ENSAE 2A

Séries temporelles linéaires

TD n°5

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

L'objectif de cette séance est de mettre en pratique les méthodes habituelles de traitement de 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.

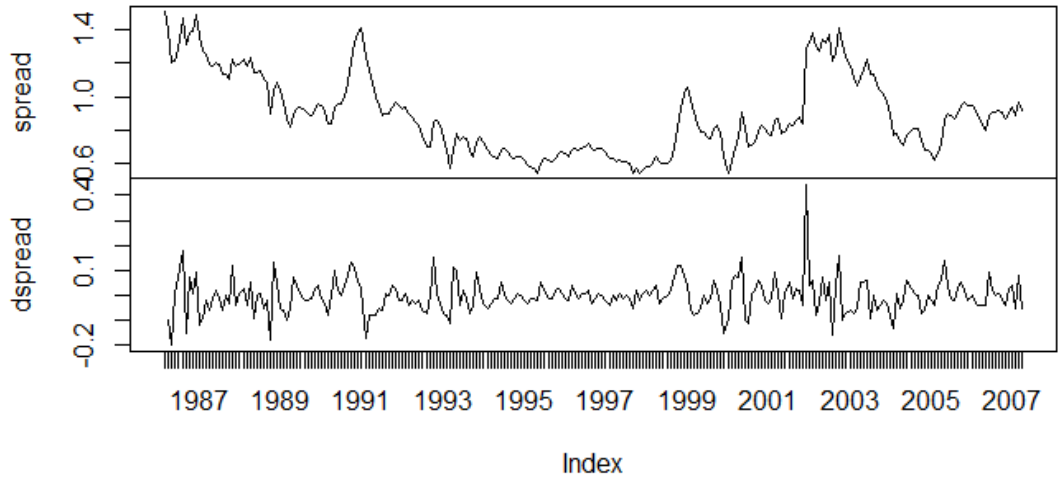
Remarque : Pour des raisons d'encodage texte, copier-coller les lignes de code peut ne pas donner le résultat voulu. Il faut bien vérifier ce qu'il en sort.

Q1. Importer la série contenue dans le fichier data_tp5.csv. La série étudiée est le spread de taux entre les obligations BAA et AAA (pour le compte d'entreprises).

```
datafile <- "data_tp5.csv"
data <- read.csv(datafile, sep=";")
```

Q2. Représenter graphiquement la série $spread$ ainsi que la série différenciée à l'ordre 1. Qu'observe-t-on ?

```
require(zoo)
require(tseries)
dates_char <- as.character(data$dates)
dates_char[1];dates_char[length(dates_char)] #affiche la première et la dernière date
dates <- as.yearmon(seq(from=1986+2/12,to=2007+3/12,by=1/12)) #index des dates pour spread
spread <- zoo(data$spread,order.by=dates)
dspread <- diff(spread,1) #différence première
plot(cbind(spread,dspread))
```



La série en niveau semble être très persistante et semble avoir une tendance non linéaire, voire non déterministe. Elle ressemble beaucoup à un marche aléatoire. Toutefois, en différence première elle semble relativement stable autour d'une constante nulle et pourraît être stationnaire. La série du spread est probablement I(1).

Q3. Conduire des tests de racine(s) unitaire(s) pour déterminer l'ordre d'intégration. Justifier le choix de votre spécification et la détermination du nombre de retards (par exemple, dans le cas des tests augmentés de Dickey-Fuller). Que peut-on en conclure ? En particulier, proposer un ordre maximum d^*.

Avant de procéder aux tests de racine unitaire, il convient de vérifier s'il y a une constante et/ou une tendance linéaire non nulle. La représentation graphique de $spread$ a montré que la tendance n'est probablement pas linéaire, mais si on devait en choisir une elle serait négative. Régressons $spread$ sur ses dates pour le vérifier :

```
summary(lm(spread ~ dates))
```

```
Call:
lm(formula = spread ~ dates)

Residuals:
     Min       1Q   Median       3Q      Max
-0.35663 -0.19280 -0.03509  0.14884  0.57802

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 20.612688   4.631245   4.451 1.29e-05 ***
dates       -0.009877   0.002319  -4.258 2.91e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2259 on 252 degrees of freedom
Multiple R-squared:  0.06713,   Adjusted R-squared:  0.06342
F-statistic: 18.13 on 1 and 252 DF,  p-value: 2.91e-05
```

Le coefficient associé à la tendance linéaire ($dates$) est bien négative, et peut-être significative (on ne peut pas vraiment le confirmer car le test n'est pas valide en présence de résidus possiblement autocorrélés). Il faudra donc se mettre dans le cas des tests de racine unitaire avec constante et éventuellement tendance non nulles.
Le test de Dickey-Fuller augmenté (ADF) dans le cas avec constante et tendance consiste en la régression suivante, pour une variable X donnée :

$$\Delta X_t = c + bt + \beta X_{t-1} + \sum_{\ell=\mathbb{1}_{k>0}}^{k} \phi_\ell \Delta X_{t-\ell} + \varepsilon_t$$

où $\beta + 1$ est l'autocorrélation à l'ordre 1 de X et k le nombre de retards nécessaires à considérer pour rendre les résidus non autocorrélés.
L'hypothèse nulle de racine unitaire $H_0 : \beta = 0$ est testée par la statistique de test $\hat{\beta}/\hat{\sigma}(\hat{\beta})$ qui suit une loi de Dickey-Fuller dépendant du nombre d'observation et du cas du test dans lequel on se place.
Pour $spread$, le test ADF donne :

```
require(fUnitRoots) #tests de racine unitaire plus modulables
adf <- adfTest(spread, lag=0, type="ct") #test ADF dans le cas avec constante et tendance
```

Avant d'interpréter le test, vérifions que les résidus du modèle de régression sont bien non autocorrélés, sans quoi le test ne serait pas valide.

```
# tests d'autocorrélation comme dans le TD4
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))
}
```

Comme la série est mensuelle, testons l'autocorrélation des résidus jusqu'à l'ordre 24 (deux ans), sans oublier de corriger les degrés de libertés du nombre de régresseurs.

```
Qtests(adf@test$lm$residuals24,length(adf@test$lm$coefficients))
```

```
      lag         pval
 [1,]   1           NA
 [2,]   2           NA
 [3,]   3           NA
 [4,]   4 0.0001653553
 [5,]   5 0.0006228068
 [6,]   6 0.0020253166
 [7,]   7 0.0048178976
 [8,]   8 0.0053287554
 [9,]   9 0.0107522173
[10,]  10 0.0144275844
[11,]  11 0.0251292647
[12,]  12 0.0402752956
[13,]  13 0.0546570005
[14,]  14 0.0763152444
[15,]  15 0.1084170353
[16,]  16 0.1459326222
[17,]  17 0.1919718170
[18,]  18 0.2314483894
[19,]  19 0.2493648556
[20,]  20 0.2971825207
[21,]  21 0.3050056430
[22,]  22 0.3387227731
[23,]  23 0.3947387545
[24,]  24 0.4451272327
```

L'absence d'autocorrélation des résidus est rejetée au moins une fois (Q(4) à Q(12)), le test ADF avec aucun retard n'est donc pas valide. Ajoutons des retards de ΔX_t jusqu'à ce que les résidus ne soient plus autocorrélés.

```
adfTest_valid <- function(series,kmax,type){ #tests ADF jusqu'à des résidus non autocorrélés
k <- 0
noautocorr <- 0
while (noautocorr==0){
cat(paste0("ADF with ",k, " lags: residuals OK? "))
adf <- adfTest(series,lags=k,type=type)
pvals <- Qtests(adf@test$lm$residuals,24,fitdf=length(adf@test$lm$coefficients))[,2]
if (sum(pvals<0.05,na.rm=T) == 0) {
noautocorr <- 1; cat("OK \n")}
else cat("nope \n")
k <- k + 1
}
return(adf)
}
adf <- adfTest_valid(spread,24,"ct")
```

```
ADF with 0 lags: residuals OK? nope
ADF with 1 lags: residuals OK? nope
ADF with 2 lags: residuals OK? nope
ADF with 3 lags: residuals OK? nope
ADF with 4 lags: residuals OK? nope
ADF with 5 lags: residuals OK? nope
ADF with 6 lags: residuals OK? nope
ADF with 7 lags: residuals OK? nope
ADF with 8 lags: residuals OK? nope
ADF with 9 lags: residuals OK? nope
ADF with 10 lags: residuals OK? nope
ADF with 11 lags: residuals OK? nope
ADF with 12 lags: residuals OK? nope
ADF with 13 lags: residuals OK? OK
```

Il a fallu considérer 13 retards au test ADF pour supprimer l'autocorrélation des résidus.

```
adf #affichage des résultats du test valide maintenu
```

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 13
  STATISTIC:
    Dickey-Fuller: -2.5543
  P VALUE:
    0.3424
```

La racine unitaire n'est pas rejetée à un seuil de 95% pour la série en niveau, la série est donc au moins I(1). Testons maintenant la racine unitaire pour la série différenciée $dspread$. La représentation graphique précédente semble montrer l'absence de constante et de tendance non nulle. Vérifions avec une régression :

```
summary(lm(dspread ~ dates[-1])) #sans la première date car on a différencié la série
Call:
lm(formula = dspread ~ dates[-1])

Residuals:
     Min       1Q   Median       3Q      Max
-0.19022 -0.03555 -0.00358  0.03433  0.43858

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.4379973  1.3709288  -1.049    0.295
dates[-1]    0.0007190  0.0006866   1.047    0.296

Residual standard error: 0.06647 on 251 degrees of freedom
Multiple R-squared:  0.00435,   Adjusted R-squared:  0.0003837
F-statistic: 1.097 on 1 and 251 DF,  p-value: 0.296
```

Il y a bien ni constante ni tendance significative. Effectuons donc le test ADF dans le cas sans constante ni tendance, en vérifiant l'absence autocorrélation des résidus.

```
adf <- adfTest_valid(dspread,24, type="nc")
ADF with 0 lags: residuals OK? OK
```

Il n'a pas été nécessaire d'inclure des retards dans le test ADF (test DF simple).

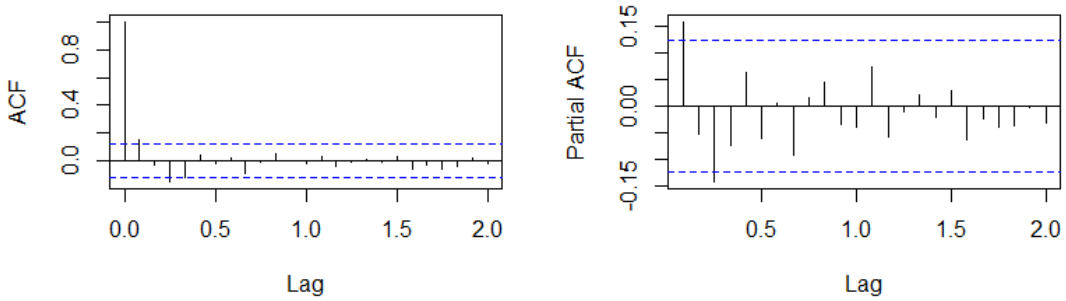
```
adf
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 0
  STATISTIC:
    Dickey-Fuller: -13.5536
  P VALUE:
    0.01
```

Le test rejette la racine unitaire (p-value$<$0.05), on dira donc que la série différenciée est "stationnaire". $spread$ est donc I(1).

Q4. Étudier les fonctions d'autocorrélation et d'autocorrélation partielle de la série retenue. En particulier, proposer des ordres maximum p^* et q^* vraisemblables pour la série étudiée.

```
x <- dspread
par(mfrow=c(1,2))
acf(x);pacf(x)
```



L'ACF est significative à l'ordre 3 au maximum, on choisira donc $q^* = 3$. La PACF est aussi significative à l'ordre 3 au maximum, on choisira donc $p^* = 3$.

Q5. En utilisant les critères d'information, proposer différentes spécifications plausibles.

Les modèles possibles sont tous les ARIMA(p,1,q) pour $spread$ où $p \leq 3$ et $q \leq 3$. Calculons les AIC et les BIC pour chacun de ces modèles.

```
mat <- matrix(NA,nrow=pmax+1,ncol=qmax+1) #matrice vide à remplir
rownames(mat) <- paste0("p=",0:pmax) #renomme les lignes
colnames(mat) <- paste0("q=",0:qmax) #renomme les colonnes
AICs <- mat #matrice des AIC non remplie
BICs <- mat #matrice des BIC non remplie
pqs <- expand.grid(0:pmax,0:qmax) #toutes les combinaisons possibles de p et q
 for (row in 1:dim(pqs)[1]){ #boucle pour chaque (p,q)
 p <- pqs[row,1] #récupère p
 q <- pqs[row,2] #récupère q
 estim <- try(arima(x,c(p,0,q),include.mean = F)) #tente d'estimer l'ARIMA
 AICs[p+1,q+1] <- if (class(estim)=="try-error") NA else estim$aic #assigne l'AIC
 BICs[p+1,q+1] <- if (class(estim)=="try-error") NA else BIC(estim) #assigne le BIC
}
AICs #affiche les AICs
          q=0       q=1       q=2       q=3
p=0 -652.4245 -657.0572 -655.1514 -657.0365
p=1 -656.8556 -655.0854 -656.4423 -657.4732
p=2 -655.4831 -657.3828 -658.2945 -656.6969
p=3 -658.7664 -657.5734 -656.5941 -657.0155
AICs==min(AICs) #affiche le modèle minimisant l'AIC
      q=0   q=1   q=2   q=3
p=0 FALSE FALSE FALSE FALSE
p=1 FALSE FALSE FALSE FALSE
p=2 FALSE FALSE FALSE FALSE
p=3  TRUE FALSE FALSE FALSE
```

L'ARIMA(3,1,0) minimise l'AIC. On le garde donc.

```
arima310 <- arima(spread,c(3,1,0),include.mean=F)


BICs #affiche les BICs
          q=0       q=1       q=2       q=3
p=0 -648.8911 -649.9904 -644.5512 -642.9029
p=1 -649.7889 -644.4853 -642.3087 -639.8063
p=2 -644.8829 -643.2493 -640.6275 -635.4966
p=3 -644.6328 -639.9064 -635.3937 -632.2818
BICs==min(BICs) #affiche le modèle minimisant le BIC
```

```
      q=0   q=1   q=2   q=3
p=0 FALSE  TRUE FALSE FALSE
p=1 FALSE FALSE FALSE FALSE
p=2 FALSE FALSE FALSE FALSE
p=3 FALSE FALSE FALSE FALSE
```

L'ARIMA(0,1,1) minimise le BIC. On le garde donc.

```
arima011 <- arima(spread,c(0,1,1),include.mean=F)
```

Q6. Pour chacun des modèles étudiés, déterminer les estimations des paramètres. Peut-on améliorer la qualité d'ajustement de ces modèles ? Expliquer soigneusement.

```
arima310
Call:
arima(x = spread, order = c(3, 1, 0), include.mean = F)

Coefficients:
         ar1      ar2      ar3
      0.1590  -0.0265  -0.1469
s.e.  0.0623   0.0643   0.0636

sigma^2 estimated as 0.004196:  log likelihood = 333.38,  aic = -658.77
```

Le coefficient AR(3) est significatif (le rapport entre le coefficient estimé et son erreur standard est plus grand en valeur absolue que 1.96), l'ARIMA(3,1,0) est donc bien ajusté.

```
arima011
Call:
arima(x = spread, order = c(0, 1, 1), include.mean = F)

Coefficients:
         ma1
      0.1609
s.e.  0.0604

sigma^2 estimated as 0.004293:  log likelihood = 330.53,  aic = -657.06
```

Le coefficient MA(1) est significatif, l'ARIMA(0,1,1) est donc bien ajusté.

Q7. Pour chacun des modèles de la question précédente, effectuer un test d'autocorrélation des résidus. Que peut-on en conclure ?

```
Qtests(arima310$residuals,24,fitdf=3)
```

```
      lag      pval
 [1,]   1        NA
 [2,]   2        NA
 [3,]   3        NA
 [4,]   4 0.1903936
 [5,]   5 0.3220626
 [6,]   6 0.3832999
 [7,]   7 0.5130327
 [8,]   8 0.3625799
 [9,]   9 0.4787474
[10,]  10 0.4546497
[11,]  11 0.5478769
[12,]  12 0.6138040
[13,]  13 0.6158028
[14,]  14 0.6548444
[15,]  15 0.7319879
[16,]  16 0.7942047
[17,]  17 0.8324315
[18,]  18 0.8613335
[19,]  19 0.8353992
[20,]  20 0.8775719
[21,]  21 0.8790861
[22,]  22 0.8941572
[23,]  23 0.9209225
[24,]  24 0.9181176
```

L'absence d'autocorrélation des résidus n'est jamais rejetée. L'ARIMA(3,1,0) est donc valide.

```
Qtests(arima011$residuals,24,fitdf=1)
      lag      pval
 [1,]   1        NA
 [2,]   2 0.91990818
 [3,]   3 0.10471573
 [4,]   4 0.06352586
 [5,]   5 0.08414332
 [6,]   6 0.13111939
 [7,]   7 0.18156508
 [8,]   8 0.13403557
 [9,]   9 0.19492880
[10,]  10 0.21834556
[11,]  11 0.29079567
[12,]  12 0.35393707
[13,]  13 0.39292541
[14,]  14 0.43916227
[15,]  15 0.51663244
[16,]  16 0.58621076
[17,]  17 0.64752868
[18,]  18 0.67787069
[19,]  19 0.66658126
[20,]  20 0.72611395
[21,]  21 0.73625772
[22,]  22 0.76759778
[23,]  23 0.79965156
[24,]  24 0.82269454
```

L'absence d'autocorrélation des résidus n'est jamais rejetée. L'ARIMA(0,1,1) est donc aussi valide.

Q8. Quel(s) modèle(s) peut-on choisir au terme des trois premières étapes de la méthode de Box-Jenkins ?

Les ARIMA(3,1,0) et ARIMA(0,1,1) pour spread sont tous les deux bien ajustés, valides, et minimisent un des critères d'information. On les choisit donc comme meilleurs modèles à ce stade.

Q9. À partir de votre meilleur modèle, déterminer les résidus ajustés. Quel est l'effet de l'observation en 2001m12 ? Comment pourrait-on prendre en compte cette observation ? Expliquer soigneusement.

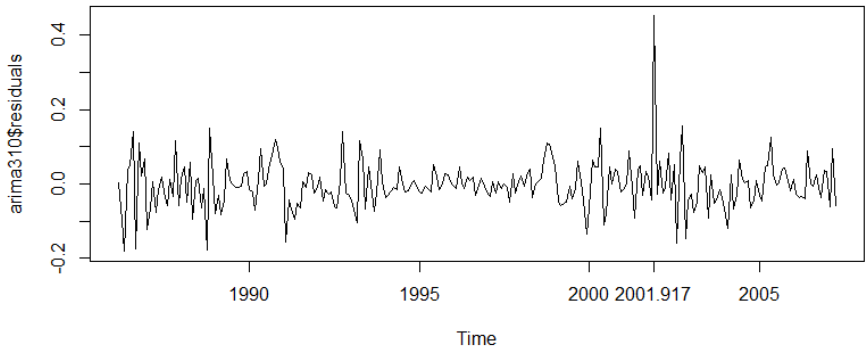
Pour choisir entre les deux, on peut garder celui qui donne la meilleur prévision dans l'échantillon par exemple, en calculant le R^2 ajusté (coefficient de détermination).

```
adj_r2 <- function(model){
ss_res <- sum(model$residuals^2) #somme des résidus au carré
 p <- model$arma[1] #récupère l'ordre AR
```

```
 q <- model$arma[2] #récupère l'ordre MA
 ss_tot <- sum(dspread[-c(1:max(p,q))]^2) #somme des observations de l'échantillon au carré
 n <- model$nobs-max(p,q) #taille de l'échantillon
 adj_r2 <- 1-(ss_res/(n-p-q-1))/(ss_tot/(n-1)) #r2 ajusté
 return(adj_r2)
}
adj_r2(arima310)
[1] -0.008974883
adj_r2(arima011)
[1] -0.02392958
```

L'ARIMA(3,1,0) a le R2 ajusté le plus important, il donne donc la meilleure prévision dans l'échantillon. On le garde comme meilleur modèle au final.

```
dev.off() #réinitialise les paramètres graphiques
plot(arima310$residuals)
axis(side=1,2001+11/12) #ajoute une légende pour décembre 2001
```



On voit une valeur extrême en décembre 2001 (crise en Argentine ?). On pourrait la prendre en compte en ajoutant une indicatrice correspondant à cette date dans la régression.

Q10. *(Optionnel)* On souhaite conduire un test de stabilité (par exemple, avant et après 1995). Comment pourrait-on procéder ?

En regardant la représentation du spread en niveau, on voit qu'il est décroissant avant 1995, croissant après. Pour prendre en compte cette non linéarité, on peut par exemple régresser séparément les deux échantillons et voir si on obtient les mêmes modèles ARIMA déjà, et comparer les coefficients avec un test d'égalité des coefficients.