

Code R

```
# Nous commençons par appeler les sous-programmes et les bibliothèques d'intérêt pour notre projet
```

```
library('ggplot2')
```

```
install.packages("forecast")
```

```
library(forecast)
```

```
install.packages("tseries")
```

```
library(tseries)
```

```
# Cette version du modèle d'attrition utilise la forme en logarithme naturel pour éliminer l'hétéroscédasticité éventuelle.
```

```
#
```

```
# Le fichier CSV est sous une forme logarithmique (taux en conversion logarithmique après multiplication par 100).
```

```
# Après lecture des données, nous traçons les données d'attrition d'intérêt pour analyser s'il n'y a pas de valeurs aberrantes que nous devons éliminer ou traiter comme variable fictive.
```

```
# Les lignes suivantes de codage nous permettent justement de le faire.
```

```
#
```

```
# Les codes en question éliminent essentiellement les données pour la période de réduction des Forces 1994-1998.
```

```
# Le code tsclean est là foncièrement pour contrôler les valeurs aberrantes.
```

```
LLcafdta <- read.csv('LogFormatt.csv', header=TRUE, stringsAsFactors=FALSE)
```

```
# Tracer les données
```

```
ggplot(LLcafdta, aes(x=FY, y=IOffr)) + geom_point()
```

```
#
```

```
# atrts <- ts(LLcafdta, start=c(1985), end=c(2017), frequency=1)
```

```
Smth_Offr = ts(LLcafdta[, c('IOffr')])
```

```
LLcafdta$Smth_Offr = tsclean(Smth_Offr)
```

```
#
```

```
Smth_ncm = ts(LLcafdta[, c('Incm')])
```

```
LLcafdta$Smth_ncm = tsclean(Smth_ncm)
```

```
#
```

```
Smth_CAF = ts(LLcfdta[, c('ICAF')])
```

```
LLcfdta$Smth_CAF= tsclean(Smth_CAF)
```

```
## On peut faire une analyse ARIMA unidimensionnelle en portant en graphique les fonctions d'autocorrélation  
intégrale (ACF) et partielle (PACF).
```

```
# d'une série en question. Si l'ACF décroît lentement et que la PACF montre une certaine autocorrélation  
(significative),
```

```
# nous pouvons, au premier décalage, prendre les différences une fois sur la série et retester les données.
```

```
#
```

```
##On utilise l'information de
```

```
## https://ocw.mit.edu/courses/economics/14-381-statistical-method-in-economics-fall-2013/study-materials/MIT14\_381F13\_EcnomtricsInR.pdf
```

```
# Test de stationnarité (tests DF et DFA bien connus)
```

```
adf.test(Smth_Offrr, alternative = "stationary")
```

```
adf.test(Smth_ncm, alternative = "stationary")
```

```
adf.test(Smth_CAF, alternative = "stationary")
```

```
#
```

```
# Tracer les données ACF et PACF
```

```
#
```

```
Acf(Smth_Offrr, main='Smoothed Officer')
```

```
Pacf(Smth_Offrr, main='Smoothed Officer')
```

```
#
```

```
Acf(Smth_ncm, main='Smoothed NCM')
```

```
Pacf(Smth_ncm, main='Smoothed NCM')
```

```
#
```

```
Acf(Smth_CAF, main='Smoothed CAF')
```

```
Pacf(Smth_CAF, main='Smoothed CAF')
```

```
#
```

#Ou on peut utiliser le programme automatisé pour générer un modèle ARIMA.

On n'oublie pas de faire sa propre évaluation indépendante de la robustesse du modèle

```
auto.arma(Smth_Offrr, seasonal=FALSE)
```

```
auto.arma(Smth_ncm, seasonal=FALSE)
```

```
auto.arma(Smth_CAF, seasonal=FALSE)
```

#

#On fait un test de diagnostic en examinant les résidus.

```
fit<-auto.arma(Smth_Offrr, seasonal=FALSE)
```

```
tsdisplay(residuals(fit), lag.max=15, main='(1,0,0) Model Residuals S_Offrr')
```

#

```
fit2<-auto.arma(Smth_ncm, seasonal=FALSE)
```

```
tsdisplay(residuals(fit2), lag.max=15, main='(1,0,0) Model Residuals S_ncm')
```

#

```
fit3<-auto.arma(Smth_CAF, seasonal=FALSE)
```

```
tsdisplay(residuals(fit2), lag.max=15, main='(1,0,0) Model Residuals S_CAF')
```

#

Le modèle pour les militaires du rang et les FAC indique des pics au décalage 4, ce qui donne lieu à une modification permettant d'inclure un tel modèle.

#

```
fit5 = arima(Smth_CAF, order=c(1,0,4))
```

```
fit5
```

```
tsdisplay(residuals(fit5), lag.max=15, main='CAF model with 1,0,4')
```

#

On peut ne pas aimer le résultat pour les militaires du rang, car les erreurs-types indiquent une quasi-région de stationnarité.

```
fit4 = arima(Smth_ncm, order=c(1,1,4))
```

```
fit4
```

```
tsdisplay(residuals(fit4), lag.max=15, main='ncm model with 1,1,4')
```

```

# On établit la prévision sous une forme logarithmique et avec des versions lissées.

fcst_SOffr <- forecast(fit, h=6)

plot(fcst_SOffr)

fcst_Sncm <- forecast(fit4, h=6)

plot(fcst_Sncm)

#

fcst_SCAF <- forecast(fit5, h=6)

plot(fcst_SCAF)

accuracy(fcst_SOffr)

accuracy(fcst_Sncm)

accuracy(fcst_SCAF)

#

fcst_SOffr

fcst_Sncm

fcst_SCAF

#

#Par attrition pendant la période d'instruction, les valeurs aberrantes sont éliminées, et on répète.

Smth_Bncm = ts(LLcafdta[, c('IBncm')])

LLcafdta$Smth_Bncm= tsclean(Smth_Bncm)

#

Smth_BCAF = ts(LLcafdta[, c('IBCAF')])

LLcafdta$Smth_BCAF= tsclean(Smth_BCAF)

#

Smth_Bof = ts(LLcafdta[, c('IBoffr')])

LLcafdta$Smth_Bof= tsclean(Smth_Bof)

#

# Tracer ACF et PACF.

#

```

```

Acf(Smth_Bof, main='Attrition during Training Officer')
Pacf(Smth_Bof, main='Attrition during Training Officer')

#

Acf(Smth_Bncm, main='Attrition during Training NCM')
Pacf(Smth_Bncm, main='Attrition during Training NCM')

#

Acf(Smth_BCAF, main='Training Attrition CAF')
Pacf(Smth_BCAF, main='Training Attrition CAF')

#

# La série n'exige pas qu'on prenne les différences, mais elle montre bel et bien un certain nombre de
propriétés en moyenne mobile.

#

fit10 = arima(Smth_Bof, order=c(0,0,2))
fit10
tsdisplay(residuals(fit10), lag.max=15, main='Officer (training) with 0,0,2')
fcst_Bof <- forecast(fit10, h=6)
plot(fcst_Bof)
accuracy(fcst_Bof)
fcst_Bof

#

fit11 = arima(Smth_Bncm, order=c(0,0,4))
fit11
tsdisplay(residuals(fit11), lag.max=15, main='ncm training with 0,0,4')
fcst_Bncm <- forecast(fit11, h=6)
plot(fcst_Bncm)
accuracy(fcst_Bncm)
fcst_Bncm

#

fit12 = arima(Smth_BCAF, order=c(0,0,4))

```

```
fit12  
  
tsdisplay(residuals(fit12), lag.max=15, main='CAF training with 0,0,4')  
  
fcst_BCAF <- forecast(fit12, h=6)  
  
plot(fcst_BCAF)  
  
accuracy(fcst_BCAF)  
  
fcst_BCAF  
  
# On peut aussi utiliser  
  
#TheilU(a,p)
```