Package MTS

Jules CORBEL & Paul GUILLOTTE

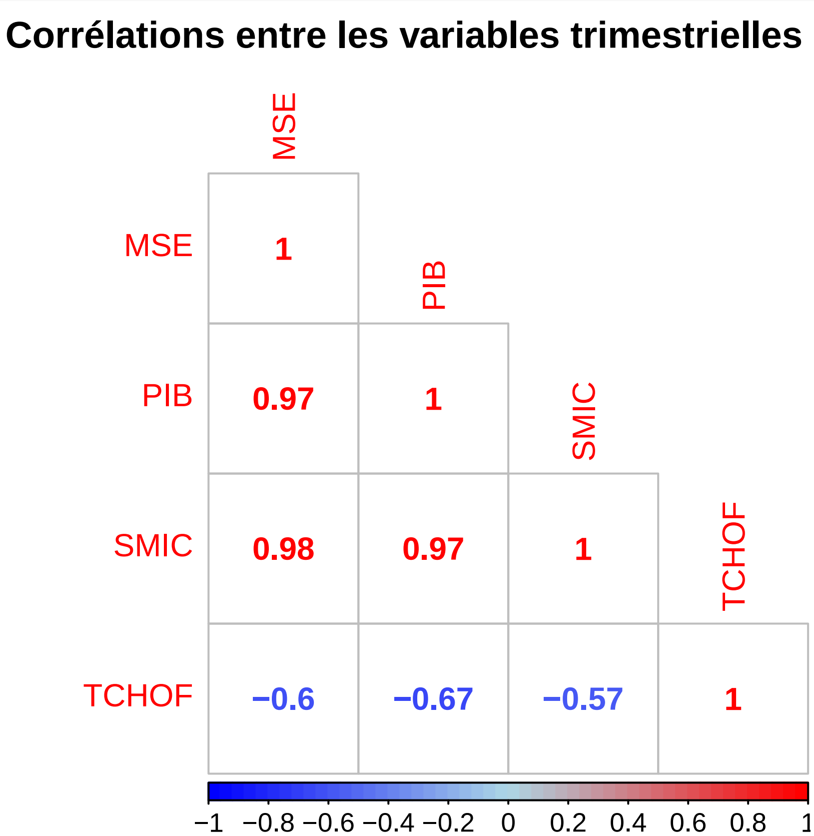
04/02/2019

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.

Visualation 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")
```

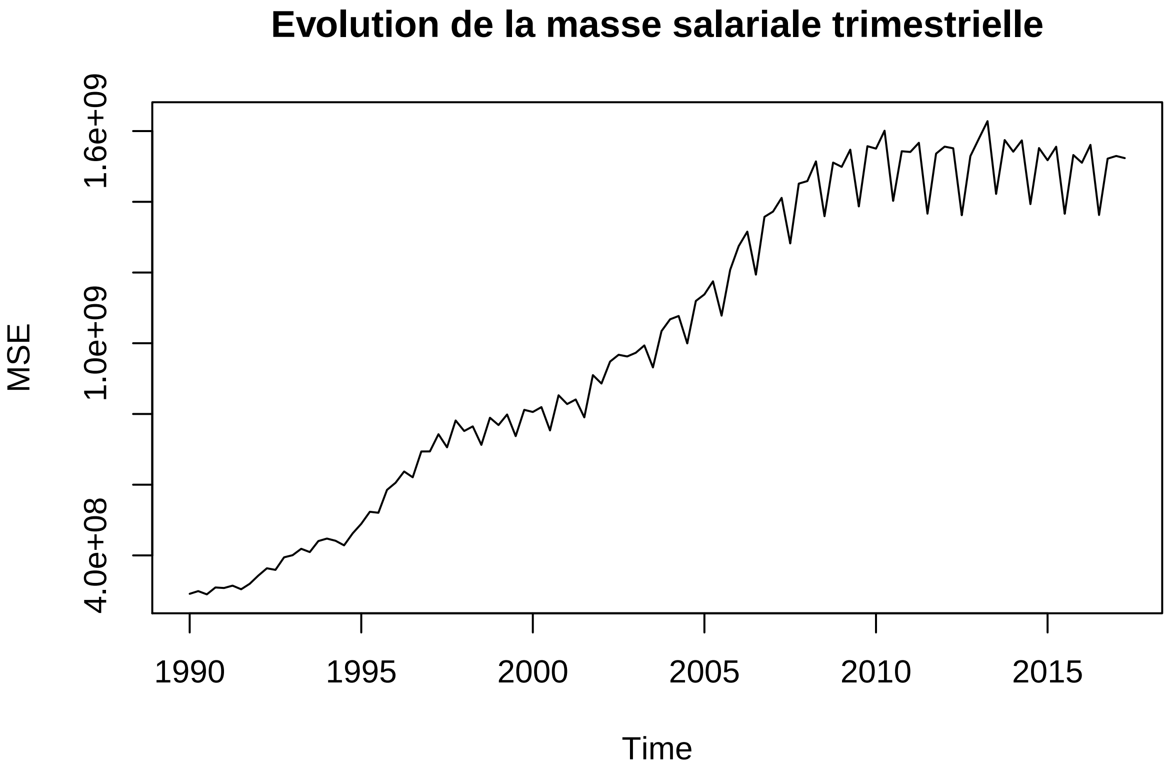


On se rend compte que le taux de chômage des femmes est corrélé négativement avec toutes les autres variables. Le trio de variables PIB, masse salariale et SMIC sont extrêmement liées entre elles.

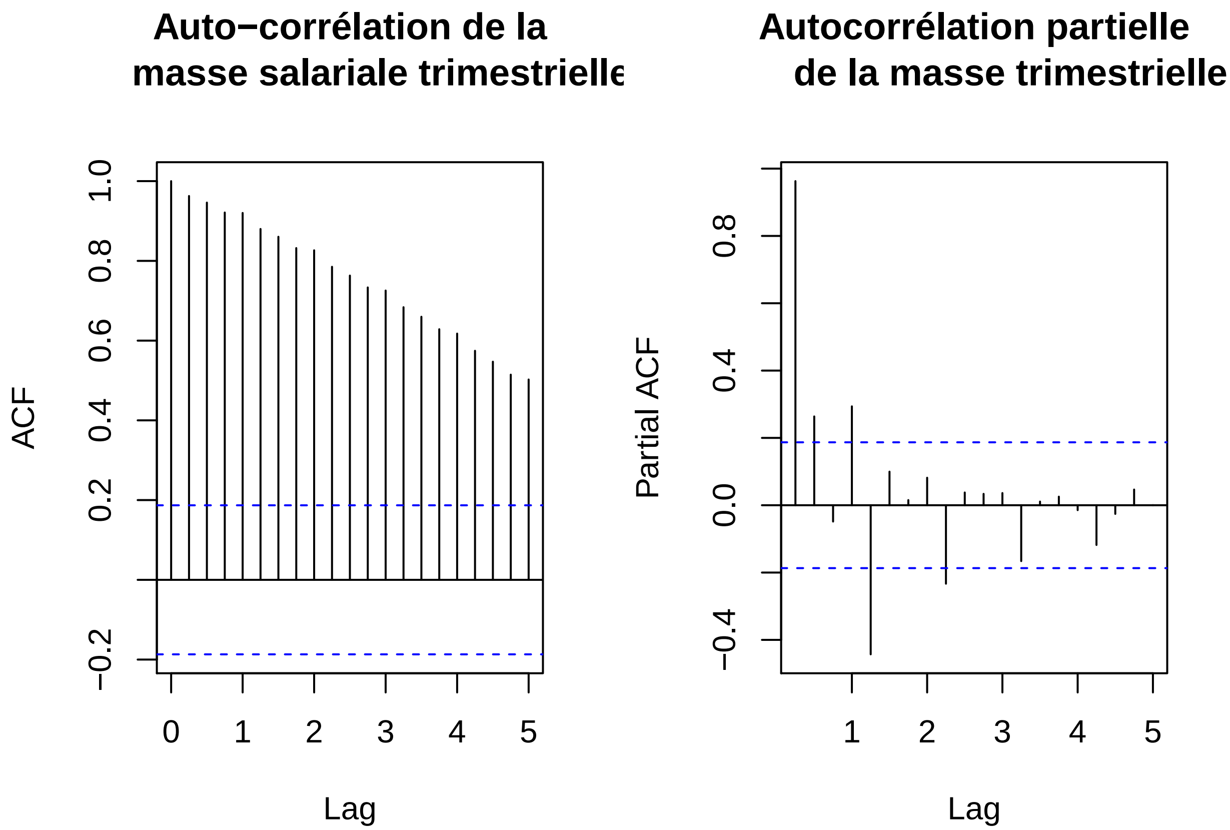
Nous allons maintenant nous attarder sur chaque série individuellement.

Masse salariale

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




```
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)
```




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

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

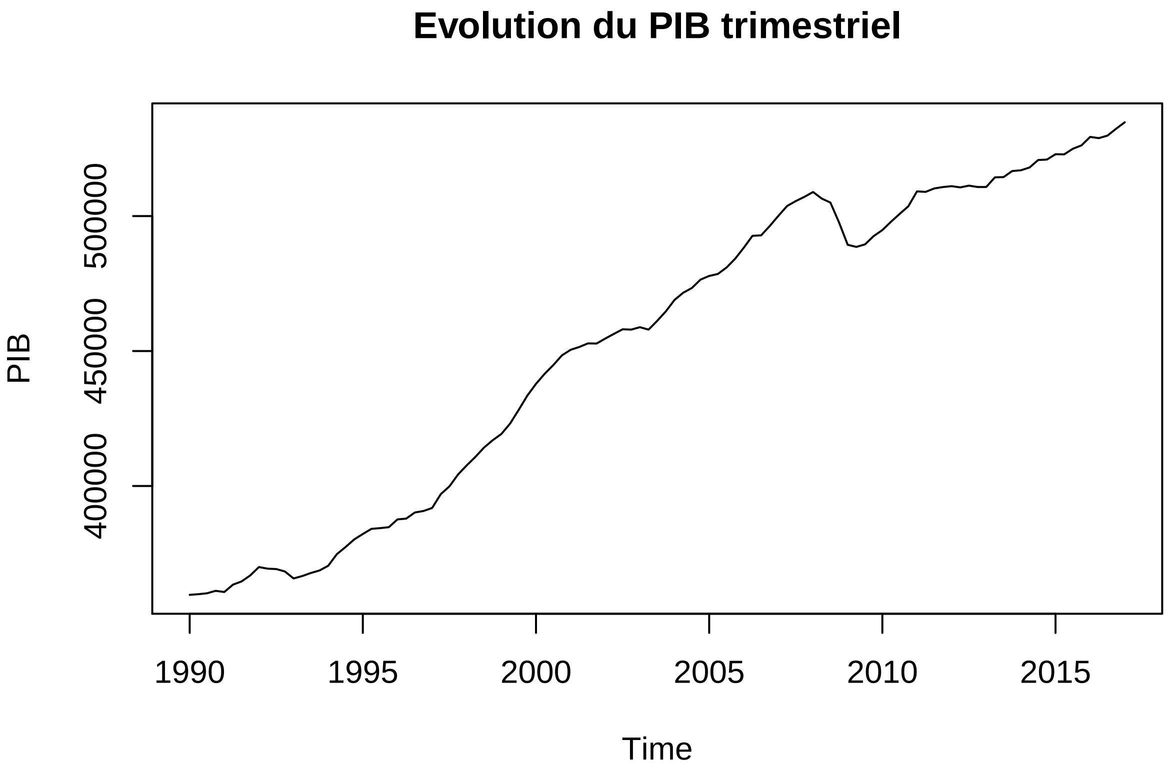
```
##
##  KPSS Test for Level Stationarity
##
## data:  MSE
## KPSS Level = 3.6772, Truncation lag parameter = 2, p-value = 0.01
```

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.

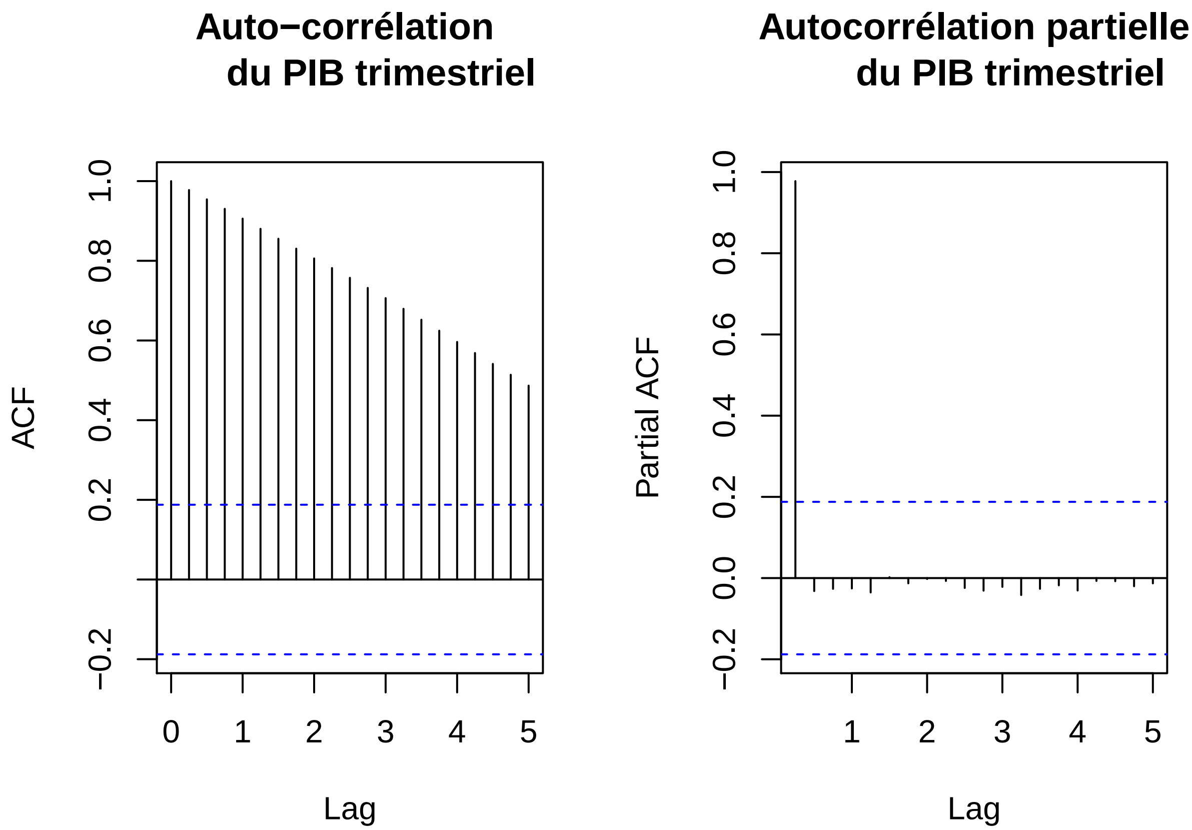
Comme la série comporte une tendance et une saisonnalité, elle ne correspond pas aux deux premières conditions de la stationnarité du second ordre, soit que la série possède une moyenne et un écart-type constants. Cela est confirmé par la fonction ACF qui décroît régulièrement. Nous effectuons également 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

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




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

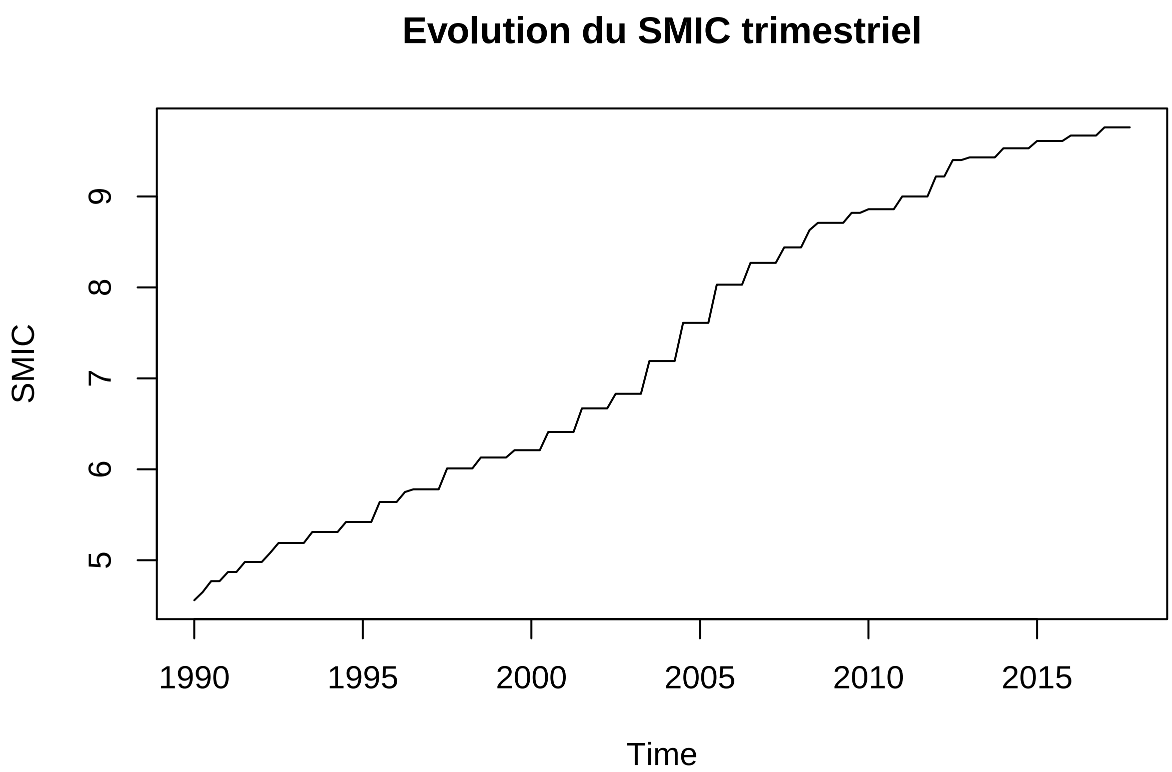
```
## Warning in kpss.test(PIB): p-value smaller than printed p-value

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

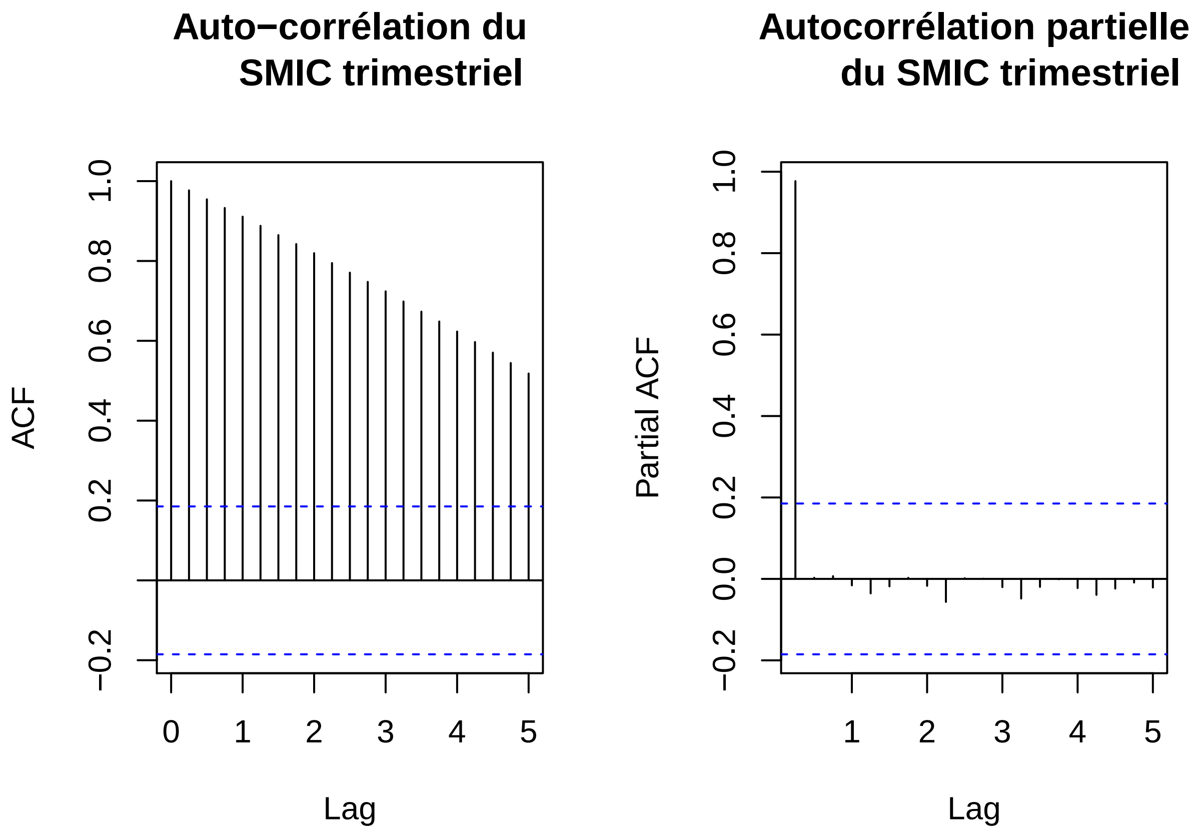
Comme pour la masse salariale, le PIB annuel possède une tendance. Cependant, il ne semble pas posséder de saisonnalité. Cette série n'est donc pas non plus stationnaire. Nous effectuons à nouveau un test de KPSS. La p-value est de 0.01 ce qui nous confirme que la série n'est pas stationnaire avec un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

SMIC

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




```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)
```

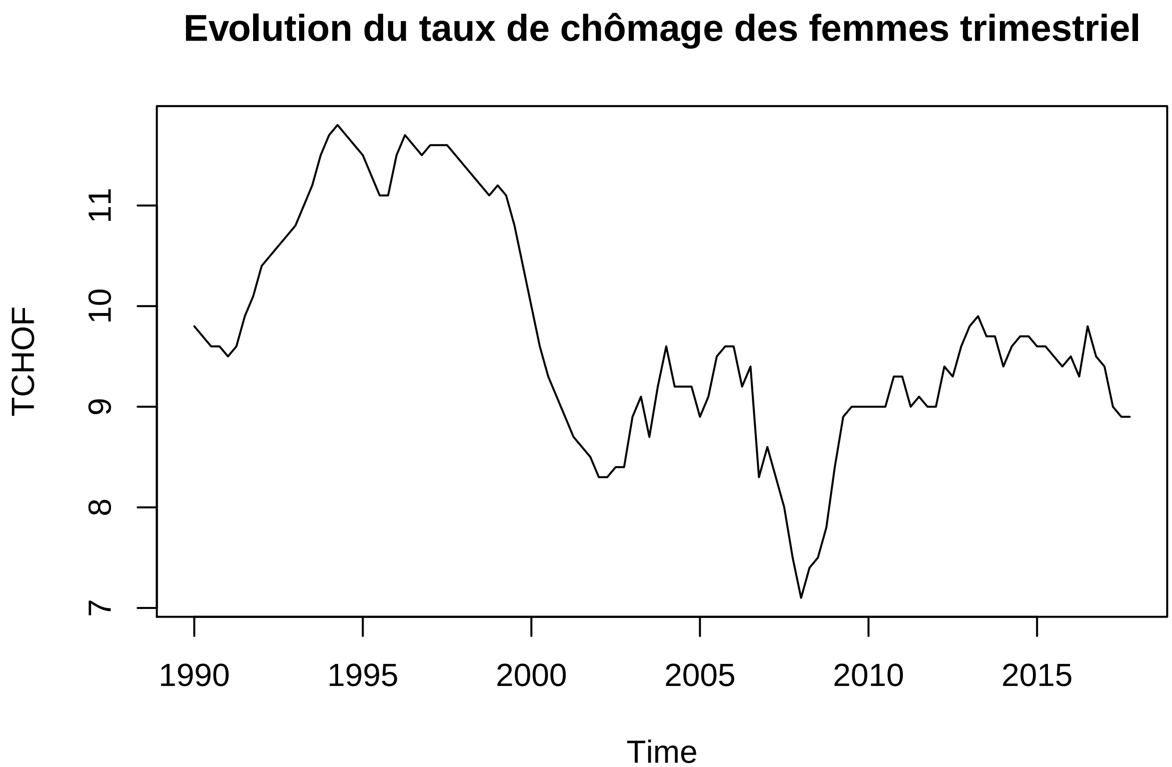
```
## Warning in kpss.test(SMIC): p-value smaller than printed p-value
```

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

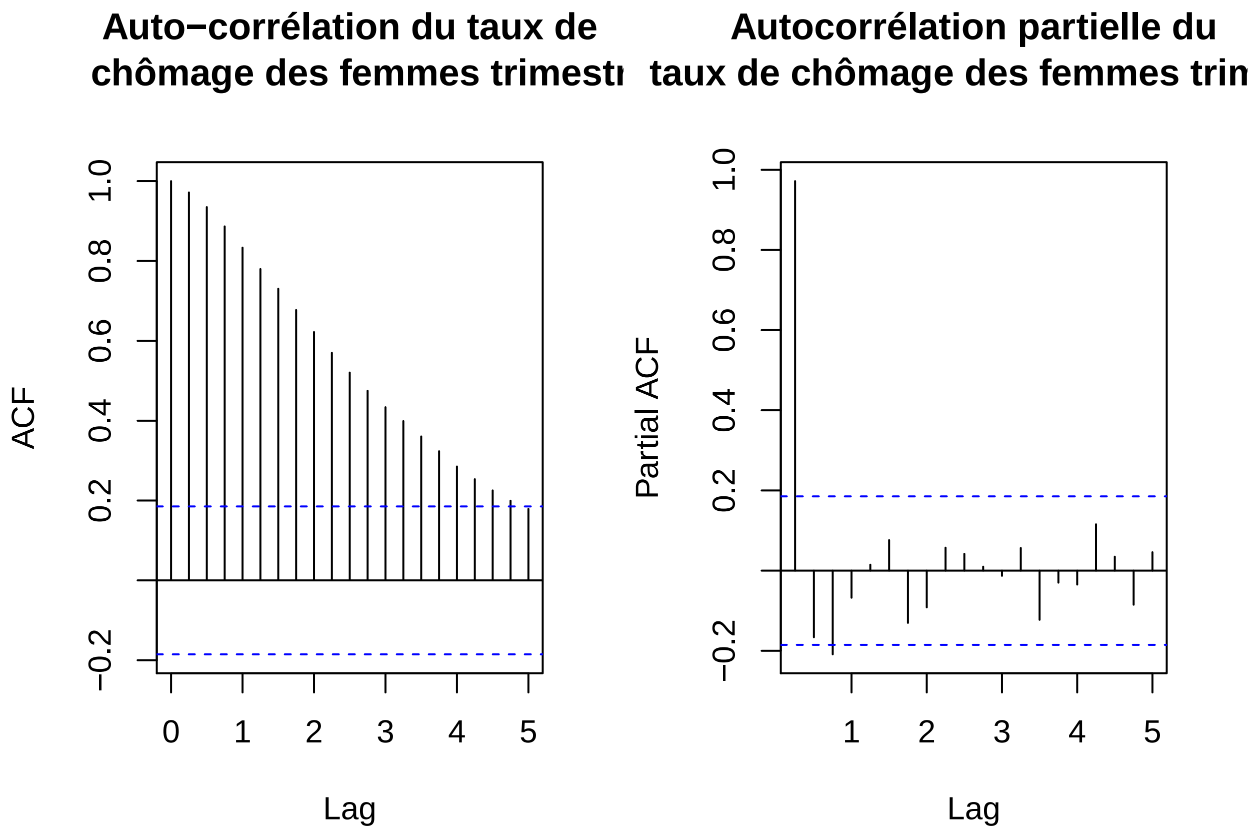
Au regard de la représentation graphique, on s'aperçoit qu'il y a bien une tendance. Pour la saisonnalité, il est plus difficile de savoir s'il en existe une ou pas, puisque la série semble augmenter seulement à certains temps.

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")
```




```
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)
```




```r
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.

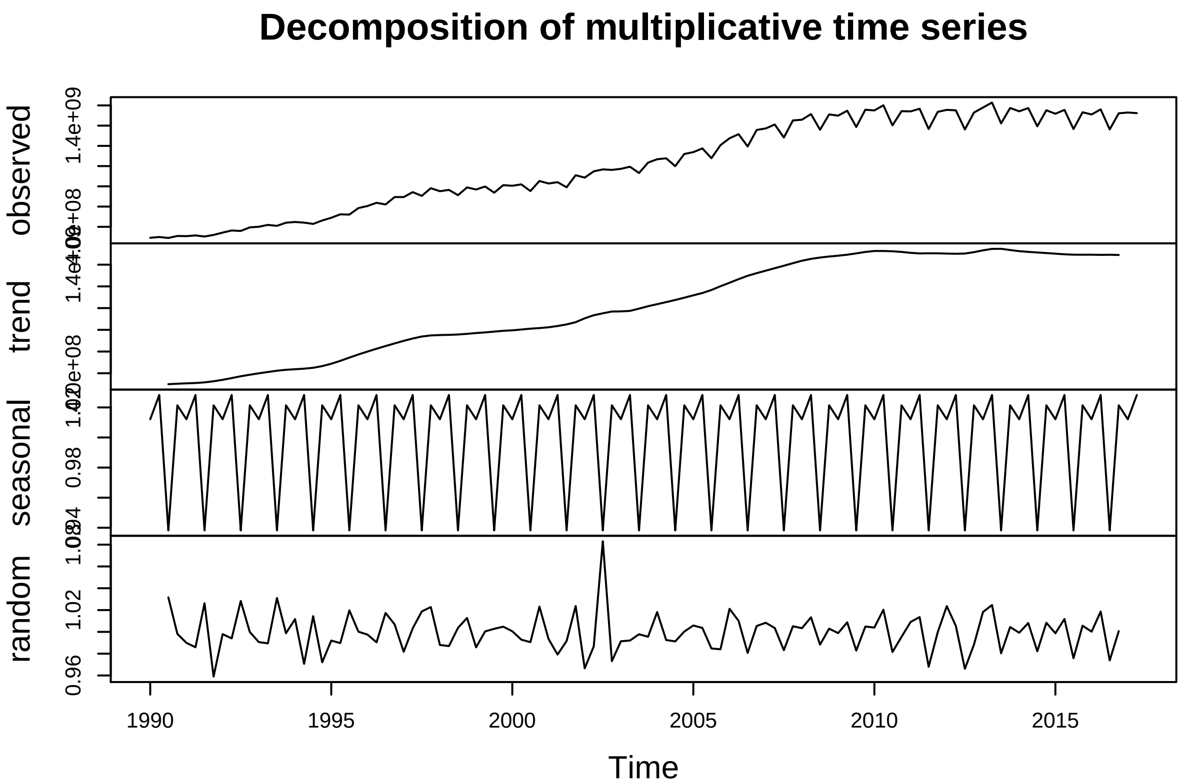
Transformation des séries

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.

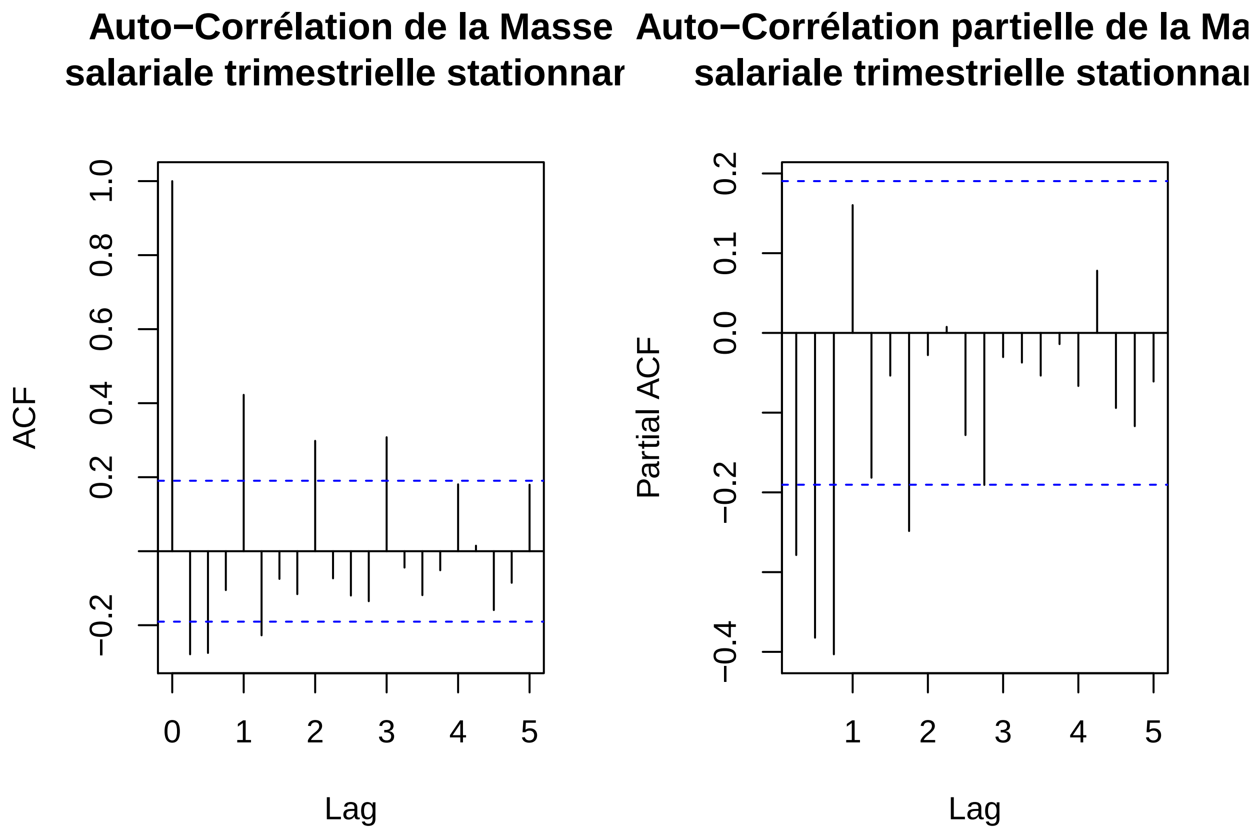
Pour chacune de ces séries, nous allons créé un échantillon d'apprentissage, qui nous permettra de construire les différents modèles, ainsi qu'un échantillon de test, qui nous permettra de comparer les prédictions des modèles construits avec des vraies valeurs. L'échantillon d'apprentissage sera composé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

Masse salariale

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




```r
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")
```

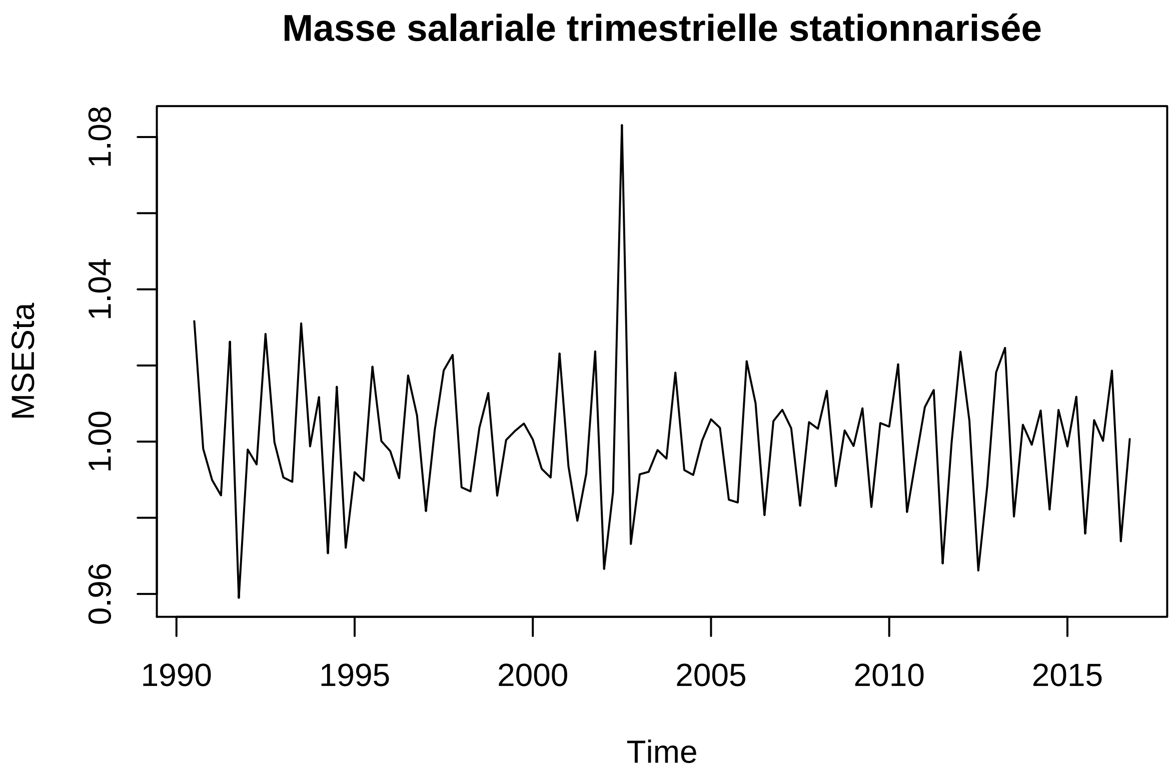



```r
par(mfrow=c(1,1))
kpss.test(MSESta)
```

```
## 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
```

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

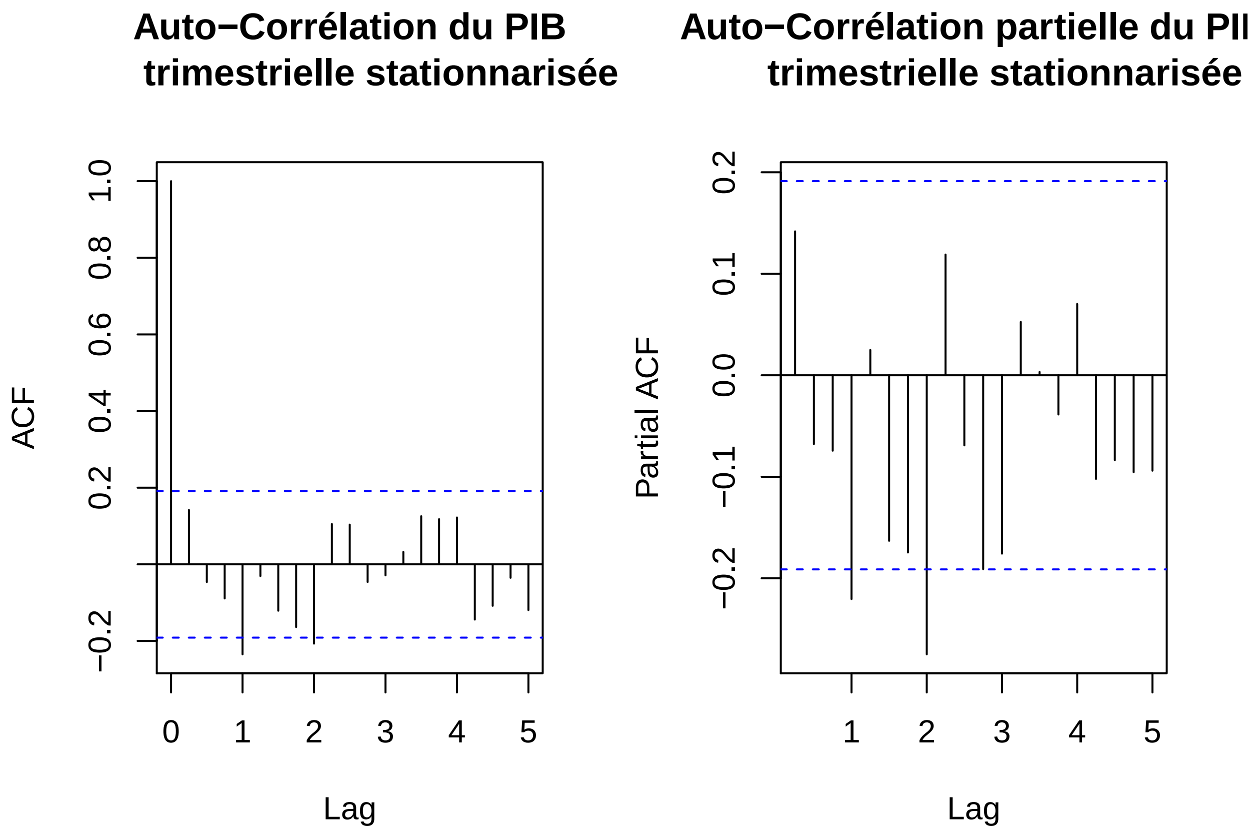



```
MSEStaTrain <- window(MSESta, end=c(2015,4))
MSEStaTest <- window(MSESta, start=2016)
MSETrain <- window(MSE, end=c(2015,4))
MSETest <- window(MSE, start=2016)
```

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")
```

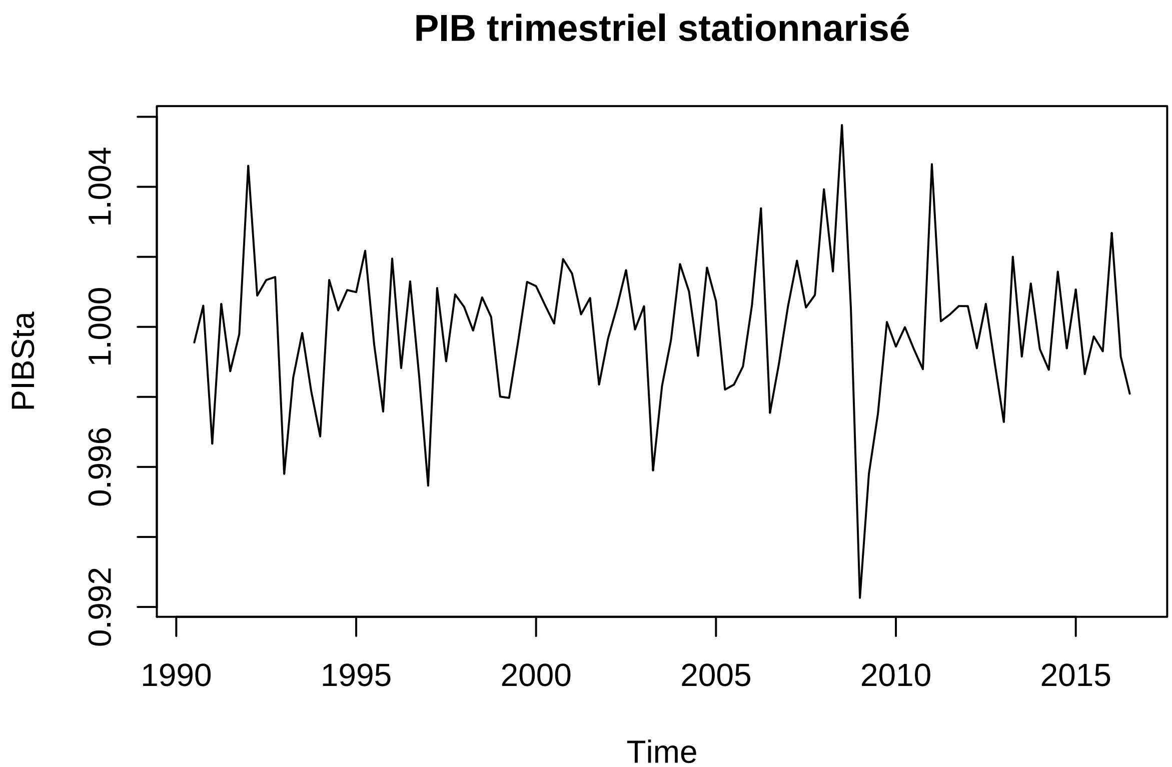



```r
par(mfrow=c(1,1))
kpss.test(PIBSta)
```

```
## 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
```

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

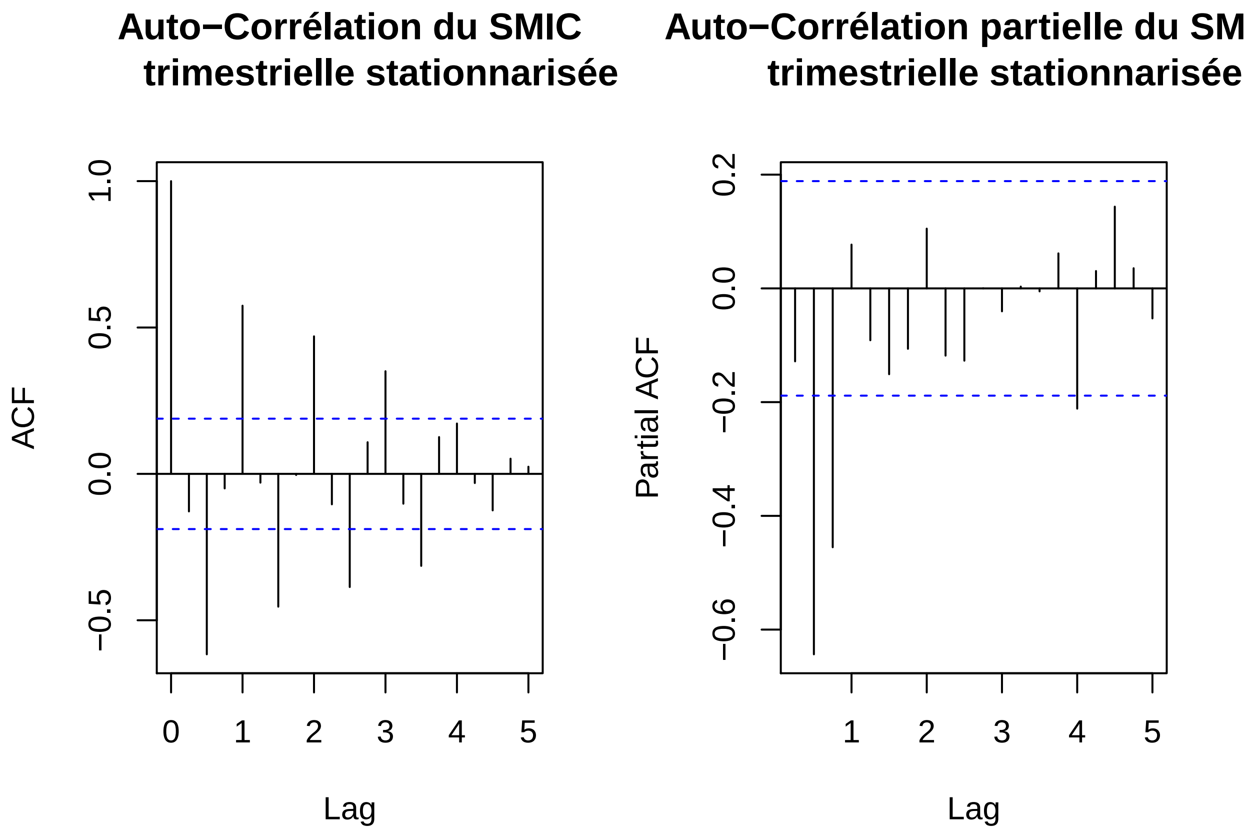



```
PIBStaTrain <- window(PIBSta, end=c(2015,4))
PIBStaTest <- window(PIBSta, start=2016)
PIBTrain <- window(PIB, end=c(2015,4))
PIBTest <- window(PIB, start=2016)
```

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")
```

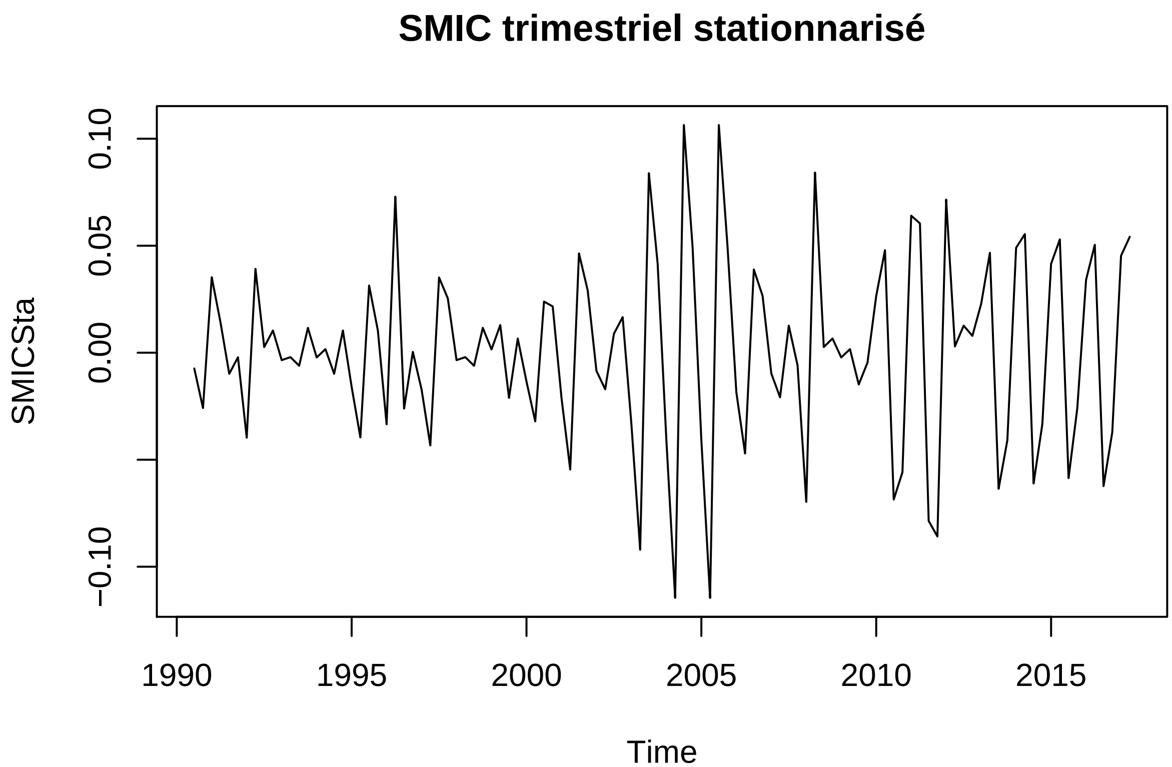



```r
par(mfrow=c(1,1))
kpss.test(SMICSta)
```

```
## 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
```

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

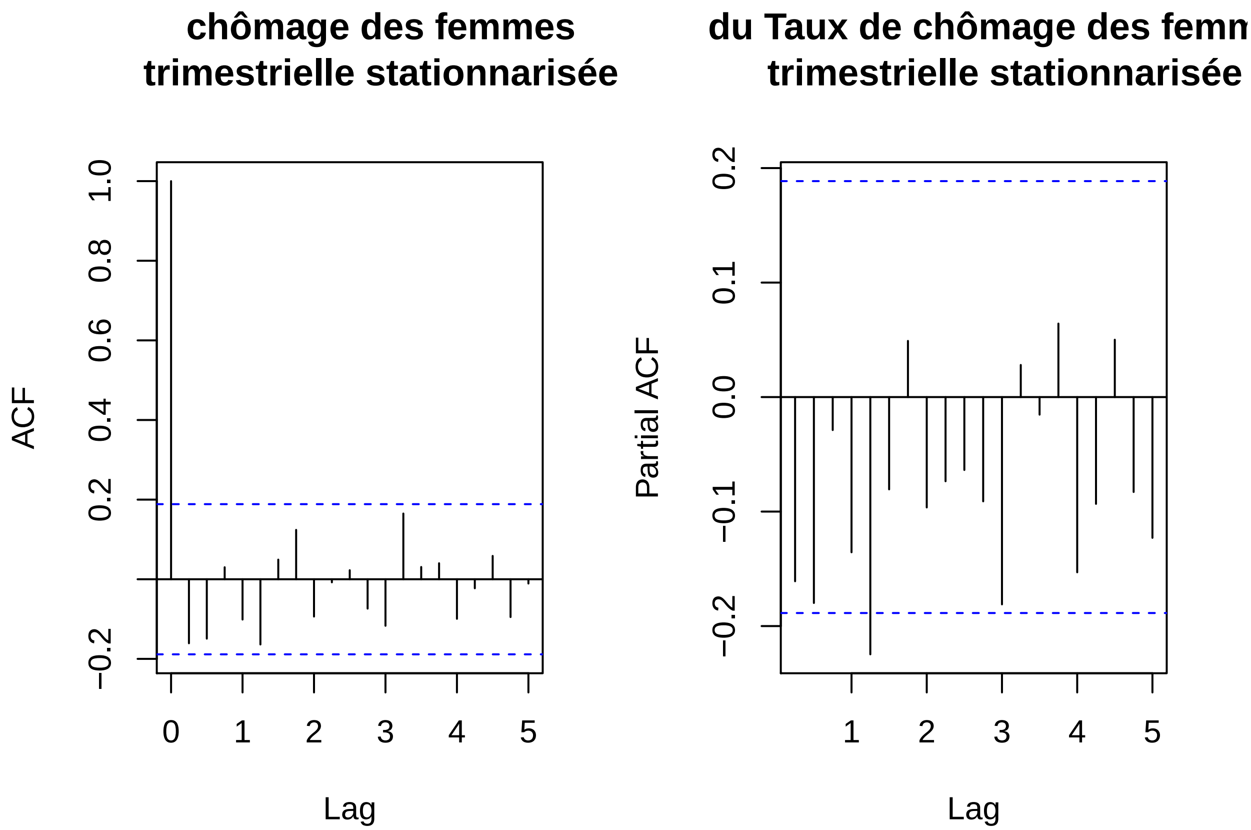



```
SMICStaTrain <- window(SMICSta, end=c(2015,4))
SMICStaTest <- window(SMICSta, start=2016)
SMICTrain <- window(SMIC, end=c(2015,4))
SMICTest <- window(SMIC, start=2016)
```

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")
```

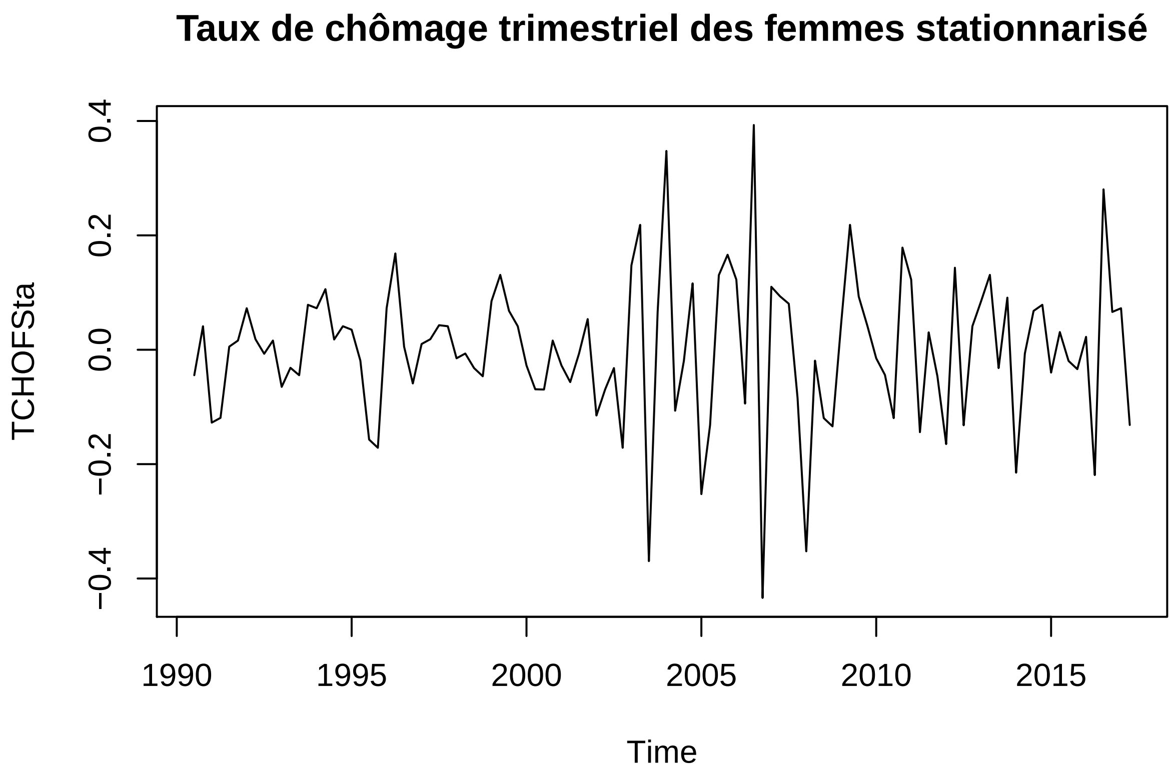



```r
par(mfrow=c(1,1))
kpss.test(TCHOFSta)
```

```
## 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
```

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




```r
TCHOFStaTrain <- window(TCHOFSta, end=c(2015,4))
TCHOFStaTest <- window(TCHOFSta, start=2016)
TCHOFTrain <- window(TCHOF, end=c(2015,4))
TCHOFTest <- window(TCHOF, 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 **MTS**, la fonction utilisée est VARorder, qui comme VARselect utilise les critères d'AIC, BIC et HQ afin de déterminer l'ordre du processus. Toutefois, le critère FPE n'est pas présent, ce qui nous conforte dans notre idée qu'il n'est pas très utile à notre étude. Les valeurs des différents indicateurs nous font retenir un ordre p=3.

```r
selec <- VARorder(cbind(MSEStaTrain, PIBStaTrain, SMICStaTrain, TCHOFStaTrain), maxp=10)
```

```
## selected order: aic =  10
## selected order: bic =  0
## selected order: hq =  3
## Summary table:
##        p      AIC      BIC       HQ    M(p) p-value
##  [1,]  0 -31.0096 -31.0096 -31.0096  0.0000  0.0000
##  [2,]  1 -30.9214 -30.5096 -30.7546 19.5070  0.2432
##  [3,]  2 -31.5822 -30.7587 -31.2487 80.4006  0.0000
##  [4,]  3 -31.9307 -30.6954 -31.4305 51.9850  0.0000
```

```
##  [5,]  4 -32.0595 -30.4124 -31.3925 32.9649  0.0075
##  [6,]  5 -32.0057 -29.9469 -31.1721 18.3312  0.3049
##  [7,]  6 -31.9908 -29.5202 -30.9904 19.8690  0.2262
##  [8,]  7 -31.9619 -29.0796 -30.7948 17.8037  0.3355
##  [9,]  8 -31.8948 -28.6007 -30.5609 14.4236  0.5672
## [10,]  9 -32.1777 -28.4719 -30.6771 32.5182  0.0086
## [11,] 10 -32.2875 -28.1699 -30.6202 21.3890  0.1640
```

Estimation du modèle

Dans le package **MTS**, la fonction utilisée pour construire des modèles VAR est *VAR*, qui prend en entrée la série temporelle multivariée et l'ordre du processus. On affiche ci-dessous les résultats renvoyés par la fonction sur le modèle

```r
modele<-VAR(cbind(MSEStaTrain, PIBStaTrain, SMICStaTrain, TCHOFStaTrain), p=3, output=F)
print("Coefficients du modèle")
```

```
## [1] "Coefficients du modèle"
```

```r
coeff<-modele$coef[2:5,]
colnames(coeff)<-rownames(coeff)<-c("MSE", "PIB", "SMIC", "TCHOF")
coeff
```

```
##                MSE           PIB        SMIC      TCHOF
## MSE   -0.56278414 -0.0168586446 -0.02625032 -0.3580146
## PIB    0.64149028  0.1814898925  0.48689607 -6.7304243
## SMIC   0.00298799 -0.0002301839 -0.53899891  0.1622353
## TCHOF -0.00732917  0.0011657405 -0.01560563 -0.2811625
```

```r
print("Erreurs standard des coefficients")
```

```
## [1] "Erreurs standard des coefficients"
```

```r
se<-modele$secoef[2:5,]
colnames(se)<-rownames(se)<-c("MSE", "PIB", "SMIC", "TCHOF")
se
```

```
##              MSE         PIB       SMIC     TCHOF
## MSE   0.09953303 0.013493075 0.20448277 0.8112122
## PIB   0.78217596 0.106034738 1.60691891 6.3748752
## SMIC  0.04600804 0.006237024 0.09451989 0.3749738
## TCHOF 0.01283070 0.001739379 0.02635967 0.1045725
```

```r
matrix(data=c(modele$aic, modele$bic, modele$hq), nrow=1, byrow=T, dimnames=list(NULL, c("AIC", "BIC",
```

```
##            AIC       BIC        HQ
## [1,] -31.79783 -30.56255 -31.29762
```

Afin de vérifier que les résidus du modèle ne présentent ni d'auto-corrélation (corrélation dans une série entre deux temps différents), ni de corrélation croisée (corrélation entre deux séries différentes).

```r
LjungBox(modele$residuals, 1:10)
```

```
##  lags  statistic  df   p-value
##     1   6.566401  16 0.9807419
##     2  11.053772  32 0.9997889
##     3  17.856956  48 0.9999783
##     4  38.659275  64 0.9949102
```

```
##     5  60.230726  80 0.9515649
##     6  77.266279  96 0.9196279
##     7  98.322899 112 0.8182994
##     8 117.417567 128 0.7384538
##     9 136.629432 144 0.6563374
##    10 149.286954 160 0.7174559
```

L'hypothèse nulle d'indépendance des résidus est conservée, ce qui nous permet de valider notre modèle.

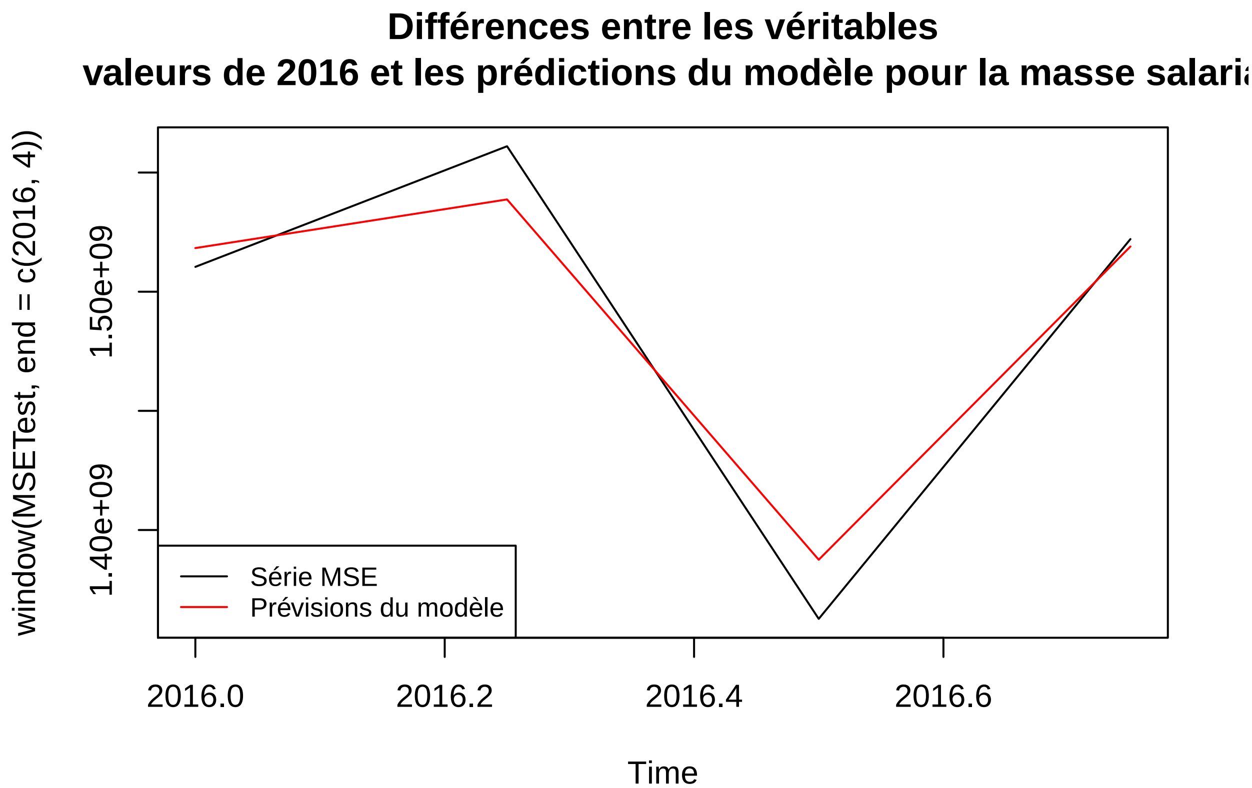
Prévisions

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 et le taux de chômage.

```
#SMIC
VARorder(cbind(MSEStaTrain, SMICStaTrain, TCHOFStaTrain), maxp=10)
modele<-VAR(cbind(MSEStaTrain, SMICStaTrain, TCHOFStaTrain), p=3)
pred <- VARpred(modele, 4)
```

```
plot(window(MSETest, end=c(2016,4)), main="Différences entre les véritables
     valeurs de 2016 et les prédictions du modèle pour la masse salariale")
#Reconstruction de la variable stationnaire
recon <- ts(pred$pred[,1], start=2016, frequency=4) * MSETrendTest * MSESeasonalTest
lines(recon, col = "red")
legend('bottomleft', legend = c('Série MSE', 'Prévisions du modèle'),
       col=c('black', 'red'), lty=1, cex=0.8)
```



Nous nous intéressons donc à l'erreur quadratique moyenne de cette prévision, qui est inférieure à celle obtenue pour le meilleur modèle effectué avec le package **vars**.

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

```
## [1] 1.977102e+14
```