

## Séries Temporelles –TP 6 : Processus SARIMA sous R

### Durée 4 Heures

#### Consignes

Ce TP doit être réalisé à l'aide du logiciel R sous Linux. A l'aide de l'éditeur de texte d'Open Office, créer un fichier (Nom\_Prénom) dans lequel vous répondrez aux questions du TP. Pour chaque question, vous incluez dans votre compte-rendu les codes R, les graphiques, les résultats, leurs interprétations. **En fin de TP, vous exporterez votre fichier en document pdf.** Seul le fichier pdf et la base de données augmentée des colonnes supplémentaires créées lors du TP seront pris en compte pour la correction. Une partie de la **note portera** sur la **clarté** de votre compte-rendu (**rédaction, propreté, lisibilité**)

Connectez vous sur votre compte Polytech et lancez Rstudio avec la commande :  
`/usr/local/rstudio/bin/rstudio/`

#### Exercice 1 – Simulation et identification de processus ARMA

On se servira des lignes de codes données en annexe.

- 1) Processus AR(p) :
  - a. Simuler 1000 valeurs d'un processus AR(1) avec un coefficient  $a_1 = 0.8$  :
 
$$X_t = 0.8X_{t-1} + \varepsilon_t$$
    - i. Représenter graphiquement les valeurs simulées
    - ii. Tracer l'ACF et le PACF et analyser
    - iii. Enregistrer les données dans un fichier.
  - b. Reprendre la question 1a) avec  $a_1 = -0.8$  ; et  $a_1 = 1.2$
  - c. Reprendre la question 1a) pour un processus AR(2) avec  $a_1 = 0$  ;  $a_2 = -0.9$
- 2) Processus MA(q)
  - a. Simuler 1000 valeurs d'un processus MA(1) avec un coefficient  $b_1 = 0.8$ 
    - i. Représenter graphiquement les valeurs simulées
    - ii. Tracer l'ACF et le PACF et analyser
    - iii. Enregistrer les données dans un fichier.
  - b. Reprendre la question 2a) avec  $b_1 = -0.8$
  - c. Reprendre la question 2a) pour un processus MA(2) avec  $b_1 = 0$  ;  $b_1 = -0.9$
- 3) Processus ARMA(p,q)
  - a. Simuler 1000 valeurs d'un processus ARMA(1,1) avec un coefficient  $a_1 = 0.8$  et  $b_1 = 0.8$ 
    - i. Représenter graphiquement les valeurs simulées
    - ii. Tracer l'ACF et le PACF et analyser
    - iii. Enregistrer les données dans un fichier.
  - b. Reprendre la question 3a) pour un processus ARMA(2,1) avec  $a_1 = 0$  ;  $a_2 = 0.8$  ;  $b_1 = 0.8$

#### Exercice 2 – DonnéeMasseSalariale

Le fichier ST-data-salaire.csv, qui vous a été envoyé, représente la masse salariale trimestrielle de 1990 à 2010 (soit 84 semestres) des employés (MSE) d'une maison d'assurance. L'objectif est de prédire la masse salariale des quatre trimestres de la dernière année, en utilisant les données des années 1990 à 2009.

#### Modélisation à l'aide d'un processus SARIMA

Dans cette partie, on cherche à modéliser la série observée pour les 80 premiers trimestres (1990 à 2009).

- 1) Représenter graphiquement les 80 premiers trimestres de la série, et donner une interprétation qualitative. (tendance, saisonnalité,...).
- 2) Calculer et représenter les fonctions ACF et PACF.
- 3) La série étudiée est-elle stationnaire ? Pourquoi ? Dites comment peut-on la rendre stationnaire ? (il ne s'agit pas de faire de calculs mais d'expliquer votre démarche).

- 4) Désaisonnaliser la série et modélisation de l'erreur  
**mse** = **decompose**(mse, type = "multiplicative")  
*#Décomposition de la série en tendance, saisonnalité et erreur.*  
*#(Le paramètre « type » peut prendre la valeur "additive" en cas de saisonnalité additive)*  
 En plus de la série d'origine « **mse** », on obtient, comme dans SPSS, trois autres séries : les coefficients saisonniers : « **mse\$seasonal** », la tendance : « **mse\$trend** » et l'erreur (la partie aléatoire) : « **mse\$random** »
  - a. Représenter graphiquement les quatre séries : **plot(mse)**
  - b. Tracer les fonctions d'auto-corrélations (ACF) et d'auto-corrélations partielles (PACF) empiriques de l'erreur : la partie aléatoire avec un retard de longueur 40.
  - c. En déduire que un ARIMA(3,0,1) peut être utilisé pour modéliser l'erreur.
  - d. Estimer les paramètres de ce modèle et tester si les résidus obtenus sont indépendants identiquement distribués.
  - e. Le modèle expert de SPSS pour modéliser l'erreur est de type SARIMA(0,0,1)(1,0,1). Estimer et afficher les paramètres de ce dernier modèle.
  - f. Calculer le BIC des deux modèles et décider lequel choisir.
  - g. Estimer la tendance par régression linéaire
- 5) Modélisation
  - a. D'après les résultats précédents, on propose de modéliser la MSE avec un modèle saisonnier SARIMA(0,1,1)(1,0,1) de période 4. Estimer les paramètres de ce modèle et analyser.  
**mse11 = arima(mse, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 4))**
  - b. Estimer les paramètres d'un lissage multiplicatif de Holt-Winters et analyser les paramètres associés.

### Prévision

L'objectif est de prédire la masse salariale des quatre trimestres de la dernière année, en utilisant les données des années 1990 à 2009.

- 1) Effectuer la prévision à l'aide du lissage multiplicatif de Winters (mettre dans l'options de SPSS : année=2010 et trimestre=4). Enregistrer les prévisions, leur intervalle de confiance et les résidus.
- 2) Effectuer la prévision à l'aide du modèle saisonnier SARIMA(0,1,1)(0,1,1) de SPSS.
- 3) Représenter alors sur un même graphique la série complète ainsi que les prédictions de la dernière année par lissage exponentiel et SARIMA.
- 4) Comment comparer les deux résultats ? Faites un zoom sur les dernières années.
- 5) Conclure.

### Exercice 3 – Données AirPassengers

On étudie la série temporelle du nombre de passagers par mois (en milliers) dans les transports aériens, de 1949 à 1960. Cette série est disponible sous R(AirPassengers).

- 1) Estimation paramétrique de la tendance
  - a. Représenter graphiquement la série. Cette série vous-t-elle stationnaire ? Présente-t-elle des tendances et saisonnalités ? Expliquez. Attention, il faut créer une série ts avant.
  - b. Estimer les paramètres d'une tendance linéaire de la forme  $\beta_0 + \beta_1 t$
  - c. Supprimer cette tendance et représenter graphiquement la série ainsi obtenue. Vérifier que la série des résidus est de moyenne nulle.
  - d. Calculer et représenter l'auto-corrélation et l'auto-corrélation partielle de la série des résidus.
  - e. Tester si la série des résidus est indépendante et identiquement distribuée.
- 2) Désaisonnaliser la série **AirPassengers**
  - a. Analyser et modéliser la tendance et comparer avec la question 1)
  - b. Analyser et modéliser la partie aléatoire : la série des résidus.
  - c. Estimer les paramètres du modèle des résidus
  - d. En déduire un modèle SARIMA pour **AirPassengers**
- 3) Modélisation
  - a. Déduire des questions 1 et 2 un modèle SARIMA pour **AirPassengers**
  - b. Proposer un lissage exponentiel approprié pour **AirPassengers**
- 4) Prévision de l'année 1960
  - a. Estimer un modèle SARIMA en utilisant les années 1949 à 1959

- b. Prévoir 1960 à l'aide du modèle SARIMA de la question 4a).
- c. Prévoir 1960 à l'aide d'un lissage exponentiel approprié pour **AirPassengers**
- d. Quelle est la meilleure prévision ? Justifier et motiver votre réponse

#### Exercice 4 : Données Varicelle

Analyser et modéliser les données varicelle.

#### ANNEXE : SIMULATION DE PROCESSUS ARMA SOUS R

##### Initialisation de paramètres

```
#seed pour aléatoire
set.seed(10)
#longueur de la série
n = 1000
#dossier où enregistrer les séries
chemin = "~/Documents/TPGIS/Séries Temporelles/TP4/"
```

##### Simulation de processus Autorégressifs

```
### simulation AR(1) avec a = 0.8
x = arima.sim(list(ar=c(0.8)), n)
#représentation graphique
plot(x, main="AR(1) avec a = c(0.8)")
#autocorrélation
acx = acf(x, lag.max=40, main="ACF d'un AR(1) avec a = c(0.8)")
#autocorrélation partielle
pacx = pacf(x, lag.max=40, main="PACF d'un AR(1) avec a = c(0.8)")
#enregistrement de la série
write.table(x, file=paste0(chemin, "/ar1.txt"), row.names = FALSE, col.names = FALSE)
### simulation AR(2) avec a = c(0,0.9)
x = arima.sim(list(ar=c(0,0.9)), n)
#représentation graphique
plot(x, main="AR(2) avec a = c(0,0.9)")
#autocorrélation
acx = acf(x, lag.max=40, main="ACF d'un AR(2) avec a = c(0,0.9)")
#autocorrélation partielle
pacx = pacf(x, lag.max=40, main="PACF d'un AR(2) avec a = c(0,0.9)")
#enregistrement de la série
write.table(x, file=paste0(chemin, "/ar2.txt"), row.names = FALSE, col.names = FALSE)
```

##### Simulation de processus moyennes mobiles

```
### simulation MA(1) avec b = c(0.8)
x = arima.sim(list(ma=c(0.8)), n)
#représentation graphique
plot(x, main="MA(1) avec b = c(0.8)")
#autocorrélation
acx = acf(x, lag.max=40, main="ACF d'un MA(1) avec b = c(0.8)")
#autocorrélation partielle
pacx = pacf(x, lag.max=40, main="PACF d'un MA(1) avec b = c(0.8)")
#enregistrement de la série
write.table(x, file=paste0(chemin, "/ma1.txt"), row.names = FALSE, col.names = FALSE)
### simulation MA(2) avec b = c(0,0.9)
x = arima.sim(list(ma=c(0,0.9)), n)
#représentation graphique
plot(x, main="MA(2) avec b = c(0,0.9)")
#autocorrélation
acx = acf(x, lag.max=40, main="ACF d'un MA(2) avec b = c(0,0.9)")
#autocorrélation partielle
pacx = pacf(x, lag.max=40, main="PACF d'un MA(2) avec b = c(0,0.9)")
#enregistrement de la série
write.table(x, file=paste0(chemin, "/ma2.txt"), row.names = FALSE, col.names = FALSE)
```

### Simulation de processus mixte ARMA

```
### simulation ARMA(1,1) avec a=c(0.8) et b = c(0.8)
x = arima.sim(list(ar=c(0.8),ma=c(0.8)), n)
#représentation graphique
plot(x, main="ARMA(1,1) avec a=c(0.8) et b = c(0.8)")
#autocorrélation
acx = acf(x, lag.max=40, main="ACF d'un ARMA(1,1) avec a=c(0.8) et b = c(0.8)")
#autocorrélation partielle
pacx = pacf(x, lag.max=40, main="PACF d'un ARMA(1,1) avec a=c(0.8) et b = c(0.8)")
#autocorrélation partielle
pacx = pacf(x, lag.max=40, main="PACF d'un ARMA(1,1) avec a=c(0.8) et b = c(0.8)")
#enregistrement de la série
write.table(x, file=paste0(chemin,"/arma11.txt"), row.names = FALSE, col.names = FALSE)
```

### Lignes de codes R pour analyser un processus SARIMA : Application à la masse salariale

Importation des données :

```
# changement environnement de travail
setwd("~/Documents/TPGIS/Séries Temporelles/TP6/")
# Importation des données
donnees = read.table("ST-data-salaire.csv", header = TRUE)
```

Modélisation : Nous créons un objet ts en enlevant les 4 derniers trimestres.

```
# création d'un objet ts
# l'option start contient l'année et le trimestre de départ
# l'option frequency définit le fait de travailler avec des trimestres
```

```
mse = ts(donnees[1:80,1], start = c(1990, 1), frequency = 4)
```

*# Aperçu des données*

```
print(mse)
```

Affichage des données et fonctions ACF et PACF

```
plot(mse, main = "masse salariale trimestrielle de 1990 à 2009")
```

*# fonction d'autocorrélation avec un lag de 40*

```
acf(mse, lag.max = 40, main = "ACF de mse")
```

*# fonction d'autocorrélation partielle avec un lag de 40*

```
pacf(mse, lag.max = 40, main = "PACF de mse")
```

Désaisonnalisation

*# Décomposition de la série en tendance, saisonnalité et erreur.*

*# Le paramètre type peut prendre la valeur "additive" en cas de saisonnalité additive.*

```
mseosais = decompose(mse, type = "multiplicative")
```

*# la série d'origine*

```
mseosais$x
```

*# les coefficients saisonniers*

```
mseosais$seasonal
```

*# la tendance*

```
mseosais$trend
```

*# la partie aléatoire de la série (sans tendance ni saisonnalité).*

*# On notera la présence de NA.*

```
mseosais$random
```

*# représentation graphique des quantités présentées ci-dessus*

```
plot(mseosais)
```

*# fonction d'autocorrélation avec un lag de 40.*

*# L'option na.action = na.pass permet de gérer les valeurs manquantes.*

```
acf(mseosais$random, lag.max = 40, main = "ACF de mse après décomposition saisonnière",
na.action = na.pass)
```

*# fonction d'autocorrélation partielle avec un lag de 40*

```
pacf(mseosais$random, lag.max = 40, main = "PACF de mse après décomposition saisonnière",
na.action = na.pass)
```

*# Estimation des paramètres d'un ARIMA(3,0,1).*

*# Le paramètre seasonal permet de spécifier l'ordre la partie saisonnière d'un SARIMA*

*# ainsi que sa période.*

```
mse31 = arima(msenosais$random, order = c(3, 0, 1),
seasonal = list(order = c(0,0,0), period = NA))
#aperçu des coefficients
print(mse31)
Testons si les résidus sont iid avec le test de Ljung-Box.
# Test de Ljung-Box
# lag = 20 spécifie que la statistique de test est calculée à partir
# des 20 premières autocorrélations.
Box.test(mse31$residuals, lag = 20)
SPSS propose un SARIMA(0,0,1)(1,0,1) de période 4.
# Estimation d'un SARIMA(0,0,1)(1,0,1) de période 4.
mse01 = arima(msenosais$random, order = c(0, 0, 1),
seasonal = list(order = c(1, 0, 1), period = 4))
# Aperçu des coefficients
print(mse01)
Afin de comparer les 2 modèles, nous allons utiliser le BIC (plus la valeur est faible, mieux c'est).
#BIC du ARIMA(3,0,1)
BIC(mse31)
# BIC du SARIMA(0,0,1)(1,0,1)
BIC(mse01)
Estimation de la tendance par régression linéaire
# Régression linéaire
mselm = lm(mse ~ donnees[1:80,2])
# Aperçu des résultats
summary(mselm)
Modélisation
# Estimation d'un SARIMA(0,1,1)(1,0,1) de période 4.
#mse11 = arima(mse, order = c(0, 1, 1),
# seasonal = list(order = c(1, 0, 1), period = 4))
La partie AR saisonnière n'est pas stationnaire, R ne peut donc pas estimer les paramètres. Le modélisateur
SPSS propose lui un SARIMA(0,1,1)(0,1,1)
# Estimation d'un SARIMA(0,1,1)(0,1,1) de période 4.
mse11 = arima(mse, order = c(0, 1, 1),
seasonal = list(order = c(0, 1, 1), period = 4))

Prévision à l'aide d'un lissage (multiplicatif) de Holt-Winter
mseHW = HoltWinters(mse, alpha = NULL, beta = NULL, gamma = NULL,
seasonal = "multiplicative")
plot(mseHW)
# On prédit 4 nouveaux trimestres à l'aide du modèle SARIMA
predmse11 = predict(mse11,4)
predmseHW = predict(mseHW,4)
# représentation graphique
plot(ts(donnees[,1], start = c(1990, 1), frequency = 4), ylab = "MSE")
lines(ts(predmse11$pred, start = c(2010, 1), frequency = 4), col="red")
lines(ts(predmseHW, start = c(2010, 1), frequency = 4), col="blue")

# zoom sur les dernières années
plot(ts(donnees[65:84,1], start = c(2006, 1), frequency = 4),
ylim = c(min(donnees[65:84,1]), max(predmseHW)), ylab = "MSE")
lines(ts(predmse11$pred, start = c(2010, 1), frequency = 4), col="red")
lines(ts(predmseHW, start = c(2010, 1), frequency = 4), col="blue")
legend("bottomright", c("SARIMA", "HW"), col=c("red", "blue"), lty=1, cex=0.7)
```