ENSAE 2A

Séries temporelles linéaires

TD n°4

Pour toute remarque, contacter `jerome.trinh@ensae.fr`

L'objectif de cette séance est de mettre en pratique les méthodes habituelles de traitement des séries temporelles univariées. Il s'agit en particulier de mettre en œuvre l'identification, l'estimation et la sélection d'un modèle pour une série brute donnée.

Remarque : Pour des raisons d'encodage texte, copier-coller les lignes de code peut ne pas donner le résultat voulu. Il faut bien vérifier ce qu'il en sort.

Q1. Ouvrir R et importer la séries "Donnees1.csv". On considèrera dans la suite xm la série privée de ses 4 dernières observations.

```
datafile <- "Donnees1.csv"
data <- read.csv(datafile,sep=";")
```

Nous utiliserons les packages "zoo" qui formalise les séries temporelles de manières pratiques, et "tseries" pour diverses fonctions sur les séries temporelles.

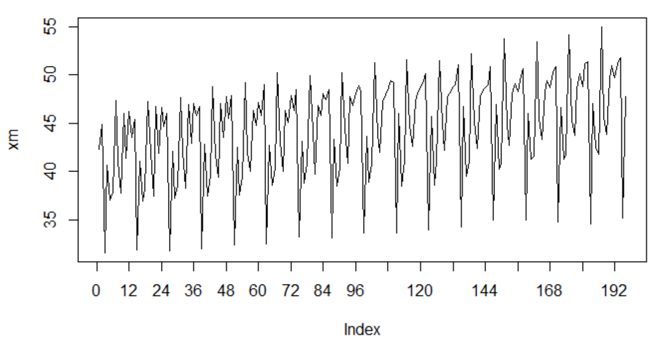
```
require(zoo)
require(tseries)
```

On définit $xm.source$ la série initiale au format "zoo", T sa longueur et xm la série privée de ses 4 dernières observations

```
xm.source <- zoo(data[[1]]) #convertit le 1er élément de data en "zoo"
T <- length(xm.source)
xm <- xm.source[1:(T-4)] #supprime les 4 dernières valeurs
```

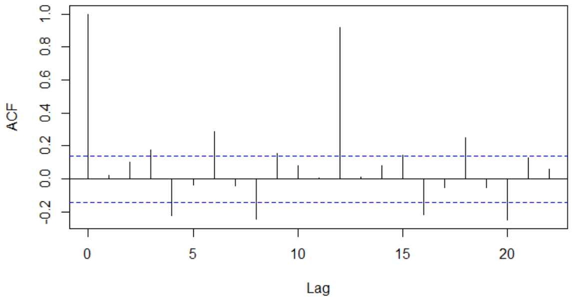
Q2. Représenter graphiquement la série xm. Qu'observe-t-on ? Comment peut-on résoudre le problème de saisonnalité de xm ? On note dans la suite $desaison$ la série obtenue par désaisonnalisation de xm, et on supposera que la $desaison$ suit un ARIMA(p, d, q).

```
plot(xm, xaxt="n") #représente xm sans l'axe des abcisses
axis(side=1,at=seq(0,196,12)) #affiche l'axe des abcisses par 12 de 0 à 196
```



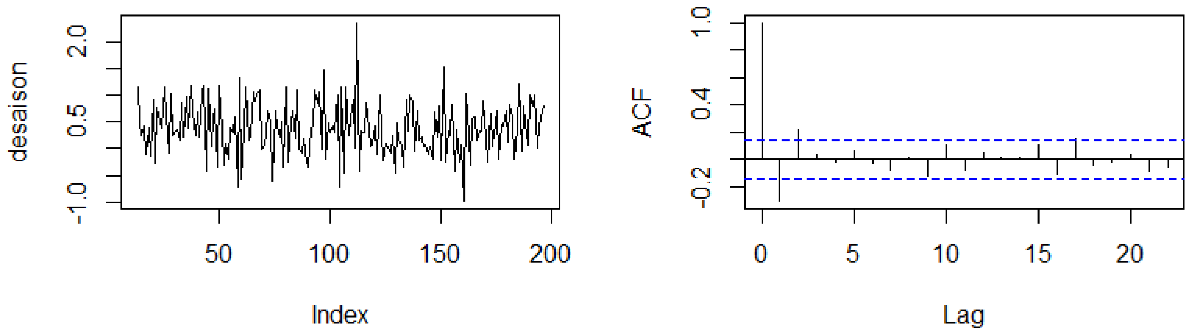
On suspecte une saisonnalité de périodicité 12, ce que l'on peut confirmer en représentant les fonctions d'auto-corrélation totale de la série.

```
acf(xm) #trace l'autocorrélogramme total
```



On observe effectivement une forte autocorrélation d'ordre 12. On peut tenter de supprimer cette saisonnalité en corrigeant la série par son 12ème retard.

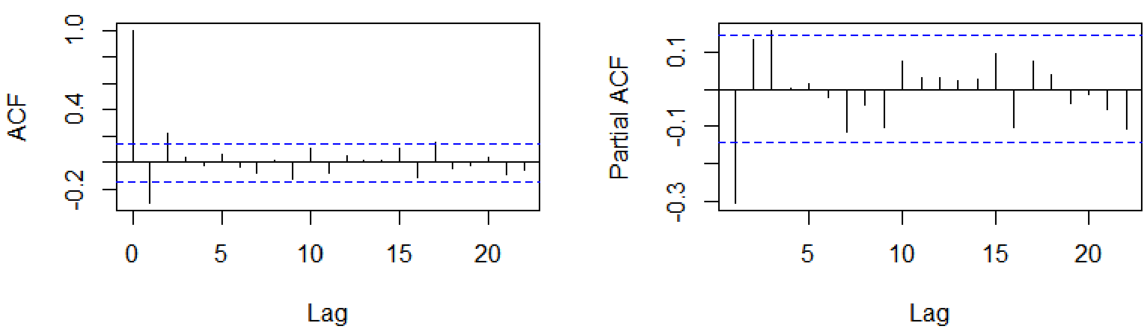
```
desaison <- xm-lag(xm,-12) #retire le 12ème retard
par(mfrow=c(1,2)) #dispose les graphiques en 1 colonne et 2 lignes
plot(desaison)
acf(desaison)
```



La saisonnalité semble avoir été corrigée.

Q3. Etudier les auto-corrélogrammes de la série $desaison$. A priori, est-elle intégrée ?

```
acf(desaison) #trace les fonctions d'autocorrélation totale
pacf(desaison) #trace les fonctions d'autocorrélation partielle
```



L'autocorrélation d'ordre 1 (totale ou partielle, c'est la même chose) est d'environ -0.3, soit petite et loin d'être égales à 1. La série semble donc stationnaire.

Q4. Effectuer le test de racine unitaire vous semblant le plus adapté à la série $desaison$. Ce test confirme-t-il les conclusions précédentes ?

La représentation de $desaison$ à la question 2 semble nous indiquer que la série ne présente pas de tendance. Nous pouvons donc effectuer par exemple comme test de racine unitaire :

— un test de Dickey-Fuller augmenté (ADF) dans le cas avec constante non nulle et sans tendance

— un test de Phillips-Perron intégrant une constante dans l'équation de régression

Pour ce TD, nous regarderons le test de Phillips-Perron.

```
pp.test(desaison) #pp test sur la série (cas général : avec constante et tendance)
```

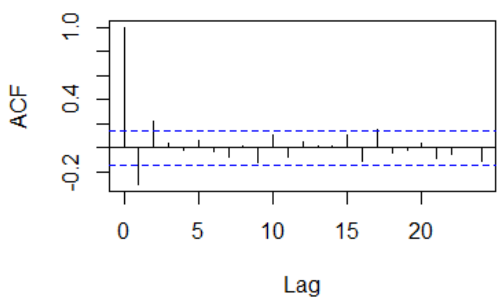
```
        Phillips-Perron Unit Root Test

data:  desaison
Dickey-Fuller Z(alpha) = -268.96, Truncation lag parameter = 4, p-value = 0.01
alternative hypothesis: stationary
```

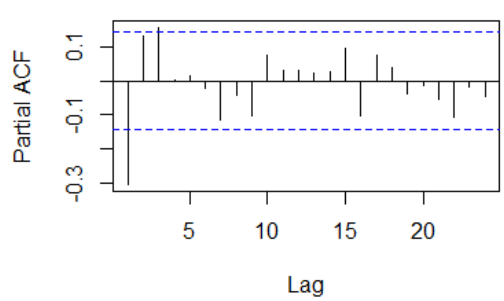
L'hypothèse nulle de racine unitaire est rejetée à un seuil de 95% (p-value=0.01¡0.05), on dira que la série est stationnaire.

Q5. Proposer les ordres maximums p^*, d^*, q^* vraisemblables pour la série $desaison$. Vérifier que le modèle ARIMA(p^*,d^*,q^*) correspondant est valide.

Soit y la série centrée. On regarde ensuite les autocorrélogrammes total et partiel de y, jusqu'à un horizon raisonnable (deux périodicités = 24).

```
y <- desaison - mean(desaison) #centre la série
acf(y,24); pacf(y,24) # le point virgule valide la commande comme un saut de ligne
```





Comme la série est stationnaire, elle est intégrée d'ordre $d = 0$.
Les fonctions d'autocorrélation totales sont significatives (i.e. plus grandes que les bornes $\pm 1,96/\sqrt{n}$ de l'intervalle de confiance d'un test de nullité de l'autocorrélation à un seuil de 95%) jusqu'à $q^* = 2$ et les autocorrélations partielles jusqu'à $p^* = 3$. Si y suit un ARIMA(p,d,q), elle suit donc au plus un ARIMA($p^* = 3$, $d^* = 0$, $q^* = 2$), que l'on peut estimer.

```
arima(y,c(3,0,2)) #régresse l'ARIMA(3,0,2) pour la série y
```

```
Call:
arima(x = y, order = c(3, 0, 2))

Coefficients:
         ar1     ar2     ar3     ma1      ma2  intercept
      -0.307  0.2091  0.1748  0.0178  -0.0390     0.0008
s.e.   0.542  0.2490  0.1604  0.5488   0.3081     0.0368

sigma^2 estimated as 0.2234:  log likelihood = -123.32,  aic = 260.64
```

Le modèle est valide si ses résidus ne sont pas autocorrélés. On peut le tester en utilisant par exemple le test de Ljung-Box de l'hypothèse nulle de nullité jointe des autocorrélations jusqu'à un ordre k donné (donc absence d'autocorrélation), dont la statistique de test :

$$Q(k) = n(n+2)\sum_{\ell=1}^{k}\frac{\hat{\rho}_\ell^2}{n-\ell}$$

suit une loi du χ_2 à $k - p - q$ degrés de liberté (simplement k degrés de libertés le test est effectué sur des données observables), où n est le nombre d'observations, k le nombre de retards à tester et $\hat{\rho}_\ell$ l'autocorrélation empirique d'ordre ℓ.
Par exemple, pour $k = 6$ (comme $p + q = 5$, on ne peut effectuer le test pour $k < 6$) :

```
arima302 <- arima(y,c(3,0,2)) #enregistre les résultats de l'estimation
Box.test(arima302$residuals, lag=6, type="Ljung-Box", fitdf=5) #test de Ljung-Box
```

```
        Box-Ljung test

data:  arima302$residuals
X-squared = 0.40607, df = 1, p-value = 0.524
```

L'hypothèse nulle n'est pas rejetée à un seuil de 95% (p-value>0.05), on dira donc que les résidus ne sont pas autocorrélés jusqu'à 6 retards. Pour nous assurer de l'absence d'autocorrélation, nous la testons pour deux périodicités, c'est-à-dire jusqu'à 24 retards ici.

```
Qtests <- function(series, k, fitdf=0) {
pvals <- apply(matrix(1:k), 1, FUN=function(l) {
pval <- if (l<=fitdf) NA else Box.test(series, lag=l, type="Ljung-Box", fitdf=fitdf)$p.value
return(c("lag"=l,"pval"=pval))
})
return(t(pvals))
}
Qtests(arima302$residuals, 24, 5) #tests de LB pour les ordres 1 à 24
```

```
      lag      pval
 [1,]   1        NA
 [2,]   2        NA
 [3,]   3        NA
 [4,]   4        NA
 [5,]   5        NA
 [6,]   6 0.5239728
 [7,]   7 0.2923972
 [8,]   8 0.3708128
 [9,]   9 0.3015618
[10,]  10 0.3096879
[11,]  11 0.4154275
[12,]  12 0.5294905
[13,]  13 0.6131866
[14,]  14 0.6179548
[15,]  15 0.6439982
[16,]  16 0.6600634
[17,]  17 0.4385724
[18,]  18 0.5197705
[19,]  19 0.5952434
[20,]  20 0.6660243
[21,]  21 0.5958363
[22,]  22 0.5016374
[23,]  23 0.5460007
[24,]  24 0.5417287
```

L'absence d'autocorrélation n'est jamais rejetée à 95% jusqu'à 24 retards. Le modèle est donc valide.

Q6. Quels sont les sous-modèles de l'ARIMA(p^*,d^*,q^*) possibles ? Comment peut-on choisir parmi ces sous-modèles ?

Les sous-modèles possibles sont les ARIMA(p,d,q) tels que $p \leq p^*$, $d \leq d^*$, $q \leq q^*$.
Dans l'absolu, on cherche un modèle :

- bien ajusté : les coefficients estimés (notamment les coefficients des ordres AR et MA les plus élevés) sont statistiquement siginificatifs
- valide : les résidus ne sont pas autocorrélés

Passons en revue chacun de ces modèles :

- $p = 3$, $q = 2$:
  Nous avons déjà montré que les résidus de l'ARIMA(3,0,2) ne sont pas autocorrélés, ce modèle est donc valide.
  Pour la significativité des coefficients, on peut vérifier qu'on a bien que le rapport entre le coefficient estimé et la variance estimée du coefficient estimé est bien supérieur à en valeur absolue à 1.96 (ou si la p-value correspondante est inférieure à 0.05) :

```
#fonction de test des significativités individuelles des coefficients
signif <- function(estim){
```

```
coef <- estim$coef
se <- sqrt(diag(estim$var.coef))
t <- coef/se
pval <- (1-pnorm(abs(t)))*2
return(rbind(coef,se,pval))
}
signif(arima302) #tests de siginificativité de l'ARIMA(3,0,2)
            ar1       ar2       ar3        ma1         ma2    intercept
coef -0.3069992 0.2090601 0.1747649 0.01781536 -0.03904574 0.0008088382
se    0.5420440 0.2489709 0.1603828 0.54880708  0.30813266 0.0368461190
pval  0.5711401 0.4010784 0.2758570 0.97410364  0.89916415 0.9824864146
```

Les coefficients des retards les plus élevés AR(3) et MA(2) ne rejettent chacun pas leur nullité à 95% (p-value>0.05), le modèle ARIMA(3,0,2) est donc mal ajusté.

On cherche à appliquer une procédure similaire aux sous-modèles ARIMA candidats.

```
##fonction d'affichage des tests pour la sélection du modèle ARIMA
arimafit <- function(estim){
adjust <- round(signif(estim),3)
pvals <- Qtests(estim$residuals,24,length(estim$coef)-1)
pvals <- matrix(apply(matrix(1:24,nrow=6),2,function(c) round(pvals[c,],3)),nrow=6)
colnames(pvals) <- rep(c("lag", "pval"),4)
cat("tests de nullité des coefficients :\n")
print(adjust)
cat("\n tests d'absence d'autocorrélation des résidus : \n")
print(pvals)
}
```

- $p = 1$, $q = 0$:

```
estim <- arima(y,c(1,0,0)); arimafit(estim)
tests de nullité des coefficients :
        ar1 intercept
coef -0.309    -0.002
se    0.071     0.027
pval  0.000     0.955

 tests d'absence d'autocorrélation des résidus :
     lag  pval lag  pval lag  pval lag  pval
[1,]   1    NA   7 0.082  13 0.209  19 0.205
[2,]   2 0.016   8 0.109  14 0.224  20 0.252
[3,]   3 0.015   9 0.077  15 0.203  21 0.184
[4,]   4 0.039  10 0.093  16 0.235  22 0.159
[5,]   5 0.064  11 0.127  17 0.128  23 0.182
[6,]   6 0.096  12 0.168  18 0.166  24 0.171
```

Le modèle est bien ajusté (le coefficient AR(1) est significatif) mais n'est pas valide (les résidus sont autocorrélés car Q(1) à Q(3) rejettent l'absence d'autocorrélation).

- $p = 2$, $q = 0$:

```
estim <- arima(y,c(2,0,0)); arimafit(estim)
tests de nullité des coefficients :
        ar1   ar2 intercept
coef -0.267 0.133    -0.001
se    0.074 0.073     0.031
pval  0.000 0.071     0.985

 tests d'absence d'autocorrélation des résidus :
     lag  pval lag  pval lag  pval lag  pval
[1,]   1    NA   7 0.230  13 0.469  19 0.393
[2,]   2    NA   8 0.295  14 0.481  20 0.444
[3,]   3 0.041   9 0.231  15 0.477  21 0.343
[4,]   4 0.116  10 0.246  16 0.485  22 0.329
[5,]   5 0.170  11 0.314  17 0.281  23 0.370
[6,]   6 0.247  12 0.396  18 0.344  24 0.363
```

Le modèle n'est pas bien ajusté (le coefficient AR(2) n'est pas significatif).

- $p = 3$, $q = 0$:

```
estim <- arima(y,c(3,0,0)); arimafit(estim)
```

```
 tests de nullité des coefficients :
          ar1   ar2   ar3 intercept
 coef -0.289 0.177 0.161     0.001
 se    0.073 0.075 0.073     0.037
 pval  0.000 0.019 0.028     0.983

  tests d'absence d'autocorrélation des résidus :
      lag  pval lag  pval lag  pval lag  pval
 [1,]   1    NA   7 0.653  13 0.790  19 0.737
 [2,]   2    NA   8 0.681  14 0.786  20 0.792
 [3,]   3    NA   9 0.567  15 0.799  21 0.730
 [4,]   4 0.697  10 0.545  16 0.804  22 0.644
 [5,]   5 0.872  11 0.640  17 0.602  23 0.682
 [6,]   6 0.922  12 0.732  18 0.674  24 0.676
```

Le modèle est bien ajusté (le coefficient AR(3) est significatif) et valide (l'absence d'autocorrélation n'est jamais rejetée). On le garde donc comme modèle candidat.

```
ar3 <- estim
```

- $p = 0$, $q = 1$:

```
estim <- arima(y,c(0,0,1)); arimafit(estim)
```

```
 tests de nullité des coefficients :
         ma1 intercept
 coef -0.222    -0.002
 se    0.059     0.028
 pval  0.000     0.957

  tests d'absence d'autocorrélation des résidus :
      lag  pval lag  pval lag  pval lag  pval
 [1,]   1    NA   7 0.024  13 0.075  19 0.073
 [2,]   2 0.001   8 0.039  14 0.093  20 0.097
 [3,]   3 0.003   9 0.025  15 0.077  21 0.072
 [4,]   4 0.007  10 0.029  16 0.087  22 0.068
 [5,]   5 0.014  11 0.040  17 0.040  23 0.085
 [6,]   6 0.024  12 0.055  18 0.055  24 0.073
```

Le modèle est bien ajusté (MA(1) est significatif) mais n'est pas valide (l'absence d'autocorrélation est rejetée par au moins un des tests).

- $p = 0$, $q = 2$:

```
estim <- arima(y,c(0,0,2)); arimafit(estim)
```

```
 tests de nullité des coefficients :
         ma1   ma2 intercept
 coef -0.297 0.254     0.000
 se    0.074 0.067     0.033
 pval  0.000 0.000     0.998

  tests d'absence d'autocorrélation des résidus :
      lag  pval lag  pval lag  pval lag  pval
 [1,]   1    NA   7 0.665  13 0.801  19 0.741
 [2,]   2    NA   8 0.701  14 0.800  20 0.790
 [3,]   3 0.629   9 0.604  15 0.803  21 0.715
 [4,]   4 0.860  10 0.579  16 0.803  22 0.659
 [5,]   5 0.769  11 0.662  17 0.614  23 0.694
 [6,]   6 0.846  12 0.746  18 0.684  24 0.692
```

Le modèle est bien ajusté (MA(2) est significatif) et valide (l'absence d'autocorrélation n'est jamais rejetée). On le garde donc comme modèle candidat.

```
ma2 <- estim
```

- $p = 1$, $q = 1$:

```
estim <- arima(y,c(1,0,1)); arimafit(estim)
```

```
tests de nullité des coefficients :
        ar1   ma1 intercept
coef -0.495 0.201    -0.001
se    0.146 0.157     0.029
pval  0.001 0.201     0.966

 tests d'absence d'autocorrélation des résidus :
     lag  pval lag  pval lag  pval lag  pval
[1,]   1    NA   7 0.106  13 0.280  19 0.255
[2,]   2    NA   8 0.144  14 0.295  20 0.305
[3,]   3 0.012   9 0.106  15 0.279  21 0.221
[4,]   4 0.043  10 0.125  16 0.307  22 0.203
[5,]   5 0.073  11 0.169  17 0.166  23 0.232
[6,]   6 0.114  12 0.225  18 0.213  24 0.223
```

Le modèle n'est pas bien ajusté (MA(1) non significatif).

- $p = 2$, $q = 1$:

```
estim <- arima(y,c(2,0,1)); arimafit(estim)
```

```
tests de nullité des coefficients :
       ar1   ar2    ma1 intercept
coef 0.202 0.302 -0.476     0.001
se   0.202 0.082  0.202     0.037
pval 0.317 0.000  0.018     0.985

 tests d'absence d'autocorrélation des résidus :
     lag  pval lag  pval lag  pval lag  pval
[1,]   1    NA   7 0.362  13 0.586  19 0.503
[2,]   2    NA   8 0.418  14 0.593  20 0.561
[3,]   3    NA   9 0.317  15 0.595  21 0.441
[4,]   4 0.221  10 0.329  16 0.603  22 0.398
[5,]   5 0.405  11 0.415  17 0.369  23 0.435
[6,]   6 0.493  12 0.510  18 0.442  24 0.428
```

Le modèle est bien ajusté (AR(2) et MA(1) significatifs) et valide (l'absence d'autocorrélation n'est jamais rejetée). On le garde donc comme modèle candidat.

```
ar2ma1 <- estim
```

- $p = 1$, $q = 2$:

```
estim <- arima(y,c(1,0,2)); arimafit(estim)
```

```
tests de nullité des coefficients :
       ar1    ma1   ma2 intercept
coef 0.338 -0.628 0.327     0.001
se   0.332  0.327 0.095     0.037
pval 0.308  0.055 0.001     0.987

 tests d'absence d'autocorrélation des résidus :
     lag  pval lag  pval lag  pval lag  pval
[1,]   1    NA   7 0.563  13 0.721  19 0.687
[2,]   2    NA   8 0.617  14 0.750  20 0.743
[3,]   3    NA   9 0.488  15 0.754  21 0.646
[4,]   4 0.571  10 0.466  16 0.746  22 0.569
[5,]   5 0.587  11 0.560  17 0.544  23 0.607
[6,]   6 0.749  12 0.650  18 0.619  24 0.604
```

Le modèle n'est pas bien ajusté (AR(2) non significatif).

- $p = 2$, $q = 2$:

```
tests de nullité des coefficients :
       ar1   ar2    ma1   ma2 intercept
coef 0.289 0.092 -0.583 0.246     0.001
se   0.268 0.177  0.266 0.176     0.037
pval 0.282 0.602  0.029 0.163     0.983

 tests d'absence d'autocorrélation des résidus :
     lag  pval lag  pval lag  pval lag  pval
[1,]   1    NA   7 0.433  13 0.669  19 0.635
[2,]   2    NA   8 0.498  14 0.691  20 0.696
[3,]   3    NA   9 0.396  15 0.703  21 0.591
[4,]   4    NA  10 0.387  16 0.702  22 0.515
[5,]   5 0.421  11 0.488  17 0.485  23 0.553
[6,]   6 0.636  12 0.590  18 0.563  24 0.552
```

Le modèle n'est pas bien ajusté (AR(2) et MA(2) non significatifs).

Nous avons donc 3 sous-modèles bien ajustés et valides : ARIMA(3,0,0), ARIMA(0,0,2) et ARIMA(2,0,1). Parmi ces derniers, on pourrait utiliser un critère d'information, tel que l'AIC ou le BIC (qui dépendent négativement de la log-vraisemblance du modèle pénalisée par le nombre de paramètres à estimer) :

```
models <- c("ar3","ma2","ar2ma1"); names(models) <- models
apply(as.matrix(models),1, function(m) c("AIC"=AIC(get(m)), "BIC"=BIC(get(m))))
         ar3      ma2   ar2ma1
AIC 260.6771 255.5774 258.6039
BIC 270.3219 268.4371 274.6786
```

L'ARIMA(0,0,2) minimise à la fois l'AIC et le BIC, il serait donc le meilleur selon ces critères.

Q7. Pour chacun de ces sous-modèles possibles, effectuez une prévision à 4 mois de la série $desaison$ et xm. Comparer les résultats obtenus avec les 4 dernières observations de "Données1.csv". Qu'en déduisez-vous sur les modèles proposés ?

```
##création de séries où à chaque colonne sera assignée la prédiction par un modèle
models <- c("ar3","ma2","ar2ma1")
preds <- zoo(matrix(NA,ncol=3,nrow=4),order.by=tail(index(xm.source),4))
colnames(preds) <- models
desaisonp <- preds #séries vierges pour les prédiction de desaison
xmp <- preds #séries vierges pour les prédiction de xm
```

Les modèles ARIMA ont été estimés sur les valeurs centrées y. On ajoute donc aux prédictions de ces modèles la moyenne empirique de $desaison$ pour obtenir la prévision de $desaison$:

$$\widehat{desaison}_t = \bar{y} + \hat{y}_t$$

On avait $desaison_t = xm_t - xm_{t-12}$, on a donc :

$$\widehat{xm}_t = xm_{t-12} + \widehat{desaison}_t$$

```
##prédiction de desaison et xm par chaque modèle
for (m in models){
pred1 <- mean(desaison) + zoo(predict(get(m),4)$pred, order.by=tail(index(xm.source),4))
pred2 <- as.numeric(tail(xm,12))[1:4] + pred1
desaisonp[,m] <- pred1
xmp[,m] <- pred2
}
obs <- tail(xm.source,4) #4 dernières observations de la série originelle
cbind(obs,xmp) #affichage des observations et des prédictions
         obs      ar3      ma2   ar2ma1
197 42.76067 42.82602 42.79939 42.83095
198 41.53919 42.30683 42.31568 42.29020
199 54.52629 55.33157 55.31540 55.31388
200 46.65552 46.11783 46.11303 46.14641
```

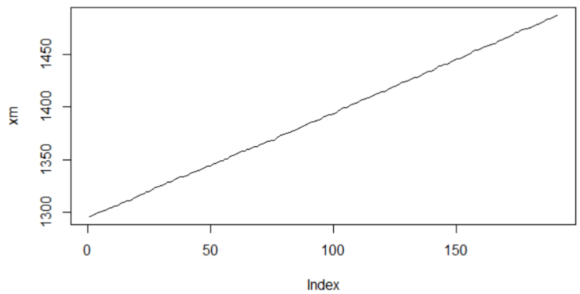
Pour comparer la performance prédictive des modèles sélectionnés, calculons le écart moyen entre l'observation et la prédiction (RMSE ou *root mean squared error*) que l'on normalisera avec l'écart-type de la série.

```
apply(xmp,2, function(x) sqrt(sum((x-obs)^2)/4)/sd(xm.source)) #calcul des RMSE
      ar3       ma2    ar2ma1
0.1188310 0.1184545 0.1155785
```

L'ARIMA(2,0,1) a le RMSE le plus faible et donne donc la meilleur prévision de la série originelle.

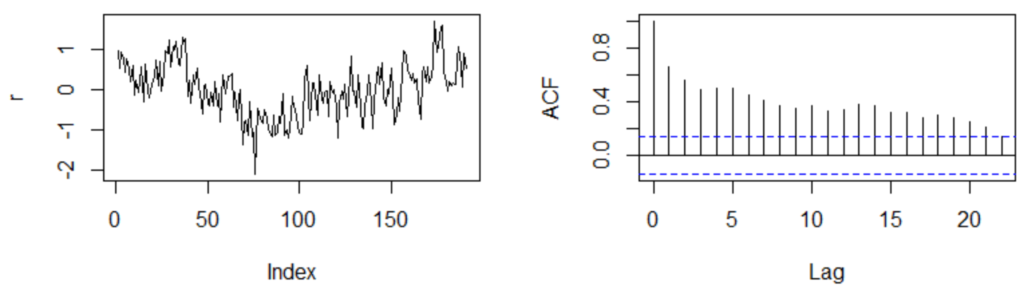
Q8. En suivant une démarche analogue, proposer un modèle pour la série contenue dans le fichier “Donnees2.csv”.

```
datafile <- "Donnees2.csv" #définit le fichier de données

data <- read.csv(datafile,sep=";") #importe un fichier .csv dans un objet de classe data.frame
xm.source <- zoo(data[[1]]) #convertit le premier élément de data en série temporelle de type "zoo"
T <- length(xm.source)
xm <- xm.source[1:(T-4)] #supprime les 4 dernières valeurs
dev.off() #réinitialise les paramètre de graphique
plot(xm)
```



La série semble présenter une tendance linéaire. Corrigeons cette tendance pour y voir plus clair.

```
trend <- 1:length(xm)
lt <- lm(xm ~ trend) #régresse la série sur une constante et une tendance linéaire
summary(lt) #affiche un résumé de la régression
r <- lt$residuals #on garde la série corrigée de sa tendance
par(mfrow=c(1,2))
plot(r)
acf(r)
```



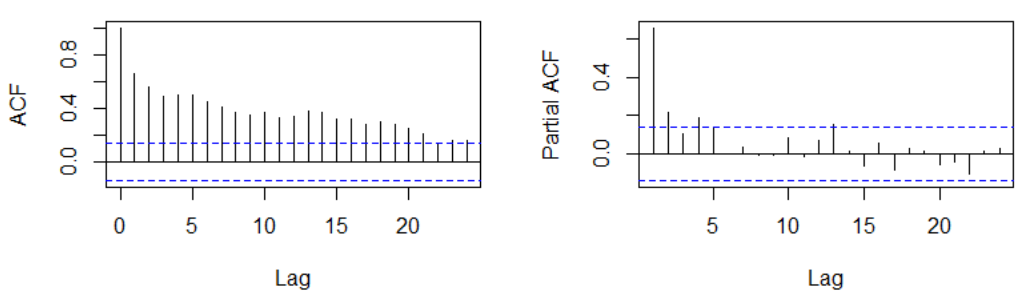
La série ne semble pas présenter de stationnarité.

```
pp.test(xm)

        Phillips-Perron Unit Root Test

 data:  xm
 Dickey-Fuller Z(alpha) = -58.438, Truncation lag parameter = 4, p-value = 0.01
 alternative hypothesis: stationary
```

La série rejette la racine unitaire, donc on dira qu’elle est stationnaire, i.e. d=0.

Remarque : la série est très persistante, ce qui posera problème par la suite

```
### pour l’intérêt de l’exercice (et des temps de calculs), on n’ira pas plus loin que 24 retards
acf(r,24);pacf(r,24)
```



On choisira $p \leq 4$ et $q \leq 21$.

```
pmax=4; qmax=21
```

Remarque : si on regardait plus retards, les autocorrélations redeviennent significatives jusqu'à un ordre très très élevé (essayez `acf(r,150)`).

```
## fonction pour estimer un arima et en vérifier l'ajustement et la validité
modelchoice <- function(p,q,data=r, k=24){
estim <- try(arima(data, c(p,0,q),optim.control=list(maxit=20000)))
if (class(estim)=="try-error") return(c("p"=p,"q"=q,"arsignif"=NA,"masignif"=NA,"resnocorr"=NA, "ok"=NA))
arsignif <- if (p==0) NA else signif(estim)[3,p]<=0.05
masignif <- if (q==0) NA else signif(estim)[3,p+q]<=0.05
resnocorr <- sum(Qtests(estim$residuals,24,length(estim$coef)-1)[,2]<=0.05,na.rm=T)==0
checks <- c(arsignif,masignif,resnocorr)
ok <- as.numeric(sum(checks,na.rm=T)==(3-sum(is.na(checks))))
return(c("p"=p,"q"=q,"arsignif"=arsignif,"masignif"=masignif,"resnocorr"=resnocorr,"ok"=ok))
}

## fonction pour estimer et vérifier tous les arima(p,q) avec p<=pmax et q<=max
armamodelchoice <- function(pmax,qmax){
pqs <- expand.grid(0:pmax,0:qmax) t(apply(matrix(1:dim(pqs)[1]),1,function(row) {
p <- pqs[row,1]; q <- pqs[row,2]
cat(paste0("Computing ARMA(",p,",",q,") \n"))
modelchoice(p,q)
}))
}

armamodels <- armamodelchoice(pmax,qmax) #estime tous les arima (patienter...)

selec <- armamodels[armamodels[,"ok"]==1&!is.na(armamodels[,"ok"]),] #modèles bien ajustés et valides
selec
     p  q arsignif masignif resnocorr ok
[1,] 2  1        1        1         1  1
[2,] 0 10       NA        1         1  1
[3,] 2 14        1        1         1  1
[4,] 4 14        1        1         1  1
[5,] 1 18        1        1         1  1
```

On a 8 modèles bien ajustés et valides.

```
pqs <- apply(selec,1,function(row) list("p"=as.numeric(row[1]),"q"=as.numeric(row[2]))) #crée
#une liste des ordres p et q des modèles candidats
names(pqs) <- paste0("arma(",selec[,1],",",selec[,2],")") #renomme les éléments de la liste
models <- lapply(pqs, function(pq) arima(r,c(pq[["p"]],0,pq[["q"]]))) #crée une liste des modèles
vapply(models, FUN.VALUE=numeric(2), function(m) c("AIC"=AIC(m),"BIC"=BIC(m))) #calcule les AIC et
#BIC des modèles candidats
    arma(2,1) arma(0,10) arma(2,14) arma(4,14) arma(1,18)
AIC  258.2550   269.7291   271.0639   270.5061   272.5334
BIC  274.5163   308.7564   329.6048   335.5516   340.8312
```

L'ARIMA(2,0,1) minimise les critères d'information.

```
rps <- lapply(models, function(m) as.zoo(predict(m,4)$pred) #prévisions de r
xmps <- lapply(rps, function(rp) rp+cbind(1,c((T-3):T))%*%lt$coefficients) #prévisions de xm
rmse <- vapply(xmps, FUN.VALUE=numeric(1), function(xmp) sqrt(sum((as.zoo(xmp)-tail(xm.source,4))^2)))
#calcule les rmse out-of-sample
rmse
 arma(2,1) arma(0,10) arma(2,14) arma(4,14) arma(1,18)
 0.7131456  1.0280417  1.1187613  1.2403910  1.2118350
```

L'ARIMA(2,0,1) fait aussi la meilleure prévision hors échantillon.