

Rapport final

Paul GUILLOTTE & Jules CORBEL

01/02/2019

Contents

Introduction	3
1 Description des jeux de données	4
2 Analyse descriptive des séries	5
2.1 Rappel sur la stationnarité du second ordre	5
2.2 Masse salariale	5
2.3 PIB	7
2.4 SMIC	8
2.5 Taux de chômage des femmes	9
2.6 Calcul des corrélations	11
2.7 Découpage des séries	12
2.8 Estimation de la valeur manquante du PIB	12
3 Modélisation vectorielle	13
3.1 Définition des modèles	13
3.2 Transformation des séries	14
3.3 Corrélation entre les variables stationnarisées	20
3.4 Mise en place de modèles VAR avec le package vars	21
3.5 Mise en place de modèles VAR avec le package MTS	30
4 Construction de modèles VARMA	35
4.1 Calcul des ordres p et q	35
4.2 Estimation du modèle	36
5 Prévisions sur les années suivantes	37
Conclusion	38
A Annexe 1 : Modélisation univariée des séries	39
A.1 Comparaison des différents modèles	39
A.2 Lissage exponentiel	39
A.3 Modèles ARMA	42
A.4 Comparaison des différents modèles	44
B Annexe 2 : Modélisation ARMA avec variables exogènes	44
B.1 Définition	44
B.2 Résultats obtenus	45
C Annexe 3 : Erreurs standard associées aux coefficients du modèle VAR d'ordre 4	46

D	Annexe 4 : Résultats obtenus pour un modèle VAR d'ordre 3 pour le modèle complet avec le package MTS	47
E	Annexe 5 : Résultats obtenus pour un modèle VAR d'ordre 2 pour le modèle complet avec le package MTS	48
F	Annexe 6 : Résultats obtenus pour un modèle VARMA(3,1) pour le modèle complet	49
G	Annexe 7 : Code R du projet	52
	References	77

Introduction

Dans le cadre de la formation Génie Informatique et Statistique (GIS), l'un des modules qui nous est proposé est la réalisation d'un projet de fin d'études (PFE) en lien avec une entreprise ou un laboratoire. Notre PFE s'effectue en lien avec l'Institut des Retraites Complémentaires des Employés de Maison (IRCEM) et a pour but de construire des modèles statistiques afin de prédire la masse salariale du secteur d'activité des services à la personne pour les années à venir.

L'IRCEM a été créée en 1973. Cet organisme à but non lucratif s'occupe de la protection sociale des employés du secteur du service à la personne. Dans ce but, il doit verser des compléments de salaire aux employés à l'aide à la personne. Il effectue alors régulièrement des prévisions de la masse salariale de ce secteur d'activités, afin d'estimer l'argent qu'il devra verser.

Notre mission pour ce projet est donc de modéliser cette masse salariale et d'effectuer des prévisions pour les années 2018 et 2019, qui seront effectuées en utilisant différentes méthodes de prédiction. La première partie du projet nous a vus nous concentrer sur des méthodes de prédiction univariées, tel que le lissage exponentiel et des modélisations basées sur des processus ARMA. Nous avons également commencé à modéliser la masse salariale à l'aide des variables auxiliaires. Ici, nous présentons nos travaux sur les modèles vectoriels (modèles VAR plus des essais sur les modèles VARMA). L'intégralité du projet s'est effectuée sur le logiciel R.

1 Description des jeux de données

Les données que nous a fourni l'IRCEM sont comprises dans deux jeux de données représentant deux ensembles de variables distincts : l'un contient des variables annuelles et l'autre des variables trimestrielles, dans les deux cas à partir de 1990. Afin d'aider à la prédiction des valeurs de la masse salariale, nous devons nous appuyer sur plusieurs variables auxiliaires. Pour les données annuelles, nous disposons de 4 variables : le SMIC horaire brut, le PIB, le taux de chômage et le montant de l'Allocation pour la Garde des Enfants à Domicile. La masse salariale annuelle est connue jusqu'en 2017, et on possède les informations sur les autres variables jusqu'à 2019. En ce qui concerne le modèle trimestriel, nous disposons de 3 variables : le SMIC, le PIB et le taux de chômage des femmes. La masse salariale trimestrielle est connue jusqu'au 2e trimestre de 2017. Le PIB trimestriel est lui connu jusqu'au 1er trimestre 2017. Pour les deux autres variables, les informations que nous possédons vont jusqu'au dernier trimestre de 2017. Le faible nombre de données (surtout pour les variables annuelles) a cependant été un obstacle. En effet, dans le jeu annuel, il n'y a 28 années, ce qui représente peu de valeurs pour créer des modèles pertinents. Nous nous sommes donc concentrés uniquement sur le jeu de données trimestriel.

2 Analyse descriptive des séries

2.1 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 sont les suivantes :

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

$$Var[y_t] = \sigma^2 \neq \infty \forall t = 1 \dots t$$

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

Nous nous intéressons dans ce rapport aux différentes séries trimestrielles à notre disposition. Dans un premier temps, nous visualiserons chacune de séries. Nous vérifierons ensuite les corrélations entre les variables deux à deux afin de nous faire une idée du lien qu'il existe entre les variables.

2.2 Masse salariale

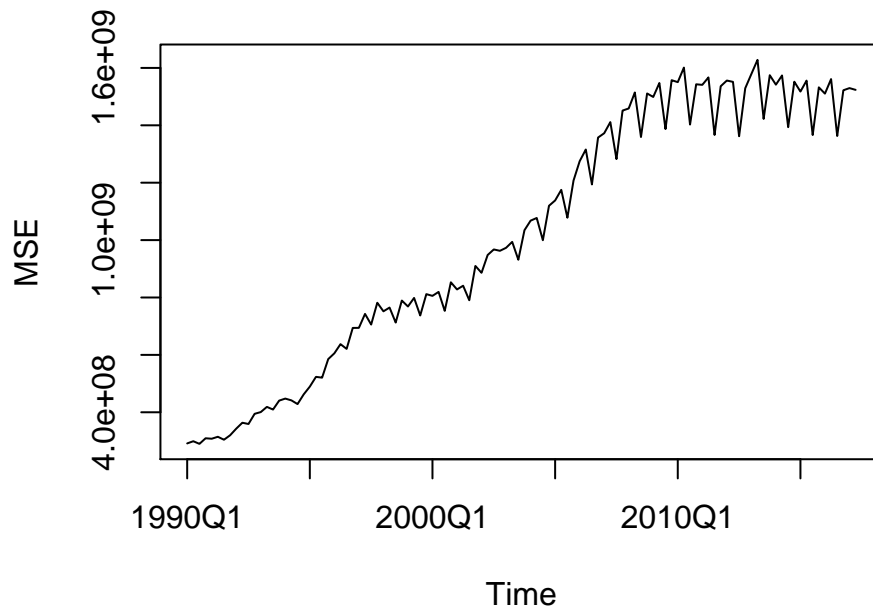


Figure 1: Evolution trimestrielle de la masse salariale

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.

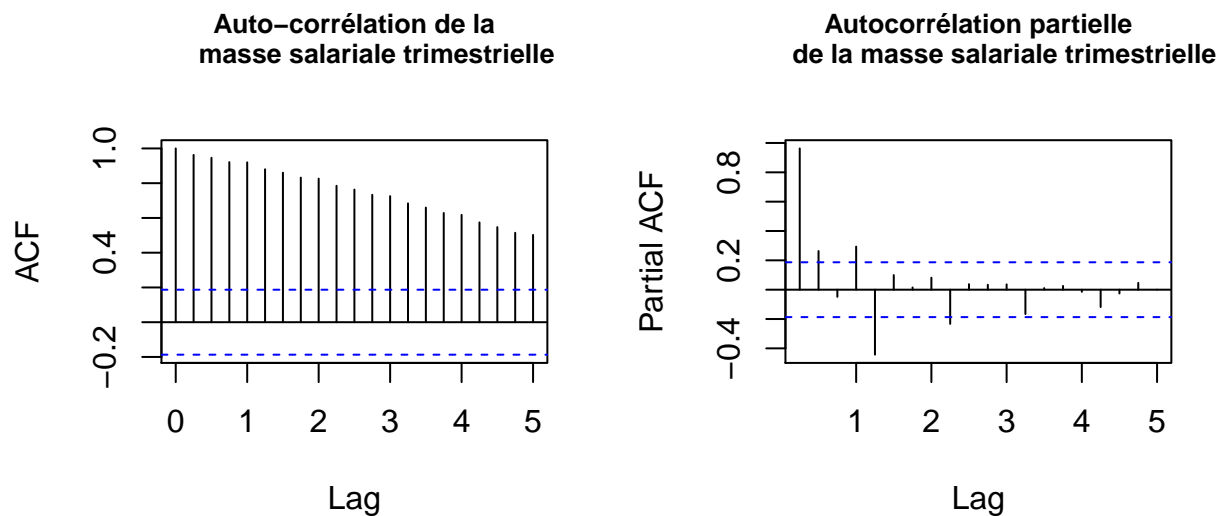


Figure 2: Fonctions d'autocorrélation de la masse salariale trimestrielle

```
## 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
## 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
```

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 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 première espèce à 5%, on conserve l'hypothèse nulle et on

conclut, à l'aide des deux tests effectués, que la série n'est pas stationnaire.

2.3 PIB

La Figure 3 nous montre l'évolution trimestrielle du PIB qui, comme pour la masse salariale possède une tendance. Cependant, elle 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.

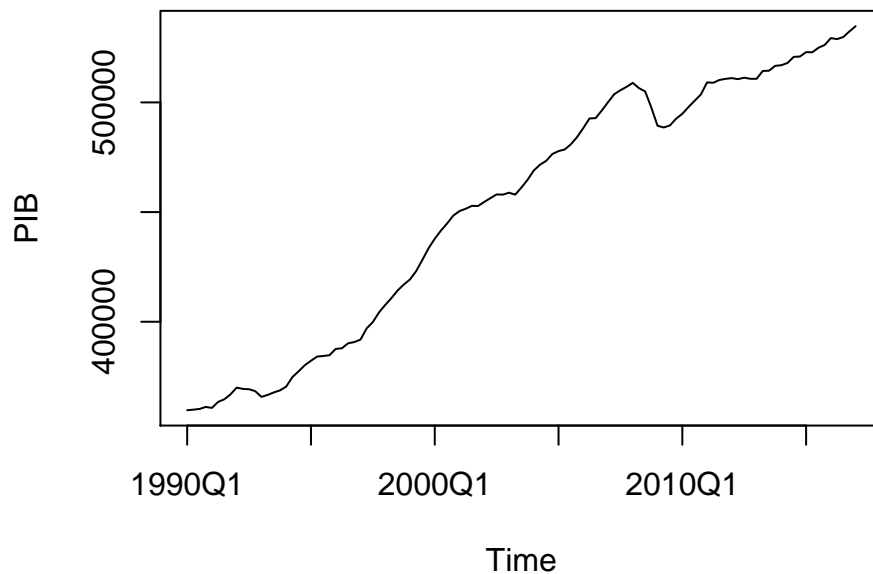


Figure 3: Evolution trimestrielle du PIB

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

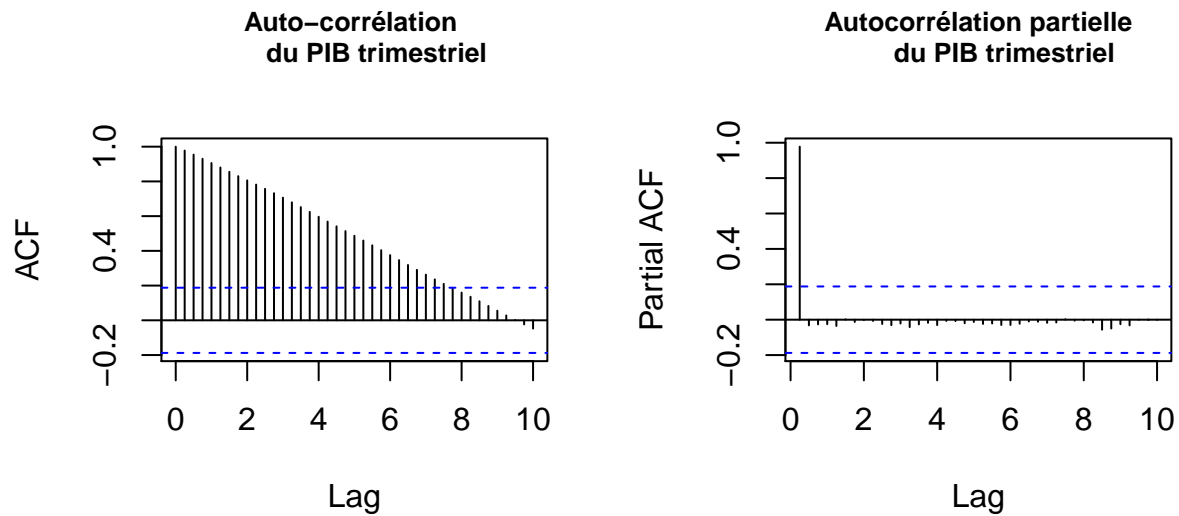


Figure 4: ACF et PACF du PIB trimestriel

2.4 SMIC

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.

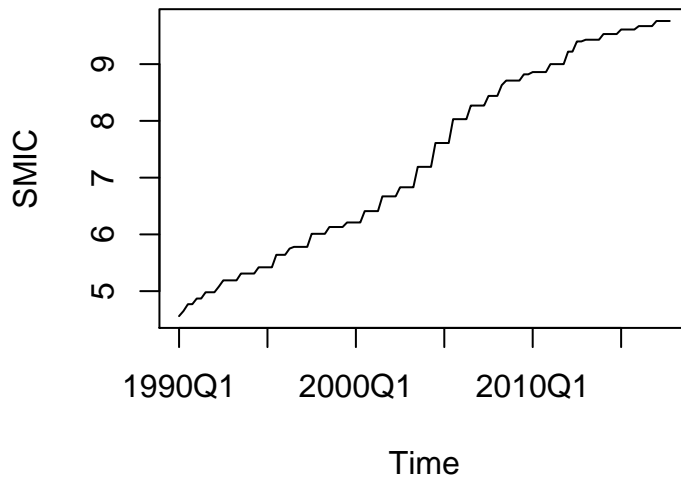


Figure 5: Evolution trimestrielle du SMIC

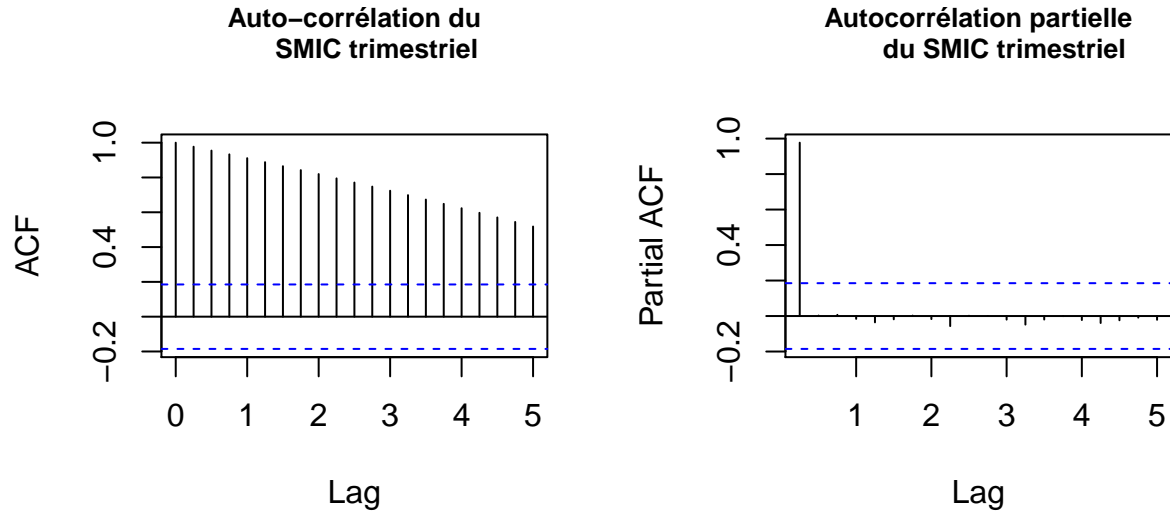


Figure 6: ACF et PACF du SMIC trimestriel

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

2.5 Taux de chômage des femmes

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 risque de première espèce de 5%.

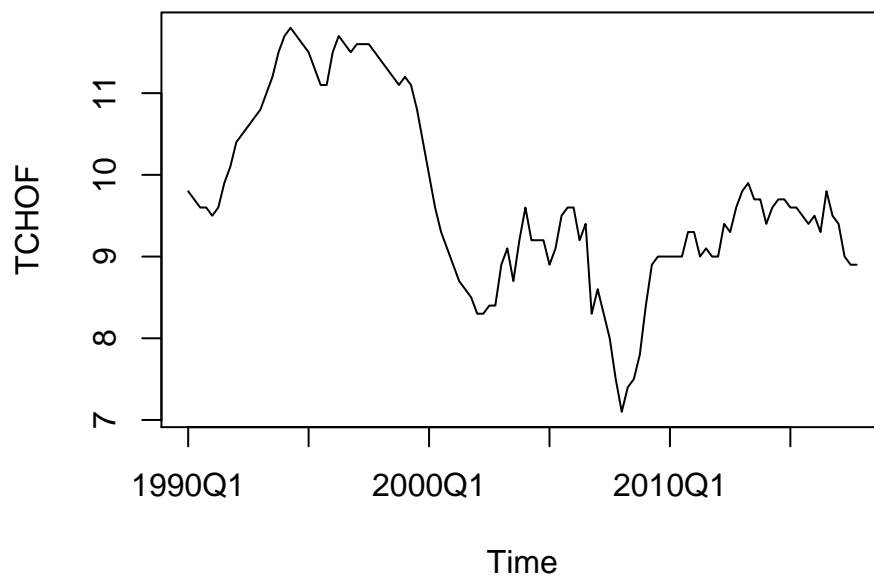


Figure 7: Evolution trimestrielle du taux de chômage des femmes

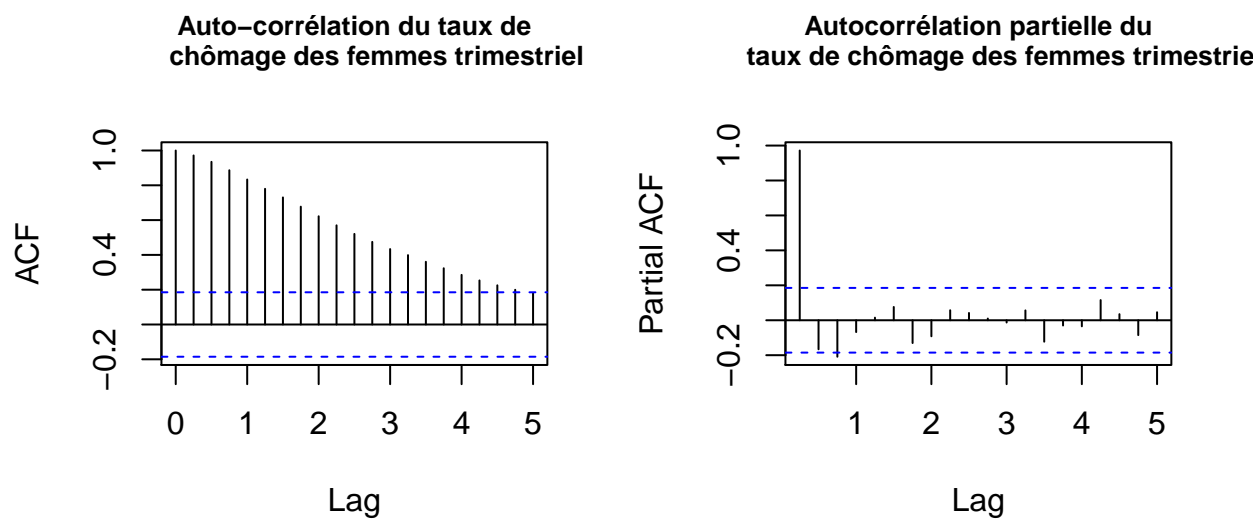


Figure 8: ACF et PACF du taux de chômage des femmes trimestriel

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

2.6 Calcul des corrélations

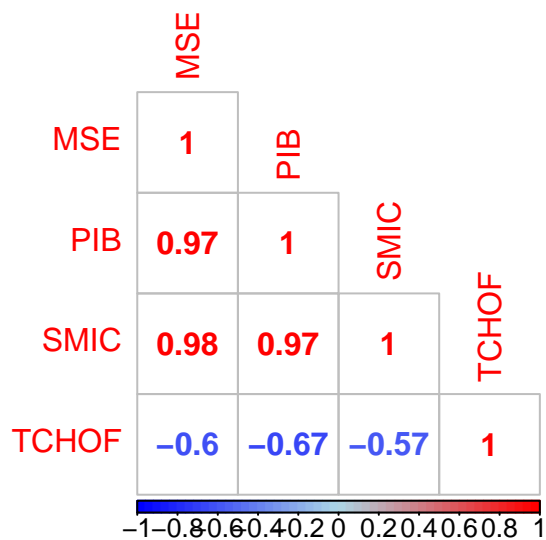


Figure 9: Corrélations entre les variables trimestrielles

```
##
## 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. Les variables PIB, masse salariale et SMIC sont extrêmement liées entre elles. En regardant le tableau des p-values associées au test de Student (H_0 : La corrélation entre les deux variables est nulle), on s'aperçoit que toutes les variables prises deux à deux présentes une corrélation.

2.7 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.

2.8 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 un modèle SARIMA correspondant à la variable PIB, étant donné que c'est celui qui donnait les meilleures prédictions (voir A).

3 Modélisation vectorielle

Dans la première partie du projet, nous avons modélisé les différentes séries séparément, à l'aide de modèles de lissage exponentiel ou des processus ARMA, dont les résultats sont présents en Annexe A. Nous avons ensuite effectué d'autres modèles ARMA sur la masse salariale en utilisant les autres variables pour l'expliquer. Ces modèles sont présents en Annexe B. Cependant, ce type de modélisation ne nous donne pas de résultats plus performants en terme de prédictions.

Pendant la deuxième partie du projet, nous nous sommes donc attachés à utiliser d'autres méthodes de modélisation. Nous nous sommes concentrés sur les modèles de type vectoriels, qui permettent donc de prédire plusieurs séries temporelles simultanément. La plus grande partie de nos travaux portent sur des modèles Vector Auto-Regressive (VAR) plus quelques tests avec l'ajout d'une partie Moving Average (MA) ce qui nous donne des modèles VARMA.

3.1 Définition des modèles

3.1.1 Ecriture

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.

3.1.2 Hypothèses

3.1.2.1 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 du 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.

3.1.2.2 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

3.2 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.

3.2.1 Masse salariale

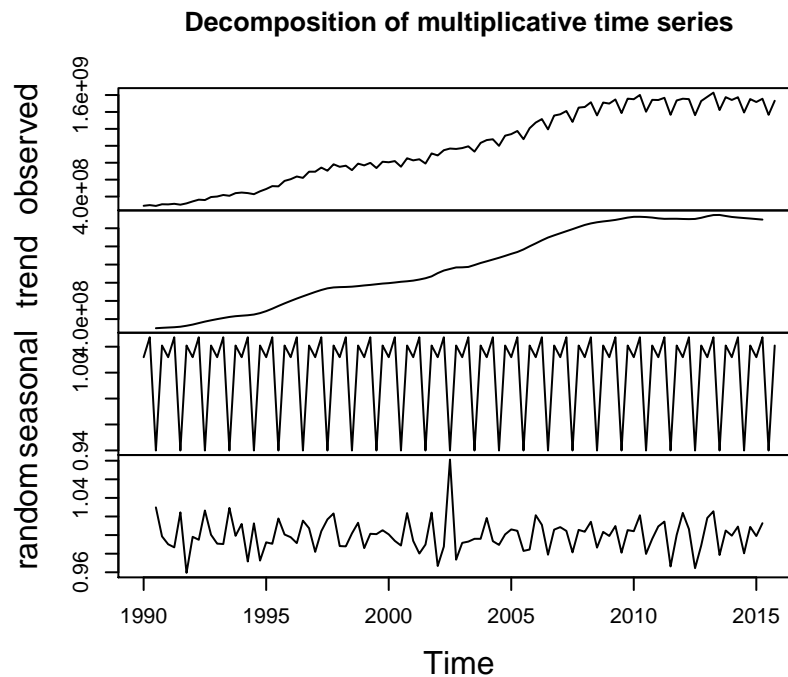


Figure 10: Décomposition de la masse salariale

Nous nous intéressons aux ACF, PACF (visibles en Figure 11) 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).

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

```
##
```

```
## KPSS Test for Level Stationarity
```

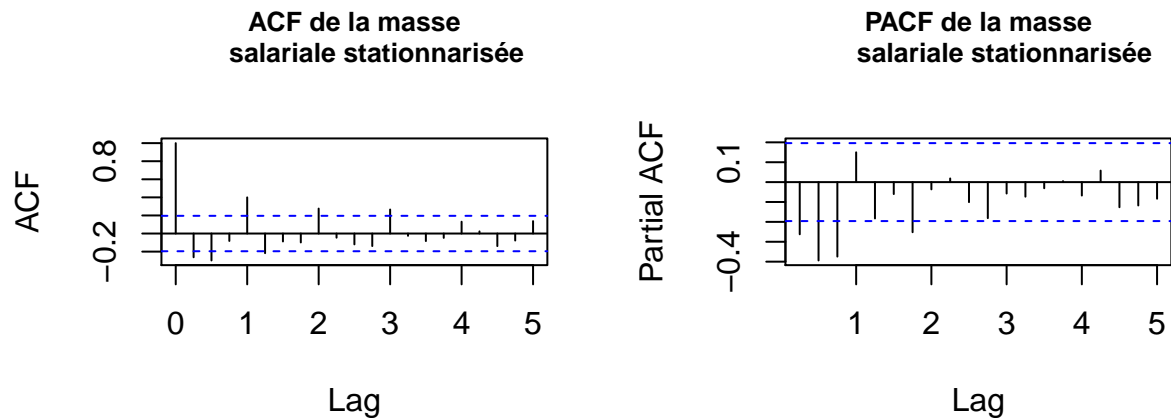


Figure 11: Fonctions d'autocorrélation de la masse salariale stationnarisée

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

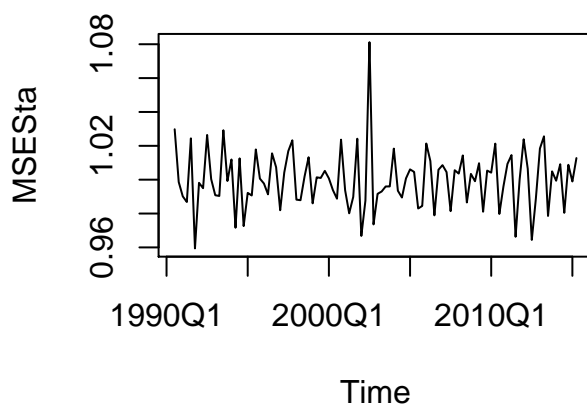


Figure 12: Masse salariale trimestrielle stationnarisée

3.2.2 PIB

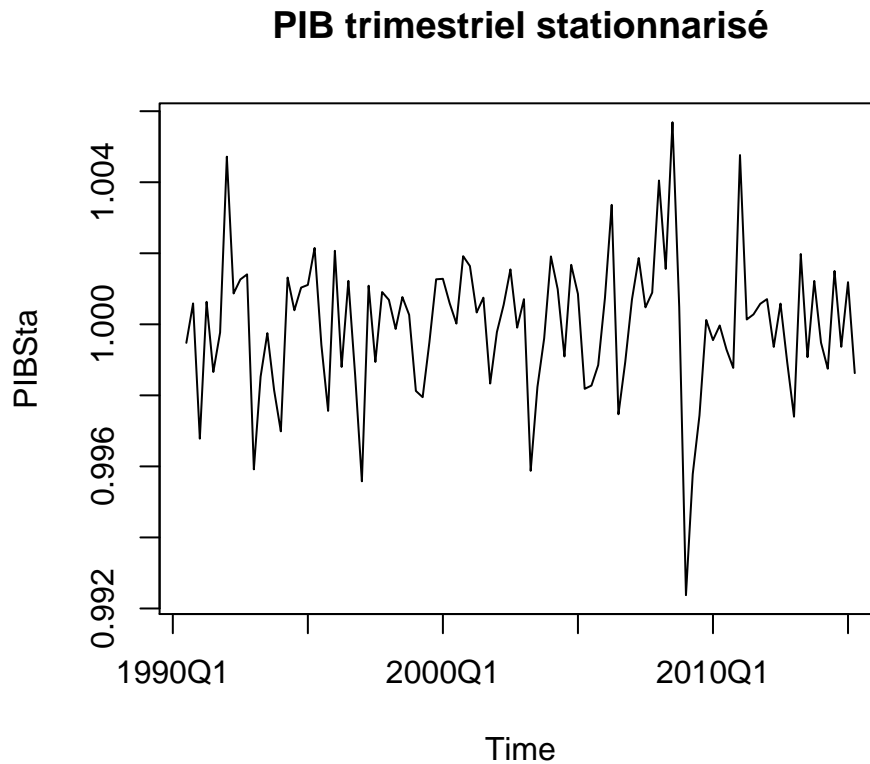


Figure 13: PIB trimestriel stationnarisé

```
## 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
## 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 (Figure 12) 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.

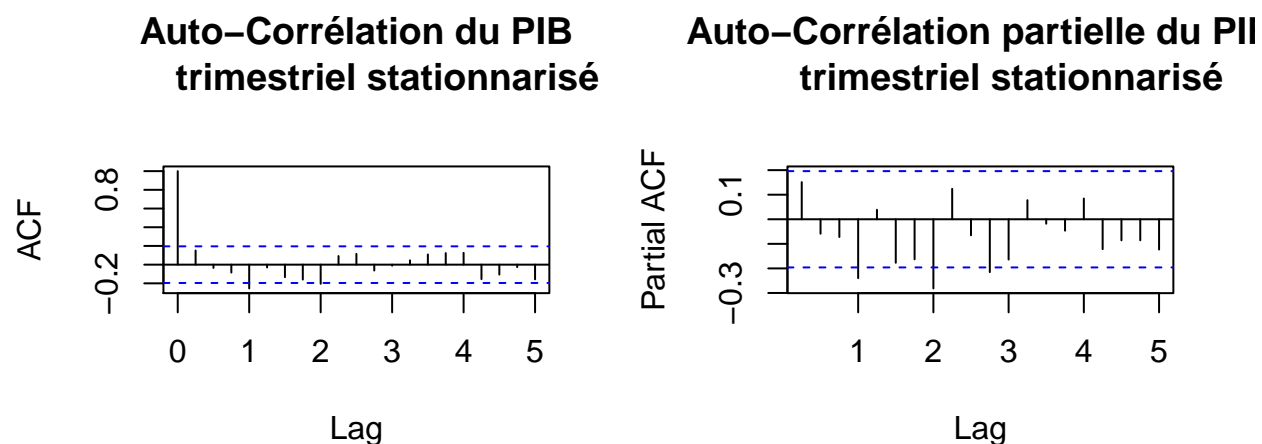


Figure 14: Fonctions d'autocorrélation du PIB stationnarisé

3.2.3 SMIC

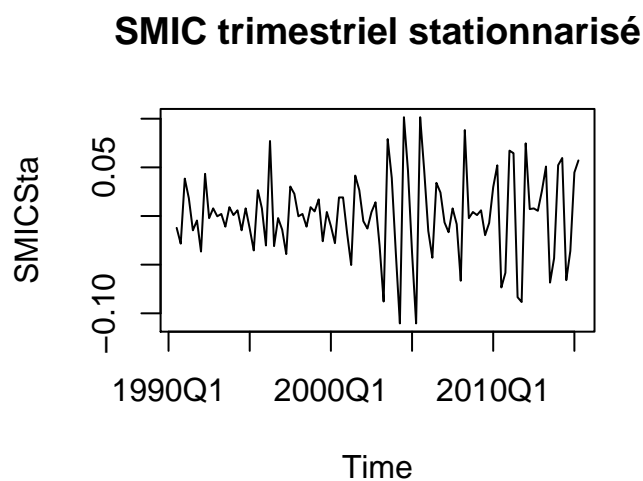


Figure 15: SMIC trimestriel stationnarisé

Comme pour la masse salariale, les ACF et PACF de la Figure 16 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.

```
## Warning in kpss.test(SMICSta): p-value greater than printed p-value
##
## KPSS Test for Level Stationarity
##
## data: SMICSta
```

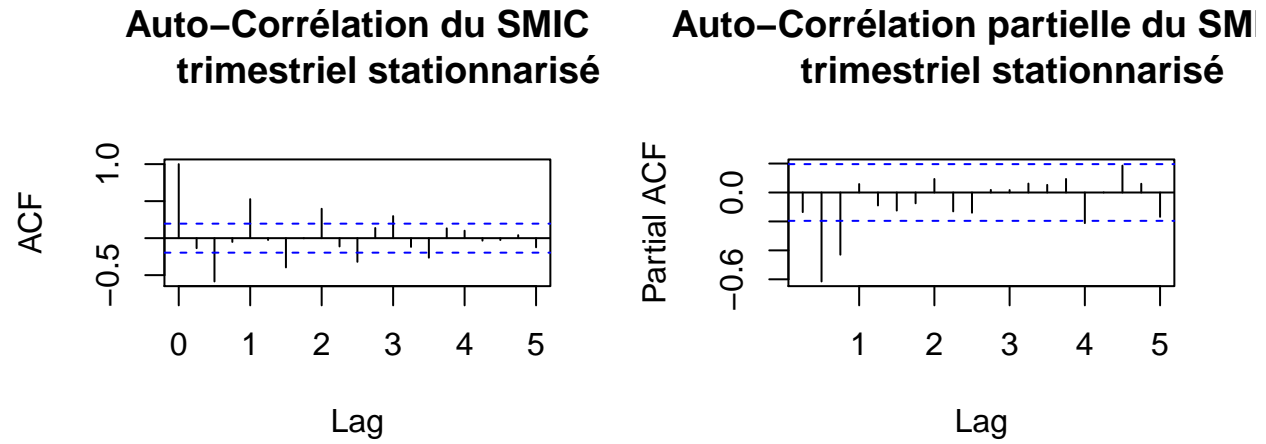


Figure 16: Fonctions d'autocorrélation du SMIC stationnarisé

```
## KPSS Level = 0.043771, Truncation lag parameter = 2, p-value = 0.1
## 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
```

3.2.4 Taux de chômage des femmes

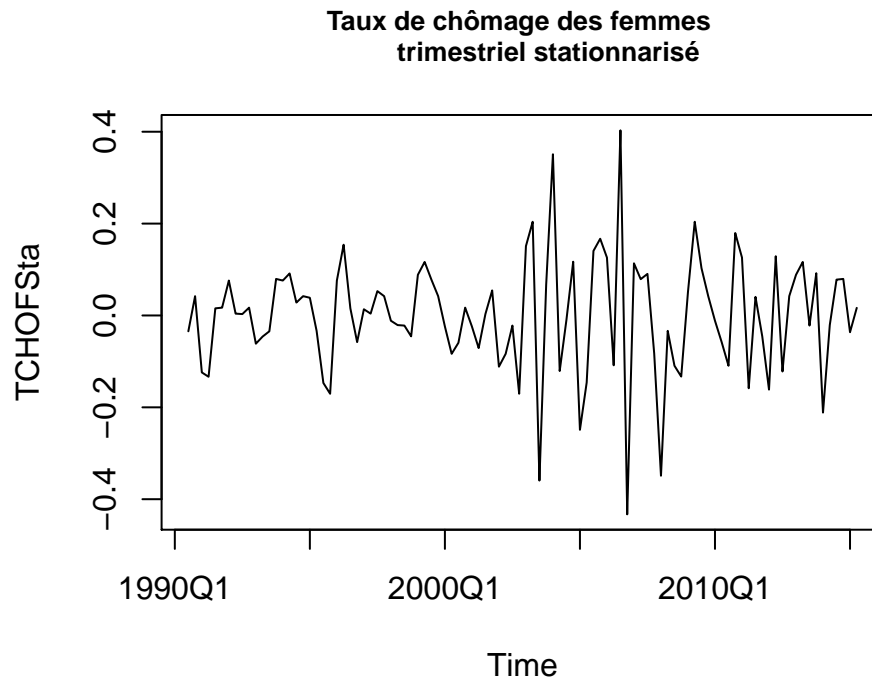


Figure 17: Taux de chômage trimestriel stationnarisé

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 présents en Figure 18, on peut conclure que la série résiduelle est stationnaire.

```
## 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
## 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
```

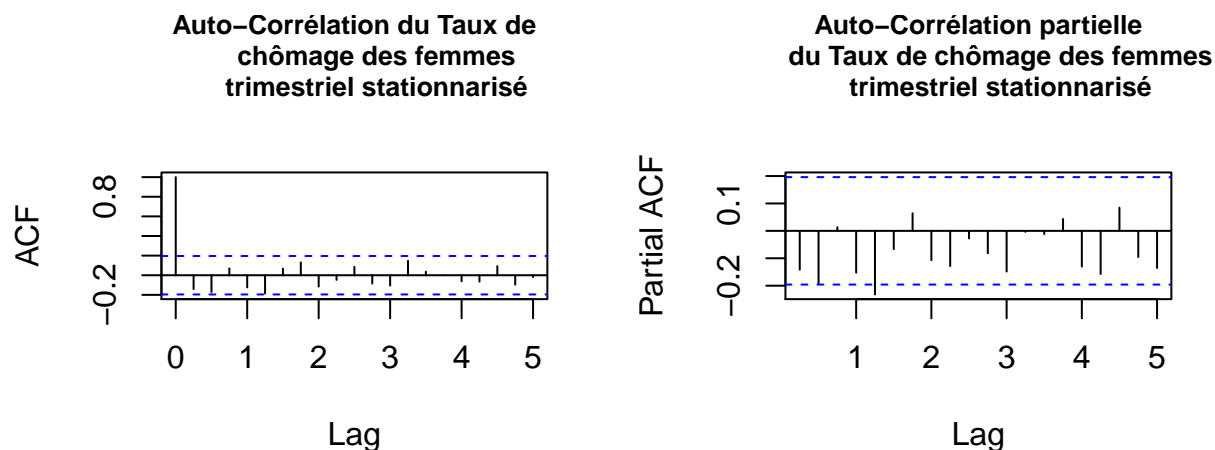


Figure 18: Fonctions d'autocorrélation du taux de chômage stationnarisé

3.3 Corrélation entre les variables stationnarisées

Corrélations entre les variables trimestrie

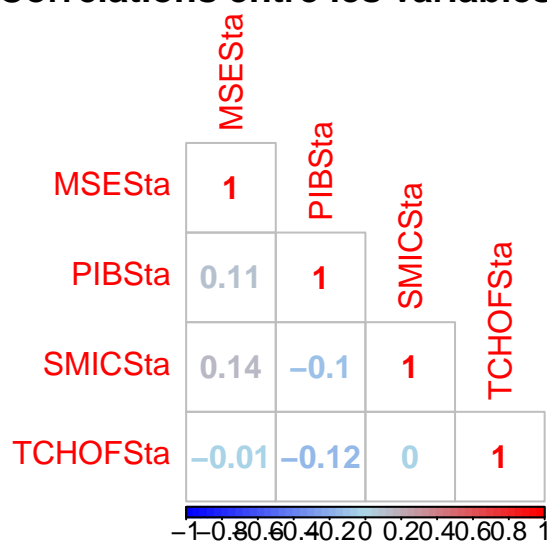


Figure 19: Corrélations entre les variables trimestrielles stationnarisées

```
##           MSE           PIB           SMIC           TCHOFSa
## 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
## TCHOFSa 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, la matrice des corrélations présentes en Figure 19 nous montre que la corrélation la plus élevée vaut 0.14 ce qui reste très faible. De plus, en regardant le tableau des p-values, l'hypothèse nulle de non significativité du coefficient de corrélation n'est rejetée pour aucun couple de variables

(avec un seuil de 5%).

Maintenant que toutes les séries ont été stationnarisées, elles peuvent être utilisées pour construire un modèle VAR.

3.4 Mise en place de modèles VAR avec le package vars

Le package **vars** a été construit par Bernhard Pfaff (Pfaff 2018). Il permet de construire des modèles vectoriels (VAR), et dispose également des différentes fonctions de diagnostics permettant de vérifier que les hypothèses du modèle sont bien remplies. La version utilisée est la version 1.5-3 et date du 6 Août 2018.

3.4.1 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 t . Dans le package **vars**, la fonction *VARselect* permet de déterminer l'ordre des modèles VAR à sélectionner 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 $\tilde{\Sigma}_u(p) = \frac{1}{T} \sum_{t=1}^T \hat{u}_t \hat{u}_t'$ (la matrice de covariance des résidus du modèle).

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.

Le critère AIC (Akaike information criterion) se calcule, dans ce package, de la manière suivante : $AIC(p) = \ln \det(\tilde{\Sigma}_u(p)) + \frac{2}{T} p K^2$. 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 efficace : 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 : $HQ(p) = \ln \det(\tilde{\Sigma}_u(p)) + \frac{2 \ln(\ln(T))}{T} p K^2$. 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 : $SC(p) = \ln \det(\tilde{\Sigma}_u(p)) + \frac{\ln(T)}{T} p K^2$. L'objectif est de minimiser ce critère. Ce critère est un autre nom du BIC.

On s'aperçoit que les différents critères à notre disposition, visibles sur les graphiques de la Figure 20, 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 définis par les différents critères : ici, nous allons donc nous intéresser aux modèles d'ordre 2, 3 et 4.

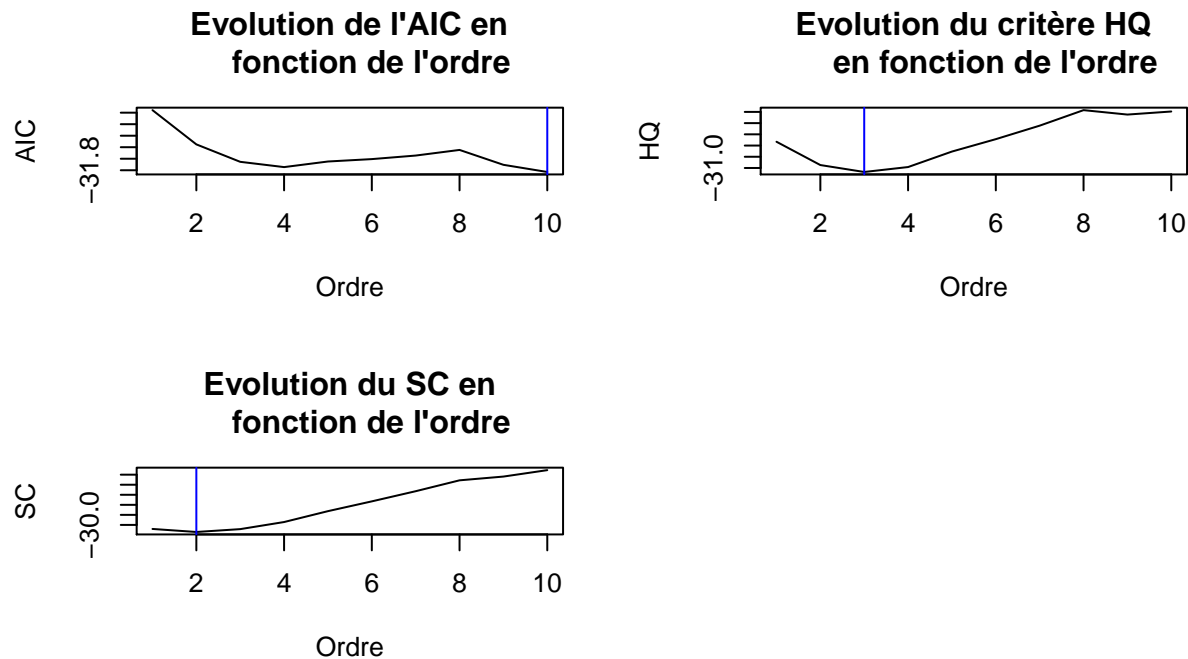


Figure 20: Critères associés au modèle complet

3.4.2 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).

3.4.2.1 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.

Les coefficients du modèle sont les suivants. Les erreurs standards associées sont présentes en Annexe C.

##	MSE	PIB	SMIC	TCHOF
## MSE	-0.49571073	0.8961454	0.066963585	-0.0127991267

```

## PIB    -0.03134947  0.1217021 -0.005125468  0.0005912438
## SMIC    0.05931412  1.1813374 -0.496926461 -0.0096012159
## TCHOF   -0.88275343 -5.5627534  0.660087437 -0.3220844544

##          MSE          PIB          SMIC          TCHOF
## MSE    -0.44837936 -0.85710828  0.059065357 -0.016289594
## PIB     -0.03290389 -0.03056789 -0.006426264 -0.001447922
## SMIC    -0.12186822 -0.59050790 -0.674266226 -0.004993847
## TCHOF    0.40185192 -17.96140240  0.821446464 -0.310004540

##          MSE          PIB          SMIC          TCHOF
## MSE    -0.27254709 -0.67990488  0.04532636  0.0043284001
## PIB     -0.02833629 -0.06107288 -0.01176086 -0.0004898846
## SMIC    -0.27835989 -0.37298995 -0.41721629 -0.0094899806
## TCHOF    0.25697997 -16.67936660  0.63877023 -0.1024043275

##          MSE          PIB          SMIC          TCHOF
## MSE     0.19486895 -0.4557750  0.09080679 -0.001506479
## PIB     -0.02602058 -0.2762433 -0.00804238  0.001727524
## SMIC     0.14617955  1.3033993  0.08614964 -0.022040339
## TCHOF   -1.24236337 -2.7851730  1.22452627 -0.152158590

```

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

```

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

```

Enfin, l'erreur quadratique moyenne de ce modèle pour les données prédites est la suivante :

```
## [1] 7.711856e+14
```

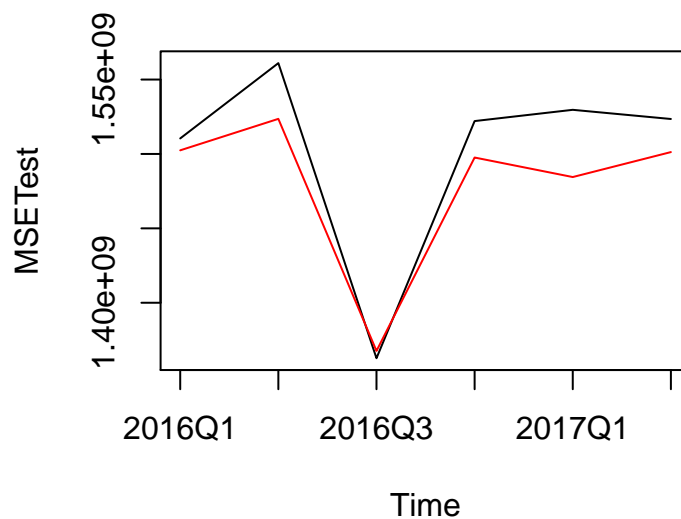


Figure 21: Comparaison entre les vraies valeur et le modèle complet pour un ordre 4

La représentation graphique de la Figure 21 nous montre des prédictions proches des vraies valeurs, à part pour le premier trimestre 2017.

3.4.2.2 Ordre 3

On s'intéresse ensuite au modèle d'ordre 3, soit celui avec le meilleur critère HQ.

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

```
##          MSE          PIB          SMIC          TCHOF
## MSE    -0.55534062  0.5875606  0.0078931593 -0.007157107
## PIB    -0.01712717  0.1810606 -0.0001595774  0.001149490
## SMIC   -0.01904054  0.4355559 -0.5342311296 -0.015680593
## TCHOF  -0.40492349 -6.7105659  0.1543612225 -0.278488213
```

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

```
##          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 :

```
## [1] 9.682886e+14
```

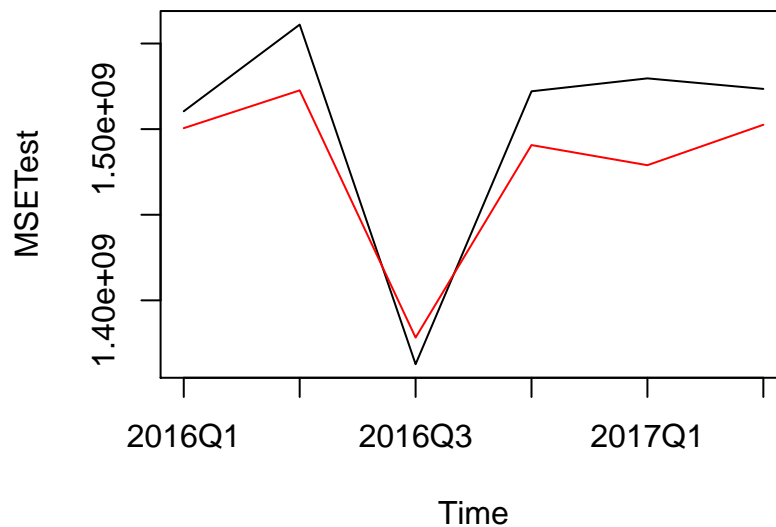


Figure 22: Comparaison entre les vraies valeur et le modèle complet pour un ordre 3

Graphiquement, la Figure 22 associée à l'ordre 3 nous donne des prédictions semblant inférieures à celles du modèle d'ordre 4. L'apport d'un lag supplémentaire dans la construction du retard a donc bien une importance dans la qualité des prédictions.

3.4.2.3 Ordre 2

Enfin, nous construisons le modèle avec le meilleur BIC, soit celui d'ordre 2.

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

##	MSE	PIB	SMIC	TCHOF
## MSE	-0.39572755	0.8295593	0.030196701	-0.009163581
## PIB	-0.01295909	0.1873301	0.006086945	0.001036053
## SMIC	0.12910889	0.9413863	-0.239926026	-0.022141074
## TCHOF	-0.51603938	-6.5905696	0.103373369	-0.217258398

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

##	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 :

```
## [1] 9.987686e+14
```

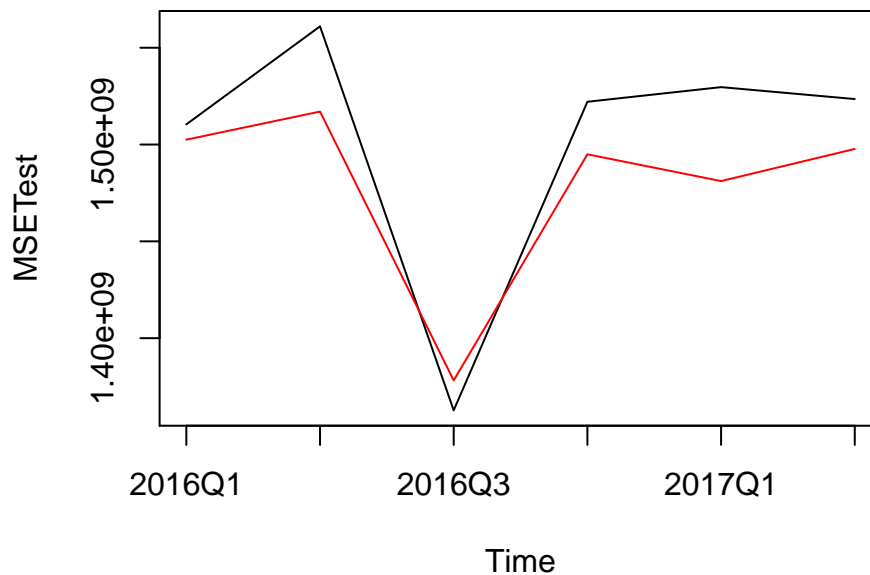


Figure 23: Comparaison entre les vraies valeur et le modèle complet pour un ordre 2

Lorsque l'on compare les erreurs quadratiques moyennes, nous remarquons que la plus faible est celle associée à un modèle d'ordre 4, soit celui avec le meilleur AIC. Il nous faut maintenant vérifier que les hypothèses associées à ce modèle soient bien vérifiées.

3.4.3 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.

```
## [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
```

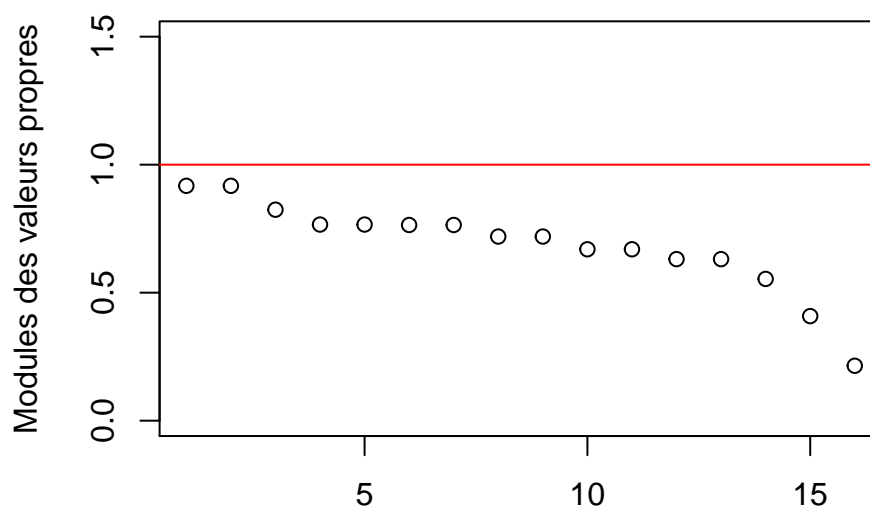


Figure 24: Valeurs propres de la matrice A

On s'aperçoit que tous les modules sont inférieurs à 1 sur le graphique de la Figure 24, le processus VAR(4) est donc stable.

3.4.4 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 : $vech(\hat{u}_t\hat{u}_t^T) = \beta_0 + B_1vech(\hat{u}_{t-1}\hat{u}_{t-1}^T) + \dots + B_qvech(\hat{u}_{t-q}\hat{u}_{t-q}^T) + v_t$. La dimension de β_O est $\frac{1}{2}K(K+1)$ et celle des matrices des coefficients B_i est $\frac{1}{2}K(K+1) \times \frac{1}{2}K(K+1)$. 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 $H_0 : B_0 = B_1 = \dots = B_q = 0$ (homoscédasticité).

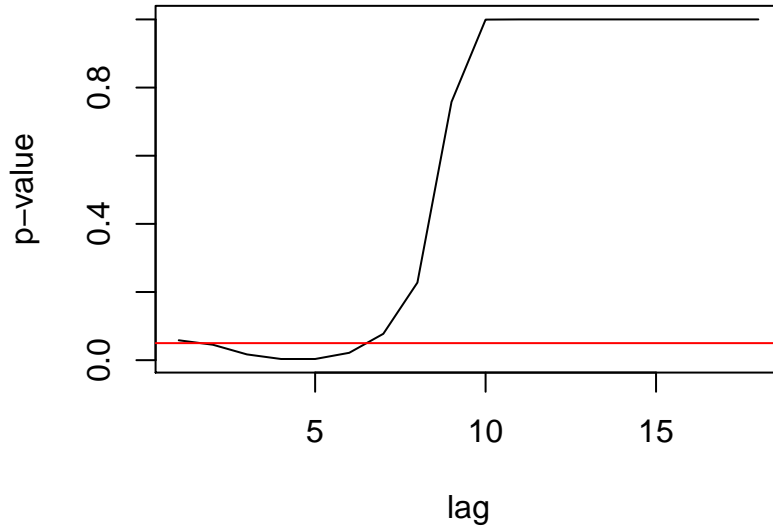


Figure 25: Evolution de la p-value en fonction du lag

On s'aperçoit au regard de la figure 25, 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.

3.4.5 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 : $s_3^2 = Tb_1^Tb_1/6$ et $s_4^2 = T(b_2 - 3K)^T(b_2 - 3K)/24$, 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 $H_0 : S = 0$ et $K = 3$.

```
##
## 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 H_0 , avec un seuil de confiance de 5%. Les résidus obtenus ne suivent pas une loi normale. Cependant ce test a tendance à rejeter facilement les jeux de données ne possédant que peu de données, ce qui est le cas ici. Les résultats de ce test ne sont donc pas forcément crédibles dans ce contexte.

3.4.6 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.

3.4.6.1 Verification de la Matrice C_0

La statistique de Portmanteau est $Q_h = T \sum_{j=1}^h \text{tr}(\hat{C}_j^T \hat{C}_0^{-1} \hat{C}_j \hat{C}_0^{-1})$, 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. $\text{tr}()$ correspond à la trace de la matrice, soit la somme des éléments diagonaux de la matrice.

```
##           MSESta       PIBSta       SMICSta       TCHOFSa
## [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

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

Même si le déterminant est très faible, il n'est pas nul. La matrice \hat{C}_0 est donc inversible.

3.4.6.2 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.

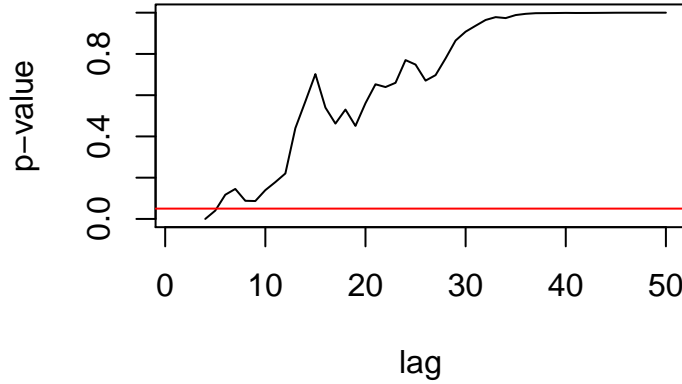


Figure 26: Evolution de la p-value en fonction du lag

Au regard de la figure 26 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.

3.4.7 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 complet, soit celui que nous avons pris en exemple. Les AIC et EQM des différents modèles sont résumés dans le tableau ci-dessous.

Variabiles utilisées	Ordre	AIC	EQM
MSE & PIB	4	-20.85328	9.153141e+14
MSE & SMIC	3	-15.37233	9.553417e+14
MSE & TCHO	3	-12.44526	9.880360e+14
MSE, PIB & SMIC	3	-27.65284	9.564856e+14
MSE, PIB & TCHO	4	-24.98222	9.020987e+14
MSE, SMIC & TCHO	4	-19.20940	7.955221e+14
MSE, PIB, SMIC & TCHO	4	-31.74520	7.711856e+14

3.5 Mise en place de modèles VAR avec le package MTS

Le package **MTS** a été créé par **Ruey S. Tsay** and **David Wood**, professeurs à l'université de Chicago (Tsay and Wood 2002). Contrairement au package **vars**, il ne se limite pas aux modèles VAR et propose également d'autres types de modèles multivariés, tels que des modèles de type VMA (Vector Moving Average), VARMA et même quelques modèles de fonctions de transfert, que nous n'avons pas eu l'occasion d'étudier. Ce package est très récent, sa première version datant du 8 octobre 2018.

Pour nous aider à comprendre l'utilisation de ce package, nous nous sommes beaucoup inspirés de l'article *Multivariate Time Series Analysis with R and Financial Applications* de **Ruey S. Tsay**

3.5.1 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. 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.

Cependant, nous avons également en notre possession un autre critère, la statistique du test de Tiao-Box ainsi que sa p -value. Ce test évalue pour un ordre i la significativité des coefficients de la matrice A_i . Ce test permet donc de déterminer si le modèle d'ordre i est meilleur que celui d'ordre $i-1$. La statistique de test est la suivante : $M(i) = -(T - K - i - \frac{3}{2}) \ln(\frac{\det(\hat{\Sigma}_i)}{\det(\hat{\Sigma}_{i-1})})$, qui suit une loi du $\chi^2_{k^2}$. Dans l'exemple que nous prenons, la première p -value supérieure à 0.05 est celle pour la matrice A_5 . L'ordre que nous devons retenir par rapport à ce test est donc 4.

```
## selected order: aic = 10
## selected order: bic = 0
## selected order: hq = 3
## Summary table:
##      p      AIC      BIC      HQ      M(p) p-value
## [1,] 0 -30.9791 -30.9791 -30.9791  0.0000 0.0000
## [2,] 1 -30.8805 -30.4637 -30.7118 18.7080 0.2841
## [3,] 2 -31.5108 -30.6772 -31.1734 76.5005 0.0000
## [4,] 3 -31.8493 -30.5988 -31.3432 50.3713 0.0000
## [5,] 4 -31.9763 -30.3090 -31.3015 32.4105 0.0088
## [6,] 5 -31.9147 -29.8306 -31.0713 17.7023 0.3416
## [7,] 6 -31.9083 -29.4074 -30.8962 20.2276 0.2101
## [8,] 7 -31.8826 -28.9648 -30.7017 17.8005 0.3357
## [9,] 8 -31.8205 -28.4859 -30.4709 14.5715 0.5562
## [10,] 9 -32.1143 -28.3628 -30.5960 32.2245 0.0093
## [11,] 10 -32.2744 -28.1061 -30.5874 23.2838 0.1064
```

Comme pour le package **VARs**, l'AIC associé aux différents modèles diminue en même temps que l'ordre augmente. Si l'on conserve un ordre raisonnable, le meilleur AIC est également pour un modèle d'ordre 4. Le BIC nous donne lui un modèle d'ordre 2 alors que le critère HQ nous conseille d'utiliser un modèle d'ordre 3. Cela correspond aux mêmes ordres que ceux donnés par le package **vars**.

3.5.2 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 d'ordre 4, soit les coefficients du modèle pour chaque ordre de retard, les erreurs standard ainsi que les valeurs des critères de qualité.

```
## Constant term:
## Estimates:  3.117817  1.364744 -1.325939  44.45549
## Std.Error:  1.664379  0.2175532  3.507354  12.73883
## AR coefficient matrix
## AR( 1 )-matrix
##      [,1] [,2] [,3] [,4]
## [1,] -0.4957  0.896  0.06696 -0.012799
## [2,] -0.0313  0.122 -0.00513  0.000591
## [3,]  0.0593  1.181 -0.49693 -0.009601
## [4,] -0.8828 -5.563  0.66009 -0.322084
## standard error
##      [,1] [,2] [,3] [,4]
## [1,] 0.1085 0.821 0.05362 0.01345
## [2,] 0.0142 0.107 0.00701 0.00176
## [3,] 0.2287 1.731 0.11300 0.02834
## [4,] 0.8307 6.288 0.41043 0.10293
## AR( 2 )-matrix
##      [,1] [,2] [,3] [,4]
## [1,] -0.4484 -0.8571  0.05907 -0.01629
## [2,] -0.0329 -0.0306 -0.00643 -0.00145
## [3,] -0.1219 -0.5905 -0.67427 -0.00499
## [4,]  0.4019 -17.9614  0.82145 -0.31000
## standard error
##      [,1] [,2] [,3] [,4]
## [1,] 0.1180 0.809 0.05463 0.01344
## [2,] 0.0154 0.106 0.00714 0.00176
## [3,] 0.2487 1.704 0.11512 0.02833
## [4,] 0.9034 6.189 0.41811 0.10289
## AR( 3 )-matrix
##      [,1] [,2] [,3] [,4]
## [1,] -0.2725 -0.6799  0.0453  0.00433
## [2,] -0.0283 -0.0611 -0.0118 -0.00049
## [3,] -0.2784 -0.3730 -0.4172 -0.00949
## [4,]  0.2570 -16.6794  0.6388 -0.10240
## standard error
##      [,1] [,2] [,3] [,4]
## [1,] 0.1183 0.841 0.05525 0.0130
## [2,] 0.0155 0.110 0.00722 0.0017
## [3,] 0.2493 1.773 0.11643 0.0275
## [4,] 0.9054 6.440 0.42288 0.0998
## AR( 4 )-matrix
```

```

##          [,1]    [,2]    [,3]    [,4]
## [1,]  0.195 -0.456  0.09081 -0.00151
## [2,] -0.026 -0.276 -0.00804  0.00173
## [3,]  0.146  1.303  0.08615 -0.02204
## [4,] -1.242 -2.785  1.22453 -0.15216
## standard error
##          [,1]    [,2]    [,3]    [,4]
## [1,]  0.1099  0.866  0.05397  0.01239
## [2,]  0.0144  0.113  0.00705  0.00162
## [3,]  0.2315  1.826  0.11372  0.02612
## [4,]  0.8409  6.631  0.41304  0.09486
##
## Residuals cov-mtx:
##          [,1]    [,2]    [,3]    [,4]
## [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
##
## det(SSE) =  4.173756e-15
## AIC =  -31.82996
## BIC =  -30.16265
## HQ  =  -31.15517

```

Les résultats associés aux modèles d'ordre 3 et 2 sont eux présents en Annexe D et en Annexe E. Comme pour le package **vars**, il nous faut maintenant vérifier les hypothèses du modèle.

3.5.3 Vérification des hypothèses

3.5.3.1 Vérification de la stabilité

Comme pour le package **vars**, nous vérifions que les modules des valeurs propres de la matrice A sont tous inférieurs à 1. R. Tsay définit cependant la matrice A différemment de B. Pfaff, soit en la retournant

$$A = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & I \\ A_p & A_{p-1} & A_{p-2} & \cdots & A_1 \end{bmatrix}$$

```

## [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

```

La Figure 27 nous montre des valeurs propres toutes inférieures à 1 en module ce qui nous permet de prouver la stabilité du modèle.

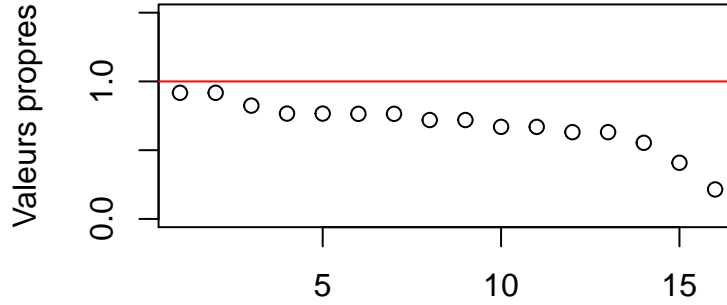


Figure 27: Valeurs propres associées à la matrice A

3.5.3.2 Autocorrélation et corrélation croisée

Nous vérifions ensuite que les résidus ne comportent ni d'autocorrélation, ni de corrélation croisée. Pour cela, on utilise la fonction *ccm*, qui vérifie les matrices de corrélation croisée pour un lag donné. On définit la matrice de variance-covariance pour un lag p comme $\hat{\Gamma}_p = \frac{1}{T} \sum_{i=p+1}^T (r_t - \bar{r})(r_{t-p} - \bar{r})$. La matrice de corrélation associée vaut donc $\rho_p = \hat{D}^{-1} \hat{\Gamma}_p \hat{D}^{-1}$.

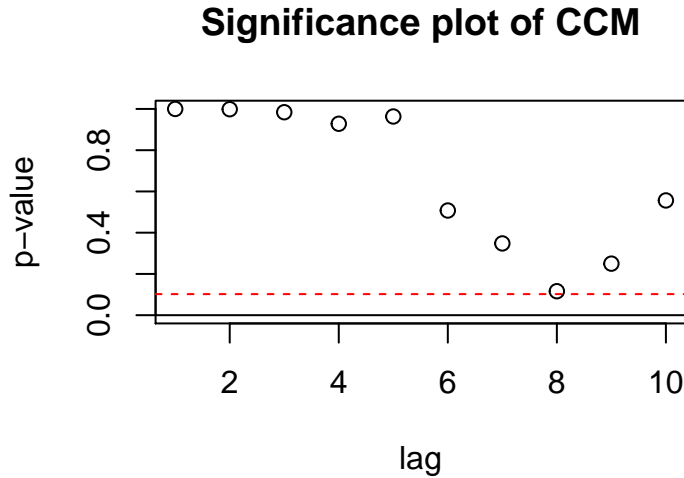


Figure 28: Significativité des corrélations croisées pour un lag de 1 à 10

La Figure 28 représente le résultat du test d'égalité de la matrice $\hat{\Gamma}_p$ à 0. On considère que la corrélation est significative si elle dépasse le seuil de $2/\sqrt{T}$ avec T la longueur de la série. On ne rejette pas l'hypothèse nulle peu importe le lag, ce qui nous indique que les résidus ne comporte pas

de corrélation croisée.

Nous effectuons ensuite le test de Ljung-Box multivarié, qui teste à la fois l'autocorrélation et la corrélation croisée. La statistique de ce test est $Q_h = T^2 \sum_{j=1}^h \frac{1}{t-j} \text{tr}(\hat{C}_j^T \hat{C}_0^{-1} \hat{C}_j \hat{C}_0^{-1})$, et elle suit une loi de $\chi^2(K^2 h)$. C'est donc la même statistique que pour le package **vars** à un coefficient près.

Ljung-Box Statistics:

##		m	Q(m)	df	p-value
##	[1,]	1.00	1.09	16.00	1.00
##	[2,]	2.00	5.29	32.00	1.00
##	[3,]	3.00	11.75	48.00	1.00
##	[4,]	4.00	20.55	64.00	1.00
##	[5,]	5.00	28.17	80.00	1.00
##	[6,]	6.00	43.72	96.00	1.00
##	[7,]	7.00	61.69	112.00	1.00
##	[8,]	8.00	85.09	128.00	1.00
##	[9,]	9.00	104.85	144.00	0.99
##	[10,]	10.00	119.74	160.00	0.99

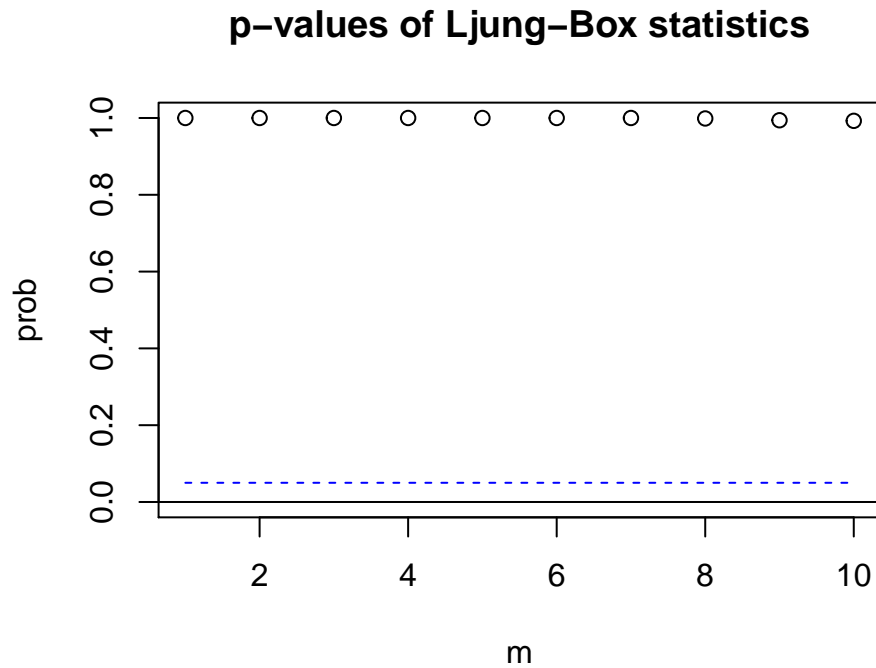


Figure 29: Evolution de la p-value en fonction du lag

Le résultat de ce test, dont la visualisation graphique est présente ci-dessus, permet également de conclure que les résidus suivent un bruit blanc.

3.5.4 Prévisions

Comme pour le package précédent, nous résumons dans le tableau suivant les informations sur les modèles construits. Encore une fois, le plus performant est le modèle comprenant toutes les variables. On note également que les EQM obtenues sont identiques à celles du package **vars**. Les AIC sont sensiblement les mêmes mais non identiques, en raison de méthodes de calcul différentes.

Variables utilisées	Ordre	AIC	EQM
MSE & PIB	4	-20.9333	9.153141e+14
MSE & SMIC	3	-15.4434	9.553417e+14
MSE & TCHOFF	6	-12.5389	9.880360e+14
MSE, PIB & SMIC	3	-27.7795	9.564856e+14
MSE, PIB & TCHOFF	4	-25.1289	9.020987e+14
MSE, SMIC & TCHOFF	4	-19.3573	7.955221e+14
MSE, PIB, SMIC & TCHOFF	4	-31.9763	7.711856e+14

4 Construction de modèles VARMA

A la fin du projet, nous avons cherché à ajouter une partie MA à nos modèles afin d'améliorer les prédictions. Le modèle s'écrit alors de la façon suivante :

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

On rajoute donc au modèle AR une partie MA représentée par les matrices $M_0 \dots M_q$. Ce type de modèle n'est que peu utilisé dans la pratique car il est difficile de satisfaire à la fois les hypothèses liées à la partie AR et celles liées à la partie MA. On lui préfère donc souvent un modèle AR, plus simple à construire.

Dans notre cas, la construction de modèle VARMA est possible grâce à la fonction *VARMA* du package **MTS**.

4.1 Calcul des ordres p et q

La méthode pour l'estimation des ordres p et q du modèles est un peu différente de celle pour un modèle VAR. En effet, nous utilisons cette fois les matrice de corrélation croisée étendues. Si le coefficient associée à la matrice de corrélation étendue est supérieur au seuil choisi (5% pour nous), alors un modèle peut être estimé. La fonction permettant de compiler les matrices de corrélation croisée étendues est *Eccm*.

```
## p-values table of Extended Cross-correlation Matrices:
## Column: MA order
## Row    : AR order
##      0      1      2      3      4      5
## 0 0.0000 0.0000 0.0000 0.0000 0.0053 0.0011
## 1 0.0000 0.0000 0.0000 0.0001 0.0132 0.0380
## 2 0.0376 0.0001 0.0000 0.0364 0.8917 0.9918
## 3 0.7560 0.6384 0.5263 0.9198 0.9927 0.9992
```

```
## 4 0.9996 0.8369 0.9676 0.9996 0.9995 0.9763
## 5 1.0000 0.9996 1.0000 1.0000 1.0000 0.9949
```

Pour le modèle complet, la matrice de corrélation croisée étendue devient significative à partir d'un modèle VARMA(3,0), soit un modèle VAR(3). Le but étant de rajouter une partie MA au modèle, les ordres retenus seront donc 3 et 1.

4.2 Estimation du modèle

Cependant, les résultats obtenus, présents en Annexe F du rapport, ne sont pas exploitables. En effet, de nombreux coefficients du modèles ne sont pas significatifs, potentiellement en raison du grand nombre de paramètres à estimer. Nous n'avons donc pas réussi à effectuer des prédictions avec ce modèle.

5 Prévisions sur les années suivantes

Maintenant que nous avons réalisé nos modèles, vérifié qu'ils satisfaisaient toutes les hypothèses de construction et comparé les prédictions de ces modèles aux valeurs de notre échantillon de test, nous effectuons quelques prédictions pour les années 2018, 2019, 2020 et 2021. Nous utilisons le modèle complet avec un ordre $p=4$ étant donné que c'est celui qui donnait les meilleurs résultats.

##		Qtr1	Qtr2	Qtr3	Qtr4
##	2016	1502432615	1523583648	1367631778	1497596174
##	2017	1484467771	1501242439	1357755499	1480917423
##	2018	1468991799	1485696650	1348625175	1467776769
##	2019	1456214113	1473382375	1340612756	1457418475
##	2020	1445681054	1463727320	1333867818	1449144583
##	2021	1437158372	1455971909	1328199051	1442510992

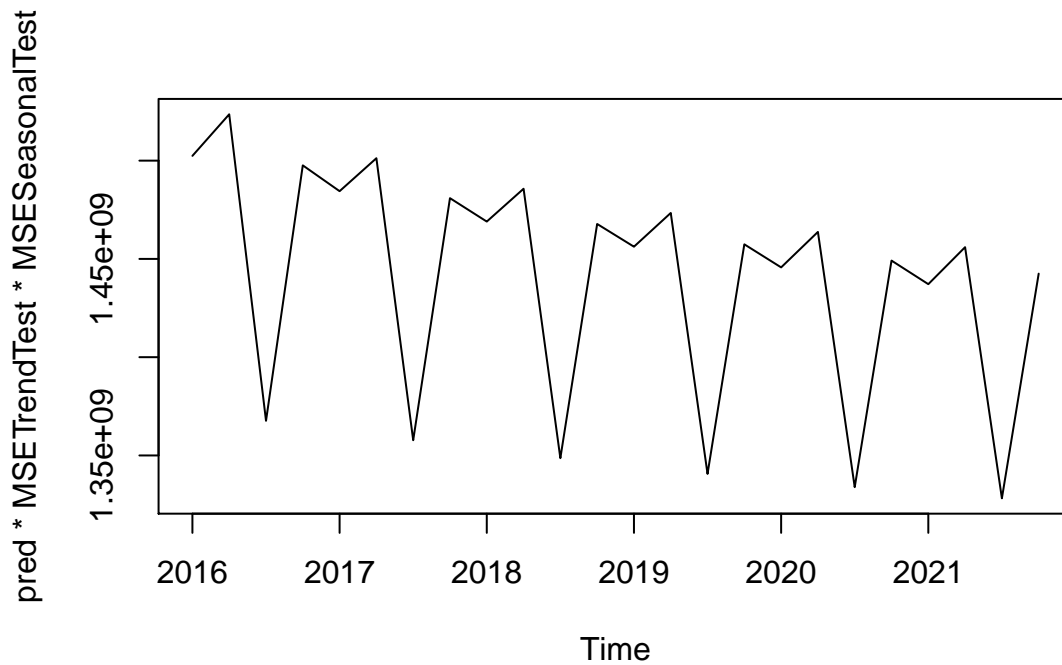


Figure 30: Prévisions du modèle VAR pour les années de 2016 à 2021

D'après le modèle, la masse salariale devrait baisser dans les années à venir, d'environ 10 millions chaque année par trimestre. Elle devrait donc par exemple être d'environ 1.44 milliard d'euros sur le trimestre prochain.

Conclusion

Nous avons donc mis en place des modèles de la famille VAR sur le jeu de données à notre disposition. L'objectif était de regarder si ce type de modèle pouvait être appliqué en respectant les différents hypothèses sous-jacentes au modèle, et de regarder la qualité des prédictions du modèle s'il était possible de le construire.

Afin de mettre en place ce type de modèles, nous avons utilisé deux packages différents, à savoir `vars` et `MTS`, et nous nous rendons compte que les résultats des deux packages sont similaires.

Nous avons également tenté d'ajouter une composante moyenne-mobile pour former des modèles VARMA et regarder la qualité de ce type de modèle par rapport aux modèles VAR que nous avons mis en place. Malheureusement, les résultats de ces modèles n'étaient pas exploitables.

Afin de vérifier si la modélisation vectorielle est la plus adaptée pour les données de l'IRCEM, il serait bon de comparer ces prédictions avec des modèles construits à l'aide de fonction de transfert, présents également dans le package `MTS`.

A Annexe 1 : Modélisation univariée des séries

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. Nous nous sommes aidés pour ce faire des travaux de Nicolas Devignes (Devignes 2018) lors d’un stage au laboratoire Paul Painlevé sur ce sujet.

A.1 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.

A.2 Lissage exponentiel

A.2.1 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 2.2 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 avons rentré.

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

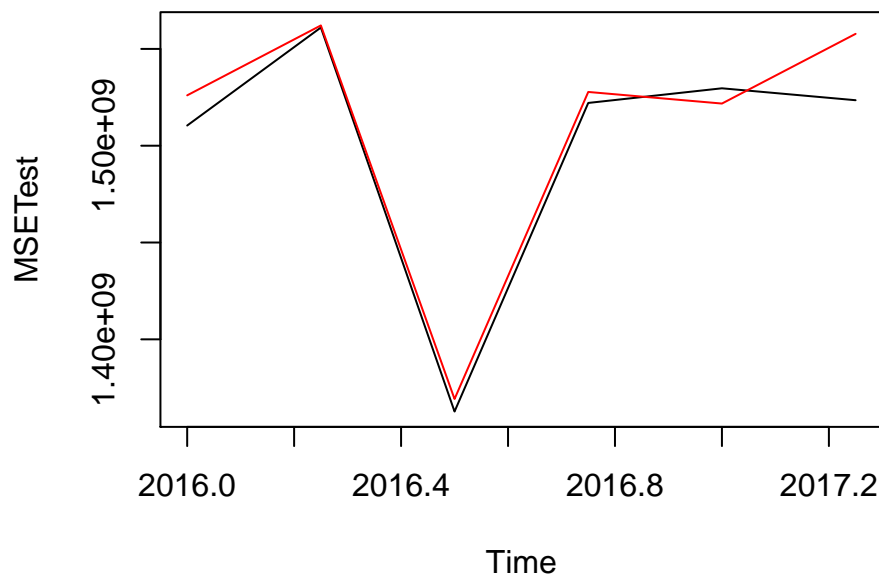


Figure 31: Comparaison entre la prédiction du lissage exponentiel et les valeurs réelles pour la masse salariale trimestrielle

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

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 31 nous montre que le modèle obtenu nous donne des prédictions très proches de la réalité.

A.2.2 Résultats obtenus

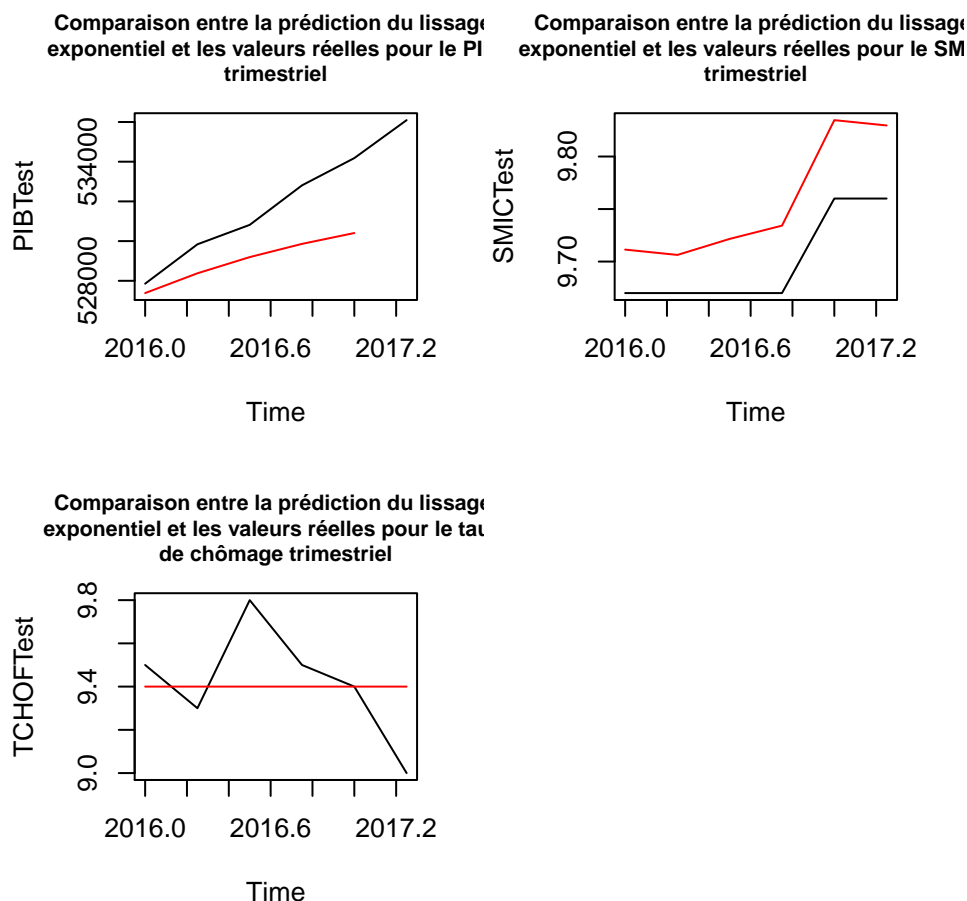


Figure 32: Résultats obtenus pour le lissage exponentiel

Les graphiques de la figure 32 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
PIB	linéaire	absente	ZAN	2053.15
SMIC	linéaire	additive	ZAA	-84.96
TCHOFT	absente	absente	ZNN	204.37

A.3 Modèles ARMA

A.3.1 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 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 les raisons évoquées dans la partie A.2.1

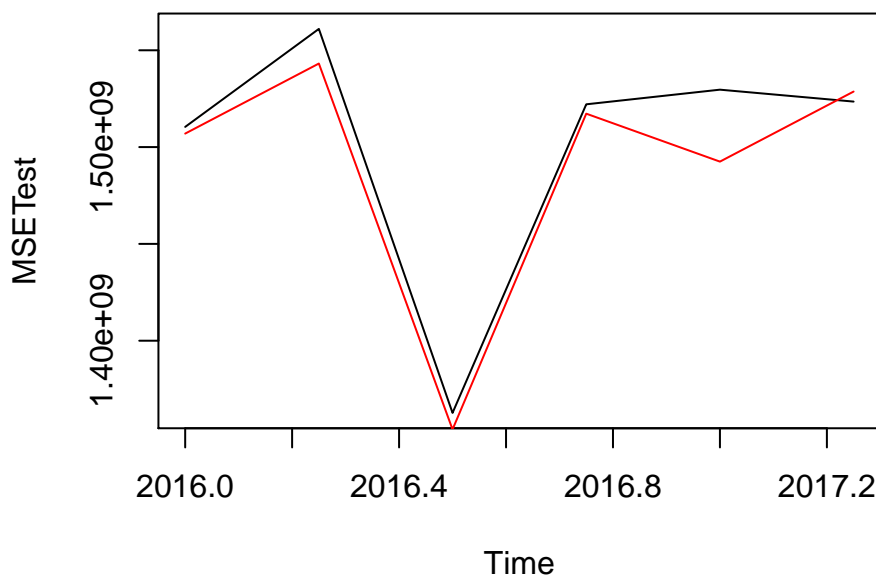


Figure 33: Comparaison entre le modèle SARIMA et les données de validation pour la masse salariale trimestrielle

Pour la MSE, on obtient par exemple un modèle $SARIMA(0, 1, 1)(0, 1, 1)_4$ ainsi qu'un AICc de 3896.01. La figure 33 nous donne des prédictions d'assez bonne qualité mais qui semblent moins bonnes que celles obtenues par lissage exponentiel.

A.3.2 Résultats obtenus

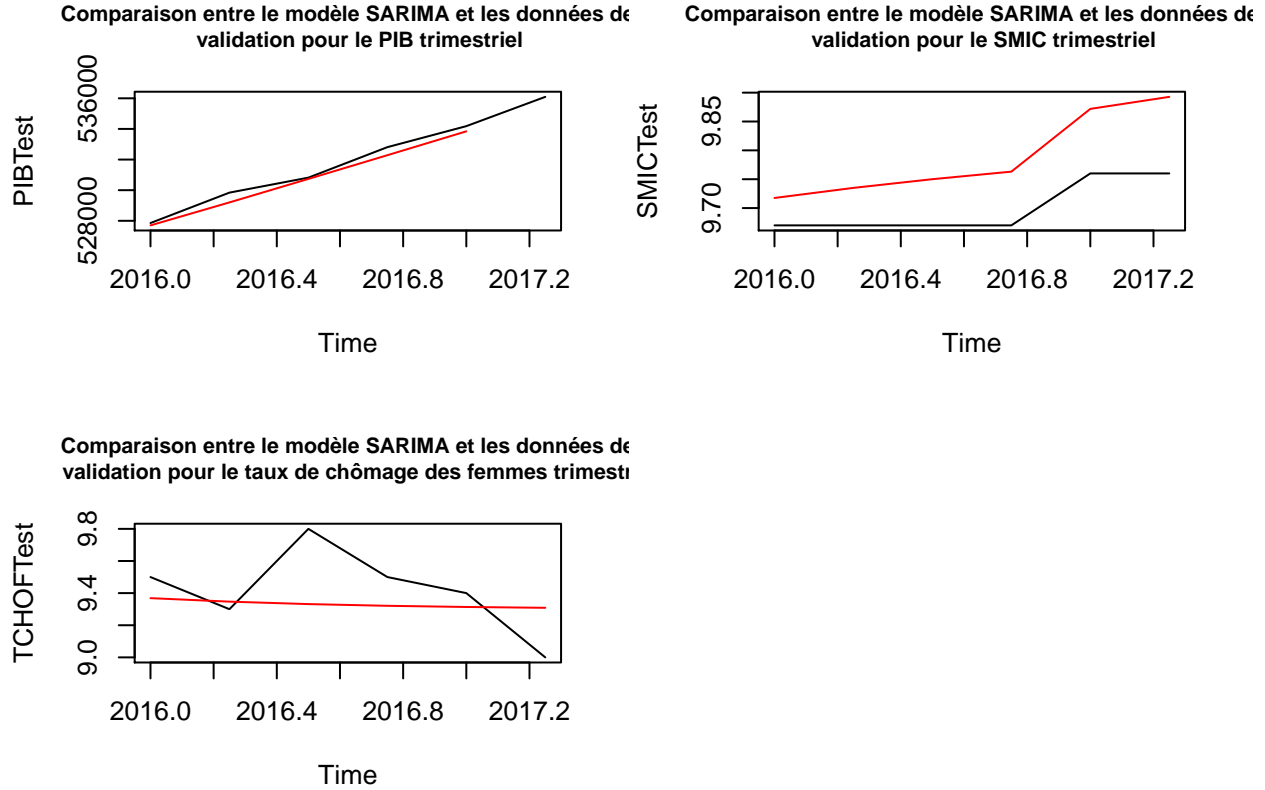


Figure 34: Résultats obtenus avec des modèles ARMA

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 2.3 et 2.5 on ne constate pas de saisonnalité dans l'analyse descriptive de la série. On peut également voir sur la figure 34 que les prédictions du PIB semblent de bien meilleure qualité

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
TCHOFT	(0,1,1)	9.76

A.4 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 utilisé 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.

##	lissage	ARMA
## MSE	2.592281e+14	3.060118e+14
## PIB	4.654031e+06	1.382004e+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 l'Annexe A

B Annexe 2 : Modélisation ARMA avec variables exogènes

B.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.

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

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

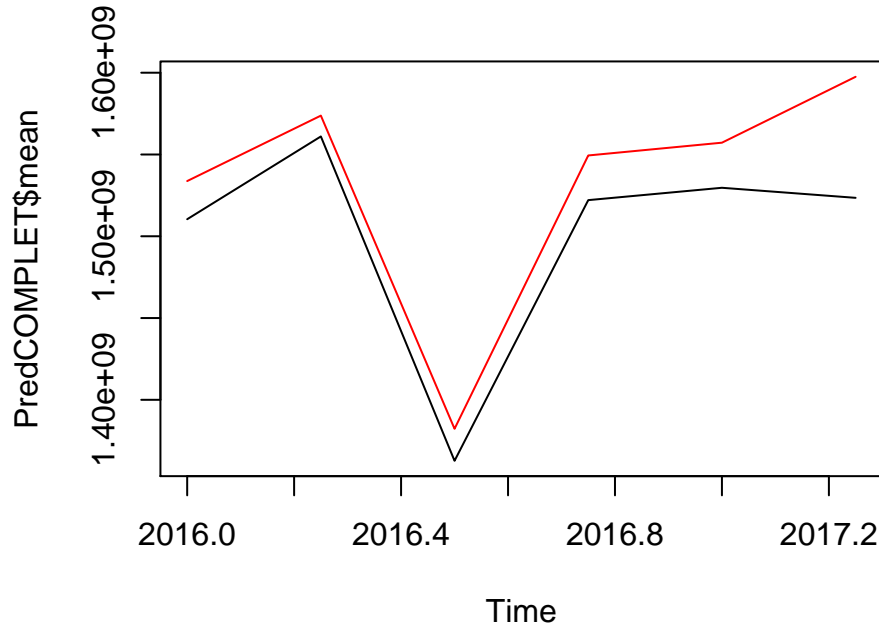


Figure 35: Masse salariale expliquée par le PIB, le SMIC et le taux de chômage vs Vraies valeurs

[1] 1.345414e+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 $\beta_3 = -1644706$

B.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 représentée dans le tableau ci-dessous.

Variable	Ordre du processus	AICc	EQM
PIB	(1,0,0)(2,1,0)	3715.99	1.170337e+15
SMIC	(0,1,0)(0,1,0)	3688.92	6.219316e+14
TCHOF	(0,0,0)(1,1,0)	3809.82	1.949846e+15
PIB & SMIC	(0,0,0)(0,1,0)	3807.37	1.330526e+15
PIB & TCHOF	(1,0,0)(0,1,0)	3721.36	7.424654e+14
SMIC & TCHOF	(0,1,0)(0,1,0)	3690.57	5.586871e+14
COMPLET	(0,0,0)(0,1,0)	3809.5	1.324892e+15

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 et le taux de chômage (dans la figure 36).

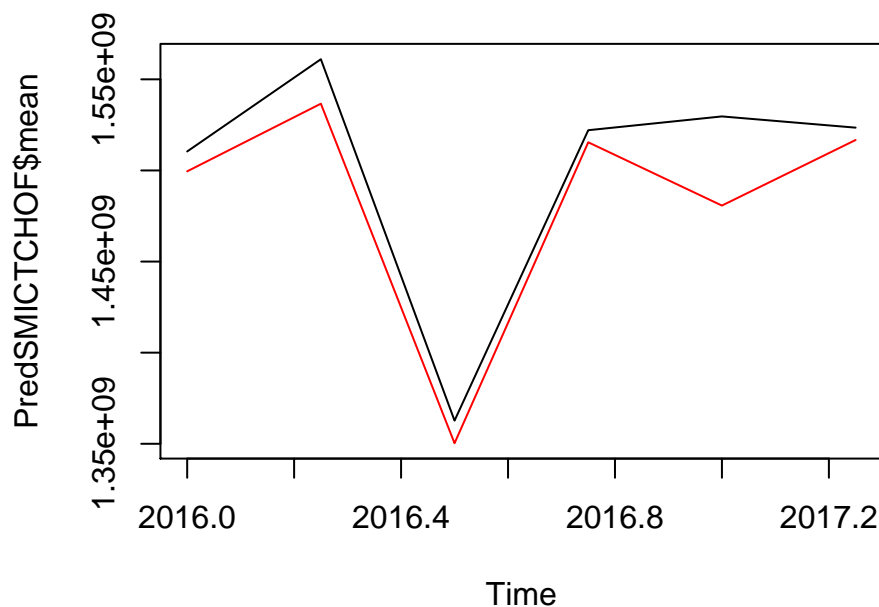


Figure 36: SARIMA expliqué par le SMIC et le taux de chômage vs Vraies valeurs

```
## [1] 5.586871e+14
```

C Annexe 3 : Erreurs standard associées aux coefficients du modèle VAR d'ordre 4

##	MSE	PIB	SMIC	TCHOF
## MSE	-0.44837936	-0.85710828	0.059065357	-0.016289594
## PIB	-0.03290389	-0.03056789	-0.006426264	-0.001447922
## SMIC	-0.12186822	-0.59050790	-0.674266226	-0.004993847
## TCHOF	0.40185192	-17.96140240	0.821446464	-0.310004540
##	MSE	PIB	SMIC	TCHOF
## MSE	-0.27254709	-0.67990488	0.04532636	0.0043284001
## PIB	-0.02833629	-0.06107288	-0.01176086	-0.0004898846
## SMIC	-0.27835989	-0.37298995	-0.41721629	-0.0094899806
## TCHOF	0.25697997	-16.67936660	0.63877023	-0.1024043275
##	MSE	PIB	SMIC	TCHOF

```
## MSE      0.19486895 -0.4557750  0.09080679 -0.001506479
## PIB      -0.02602058 -0.2762433 -0.00804238  0.001727524
## SMIC      0.14617955  1.3033993  0.08614964 -0.022040339
## TCHOF     -1.24236337 -2.7851730  1.22452627 -0.152158590
```

D Annexe 4 : Résultats obtenus pour un modèle VAR d'ordre 3 pour le modèle complet avec le package MTS

```
## Loading required package: MTS
## Warning: package 'MTS' was built under R version 3.4.4
##
## Attaching package: 'MTS'
## The following object is masked from 'package:vars':
##
##      VAR
##
## Constant term:
## Estimates:  3.217852 0.9639509 1.007265 36.54962
## Std.Error:  1.298974 0.1768625 2.680203 10.60551
## AR coefficient matrix
## AR( 1 )-matrix
##      [,1] [,2] [,3] [,4]
## [1,] -0.5553 0.588 0.00789 -0.00716
## [2,] -0.0171 0.181 -0.00016 0.00115
## [3,] -0.0190 0.436 -0.53423 -0.01568
## [4,] -0.4049 -6.711 0.15436 -0.27849
## standard error
##      [,1] [,2] [,3] [,4]
## [1,] 0.1012 0.791 0.04675 0.01296
## [2,] 0.0138 0.108 0.00636 0.00176
## [3,] 0.2088 1.632 0.09645 0.02674
## [4,] 0.8264 6.458 0.38167 0.10581
## AR( 2 )-matrix
##      [,1] [,2] [,3] [,4]
## [1,] -0.5372 -0.7716 -0.014791 -0.011517
## [2,] -0.0117 -0.0185 -0.000407 -0.000732
## [3,] -0.2281 -0.4727 -0.739666 -0.003983
## [4,] 1.1588 -17.1794 0.026010 -0.260963
## standard error
##      [,1] [,2] [,3] [,4]
## [1,] 0.1015 0.81 0.03868 0.0125
## [2,] 0.0138 0.11 0.00527 0.0017
## [3,] 0.2095 1.67 0.07982 0.0258
## [4,] 0.8290 6.61 0.31584 0.1021
## AR( 3 )-matrix
```

```

##           [,1]      [,2]      [,3]      [,4]
## [1,] -0.37851  -0.5634 -0.01094  0.004835
## [2,] -0.00979  -0.0879 -0.00835  0.000458
## [3,] -0.33392  -0.3883 -0.45705 -0.013429
## [4,]  0.76852 -14.1815  0.06093 -0.025304
## standard error
##           [,1]      [,2]      [,3]      [,4]
## [1,]  0.1004  0.839  0.0470  0.01241
## [2,]  0.0137  0.114  0.0064  0.00169
## [3,]  0.2071  1.731  0.0971  0.02560
## [4,]  0.8196  6.848  0.3841  0.10129
##
## Residuals cov-mtx:
##           [,1]      [,2]      [,3]      [,4]
## [1,]  1.902501e-04  1.717859e-07  1.508136e-05 -3.048385e-05
## [2,]  1.717859e-07  3.526920e-06 -1.012623e-05 -3.262224e-05
## [3,]  1.508136e-05 -1.012623e-05  8.099521e-04  2.986384e-04
## [4,] -3.048385e-05 -3.262224e-05  2.986384e-04  1.268199e-02
##
## det(SSE) =  6.444825e-15
## AIC =  -31.7155
## BIC =  -30.46502
## HQ  =  -31.20941

```

E Annexe 5 : Résultats obtenus pour un modèle VAR d'ordre 2 pour le modèle complet avec le package MTS

```

## Constant term:
## Estimates:  1.923469  0.8755149  0.6145106  24.51231
## Std.Error:  1.075733  0.1383336  2.345584  8.370214
## AR coefficient matrix
## AR( 1 )-matrix
##           [,1]      [,2]      [,3]      [,4]
## [1,] -0.396  0.830  0.03020 -0.00916
## [2,] -0.013  0.187  0.00609  0.00104
## [3,]  0.129  0.941 -0.23993 -0.02214
## [4,] -0.516 -6.591  0.10337 -0.21726
## standard error
##           [,1]      [,2]      [,3]      [,4]
## [1,]  0.0977  0.832  0.03839  0.01283
## [2,]  0.0126  0.107  0.00494  0.00165
## [3,]  0.2130  1.815  0.08371  0.02798
## [4,]  0.7601  6.476  0.29873  0.09985
## AR( 2 )-matrix
##           [,1]      [,2]      [,3]      [,4]
## [1,] -0.3884  -0.9693 -0.02420 -0.005659

```



```

## [2,] -0.0123 -0.0376 0.00169 -0.000400
## [3,] -0.1219 -1.5621 -0.64086 0.000695
## [4,] 0.7849 -18.1915 0.03345 -0.225798
## standard error
##      [,1] [,2] [,3] [,4]
## [1,] 0.0969 0.841 0.03883 0.01272
## [2,] 0.0125 0.108 0.00499 0.00164
## [3,] 0.2112 1.835 0.08467 0.02773
## [4,] 0.7538 6.547 0.30214 0.09896
##
## Residuals cov-mtx:
##      [,1] [,2] [,3] [,4]
## [1,] 2.239911e-04 1.481068e-06 5.388294e-05 -6.162593e-05
## [2,] 1.481068e-06 3.704053e-06 -6.133443e-06 -2.671666e-05
## [3,] 5.388294e-05 -6.133443e-06 1.064936e-03 1.240050e-04
## [4,] -6.162593e-05 -2.671666e-05 1.240050e-04 1.356110e-02
##
## det(SSE) = 1.149382e-14
## AIC = -31.45697
## BIC = -30.62331
## HQ = -31.11957

```

F Annexe 6 : Résultats obtenus pour un modèle VARMA(3,1) pour le modèle complet

```

## Number of parameters: 68
## initial estimates: 4.9726 0.8772 -0.946 43.7282 -0.3981 -0.8484 -0.0838 -0.0103 -0.5111 -0.0
## Par. lower-bounds: 2.1854 0.4817 -7.1093 19.5479 -0.7005 -2.7724 -0.1968 -0.04 -0.7174 -1.0
## Par. upper-bounds: 7.7597 1.2726 5.2174 67.9086 -0.0957 1.0756 0.0291 0.0194 -0.3048 1.203
## Final Estimates: 4.966752 0.8671174 -0.9442231 43.72315 -0.3979297 -0.8491019 -0.0815316
## Warning in sqrt(diag(solve(Hessian))): production de NaN
##
## Coefficient(s):
##      Estimate Std. Error t value Pr(>|t|)
## MSESta 4.967e+00 7.002e-01 7.093 1.31e-12 ***
## PIBSta 8.671e-01 1.129e-02 76.782 < 2e-16 ***
## SMICSta -9.442e-01 7.123e-01 -1.326 0.184964
## TCHOFSta 4.372e+01 6.236e+00 7.011 2.36e-12 ***
## MSESta -3.979e-01 1.536e-01 -2.590 0.009586 **
## PIBSta -8.491e-01 3.117e-01 -2.724 0.006451 **
## SMICSta -8.153e-02 4.714e-02 -1.730 0.083678 .
## TCHOFSta -2.464e-02 1.903e-02 -1.295 0.195363
## MSESta -5.094e-01 1.204e-01 -4.231 2.33e-05 ***
## PIBSta -3.442e-01 5.722e-01 -0.602 0.547491
## SMICSta -3.306e-02 4.292e-02 -0.770 0.441224

```

##	TCHOFSta	-1.880e-02	1.427e-02	-1.317	0.187744	
##	MSESta	-3.066e-01	1.410e-01	-2.174	0.029674	*
##	PIBSta	-1.558e+00	4.696e-01	-3.317	0.000909	***
##	SMICSta	-5.937e-02	5.272e-02	-1.126	0.260080	
##	TCHOFSta	5.289e-03	1.423e-02	0.372	0.710211	
##	MSESta	-1.119e-02	1.782e-02	-0.628	0.530263	
##	PIBSta	2.802e-01	1.332e-02	21.039	< 2e-16	***
##	SMICSta	1.284e-02	8.390e-03	1.531	0.125843	
##	TCHOFSta	-3.403e-03	2.443e-03	-1.393	0.163656	
##	MSESta	-2.036e-02	1.899e-02	-1.072	0.283789	
##	PIBSta	-9.599e-02	7.650e-03	-12.548	< 2e-16	***
##	SMICSta	-8.384e-04	7.353e-03	-0.114	0.909222	
##	TCHOFSta	2.298e-04	2.377e-03	0.097	0.922988	
##	MSESta	2.834e-02	1.667e-02	1.700	0.089136	.
##	PIBSta	-4.772e-02	1.212e-02	-3.938	8.21e-05	***
##	SMICSta	-5.811e-03	7.569e-03	-0.768	0.442701	
##	TCHOFSta	2.129e-03	1.953e-03	1.090	0.275512	
##	MSESta	-1.471e-01	3.454e-01	-0.426	0.670168	
##	PIBSta	3.356e+00	NA	NA	NA	
##	SMICSta	-5.691e-01	1.579e-01	-3.604	0.000313	***
##	TCHOFSta	-9.785e-03	4.269e-02	-0.229	0.818716	
##	MSESta	-2.251e-01	4.090e-01	-0.550	0.582127	
##	PIBSta	-1.141e+00	9.042e-01	-1.262	0.207119	
##	SMICSta	-7.717e-01	1.521e-01	-5.073	3.91e-07	***
##	TCHOFSta	-1.291e-02	4.956e-02	-0.261	0.794440	
##	MSESta	-4.487e-01	3.331e-01	-1.347	0.177983	
##	PIBSta	-4.569e-01	5.284e-01	-0.865	0.387236	
##	SMICSta	-4.799e-01	1.535e-01	-3.127	0.001768	**
##	TCHOFSta	-1.936e-02	4.431e-02	-0.437	0.662211	
##	MSESta	1.872e-01	8.303e-01	0.225	0.821671	
##	PIBSta	-1.081e+01	3.755e+00	-2.878	0.004005	**
##	SMICSta	-1.121e-01	3.832e-01	-0.292	0.769910	
##	TCHOFSta	-3.008e-01	1.154e-01	-2.607	0.009121	**
##	MSESta	1.629e+00	9.702e-01	1.679	0.093193	.
##	PIBSta	-2.041e+01	4.320e+00	-4.724	2.32e-06	***
##	SMICSta	-6.288e-02	3.689e-01	-0.170	0.864672	
##	TCHOFSta	-2.836e-01	1.190e-01	-2.383	0.017191	*
##	MSESta	1.541e+00	9.351e-01	1.648	0.099412	.
##	PIBSta	-1.590e+01	3.116e+00	-5.102	3.36e-07	***
##	SMICSta	-9.026e-02	4.114e-01	-0.219	0.826344	
##	TCHOFSta	-3.977e-02	1.125e-01	-0.354	0.723709	
##		-2.472e-01	1.135e-01	-2.178	0.029423	*
##		2.687e+00	5.575e-01	4.820	1.44e-06	***
##		1.391e-01	NA	NA	NA	
##		1.615e-02	1.477e-02	1.093	0.274222	
##		4.487e-03	NA	NA	NA	
##		-2.187e-01	6.839e-02	-3.197	0.001389	**
##		-2.149e-02	NA	NA	NA	

```

##          3.349e-03          NA          NA          NA
##        -4.620e-02    9.016e-03    -5.124  2.99e-07 ***
##        -7.399e+00          NA          NA          NA
##          1.391e-01    8.062e-03    17.249 < 2e-16 ***
##        -6.520e-02    4.158e-03   -15.681 < 2e-16 ***
##        -9.380e-01    2.546e-01    -3.684  0.000230 ***
##        -7.022e+00    3.771e+00    -1.862  0.062556 .
##          9.064e-01          NA          NA          NA
##        -5.974e-02          NA          NA          NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---
## Estimates in matrix form:
## Constant term:
## Estimates:  4.966752 0.8671174 -0.9442231 43.72315
## AR coefficient matrix
## AR( 1 )-matrix
##          [,1]    [,2]    [,3]    [,4]
## [1,] -0.3979  -0.849  -0.0815 -0.02464
## [2,] -0.0112   0.280   0.0128 -0.00340
## [3,] -0.1471   3.356  -0.5691 -0.00979
## [4,]  0.1872 -10.807  -0.1121 -0.30080
## AR( 2 )-matrix
##          [,1]    [,2]    [,3]    [,4]
## [1,] -0.5094  -0.344  -0.033057 -0.01880
## [2,] -0.0204  -0.096  -0.000838  0.00023
## [3,] -0.2251  -1.141  -0.771670 -0.01291
## [4,]  1.6288 -20.408  -0.062880 -0.28361
## AR( 3 )-matrix
##          [,1]    [,2]    [,3]    [,4]
## [1,] -0.3066  -1.5579 -0.05937  0.00529
## [2,]  0.0283  -0.0477 -0.00581  0.00213
## [3,] -0.4487  -0.4569 -0.47994 -0.01936
## [4,]  1.5408 -15.8954 -0.09026 -0.03977
## MA coefficient matrix
## MA( 1 )-matrix
##          [,1]    [,2]    [,3]    [,4]
## [1,]  0.24717 -2.687  -0.1391 -0.01615
## [2,] -0.00449  0.219   0.0215 -0.00335
## [3,]  0.04620  7.399  -0.1391  0.06520
## [4,]  0.93802  7.022  -0.9064  0.05974
##
## Residuals cov-matrix:
##          [,1]          [,2]          [,3]          [,4]
## [1,]  2.426715e-04  6.285942e-06 -1.024122e-04 -0.0001170925
## [2,]  6.285942e-06  7.605929e-06 -9.222828e-05 -0.0001004204
## [3,] -1.024122e-04 -9.222828e-05  2.764500e-03  0.0024070658
## [4,] -1.170925e-04 -1.004204e-04  2.407066e-03  0.0159049977

```

```
## ----
## aic= -29.47208
## bic= -27.70056
```

G Annexe 7 : Code R du projet

```
##### Fonctions créées pour optimiser le traitement #####

EQM<-function(serie, prediction){
  return(sum((serie - prediction)^2)/length(serie))
}

stabilityvars<-function(modele){
  k<-modele$K
  p<-modele$p
  A <- matrix(0,nrow=p*k, ncol=p*k)
  deb<-nrow(A)-k+1
  for(i in seq(1,(nrow(A)-2*k+1),k)){
    A[(i+k):(i+k*2-1),i:(i+k-1)] = diag(k)
  }
  for(i in seq(1,deb,k)){
    A[1:k,i:(i+k-1)] = getCoefMatrix(modele, i)
  }
  vp<-eigen(A)$values
  print(Mod(vp))
  plot(seq(1,nrow(A)), Mod(vp), xlab="",
        ylab="Modules des valeurs propres", ylim=c(0,1.5))
  abline(h=1, col="red")
}

getCoefMatrix<-function(modele, p){
  k<-modele$K
  result<-matrix(nrow=k, ncol=k)
  for(i in 1:length(modele$varresult)){
    result[i,] = modele$varresult[[i]]$coefficients[1:(1+k-1)]
  }
  return(result)
}

stabilityMTS<-function(modele){
  k<-ncol(modele$data)
  p<-modele$order
  A <- matrix(0,nrow=p*k, ncol=p*k)
  deb<-nrow(A)-k+1
  for(i in seq(1,(nrow(A)-2*k+1),k)){
    A[i:(i+k-1),(i+k):(i+k*2-1)] = diag(k)
  }
}
```

```

}
j = ncol(A)
for(i in seq(1,deb,k)){
  A[deb:nrow(A),(j-k+1):j] = t(modele$coef[(i+1):(i+k),])
  j = j-k
}
vp<-eigen(A)$values
print(Mod(vp))
plot(seq(1,nrow(A)), Mod(vp), xlab="",
      ylab="Modules des valeurs propres", ylim=c(0,1.5))
abline(h=1, col="red")
}

cor.mtest <- function(mat, conf.level = 0.95){
  mat <- as.matrix(mat)
  n <- ncol(mat)
  p.mat <- lowCI.mat <- uppCI.mat <- matrix(NA, n, n)
  diag(p.mat) <- 0
  diag(lowCI.mat) <- diag(uppCI.mat) <- 1
  for(i in 1:(n-1)){
    for(j in (i+1):n){
      tmp <- cor.test(mat[,i], mat[,j], conf.level = conf.level)
      p.mat[i,j] <- p.mat[j,i] <- tmp$p.value
      lowCI.mat[i,j] <- lowCI.mat[j,i] <- tmp$conf.int[1]
      uppCI.mat[i,j] <- uppCI.mat[j,i] <- tmp$conf.int[2]
    }
  }
  return(list(p.mat, lowCI.mat, uppCI.mat))
}

archTest<-function(modele){
  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")
}

calculC0<-function(modele){
  C0 <- matrix(nrow = modele$K, ncol=modele$K, 0)
  for(i in 1:nrow(residuals(modele))){
    C0 <- C0 + residuals(modele)[i,]%*%t(residuals(modele)[i,])
  }
  C0 <- (1/nrow(residuals(modele))) * C0
  C0
  d <- det(C0)
}

```

```

names(d) <- "Déterminant de la matrice"
d
}
#####

##### Importation des données #####
trim <- read.csv("~/PFE_Time_Series/Data/Data_Trim.csv", sep=";", dec=",")
#####

##### Analyse descriptive des séries #####

#MSE
#Transformation de la masse salariale en série temporelle
MSE <- ts(trim$MSE, start = 1990, end = c(2017, 2), frequency=4)
#Affichage de la série temporelle
plot(MSE, main="Evolution trimestrielle de la masse salariale", xaxt="n", cex.main=0.9)
#Modification de l'axe x pour afficher le nom des trimestres
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
#Division de l'espace d'affichage des graphiques en deux
par(mfrow=c(1,2), cex.main=0.8)
#Calcul de l'ACF et des PACF de la masse salariale
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)
#Réalisation de tests de stationnarité et de racines unitaires
kpss.test(MSE)
adf.test(MSE)

#PIB
#Transformation du PIB en série temporelle
PIB <- ts(trim$PIB, start = 1990, end = c(2017, 1), frequency=4)
#Affichage du PIB
plot(PIB, main="Evolution trimestrielle du PIB", xaxt="n", cex.main=0.9)
#Modification de l'axe x
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
#Division de l'espace d'affichage des graphiques en deux
par(mfrow=c(1,2), cex.main=0.8)
#Calcul de l'ACF et la PACF du PIB
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))
#Réalisation des tests de stationnarité et de racines unitaires
kpss.test(PIB)
adf.test(PIB)

#SMIC
#Transformation du SMIC en série temporelle
SMIC <- ts(trim$SMIC, start = c(1990,1), end = c(2017, 4), frequency = 4)

```

```

#Affichage du SMIC
plot(SMIC, main="Evolution trimestrielle du SMIC", xaxt="n", cex.main=0.9)
#Modification de l'axe x
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
#Division de l'espace d'affichage des graphiques en deux
par(mfrow=c(1,2), cex.main=0.8)
#Calcul de l'ACF et la PACF du SMIC
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))
#Réalisation des tests de stationnarité et de racines unitaires
kpss.test(SMIC)
adf.test(SMIC)

#TCHOF
#Création du taux de chômage en série temporelle
TCHOF <- ts(trim$TCHOF, start = c(1990,1), end = c(2017, 4), frequency = 4)
#Affichage du taux de chômage
plot(TCHOF, main="Evolution trimestrielle du taux de chômage des femmes", xaxt="n", cex.main=0.9)
#Modification de l'axe x
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
#Division de l'espace d'affichage des graphiques en deux
par(mfrow=c(1,2), cex.main=0.8)
#Calcul de l'ACF et de la PACF du taux de chômage
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))
#Réalisation des tests de stationnarité et de racines unitaires
kpss.test(TCHOF)
adf.test(TCHOF)

#Corrélations

#Matrice 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))
#Calcul des p-values associées aux coefficients
corr <- cor.mtest(trim[1:109,-1], 0.95)[[1]]
#Ajout des libellés à la matrice des p-values
rownames(corr) <- c("MSE", "PIB", "SMIC", "TCHOF")
colnames(corr) <- c("MSE", "PIB", "SMIC", "TCHOF")
#Affichage des p-values
corr

#Découpage des séries en échantillons d'apprentissage et de test
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))

#####

##### Modélisation individuelle #####

##Lissage exponentiel

#MSE
#Création du lissage exponentiel de la masse salariale
LEMSE<-ets(MSETrain, "ZAM")
#Affichage des différents coefficients du lissage
print(LEMSE)
#Prédiction des 6 prochaines valeurs
PredLEMSE <- forecast(LEMSE, h = 6)
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(MSETest, main="Comparaison entre la prédiction du lissage exponentiel et les valeurs réelles",
lines(PredLEMSE$mean, col="red"))
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(MSETest, PredLEMSE$mean)

#PIB
#Création du lissage exponentiel du PIB
LEPIB<-ets(PIBTrain, "ZAN")
#Affichage des différents coefficients du lissage
print(LEPIB)
#Prédiction des 5 prochaines valeurs
PredLEPIB <- forecast(LEPIB, h = 5)
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(PIBTest, ylim=c(min(PIBTest,PredLEPIB$mean),max(PIBTest,PredLEPIB$mean)), main= "Comparaison entre la prédiction du lissage exponentiel et les valeurs réelles",
lines(PredLEPIB$mean, col="red"))
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(PIBTest, PredLEPIB$mean)

#SMIC
#Création du lissage exponentiel du SMIC
LESMIC<-ets(SMICTrain, "ZAA")
#Affichage des différents coefficients du lissage
print(LESMIC)
#Prédiction des 6 prochaines valeurs
PredLESMIC <- forecast(LESMIC, h = 6)

```



```

#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(SMICTest, ylim=c(min(SMICTest,PredLESMIC$mean),max(SMICTest,PredLESMIC$mean)), main="Comparaison entre le modèle SARIMA et les données de validation pour la masse salariale",
lines(PredLESMIC$mean, col="red")
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(SMICTest, PredLESMIC$mean)

#Taux de chômage des femmes
#Création du lissage exponentiel du taux de chômage
LETCHOF<-ets(TCHOFTrain, "ZNN")
#Affichage des différents coefficients
print(LETCHOF)
#Prédiction des 6 prochaines valeurs
PredLETCHOF <- forecast(LETCHOF, h = 6)
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(TCHOFTest, ylim=c(min(TCHOFTest,PredLETCHOF$mean),max(TCHOFTest,PredLETCHOF$mean)), main="Comparaison entre le modèle SARIMA et les données de validation pour le PIB trimestriel",
lines(PredLETCHOF$mean, col="red")
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(TCHOFTest, PredLETCHOF$mean)

#Modèles ARMA

#MSE
#Création du modèle SARIMA de la masse salariale
ARIMAMSE<-auto.arima(MSETrain, ic="aicc")
#Affichage des ordres et des coefficients du modèle
print(ARIMAMSE)
#Prédiction des 6 prochaines valeurs
PredARIMAMSE<- forecast(ARIMAMSE, h=6)
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(MSETest, main="Comparaison entre le modèle SARIMA et les données de validation pour la masse salariale",
lines(PredARIMAMSE$mean, col="red")
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(MSETest, PredARIMAMSE$mean)

#PIB
#Création du modèle ARIMA du PIB
ARIMAPIB<-auto.arima(PIBTrain, ic="aicc", seasonal=F)
#Affichage des ordres et des coefficients du modèle
print(ARIMAPIB)
#Prédiction des 5 prochaines valeurs
PredARIMAPIB<- forecast(ARIMAPIB, h=5)
#Affichage de l'échantillon de test et des prédictions afin de les comparer
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")
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test

```

```

EQM(PIBTest, PredARIMAPIB$mean)

#SMIC
#Création du modèle SARIMA du SMIC
ARIMASMIC<-auto.arima(SMICTrain, ic="aicc")
#Affichage des ordres et des coefficients du modèle
print(ARIMASMIC)
#Prédiction des 6 prochaines valeurs
PredARIMASMIC<- forecast(ARIMASMIC, h=6)
#Affichage de l'échantillon de test et des prédictions afin de les comparer
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$mean, col="red")
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(SMICTest, PredARIMASMIC$mean)

#TCHOF
#Création du modèle ARIMA du taux de chômage
ARIMATCHOF<-auto.arima(TCHOFTrain, ic="aicc", seasonal=F)
#Affichage des ordres et des coefficients du modèle
print(ARIMATCHOF)
#Prédiction des 6 prochaines valeurs
PredARIMATCHOF<- forecast(ARIMATCHOF, h=6)
#Affichage de l'échantillon de test et des prédictions afin de les comparer
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$mean, col="red")
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(TCHOFTest, PredARIMATCHOF$mean)

#Comparaison des résultats
#Création d'une matrice stockant dans une colonne les EQM des prédictions des lissages et dans
#prédictions des SARIMA
resultats<-matrix(nrow=4, ncol=2, dimnames = list(c("MSE", "PIB", "SMIC", "TCHOF"),
                                                  c("lissage", "ARMA")))

#Remplissage de la matrice
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)
#Affichage de la matrice
resultats

```

```

#Prédiction de la valeur du PIB pour 2017Q2
#Prédiction de la valeur manquante du PIB
PredARIMAPIB<- forecast(ARIMAPIB, h=6)
#Stockage de la valeur en question
new.value <- PredARIMAPIB$mean[6]
#Création du nouveau jeu de test du PIB avec la valeur estimée
PIBTest<-ts(c(PredARIMAPIB$mean, new.value), start = 2016, end = c(2017, 2), frequency=4)

#####

##### Modélisation ARMA avec variables exogènes de la MSE #####

#PIB
#Création d'un SARIMA du la masse salariale prenant en compte le PIB
SARIMAPIB <- auto.arima(MSETrain, xreg = cbind(PIBTrain))
SARIMAPIB
#Prédiction des prochaines valeurs
PredPIB <- forecast(SARIMAPIB, xreg = cbind(PIBTest))
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(PredPIB$mean, col="red",
      ylim=c(min(MSETest,PredPIB$mean), max(MSETest,PredPIB$mean)),
      main = "Masse salariale expliquée par le PIB vs Vraies valeurs")
lines(MSETest)
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(PredPIB$mean, MSETest)

#SMIC
#Création d'un SARIMA du la masse salariale prenant en compte le SMIC
SARIMASMIC <- auto.arima(MSETrain, xreg = cbind(SMICTrain))
SARIMASMIC
#Prédiction des prochaines valeurs
PredSMIC <- forecast(SARIMASMIC, xreg = cbind(SMICTest))
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(PredSMIC$mean, col="red",
      ylim=c(min(MSETest,PredSMIC$mean), max(MSETest,PredSMIC$mean)),
      main = "Masse salariale expliqué par le SMIC vs Vraies valeurs")
lines(MSETest)
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(PredSMIC$mean, MSETest)

#TCHOF
#Création d'un SARIMA du la masse salariale prenant en compte le taux de chômage
SARIMATCHOF <- auto.arima(MSETrain, xreg = cbind(TCHOFTrain))
SARIMATCHOF
#Prédiction des prochaines valeurs
PredTCHOF <- forecast(SARIMATCHOF, xreg = cbind(TCHOFTest))
#Affichage de l'échantillon de test et des prédictions afin de les comparer

```

```

plot(PredTCHOF$mean, col="red",
     ylim=c(min(MSETest,PredTCHOF$mean), max(MSETest,PredTCHOF$mean)),
     main = "Masse salariale expliquée par le taux de chômage vs Vraies valeurs")
lines(MSETest)
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(PredTCHOF$mean, MSETest)

#PIB & SMIC
#Création d'un SARIMA du la masse salariale prenant en compte le PIB et le SMIC
SARIMAPIBSMIC <- auto.arima(MSETrain, xreg = cbind(PIBTrain, SMICTrain))
SARIMAPIBSMIC
#Prédiction des prochaines valeurs
PredPIBSMIC <- forecast(SARIMAPIBSMIC, xreg = cbind(PIBTest, SMICTest))
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(PredPIBSMIC$mean, col="red",
     ylim=c(min(MSETest,PredPIBSMIC$mean), max(MSETest,PredPIBSMIC$mean)),
     main = "Masse salariale expliquée par le PIB et le SMIC vs Vraies valeurs")
lines(MSETest)
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(PredPIBSMIC$mean, MSETest)

#PIB & TCHOF
#Création d'un SARIMA du la masse salariale prenant en compte le PIB et le taux de chômage
SARIMAPIBTCHOF <- auto.arima(MSETrain, xreg = cbind(PIBTrain, TCHOFTrain))
SARIMAPIBTCHOF
#Prédiction des prochaines valeurs
PredPIBTCHOF <- forecast(SARIMAPIBTCHOF, xreg = cbind(PIBTest, TCHOFTest))
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(PredPIBTCHOF$mean, col="red",
     ylim=c(min(MSETest,PredPIBTCHOF$mean), max(MSETest,PredPIBTCHOF$mean)),
     main = "Masse salariale expliquée par le PIB et le taux de chômage vs Vraies valeurs")
lines(MSETest)
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(PredPIBTCHOF$mean, MSETest)

#SMIC & TCHOF
#Création d'un SARIMA du la masse salariale prenant en compte le SMIC et le taux de chômage
SARIMASMICTCHOF <- auto.arima(MSETrain, xreg = cbind(SMICTrain, TCHOFTrain))
SARIMASMICTCHOF
#Prédiction des prochaines valeurs
PredSMICTCHOF <- forecast(SARIMASMICTCHOF, xreg = cbind(SMICTest, TCHOFTest))
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(PredSMICTCHOF$mean, col="red",
     ylim=c(min(MSETest,PredSMICTCHOF$mean), max(MSETest,PredSMICTCHOF$mean)),
     main = "Masse salariale expliquée par le SMIC et le taux de chômage vs Vraies valeurs")
lines(MSETest)
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test

```

```

EQM(PredSMICTCHOF$mean, MSETest)

#PIB, SMIC & TCHOF
#Création d'un SARIMA de la masse salariale prenant en compte toutes les variables exogènes
SARIMACOMPLET <- auto.arima(MSETrain, xreg = cbind(PIBTrain, SMICTrain, TCHOFTrain))
SARIMACOMPLET
#Prédiction des prochaines valeurs
PredCOMPLET <- forecast(SARIMACOMPLET, xreg = cbind(PIBTest, SMICTest, TCHOFTest))
#Affichage de l'échantillon de test et des prédictions afin de les comparer
plot(PredCOMPLET$mean, col="red",
      ylim=c(min(MSETest, PredCOMPLET$mean), max(MSETest, PredCOMPLET$mean)),
      main = "Masse salariale expliquée par le PIB, le SMIC et le taux de chômage vs Vraies valeurs")
lines(MSETest)
#Calcul de l'EQM entre les prédictions et les valeurs de l'échantillon de test
EQM(PredCOMPLET$mean, MSETest)

#Comparaison des résultats
#Création d'un tableau stockant les EQM des différents modèles
resultats<-matrix(nrow=7, ncol=1, dimnames = list(c("PIB", "SMIC", "TCHOF", "PIB & SMIC", "PIB & TCHOF", "SMIC & TCHOF", "PIB & SMIC & TCHOF"),
                                                  c("EQM")))

#Remplissage du tableau
resultats[1,1] = EQM(MSETest, PredPIB$mean)
resultats[2,1] = EQM(MSETest, PredSMIC$mean)
resultats[3,1] = EQM(MSETest, PredTCHOF$mean)
resultats[4,1] = EQM(MSETest, PredPIBSMIC$mean)
resultats[5,1] = EQM(MSETest, PredPIBTCHOF$mean)
resultats[6,1] = EQM(MSETest, PredSMICTCHOF$mean)
resultats[7,1] = EQM(MSETest, PredCOMPLET$mean)
resultats

#####

##### Stationnarisation des séries #####

#MSE
par(cex.main=0.8)
#Affichage de la décomposition de la masse salariale
plot(decompose(MSETrain, "multiplicative"))
#Stockage des différentes composantes de la masse salariale
MSESta <- na.omit(decompose(MSETrain, "multiplicative")$random)
MSETrend<-window(decompose(MSETrain, "multiplicative")$trend)
MSESeasonal<-window(decompose(MSETrain, "multiplicative")$seasonal)
#Affichage des résidus de la masse salariale
plot(MSESta, main="Masse salariale trimestrielle stationnarisée", xaxt="n")
#Modification de l'axe x
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
#Division de l'espace d'affichage des graphiques en deux

```

```

par(mfrow=c(1,2), cex.main=0.8)
#Calcul de l'ACF et la PACF de la masse salariale stationnarisée
acf(MSESta, main="ACF de la masse salariale stationnarisée")
pacf(MSESta, main="PACF de la masse salariale stationnarisée")
par(mfrow=c(1,1))
#Réalisation des tests de stationnarité et de racines unitaires
kpss.test(MSESta)
adf.test(MSESta)
#Création de vecteurs contenant la tendance et la saisonnalité des futures valeurs
MSETrendTest <- window(forecast(na.omit(MSETrend), h=8)$mean, start=2016)
MSESeasonalTest<-ts(c(MSESeasonal[1:4],MSESeasonal[1:2]), start=2016,frequency=4)

#PIB
#Stockage de la composante résiduelle du PIB
PIBSta <- na.omit(decompose(PIBTrain, "multiplicative")$random)
#Affichage des résidus du PIB
plot(PIBSta, main="PIB trimestriel stationnarisé", xaxt="n")
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
#Division de l'espace d'affichage des graphiques en deux
par(mfrow=c(1,2))
#Calcul de l'ACF et la PACF du PIB stationnarisée
acf(PIBSta, main="Auto-Corrélation du PIB trimestriel stationnarisé")
pacf(PIBSta, main="Auto-Corrélation partielle du PIB trimestriel stationnarisé")
par(mfrow=c(1,1))
#Réalisation des tests de stationnarité et de racines unitaires
kpss.test(PIBSta)
adf.test(PIBSta)

#SMIC
#Stockage de la composante résiduelle du SMIC
SMICSta <- na.omit(decompose(SMICTrain)$random)
#Affichage des résidus du SMIC
plot(SMICSta, main="SMIC trimestriel stationnarisé", xaxt="n")
axis(side=1, at=seq(1990,2015,5), labels=c("1990Q1", "1995Q1", "2000Q1", "2005Q1", "2010Q1", "2015Q1"))
#Division de l'espace d'affichage des graphiques en deux
par(mfrow=c(1,2))
#Calcul de l'ACF et la PACF du SMIC stationnarisée
acf(SMICSta, main="Auto-Corrélation du SMIC trimestriellstationnarisé")
pacf(SMICSta, main="Auto-Corrélation partielle du SMIC trimestriel stationnarisé")
par(mfrow=c(1,1))
#Réalisation des tests de stationnarité et de racines unitaires
kpss.test(SMICSta)
adf.test(SMICSta)

#TCHOF
#Stockage de la composante résiduelle du taux de chômage
TCHOFSta <- na.omit(decompose(TCHOFTrain)$random)

```

```

#Affichage des résidus du taux de chômage
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"))
#Division de l'espace d'affichage des graphiques en deux
par(mfrow=c(1,2))
#Calcul de l'ACF et la PACF du taux de chômage stationnarisée
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 trimestriel stationnarisée")
par(mfrow=c(1,1))
#Réalisation des tests de stationnarité et de racines unitaires
kpss.test(TCHOFSta)
adf.test(TCHOFSta)

#####

##### Modèles VAR avec vars #####
detach("package:MTS", unload=TRUE)

#PIB
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARselect(cbind(MSESta, PIBSta), lag.max=10)

#Ordre 4
#Réalisation d'un modèle d'ordre 4 avec la masse salariale et le PIB
modele<-VAR(cbind(MSESta, PIBSta), p=4, type="const")
#vérification de la stabilité du modèle
stabilityvars(modele)
#Test d'homoscédasticité
archTest(modele)
#Test de normalité
normality.test(modele)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculC0(modele)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele)

#Ordre3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale et le PIB
modele2<-VAR(cbind(MSESta, PIBSta), p=3, type="const")
#vérification de la stabilité du modèle
stabilityvars(modele2)
#Test d'homoscédasticité
archTest(modele2)
#Test de normalité
normality.test(modele2)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculC0(modele2)

```



```

#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele2)

#Ordre 2
#Réalisation d'un modèle d'ordre 2 avec la masse salariale et le PIB
modele3<-VAR(cbind(MSESta, PIBSta), p=2, type="const")
#vérification de la stabilité du modèle
stabilityvars(modele3)
#Test d'homoscédasticité
archTest(modele3)
#Test de normalité
normality.test(modele3)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculC0(modele3)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele3)

#Prédictions avec le modèle d'ordre 4 et calcul de l'EQM
PredPIB<-forecast(modele, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredPIB*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
#Prédictions avec le modèle d'ordre 3 et calcul de l'EQM
PredPIB2<-forecast(modele2, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredPIB2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
#Prédictions avec le modèle d'ordre 2 et calcul de l'EQM
PredPIB3<-forecast(modele3, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredPIB3*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredPIB, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#SMIC
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARselect(cbind(MSESta, SMICSta), lag.max=10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale et le SMIC
modele<-VAR(cbind(MSESta, SMICSta), p=3, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele)
#Test d'homoscédasticité
archTest(modele)
#Test de normalité
normality.test(modele)$jb.mul$JB
#Vérification de l'inversibilité de la matrice

```



```

calculCO(modele)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele)

#Prédictions
PredSMIC<-forecast(modele, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredCOMPLET*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredSMIC, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#TCHOF
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARselect(cbind(MSESta, TCHOFSta), lag.max=10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale et le taux de chômage
modele<-VAR(cbind(MSESta, TCHOFSta), p=3, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele)
#Test d'homoscédasticité
archTest(modele)
#Test de normalité
normality.test(modele)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculCO(modele)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele)

#Ordre 2
#Réalisation d'un modèle d'ordre 2 avec la masse salariale et le taux de chômage
modele2<-VAR(cbind(MSESta, PIBSta), p=2, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele2)
#Test d'homoscédasticité
archTest(modele2)
#Test de normalité
normality.test(modele2)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculCO(modele2)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele2)

#Prédictions avec le modèle d'ordre 3 et calcul de l'EQM

```

```

PredTCHOF<-forecast(modele, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
#Prédictions avec le modèle d'ordre 2 et calcul de l'EQM
PredTCHOF2<-forecast(modele2, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredTCHOF2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredTCHOF, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#PIB & SMIC
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARselect(cbind(MSESta, PIBSta, SMICSta), lag.max=10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale, le PIB et le SMIC
modele<-VAR(cbind(MSESta, PIBSta, SMICSta), p=3, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele)
#Test d'homoscédasticité
archTest(modele)
#Test de normalité
normality.test(modele)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculCO(modele)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele)

#Prédictions
PredPIBSMIC<-forecast(modele, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredPIBSMIC*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredPIBSMIC, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#PIB & TCHOF
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARselect(cbind(MSESta, PIBSta, TCHOFSta), lag.max=10)

#Ordre 4
#Réalisation d'un modèle d'ordre 4 avec la masse salariale et le taux de chômage et le PIB
modele<-VAR(cbind(MSESta, PIBSta, TCHOFSta), p=4, type="const")

```

```

#Vérification de la stabilité du modèle
stabilityvars(modele)
#Test d'homoscédasticité
archTest(modele)
#Test de normalité
normality.test(modele)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculC0(modele)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale et le taux de chômage et le PIB
modele2<-VAR(cbind(MSESta, PIBSta, TCHOFSta), p=3, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele2)
#Test d'homoscédasticité
archTest(modele2)
#Test de normalité
normality.test(modele2)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculC0(modele2)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele2)

#Prédictions
PredPIBTCHOF<-forecast(modele, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredPIBTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredPIBTCHOF2<-forecast(modele2, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredPIBTCHOF2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredPIBTCHOF, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#SMIC & TCHOF
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARselect(cbind(MSESta, SMICSta, TCHOFSta), lag.max=10)

#Ordre 4
#Réalisation d'un modèle d'ordre 4 avec la masse salariale et le taux de chômage et le SMIC
modele<-VAR(cbind(MSESta, SMICSta, TCHOFSta), p=4, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele)
#Test d'homoscédasticité

```

```

archTest(modele)
#Test de normalité
normality.test(modele)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculCO(modele)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele)

#Ordre3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale et le taux de chômage et le SMIC
modele2<-VAR(cbind(MSESta, SMICSta, TCHOFSta), p=3, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele2)
#Test d'homoscédasticité
archTest(modele2)
#Test de normalité
normality.test(modele2)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculCO(modele2)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele2)

#Prédictions
PredSMICTCHOF<-forecast(modele, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredSMICTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredSMICTCHOF2<-forecast(modele2, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredSMICTCHOF2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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", "2017Q2", "2017Q3", "2017Q4"), las=1)
lines(window(PredSMICTCHOF, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red", lty=2)

#Modèle complet
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARselect(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), lag.max=10)

#Ordre 4
#Réalisation d'un modèle d'ordre 4 avec toutes les variables
modele<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=4, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele)
#Test d'homoscédasticité
archTest(modele)
#Test de normalité
normality.test(modele)$jb.mul$JB

```

```

#Vérification de l'inversibilité de la matrice
calculC0(modele)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele)

#Ordre3
#Réalisation d'un modèle d'ordre 3 avec toutes les variables
modele2<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSa), p=3, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele2)
#Test d'homoscédasticité
archTest(modele2)
#Test de normalité
normality.test(modele2)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculC0(modele2)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele2)

#Ordre 2
#Réalisation d'un modèle d'ordre 2 avec toutes les variables
modele3<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSa), p=2, type="const")
#Vérification de la stabilité du modèle
stabilityvars(modele3)
#Test d'homoscédasticité
archTest(modele3)
#Test de normalité
normality.test(modele3)$jb.mul$JB
#Vérification de l'inversibilité de la matrice
calculC0(modele3)
#Test d'auto-corrélation et de corrélation croisée
Portmanteau(modele3)

#Prédictions
PredCOMPLET<-forecast(modele, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredCOMPLET*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredCOMPLET2<-forecast(modele2, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredCOMPLET2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredCOMPLET3<-forecast(modele3, h=8)$forecast$MSESta$mean
EQM(MSETest, window(PredCOMPLET3*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredCOMPLET, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

```

```

#Comparaison des résultats
resultats<-matrix(nrow=7, ncol=1, dimnames = list(c("PIB", "SMIC", "TCHOF", "PIB & SMIC", "PIB
                                                    c("EQM"))))

resultats[1,1] = EQM(MSETest, window(PredPIB*MSETrendTest*MSESeasonalTest, start=2016, end=c(20
resultats[2,1] = EQM(MSETest, window(PredSMIC*MSETrendTest*MSESeasonalTest, start=2016, end=c(
resultats[3,1] = EQM(MSETest, window(PredTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end=c
resultats[4,1] = EQM(MSETest, window(PredPIBSMIC*MSETrendTest*MSESeasonalTest, start=2016, end
resultats[5,1] = EQM(MSETest, window(PredPIBTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end
resultats[6,1] = EQM(MSETest, window(PredSMICTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end
resultats[7,1] = EQM(MSETest, window(PredCOMPLET*MSETrendTest*MSESeasonalTest, start=2016, end
resultats

#####

##### Modèles VAR avec MTS #####

require(MTS)

#PIB
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARorder(cbind(MSESta, PIBSta), maxp = 10)

#Ordre 4
#Réalisation d'un modèle d'ordre 4 avec la masse salariale et le PIB
modele<-VAR(cbind(MSESta, PIBSta), p=4)
#Vérification de la stabilité du modèle
stabilityMTS(modele)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele)
#test du Portmanteau
mq(modele$residuals, lag=10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale et le PIB
modele2<-VAR(cbind(MSESta, PIBSta), p=3)
#Vérification de la stabilité du modèle
stabilityMTS(modele2)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele2)
#test du Portmanteau
mq(modele2$residuals, lag=10)

#Prédictions
PredPIB<-window(ts(VARpred(modele, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredPIB*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredPIB2<-window(ts(VARpred(modele2, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredPIB2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

```

```

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredPIB, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#SMIC
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARorder(cbind(MSESta, SMICSta), maxp = 10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale et le SMIC
modele<-VAR(cbind(MSESta, SMICSta), p=3)
#Vérification de la stabilité du modèle
stabilityMTS(modele)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele)
#test du Portmanteau
mq(modele$residuals, lag=10)

#Prédictions
PredSMIC<-window(ts(VARpred(modele, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredSMIC*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredPIB, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#TCHOF
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARorder(cbind(MSESta, TCHOFSta), maxp = 10)

#Ordre 6
#Réalisation d'un modèle d'ordre 6 avec la masse salariale et le taux de chômage
modele<-VAR(cbind(MSESta, TCHOFSta), p=6)
#Vérification de la stabilité du modèle
stabilityMTS(modele)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele)
#test du Portmanteau
mq(modele$residuals, lag=10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale et le taux de chômage
modele2<-VAR(cbind(MSESta, TCHOFSta), p=3)

```



```

#Vérification de la stabilité du modèle
stabilityMTS(modele2)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele2)
#test du Portmanteau
mq(modele2$residuals, lag=10)

#Ordre 2
#Réalisation d'un modèle d'ordre 2 avec la masse salariale et le taux de chômage
modele3<-VAR(cbind(MSESta, TCHOFSta), p=2)
#Vérification de la stabilité du modèle
stabilityMTS(modele3)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele3)
#test du Portmanteau
mq(modele2$residuals, lag=10)

#Prédictions
PredTCHOF<-window(ts(VARpred(modele, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredTCHOF2<-window(ts(VARpred(modele2, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredTCHOF2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredTCHOF3<-window(ts(VARpred(modele3, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredTCHOF3*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredTCHOF, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#PIB & SMIC
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARorder(cbind(MSESta, PIBSta, SMICSta), maxp = 10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale, le PIB et le SMIC
modele<-VAR(cbind(MSESta, PIBSta, SMICSta), p=3)
#Vérification de la stabilité du modèle
stabilityMTS(modele)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele)
#test du Portmanteau
mq(modele$residuals, lag=10)

#Prédictions
PredPIBSMIC<-window(ts(VARpred(modele, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)

```



```

EQM(MSETest, window(PredPIBSMIC*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredPIBSMIC, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#PIB & TCHOF
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARorder(cbind(MSESta, PIBSta, TCHOFSta), maxp = 10)

#Ordre 4
#Réalisation d'un modèle d'ordre 4 avec la masse salariale, le PIB et le taux de chômage
modele<-VAR(cbind(MSESta, PIBSta, TCHOFSta), p=4)
#Vérification de la stabilité du modèle
stabilityMTS(modele)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele)
#test du Portmanteau
mq(modele$residuals, lag=10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale, le PIB et le taux de chômage
modele2<-VAR(cbind(MSESta, PIBSta, TCHOFSta), p=3)
#Vérification de la stabilité du modèle
stabilityMTS(modele2)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele2)
#test du Portmanteau
mq(modele2$residuals, lag=10)

#Ordre 2
#Réalisation d'un modèle d'ordre 2 avec la masse salariale, le PIB et le taux de chômage
modele3<-VAR(cbind(MSESta, PIBSta, TCHOFSta), p=2)
#Vérification de la stabilité du modèle
stabilityMTS(modele3)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele3)
#test du Portmanteau
mq(modele3$residuals, lag=10)

#Prédictions
PredPIBTCHOF<-window(ts(VARpred(modele, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016, end=c(2017,2))
EQM(MSETest, window(PredPIBTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredPIBTCHOF2<-window(ts(VARpred(modele2, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016, end=c(2017,2))
EQM(MSETest, window(PredPIBTCHOF2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

```

```

PredPIBTCHOF3<-window(ts(VARpred(modele3, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016, end=c(2017,2))
EQM(MSETest, window(PredPIBTCHOF3*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredPIB, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#SMIC & TCHOF
#Selection de l'ordre du modèle VAR en fonction de différents critères
VARorder(cbind(MSESta, SMICSta, TCHOFSta), maxp = 10)

#Ordre 4
#Réalisation d'un modèle d'ordre 4 avec la masse salariale, le SMIC et le taux de chômage
modele<-VAR(cbind(MSESta, SMICSta, TCHOFSta), p=4)
#Vérification de la stabilité du modèle
stabilityMTS(modele)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele)
#test du Portmanteau
mq(modele$residuals, lag=10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec la masse salariale, le SMIC et le taux de chômage
modele2<-VAR(cbind(MSESta, SMICSta, TCHOFSta), p=3)
#Vérification de la stabilité du modèle
stabilityMTS(modele2)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele2)
#test du Portmanteau
mq(modele2$residuals, lag=10)

#Prédictions
PredSMICTCHOF<-window(ts(VARpred(modele, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016, end=c(2017,2))
EQM(MSETest, window(PredSMICTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredSMICTCHOF2<-window(ts(VARpred(modele2, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016, end=c(2017,2))
EQM(MSETest, window(PredSMICTCHOF2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredSMICTCHOF, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red")

#PIB, SMIC & TCHOF
#Selection de l'ordre du modèle VAR en fonction de différents critères

```

```

VARorder(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), maxp = 10)

#Ordre 4
#Réalisation d'un modèle d'ordre 4 avec toutes les variables
modele<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=4)
#Vérification de la stabilité du modèle
stabilityMTS(modele)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele)
#test du Portmanteau
mq(modele$residuals, lag=10)

#Ordre 3
#Réalisation d'un modèle d'ordre 3 avec toutes les variables
modele2<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=3)
#Vérification de la stabilité du modèle
stabilityMTS(modele2)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele2)
#test du Portmanteau
mq(modele2$residuals, lag=10)

#Ordre 2
#Réalisation d'un modèle d'ordre 2 avec toutes les variables
modele3<-VAR(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=2)
#Vérification de la stabilité du modèle
stabilityMTS(modele3)
#Test d'auto-corrélation et de corrélation croisée
crossCorr(modele3)
#test du Portmanteau
mq(modele3$residuals, lag=10)

#Prédictions
PredCOMPLET<-window(ts(VARpred(modele, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredCOMPLET*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredCOMPLET2<-window(ts(VARpred(modele2, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredCOMPLET2*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))
PredCOMPLET3<-window(ts(VARpred(modele3, 8)$pred[,1], start=c(2015,3), frequency=4), start=2016)
EQM(MSETest, window(PredCOMPLET3*MSETrendTest*MSESeasonalTest, start=2016, end=c(2017,2)))

#Affichage des valeurs de l'échantillon de test et comparaison avec les prédictions
plot(MSETest, xlim=c(2016,2017.25), main="Différences entre les véritables
      valeurs 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"))
lines(window(PredCOMPLET, start=2016, end=c(2017,2))*MSETrendTest*MSESeasonalTest, col = "red", lty=2)

#Comparaison des résultats

```

```

resultats<-matrix(nrow=7, ncol=1, dimnames = list(c("PIB", "SMIC", "TCHOF", "PIB & SMIC", "PIB
resultats[1,1] = EQM(MSETest, window(PredPIB*MSETrendTest*MSESeasonalTest, start=2016, end=c(20
resultats[2,1] = EQM(MSETest, window(PredSMIC*MSETrendTest*MSESeasonalTest, start=2016, end=c(
resultats[3,1] = EQM(MSETest, window(PredTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end=c
resultats[4,1] = EQM(MSETest, window(PredPIBSMIC*MSETrendTest*MSESeasonalTest, start=2016, end
resultats[5,1] = EQM(MSETest, window(PredPIBTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end
resultats[6,1] = EQM(MSETest, window(PredSMICTCHOF*MSETrendTest*MSESeasonalTest, start=2016, end
resultats[7,1] = EQM(MSETest, window(PredCOMPLET*MSETrendTest*MSESeasonalTest, start=2016, end
resultats

#####

##### Modèles VARMA avec MTS #####

#Estimation de l'ordre du modèle
Eccm(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), maxq=5)
#Réalisation d'un modèle d'ordre p=3 et q=1 avec toutes les variables
modele<-VARMA(cbind(MSESta, PIBSta, SMICSta, TCHOFSta), p=3, q=1)
#Elimination des paramètres non significatifs du modèle
modele<-refVARMA(modele)

#####

```

References

- Devignes, Nicolas. 2018. *Prevision de La Masse Salariale En Presence d'informations Auxiliaires*.
- Dufour, Jean-Marie, and Denis Pelletier. 2005. *Practical Methods for Modelling Weak Varma Processes: Identification, Estimation and Specification with a Macroeconomic Application*. Département de sciences économiques; CIREQ, Université de Montréal, Montréal, Canada.
- Hyndman, J., Rob, and Yeasmin Khandakar. 2008. “Automatic Time Series Forecasting: The Forecast Package for R.” *Journal of Statistic Software*, July.
- Pfaff, Bernhard. 2018. *VAR, Svar and Svec Models: Implementation Within R Package Vars*.
- Tsay, Ruey S., and David Wood. 2002. *Analysis of Financial Time Series*. Wiley-Science Interproduction.