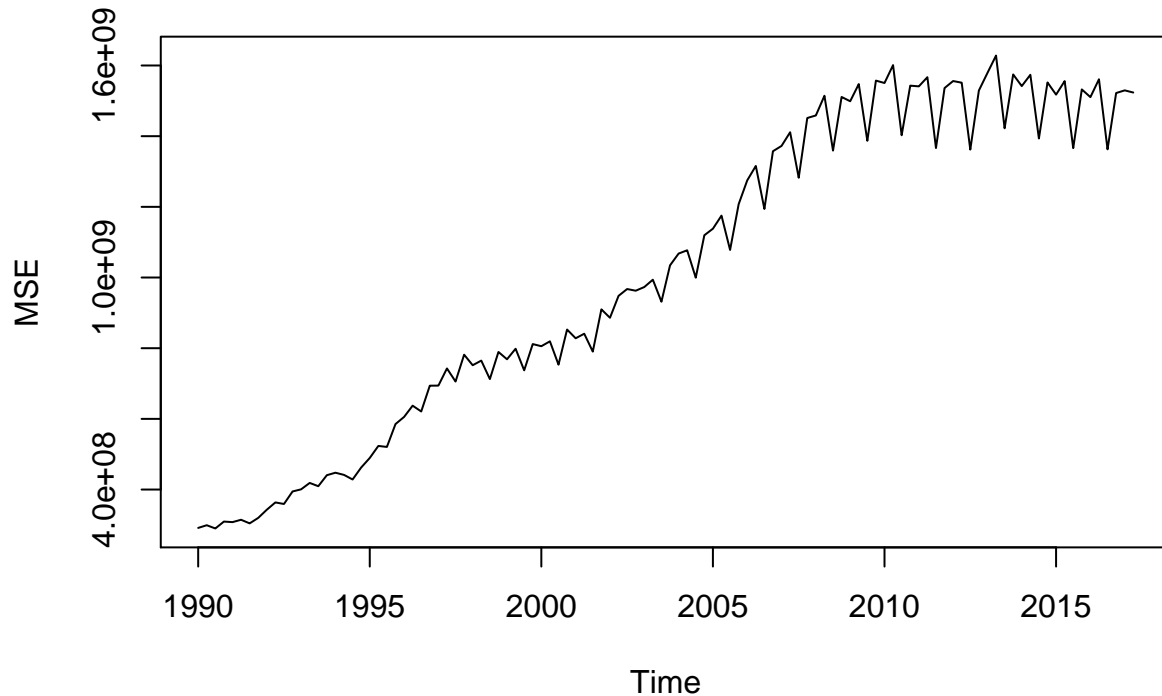
On se rend compte que le taux de chômage des femmes est corrélé négativement avec toutes les autres variables. Le trio de variables PIB, masse salariale et SMIC sont extrêmement liées entre elles.

Nous allons maintenant nous attarder sur chaque série individuellement.

## Masse salariale

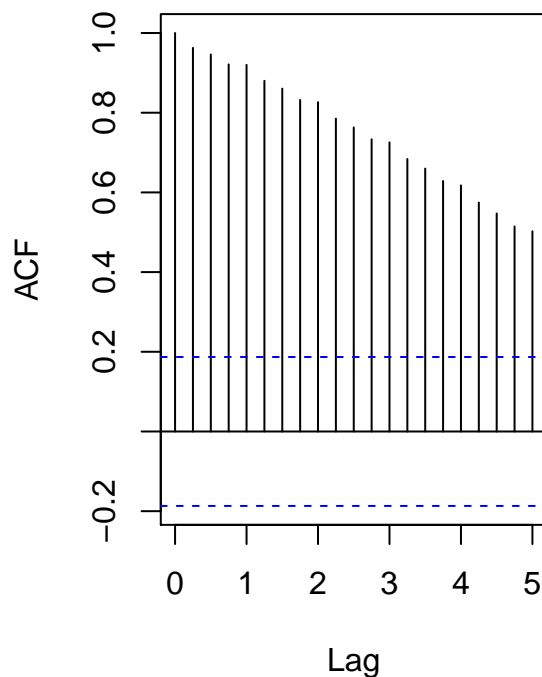
```
MSE <- ts(trim$MSE, start = 1990, end = c(2017, 2), frequency=4)
plot(MSE, main="Evolution de la masse salariale trimestrielle")
```

### Evolution de la masse salariale trimestrielle

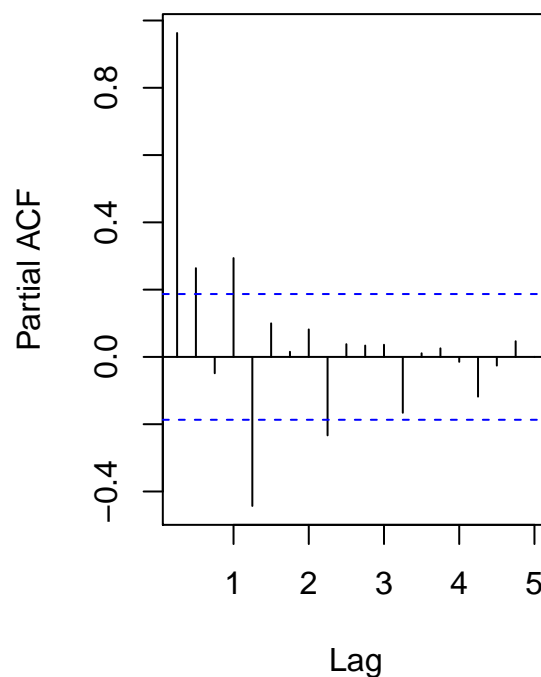


```
par(mfrow=c(1,2))
acf(MSE, main="Auto-corrélation de la
masse salariale trimestrielle", lag.max=20)
pacf(MSE, main="Autocorrélation partielle
de la masse trimestrielle", lag.max=20)
```

### Auto-corrélation de la masse salariale trimestrielle



### Autocorrélation partielle de la masse trimestrielle



```
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
```

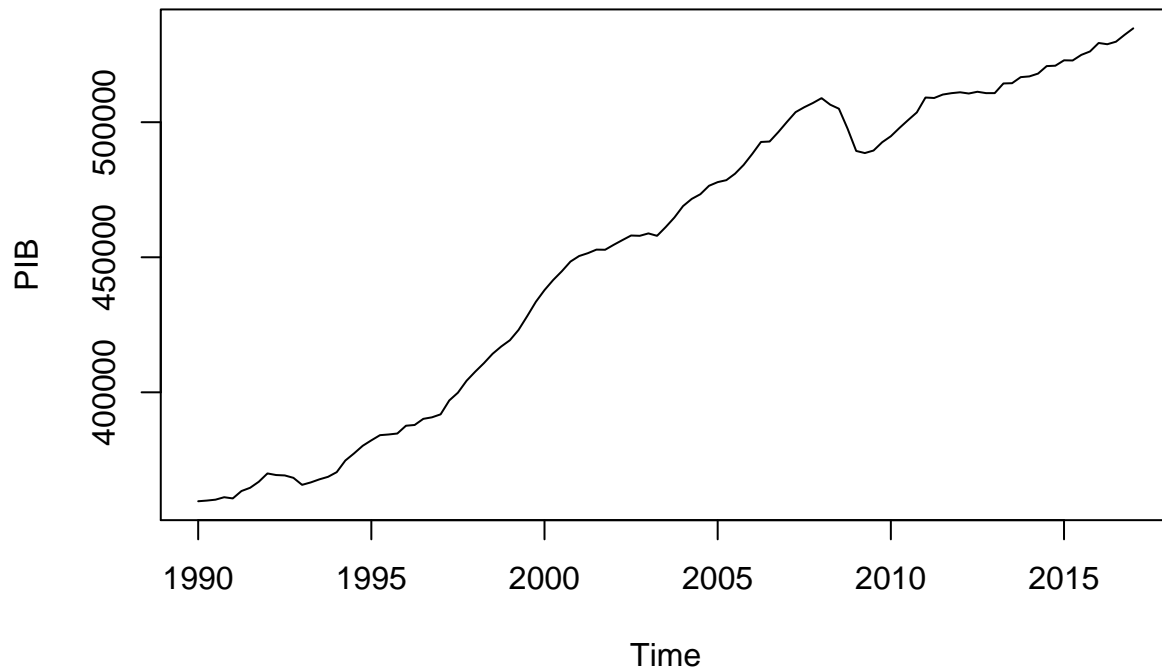
La masse salariale trimestrielle possède une composante de tendance de 1990 à 2010. La série tend par la suite à stagner. Nous remarquons également une saisonnalité sur cette série, qui est de plus en plus marquée à mesure que le temps passe.

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 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 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%.

## PIB

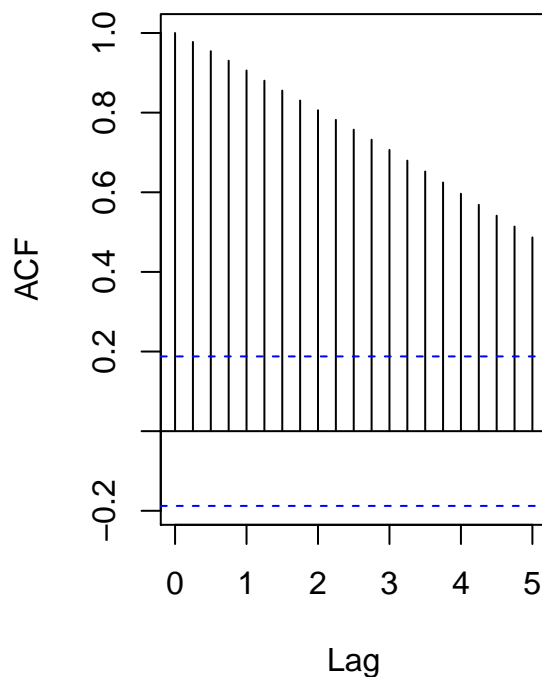
```
PIB <- ts(trim$PIB, start = 1990, end = c(2017, 1), frequency=4)
plot(PIB, main="Evolution du PIB trimestriel")
```

## Evolution du PIB trimestriel

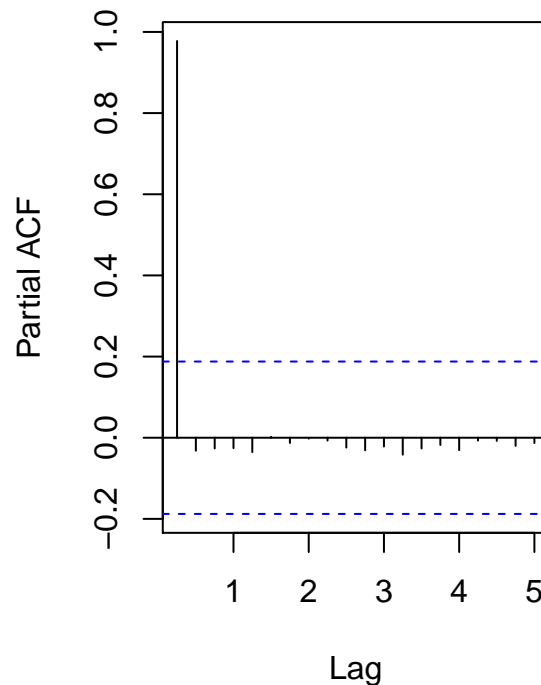


```
par(mfrow=c(1,2))
acf(PIB, main="Auto-corrélation
du PIB trimestriel", lag.max=20)
pacf(PIB, main="Autocorrélation partielle
du PIB trimestriel", lag.max=20)
```

**Auto-corrélation  
du PIB trimestriel**



**Autocorrélation partielle  
du PIB trimestriel**



```
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
```

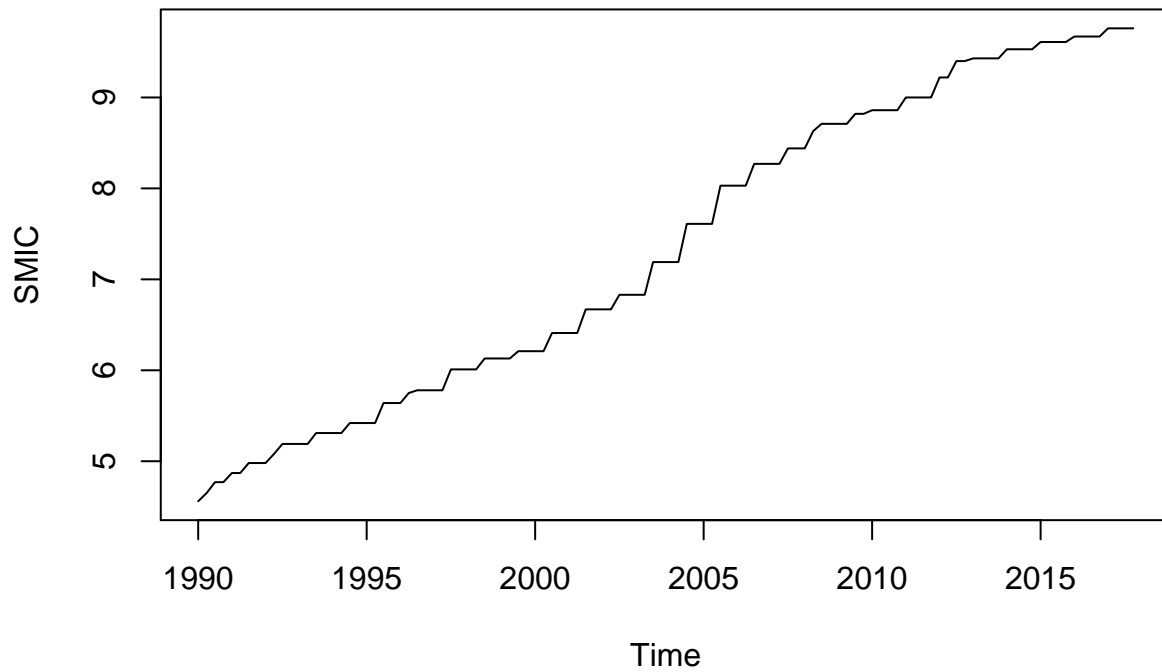
```
## KPSS Level = 3.6473, Truncation lag parameter = 2, p-value = 0.01
```

Comme pour la masse salariale, le PIB annuel possède une tendance. Cependant, il ne semble pas posséder de saisonnalité. Cette série n'est 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%.

## SMIC

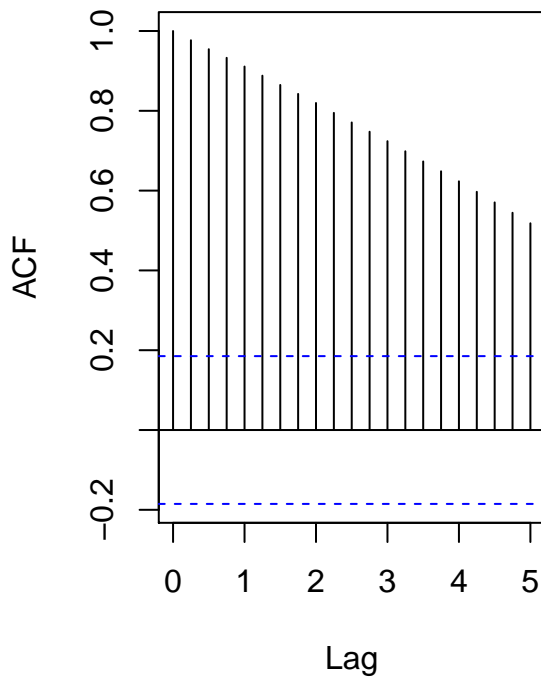
```
SMIC <- ts(trim$SMIC, start = c(1990,1), end = c(2017, 4), frequency = 4)
plot(SMIC, main="Evolution du SMIC trimestriel")
```

## Evolution du SMIC trimestriel

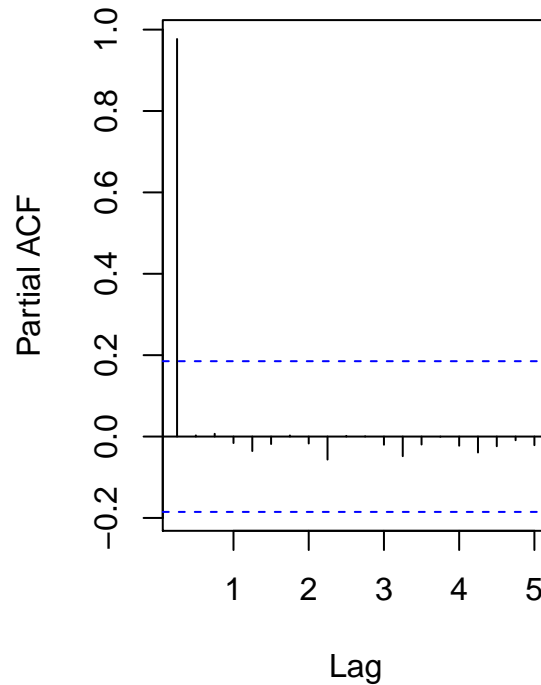


```
par(mfrow=c(1,2))
acf(SMIC, main="Auto-corrélation du
SMIC trimestriel", lag.max=20)
pacf(SMIC, main="Autocorrélation partielle
du SMIC trimestriel", lag.max=20)
```

### Auto-corrélation du SMIC trimestriel



### Autocorrélation partielle du SMIC trimestriel



```
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
```

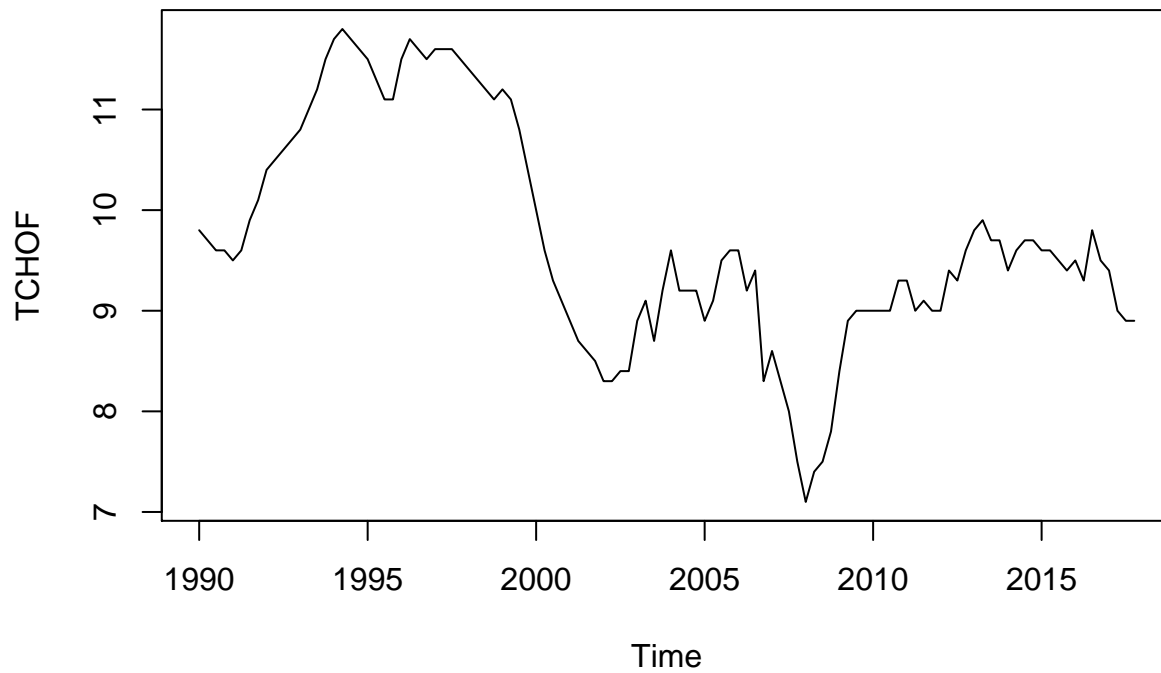
```
## KPSS Level = 3.8382, Truncation lag parameter = 2, p-value = 0.01
```

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.

### Taux de chômage des femmes

```
TCHOF <- ts(trim$TCHOF, start = c(1990,1), end = c(2017, 4), frequency = 4)
plot(TCHOF, main="Evolution du taux de chômage des femmes trimestriel")
```

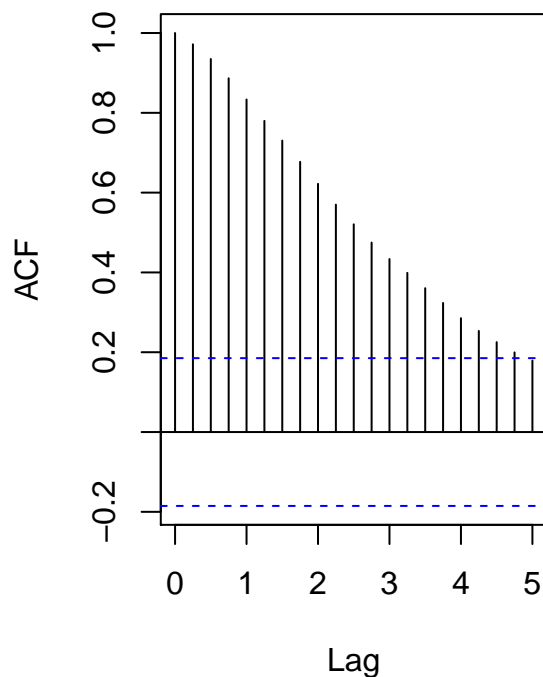
## Evolution du taux de chômage des femmes trimestriel



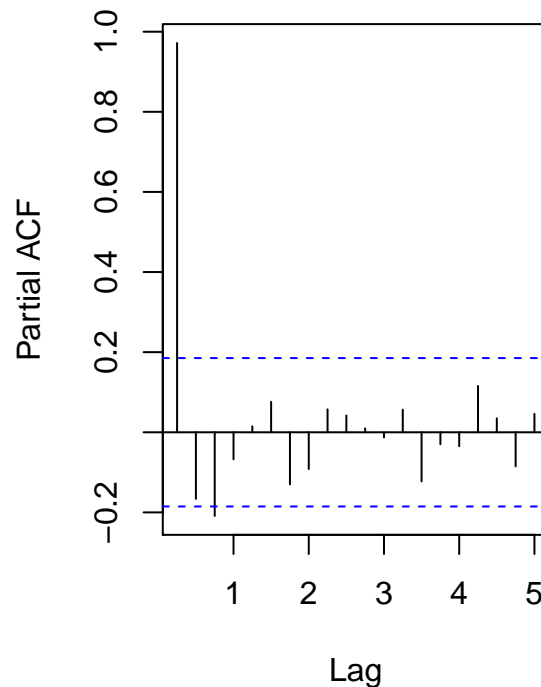
```
par(mfrow=c(1,2))
acf(TCHOF, main="Auto-corrélation du taux de
  chômage des femmes trimestriel", lag.max=20)
pacf(TCHOF, main="Autocorrélation partielle du
  taux de chômage des femmes trimestriel", lag.max=20)
```



## Auto-corrélation du taux de chômage des femmes trimestriel



## Autocorrélation partielle du taux de chômage des femmes trimestriel



```
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
```

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. Le test KPSS nous confirme que la série n'est pas stationnaire.

## Transformation des séries

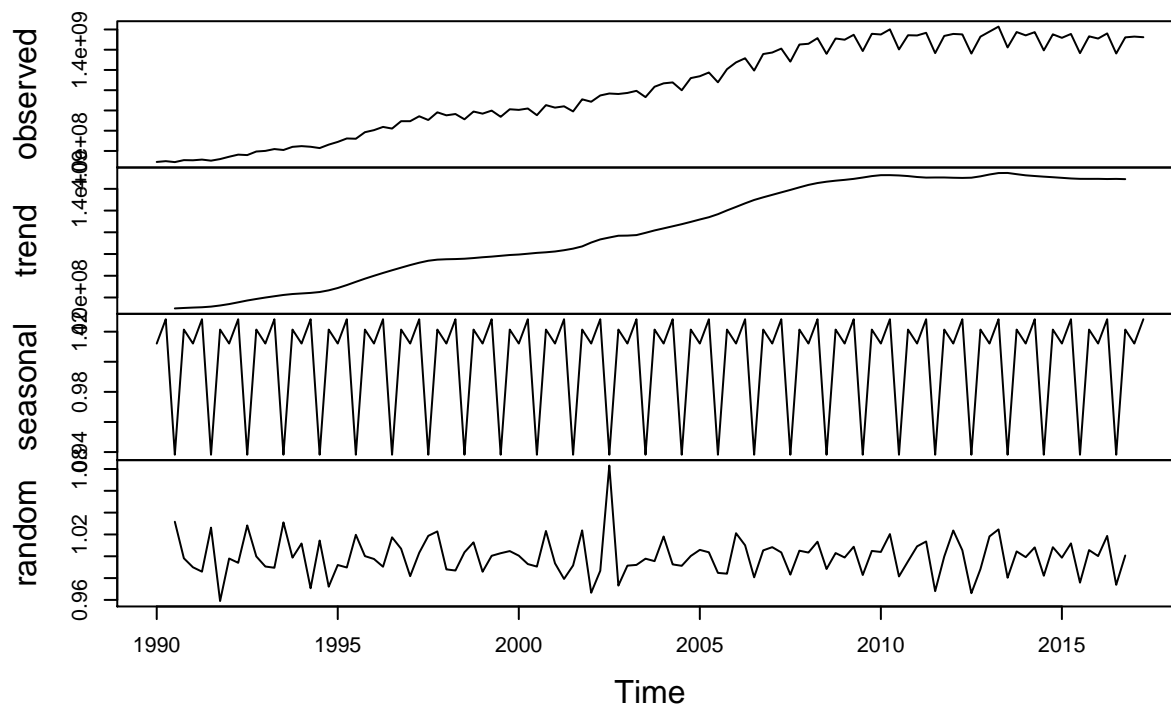
Nous allons maintenant transformer les séries pour les rendre stationnaires, afin de pouvoir appliquer les différents modèles ensuite. Afin de stationnariser les séries, nous utiliserons la fonction `decompose` qui permet de découper la série en trois : la tendance, la saisonnalité et les résidus, afin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

Pour chacune de ces 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 jusqu'au 4e trimestre 2015, et celui de test de toutes les valeurs à partir du 1er trimestre 2016.

## Masse salariale

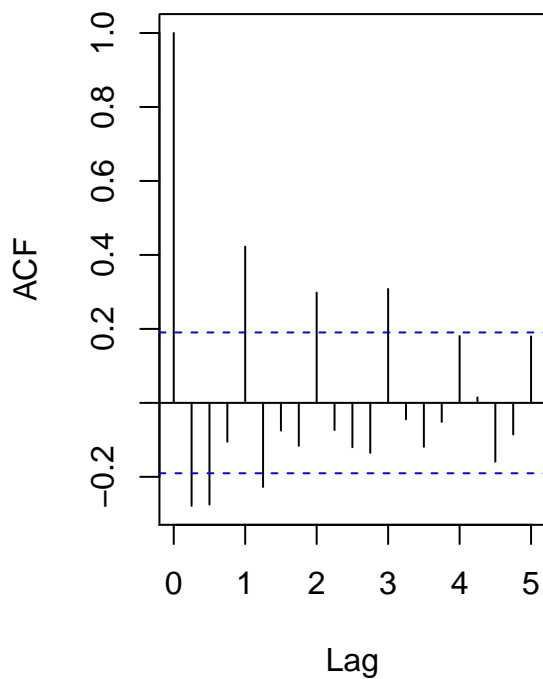
```
plot(decompose(MSE, "multiplicative"))
```

### Decomposition of multiplicative time series

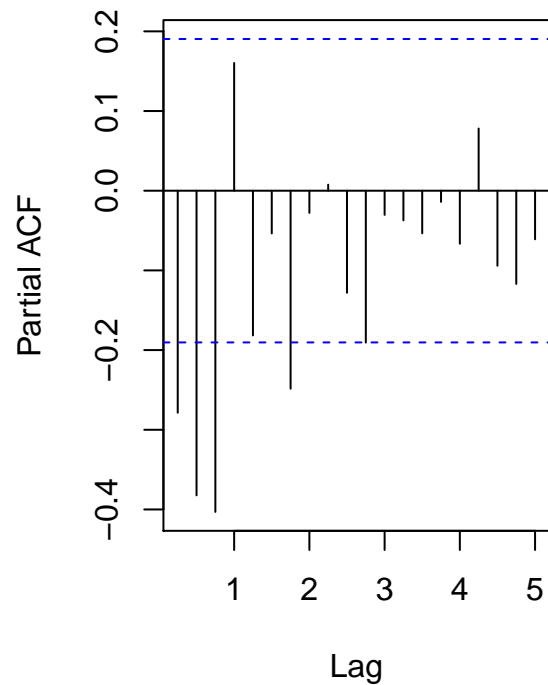


```
MSESta <- na.omit(decompose(MSE, "multiplicative")$random)
MSETrendTest <- window(decompose(MSE, "multiplicative")$trend, start=2016, end=c(2016,4))
MSESeasonalTest <- window(decompose(MSE, "multiplicative")$seasonal, start=2016, end=c(2016,4))
par(mfrow=c(1,2))
acf(MSESta, main="Auto-Corrélation de la Masse  
salariale trimestrielle stationnarisée")
pacf(MSESta, main="Auto-Corrélation partielle de la Masse  
salariale trimestrielle stationnarisée")
```

## Auto-Corrélation de la Masse salariale trimestrielle stationnar



## Auto-Corrélation partielle de la Ma salariale trimestrielle stationnar



```
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
```

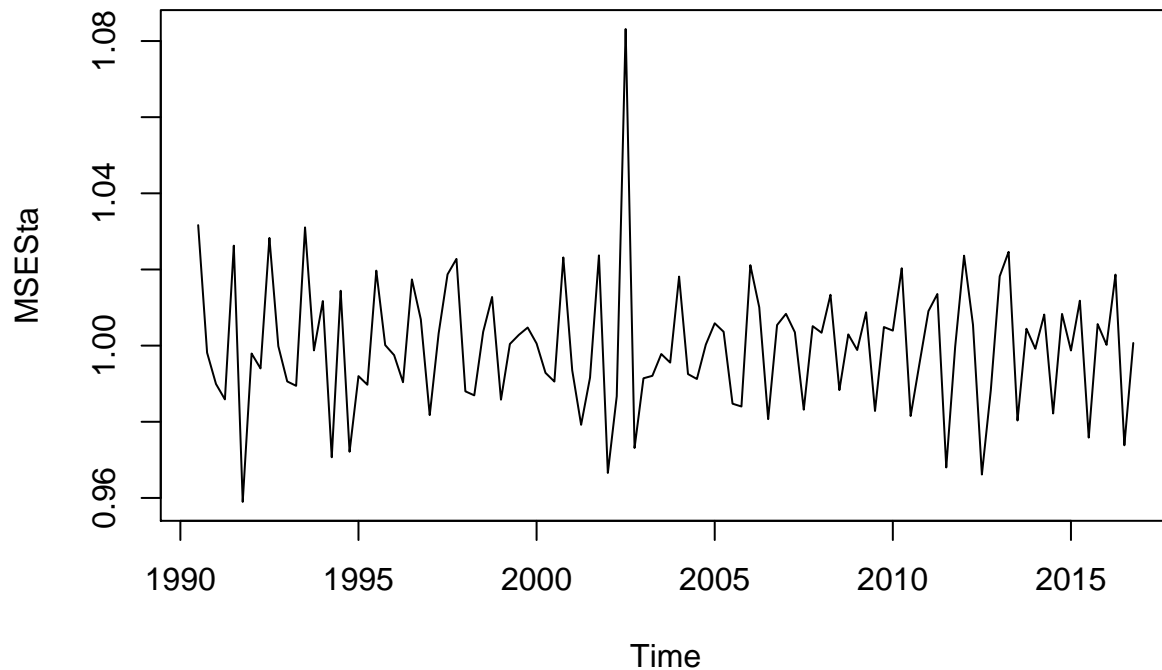
```
##
```

```
## data: MSESta
```

```
## KPSS Level = 0.021894, Truncation lag parameter = 2, p-value = 0.1
```

```
plot(MSESta, main="Masse salariale trimestrielle stationnarisée")
```

## Masse salariale trimestrielle stationnarisée



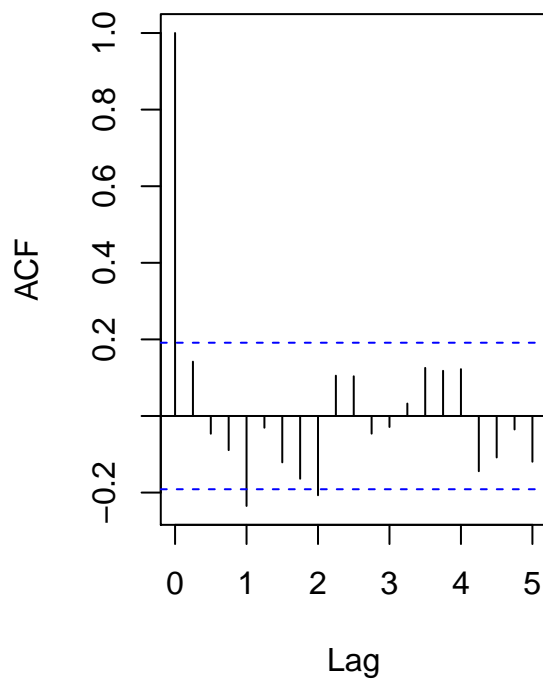
```
MSEStaTrain <- window(MSESta, end=c(2015,4))
MSEStaTest <- window(MSESta, start=2016)
MSETrain <- window(MSE, end=c(2015,4))
MSETest <- window(MSE, start=2016)
```

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 du test de KPSS nous amène à conserver l'hypothèse de stationnarité de la série avec un seuil de confiance à 5%.

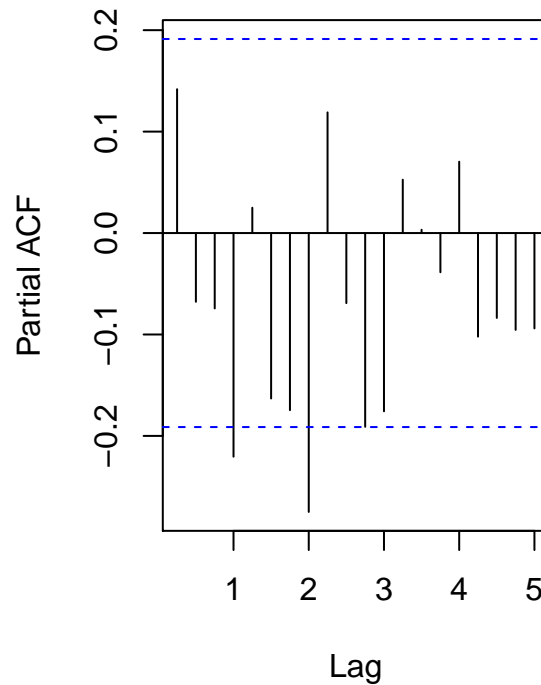
## PIB

```
PIBSta <- na.omit(decompose(PIB, "multiplicative")$random)
par(mfrow=c(1,2))
acf(PIBSta, main="Auto-Corrélation du PIB
    trimestrielle stationnarisée")
pacf(PIBSta, main="Auto-Corrélation partielle du PIB
    trimestrielle stationnarisée")
```

**Auto-Corrélation du PIB  
trimestrielle stationnarisée**



**Auto-Corrélation partielle du PII  
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
```

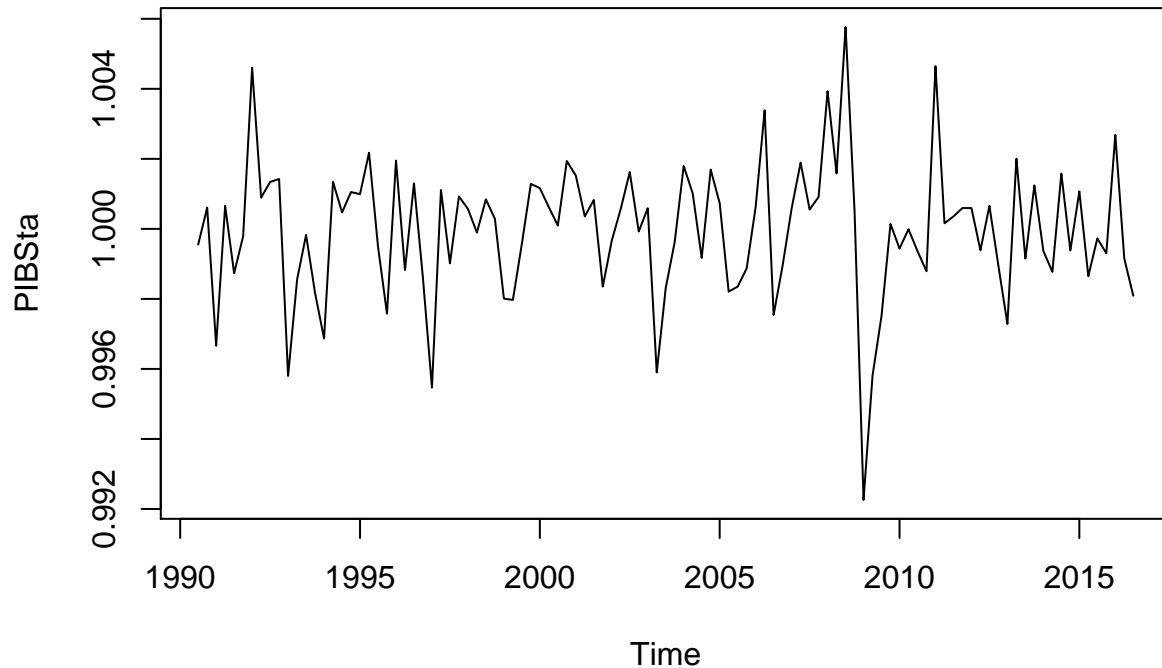
```
##
```

```
## data: PIBSta
```

```
## KPSS Level = 0.024935, Truncation lag parameter = 2, p-value = 0.1
```

```
plot(PIBSta, main="PIB trimestriel stationnarisé")
```

## PIB trimestriel stationnarisé



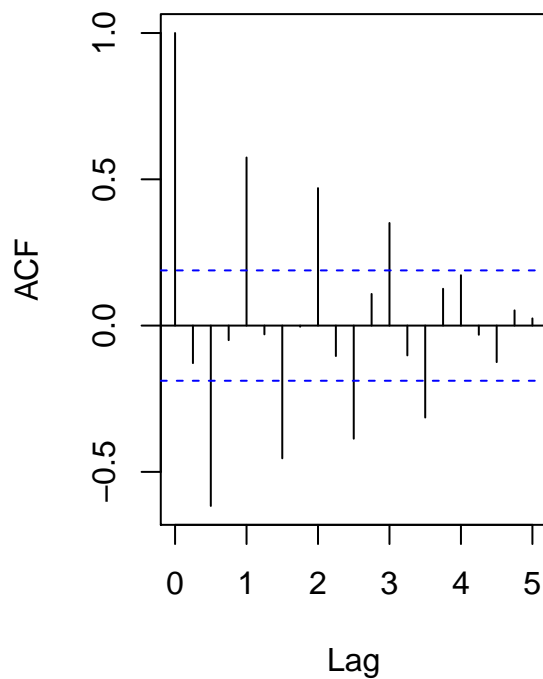
```
PIBStaTrain <- window(PIBSta, end=c(2015,4))
PIBStaTest <- window(PIBSta, start=2016)
PIBTrain <- window(PIB, end=c(2015,4))
PIBTest <- window(PIB, start=2016)
```

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. Au regard de ces différentes informations, nous pouvons conclure à la stationnarité des résidus.

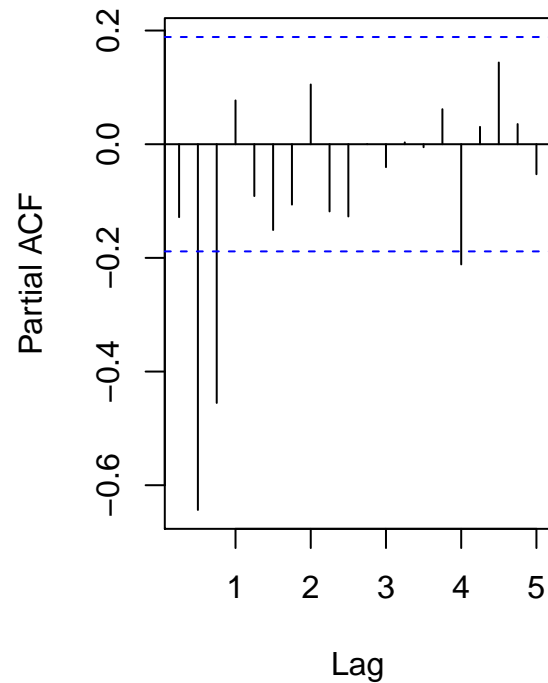
## SMIC

```
SMICSta <- na.omit(decompose(SMIC)$random)
par(mfrow=c(1,2))
acf(SMICSta, main="Auto-Corrélation du SMIC
trimestrielle stationnarisée")
pacf(SMICSta, main="Auto-Corrélation partielle du SMIC
trimestrielle stationnarisée")
```

**Auto-Corrélation du SMIC  
trimestrielle stationnarisée**



**Auto-Corrélation partielle du SM  
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
```

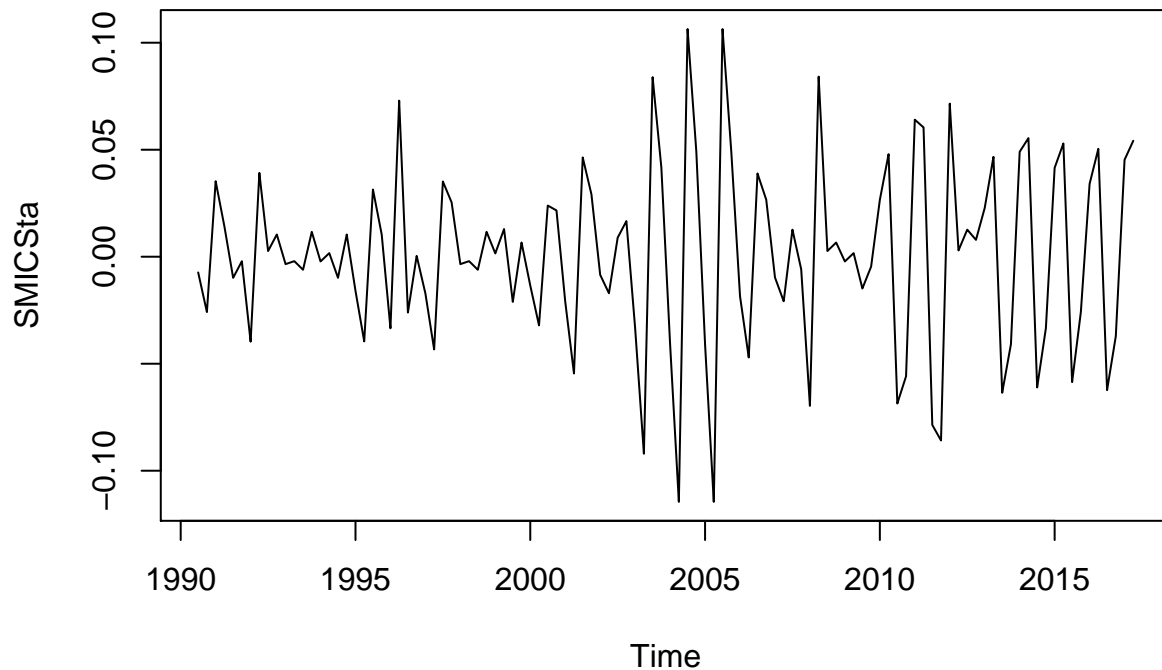
```
##
```

```
## data: SMICSta
```

```
## KPSS Level = 0.034306, Truncation lag parameter = 2, p-value = 0.1
```

```
plot(SMICSta, main="SMIC trimestriel stationnarisé")
```

## SMIC trimestriel stationnarisé



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

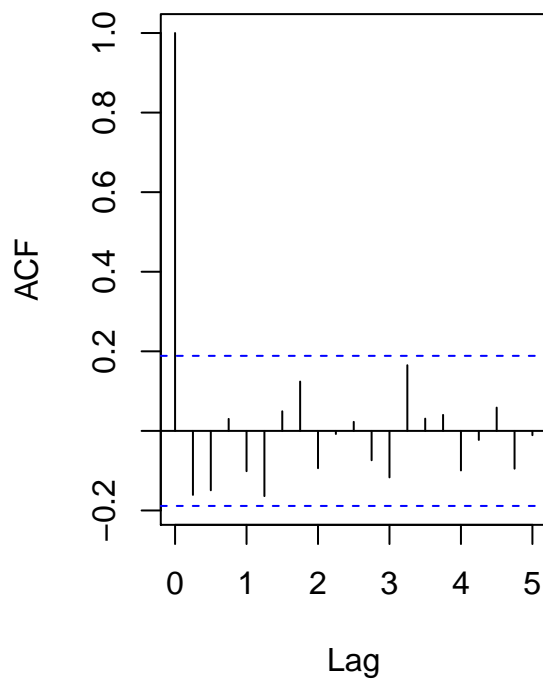
Comme pour la masse salariale, les ACF et PACF semblent montrer que la série résiduelle pourrait ne pas être stationnaire. Cependant le test de KPSS nous permet de conserver l'hypothèse de stationnarité des résidus.

## Taux de chômage des femmes

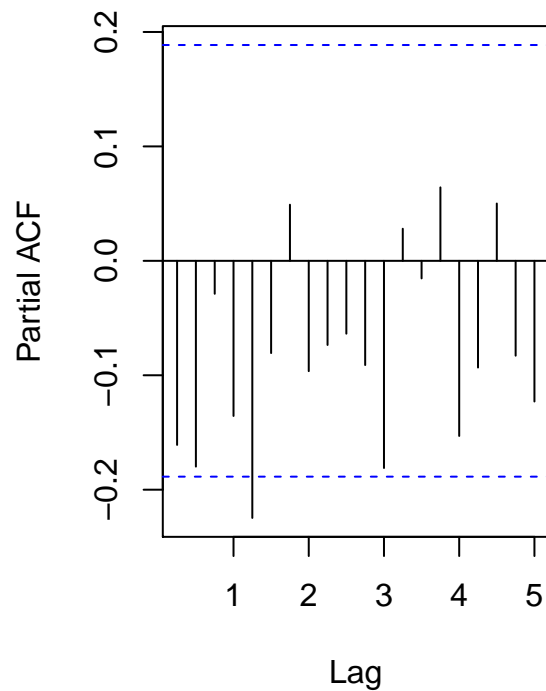
```
TCHOFSta <- na.omit(decompose(TCHOF)$random)
par(mfrow=c(1,2))
acf(TCHOFSta, main="Auto-Corrélation du Taux de
chômage des femmes
trimestrielle stationnarisée")
pacf(TCHOFSta, main="Auto-Corrélation partielle
du Taux de chômage des femmes
trimestrielle stationnarisée")
```



**chômage des femmes  
trimestrielle stationnarisée**



**du Taux de chômage des femnr  
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
```

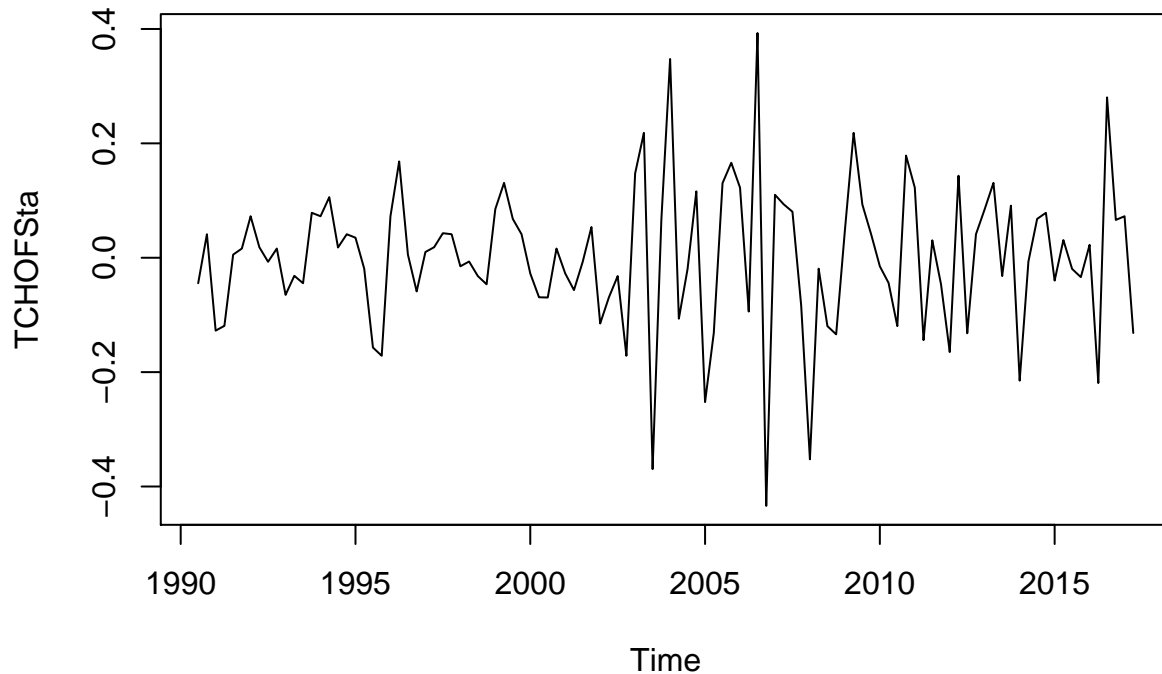
```
##
```

```
## data: TCHOFSta
```

```
## KPSS Level = 0.02222, Truncation lag parameter = 2, p-value = 0.1
```

```
plot(TCHOFSta, main="Taux de chômage trimestriel des femmes stationnarisé")
```

## Taux de chômage trimestriel des femmes stationnarisé



```
TCHOFSStaTrain <- window(TCHOFSSta, end=c(2015,4))
TCHOFSStaTest  <- window(TCHOFSSta, start=2016)
TCHOFTTrain    <- window(TCHOFSSta, end=c(2015,4))
TCHOFTTest     <- window(TCHOFSSta, start=2016)
```

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

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

## 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 valeurs de 4 critères (AIC, HQ, SC et FPE) pour différentes valeurs de l'ordre  $p$ .

Pour les critères suivants,  $n$  correspond à l'ordre de différenciation,  $T$  le nombre d'observations et  $K$  le nombre de variables et  $\tilde{\Sigma}_u(n) = T^{-1} \sum_{t=1}^T \hat{u}_t \hat{u}_t'$ .

Le critère AIC (Akaike information criterion) se calcule, dans ce package, de la manière suivante :  $AIC(n) = \ln \det(\tilde{\Sigma}_u(n)) + \frac{2}{T} n K^2$ . L'objectif est de minimiser ce critère.

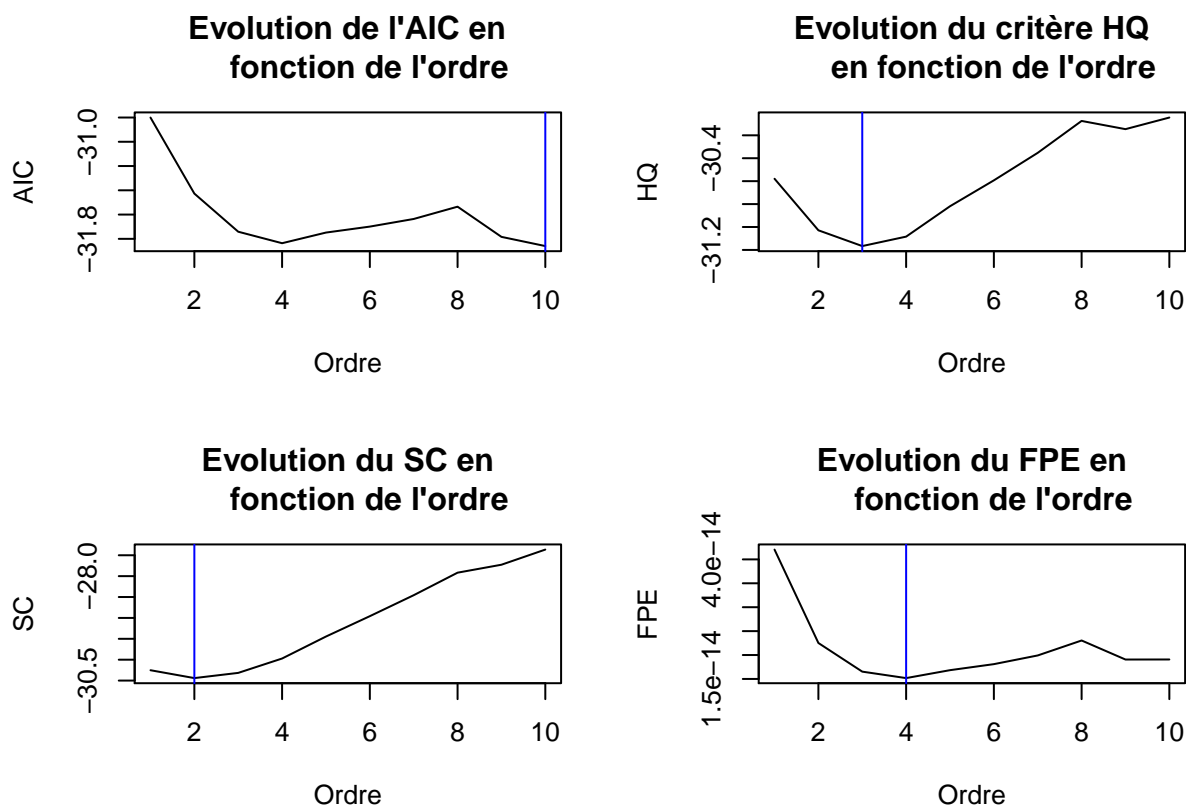
Le critère HQ (Hannan-Quinn criterion) se calcule, dans ce package, de la manière suivante :  $HQ(n) = \ln \det(\tilde{\Sigma}_u(n)) + \frac{2 \ln(\ln(T))}{T} n K^2$ . L'objectif est de minimiser ce critère. Contrairement à l'AIC, ce critère n'est pas asymptotiquement efficace.

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

Le dernier critère, le critère FPE (Final Prediction Error), est un critère à minimiser. Cependant nous ne comprenons pas son fonctionnement. Nous ne l'utiliserons donc pas.

Dans cette partie, nous développerons le fonctionnement 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(MSEStaTrain, PIBStaTrain, SMICStaTrain, TCHOFSStaTrain), 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")
plot(seq(1:10),t(selec$criteria[4,]), type="l", main="Evolution du FPE en
fonction de l'ordre",
xlab="Ordre", ylab="FPE")
abline(v=which.min(selec$criteria[4,]), 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, veiller à minimiser le critère AIC, tout en étant attentif aux autres critères. C'est pourquoi, ici, nous décidons de conserver un ordre 4.

## Estimation du modèle

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, qui nous donne donc un ordre de 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  $u_t$  le vecteur des résidus. Dans notre cas, nous aurons donc 3 matrices de taille 5x5.

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  $\mu$ .

```
modele<-VAR(cbind(MSEStaTrain, PIBStaTrain, SMICStaTrain, TCHOFStaTrain), p=4, type="const")
```

Les coefficients du modèle sont les suivants

```
A1<-cbind(modele$varresult$MSEStaTrain$coefficients[1:4],
           modele$varresult$PIBStaTrain$coefficients[1:4],
           modele$varresult$SMICStaTrain$coefficients[1:4],
           modele$varresult$TCHOFStaTrain$coefficients[1:4])
colnames(A1)<-rownames(A1)<-c("MSE", "PIB", "SMIC", "TCHOF")
A1
```

##		MSE	PIB	SMIC	TCHOF
## MSE		-0.49912624	-0.0308097020	0.05328488	-0.8445102
## PIB		0.92777848	0.1257671971	1.21484663	-5.9314489
## SMIC		0.06599595	-0.0051246390	-0.49802200	0.6626694
## TCHOF		-0.01352986	0.0006699293	-0.01008137	-0.3235506

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(MSEStaTrain, PIBStaTrain, SMICStaTrain, TCHOFStaTrain))
selection$criteria[,4]
```

##	AIC(n)	HQ(n)	SC(n)	FPE(n)
##	-3.183609e+01	-3.108380e+01	-2.997216e+01	1.517819e-14

## 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(MSEStaTrain, SMICStaTrain), lag.max=10)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      3      3      3      3
##
## $criteria
##           1           2           3           4
## AIC(n) -1.437706e+01 -1.504929e+01 -1.541342e+01 -1.538725e+01
## HQ(n)  -1.431068e+01 -1.493865e+01 -1.525853e+01 -1.518811e+01
## SC(n)  -1.421259e+01 -1.477518e+01 -1.502967e+01 -1.489386e+01
## FPE(n)  5.703539e-07  2.912536e-07  2.024391e-07  2.079440e-07
##           5           6           7           8
## AIC(n) -1.535114e+01 -1.530767e+01 -1.535484e+01 -1.529623e+01
## HQ(n)  -1.510775e+01 -1.502003e+01 -1.502295e+01 -1.492008e+01
## SC(n)  -1.474810e+01 -1.459500e+01 -1.453252e+01 -1.436427e+01
## FPE(n)  2.158164e-07  2.257434e-07  2.157878e-07  2.294353e-07
##           9          10
## AIC(n) -1.525746e+01 -1.526620e+01
## HQ(n)  -1.483706e+01 -1.480155e+01
## SC(n)  -1.421586e+01 -1.411495e+01
## FPE(n)  2.393329e-07  2.382777e-07
```

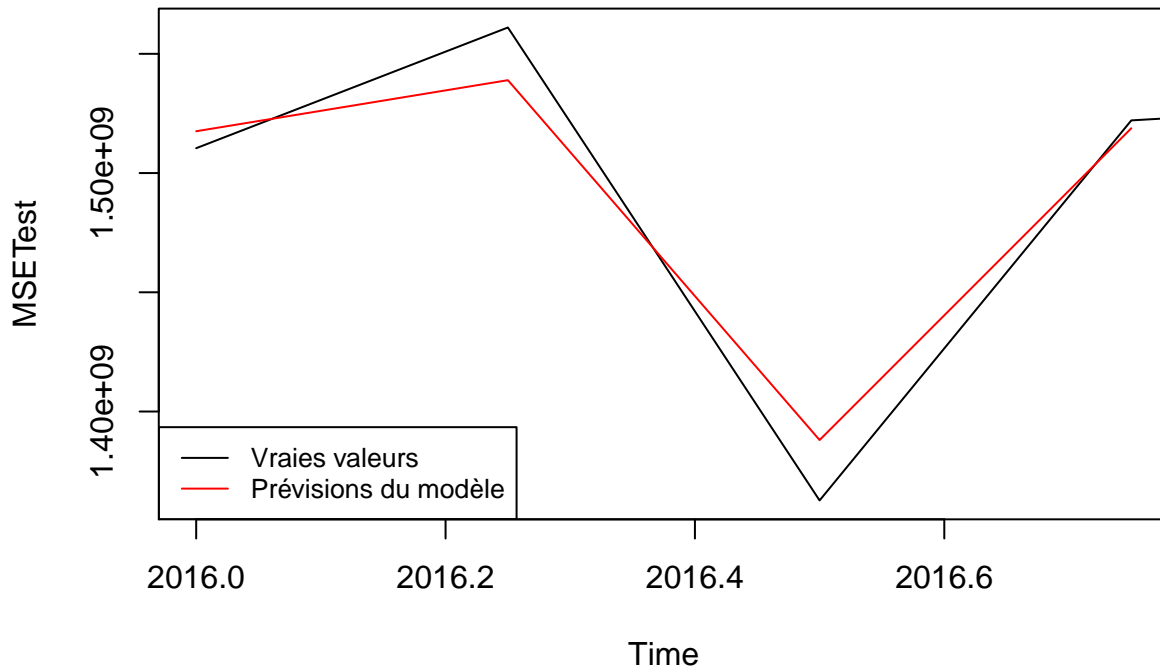
```
VAR(cbind(MSEStaTrain, SMICStaTrain), p=3, type="const")
```

```
##
## VAR Estimation Results:
## =====
##
## Estimated coefficients for equation MSEStaTrain:
## =====
## Call:
## MSEStaTrain = MSEStaTrain.l1 + SMICStaTrain.l1 + MSEStaTrain.l2 + SMICStaTrain.l2 + MSEStaTrain.l3 +
##
## MSEStaTrain.l1 SMICStaTrain.l1 MSEStaTrain.l2 SMICStaTrain.l2
## -0.554842568      0.001362944    -0.550136180    -0.014196261
## MSEStaTrain.l3 SMICStaTrain.l3          const
## -0.395923428    -0.013169066      2.500225678
##
##
## Estimated coefficients for equation SMICStaTrain:
## =====
## Call:
## SMICStaTrain = MSEStaTrain.l1 + SMICStaTrain.l1 + MSEStaTrain.l2 + SMICStaTrain.l2 + MSEStaTrain.l3 +
##
## MSEStaTrain.l1 SMICStaTrain.l1 MSEStaTrain.l2 SMICStaTrain.l2
## -0.01867915    -0.54580898    -0.22996141    -0.74042266
## MSEStaTrain.l3 SMICStaTrain.l3          const
## -0.35846413    -0.46120378      0.60780380
```

```
VARSMICSta <- forecast(VAR(cbind(MSEStaTrain, SMICStaTrain), p=3, type="const"))
plot(MSETest, xlim=c(2016,2016.75), main="Différences entre les véritables
valeurs de 2016 et les prédictions du modèle pour la masse salariale")
#Reconstruction de la variable stationnaire
```

```
recon <- VARSMICSta$forecast$MSEStaTrain$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)
```

## Différences entre les véritables valeurs de 2016 et les prédictions du modèle pour la masse salariale



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

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

```
## [1] 1.984486e+14
```