R-Code

```
# We begin by calling routines and libraries relevant to our project

 library('ggplot2')

install.packages("forecast")

library(forecast)

install.packages("tseries")

library(tseries)

#  This version of the attrition model uses the natural log form to remove potential heteroskedasticity

#

# The CSV file is in logs (rate converted to ln after multiplying by 100)

# After reading the data we plot the attrition data of interest to analyze if #there are any outliers that we need to remove or treat as dummy.  The #following lines of codes just do that

#

#These codes basically remove the data for the force reduction period 1994-1998

#tsclean code basically checks for outliers

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

# Plot the data

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

#

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

Smth_Offr = ts(LLcafdta[, c('lOffr')])

LLcafdta$Smth_Offr= tsclean(Smth_Offr)

#

Smth_ncm = ts(LLcafdta[, c('lncm')])

LLcafdta$Smth_ncm= tsclean(Smth_ncm)

#

Smth_CAF = ts(LLcafdta[, c('lCAF')])

LLcafdta$Smth_CAF= tsclean(Smth_CAF)
```

One can do univariate ARIMA by plotting the autocorrelation and partial autocorrelation functions

of a series in question. If for example, the ACF is decaying slowly and the PCAF show an autocorrelation (significant)

at lag one then we can difference the series once and retest the data.

#

##Using information from

https://ocw.mit.edu/courses/economics/14-381-statistical-method-in-economics-fall-2013/study-materials/MIT14_381F13_EcnomtrisInR.pdf

Stationarity Test (well known DF, ADF tests)

adf.test(Smth_Offr, alternative = "stationary")

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

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

#

Plot ACF and PACF

#

Acf(Smth_Offr, main='Smoothed Officer')

Pacf(Smth_Offr, 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')

#

#Or you can use the automated package to generate an ARIMA model

do not forget to do your own independent assessment of model #robustness

auto.arima(Smth_Offr, seasonal=FALSE)

auto.arima(Smth_ncm, seasonal=FALSE)

```
auto.arima(Smth_CAF, seasonal=FALSE)
#
#Diagnostics by looking at the residuals
fit<-auto.arima(Smth_Offr, seasonal=FALSE)
tsdisplay(residuals(fit), lag.max=15, main='(1,0,0) Model Residuals S_Offr')
#
fit2<-auto.arima(Smth_ncm, seasonal=FALSE)
tsdisplay(residuals(fit2), lag.max=15, main='(1,0,0) Model Residuals S_ncm')
#
fit3<-auto.arima(Smth_CAF, seasonal=FALSE)
tsdisplay(residuals(fit2), lag.max=15, main='(1,0,0) Model Residuals S_CAF')
#
# Model for ncm and CAF show spikes at lag 4 so we adjust model to #include such model
#
fit5 = arima(Smth_CAF, order=c(1,0,4))
fit5
tsdisplay(residuals(fit5), lag.max=15, main='CAF model with 1,0,4')
#
# Don't like the result for NCM as the standard errors are indicating near region of stationarity
fit4 = arima(Smth_ncm, order=c(1,1,4))
fit4
tsdisplay(residuals(fit4), lag.max=15, main='ncm model with 1,1,4')

# Forecast using Logs, smoothed versions
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
#
#Attrition During Training Remove outliers and repeat the same
Smth_Bncm = ts(LLcafdta[, c('lBncm')])
LLcafdta$Smth_Bncm= tsclean(Smth_Bncm)
#
Smth_BCAF = ts(LLcafdta[, c('lBCAF')])
LLcafdta$Smth_BCAF= tsclean(Smth_BCAF)
#
Smth_Bof = ts(LLcafdta[, c('lBoffr')])
LLcafdta$Smth_Bof= tsclean(Smth_Bof)
#
# Plot ACF and 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')
#
# The series does not require differencing but does show some MA properties
#
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

# You can also use

#TheilU(a,p)
```