HW3

```
#Initialization
rm(list=ls())
library(TSA)
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.6.2
```

```
library(vars)
```

```
## Warning: package 'vars' was built under R version 3.6.2
```

```
## Loading required package: MASS
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 3.6.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

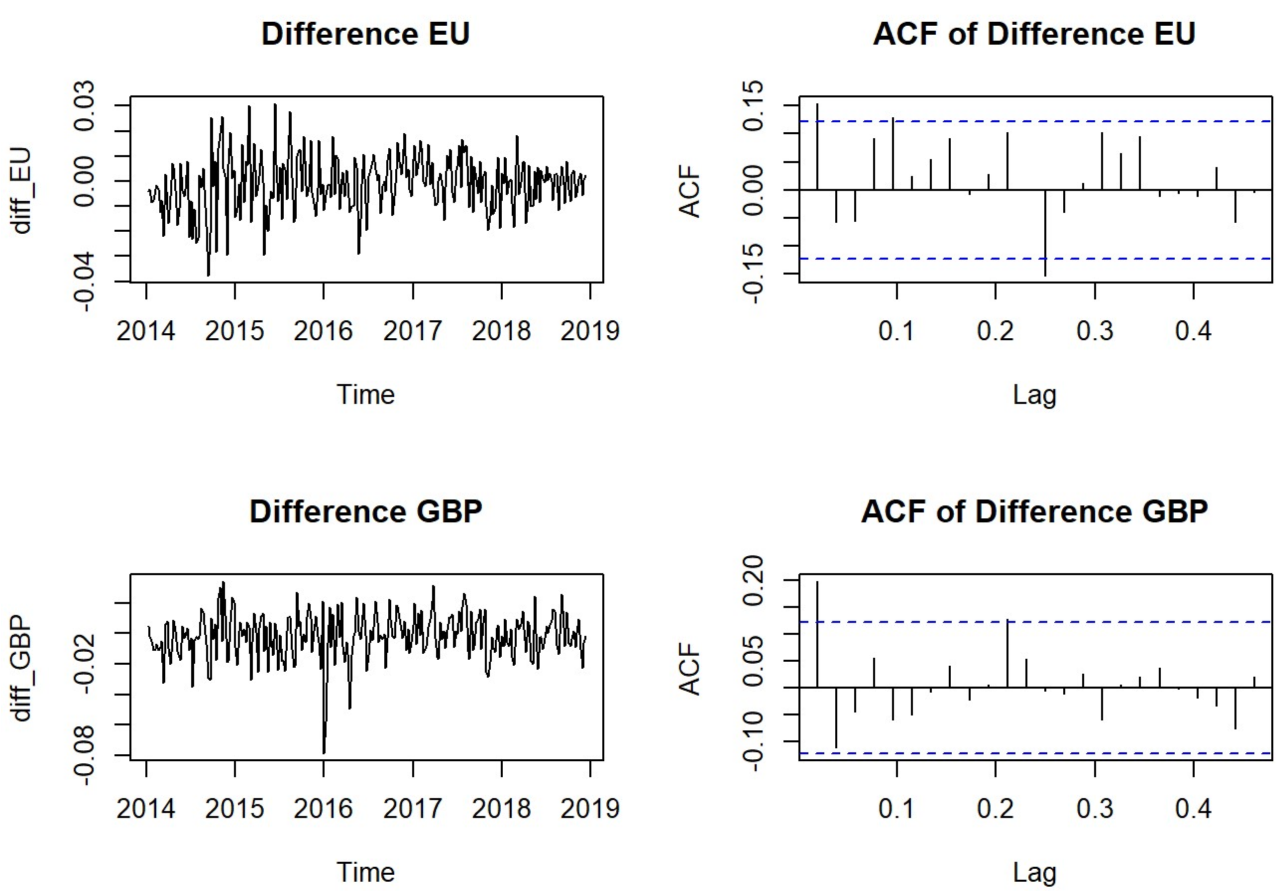
```
fname <- file.choose()
data <- read.csv(fname)
```

```
train <- data[0:(length(data[, 1])-8), ]                              # [0:258, ]
train.ts <- ts(train[, c(2,3,4,5)], start=c(2014, 1), freq=52)    # (258,4)
test <- data[(length(data[,1])-7):length(data[,1]), ]             # [259:266, ]

#You'll need this for plotting predictions
whole <- ts(data$USD.EU, start=c(2014, 1), freq=52)               # ts 1:266
times = time(whole)                                               # time()
times.test = tail(times, 8)
rm(whole)
```

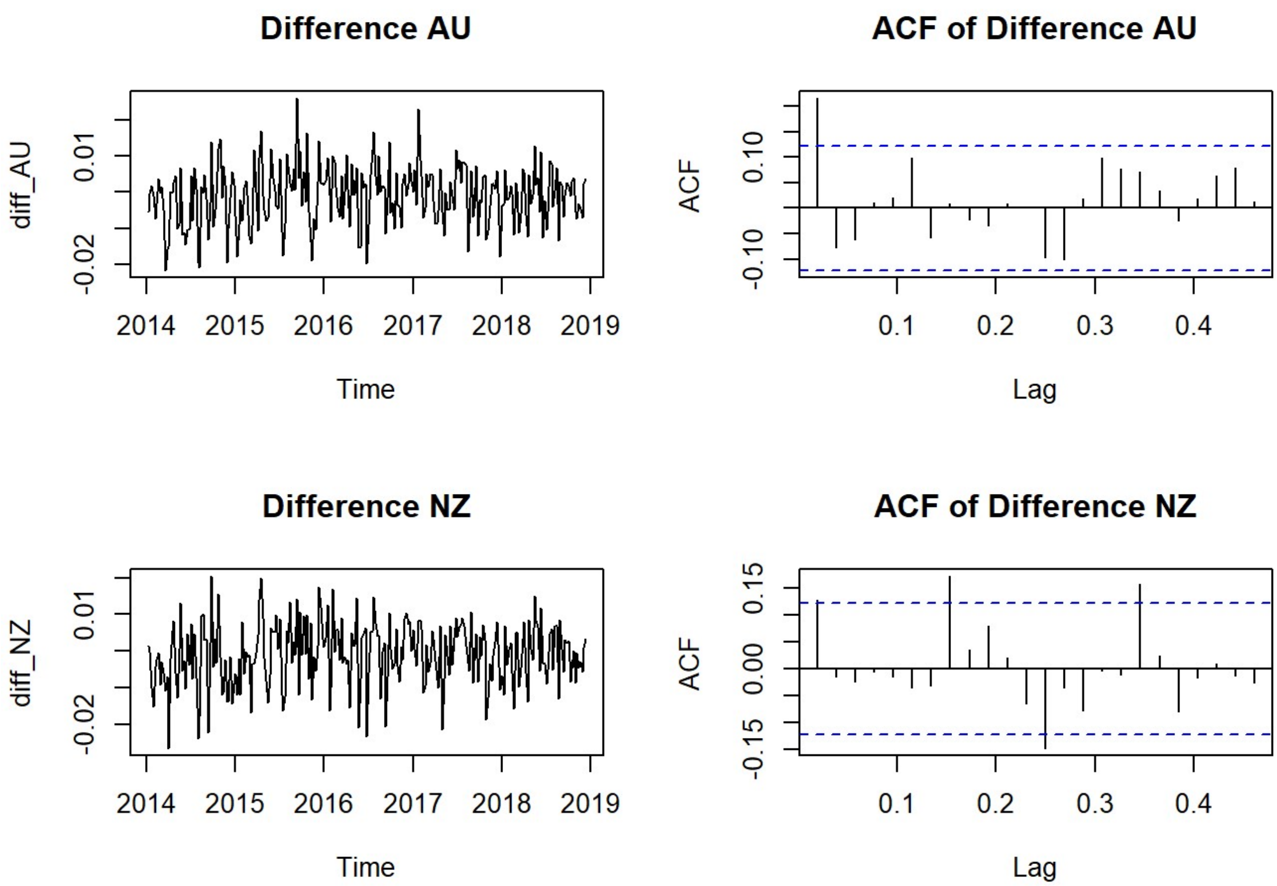
```
#USD/EUR
EU.train <- ts(train$USD.EU, start=c(2014,1), freq=52)          # ts
EU.test <- test$USD.EU                                          # num

#USD/GBP
GBP.train <- ts(train$USD.GBP, start=c(2014,1), freq=52)
GBP.test <- test$USD.GBP

#USD/AU
AU.train <- ts(train$USD.AU, start=c(2014,1), freq=52)
AU.test <- test$USD.AU

#USD/NZ
NZ.train <- ts(train$USD.NZ, start=c(2014,1),freq=52)
NZ.test <- test$USD.NZ
```

** Question 1:**

```
# Take the first order difference
diff_EU = diff(EU.train)
diff_GBP = diff(GBP.train)
diff_AU = diff(AU.train)
diff_NZ = diff(NZ.train)
#ddata.ts = ts.union(diff_EU, diff_GBP, diff_AU, diff_NZ)
#plot(ddata.ts, xlab='time', main='1st Order Differences of Currency Conversion')
par(mfrow=c(2,2))
ts.plot(diff_EU, main = "Difference EU")
acf(diff_EU, main = "ACF of Difference EU" )
ts.plot(diff_GBP, main = "Difference GBP")
acf(diff_GBP, main = "ACF of Difference GBP" )
```




```
ts.plot(diff_AU, main = "Difference AU")
acf(diff_AU, main = "ACF of Difference AU" )
ts.plot(diff_NZ, main = "Difference NZ")
acf(diff_NZ, main = "ACF of Difference NZ" )
```



Answer: For the time series plot, each difference seems to have constant mean, but variance has some changes at beginning and end. For ACF plot, it has no correlations for difference AU and difference GBP, also seems to have a little correlations for difference EU and difference NZ (beyond the confidence line). Thus it closes to stationaty.

Question 2: Unvariate Analysis

```
# (1). For EU.train
eacf(diff_EU, ar.max = 7, ma.max = 7)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7
## 0 x o o o x o o o
## 1 x o o o o o o o
## 2 x x o o x o o o
## 3 x x x o o o o o
## 4 x x x x o o o o
## 5 o x x x o o o o
## 6 o x x o o o o o
## 7 x x x o o o o o
```

```
#ARIMA order
test_modelA <- function(p,d,q){
  mod = arima(EU.train, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  print(paste(p,d,q,current.aic,sep=" "))
  return(df)
}

orders = data.frame(Inf,Inf,Inf,Inf)
names(orders) <- c("p","d","q","AIC")

for (p in 0:5){
  for (d in 0:1){
    for (q in 0:5) {
        possibleError <- tryCatch(
          orders<-rbind(orders,test_modelA(p,d,q)),
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next
    }
  }
}
```

```
## [1] "0 0 0 -667.907186624859"
## [1] "0 0 1 -965.020228901173"
## [1] "0 0 2 -1166.94962687624"
## [1] "0 0 3 -1307.51450667435"
## [1] "0 0 4 -1350.96966029843"
## [1] "0 0 5 -1407.56888935528"
## [1] "0 1 0 -1585.88895547573"
## [1] "0 1 1 -1591.31848875027"
## [1] "0 1 2 -1589.49861468133"
## [1] "0 1 3 -1588.56349192871"
## [1] "0 1 4 -1587.26880902788"
## [1] "0 1 5 -1591.33234256355"
## [1] "1 0 0 -1583.99000078046"
## [1] "1 0 1 -1589.763925425"
## [1] "1 0 2 -1587.86331664296"
## [1] "1 0 3 -1590.24433938501"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "1 0 4 -1584.62651637402"
## [1] "1 0 5 -1578.7132299443"
## [1] "1 1 0 -1590.53007899432"
## [1] "1 1 1 -1589.41792144268"
## [1] "1 1 2 -1587.63186234235"
## [1] "1 1 3 -1586.70459237841"
## [1] "1 1 4 -1590.58112337327"
## [1] "1 1 5 -1590.481319555"
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in log(s2): NaNs produced
```

```
## [1] "2 0 0 -1589.01607137189"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "2 0 1 -1587.81398114267"
## [1] "2 0 2 -1562.87958365734"
## [1] "2 0 3 -1582.23643011729"
## [1] "2 0 4 -1581.07790073901"
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "2 0 5 -1589.15739112476"
## [1] "2 1 0 -1590.03130289968"
## [1] "2 1 1 -1588.07993701488"
## [1] "2 1 2 -1597.5778939598"
## [1] "2 1 3 -1596.07487874517"
## [1] "2 1 4 -1590.05807424571"
## [1] "2 1 5 -1589.00713614334"
## [1] "3 0 0 -1588.32237120851"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 1 -1585.77299248967"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 2 -1595.80555432916"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 3 -1594.2452674495"
```

```
## Warning in log(s2): NaNs produced
```

```
## [1] "3 0 4 -1584.41020789592"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 5 -1591.02615322078"
## [1] "3 1 0 -1588.23056012765"
## [1] "3 1 1 -1586.68497286137"
## [1] "3 1 2 -1596.04921186624"
## [1] "3 1 3 -1596.82331479248"
## [1] "3 1 4 -1594.82630535245"
## [1] "3 1 5 -1595.13353147381"
## [1] "4 0 0 -1586.47097623539"
## [1] "4 0 1 -1585.42663541603"
## [1] "4 0 2 -1583.41287592177"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 0 3 -1593.18544844581"
```

```
## Warning in log(s2): NaNs produced
```

```
## [1] "4 0 4 -1586.94565022253"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 0 5 -1592.33399501165"
## [1] "4 1 0 -1589.34814478481"
## [1] "4 1 1 -1591.01047923785"
## [1] "4 1 2 -1589.01416049708"
## [1] "4 1 3 -1587.461905832"
## [1] "4 1 4 -1593.1122948953"
## [1] "4 1 5 -1595.65673150136"
## [1] "5 0 0 -1587.84675736059"
## [1] "5 0 1 -1583.43203644337"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 2 -1593.08954498274"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 3 -1591.28119448373"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 4 -1588.99613331158"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 5 -1588.82691105671"
## [1] "5 1 0 -1589.91629503787"
## [1] "5 1 1 -1587.91782570337"
## [1] "5 1 2 -1587.2360885225"
## [1] "5 1 3 -1595.19587050724"
## [1] "5 1 4 -1595.5478202137"
## [1] "5 1 5 -1595.92221828407"
```

```
orders <- orders[order(-orders$AIC),]
tail(orders)
```

```
##    p d q       AIC
## 40 3 0 2 -1595.806
## 73 5 1 5 -1595.922
## 46 3 1 2 -1596.049
## 35 2 1 3 -1596.075
## 47 3 1 3 -1596.823
## 34 2 1 2 -1597.578
```

For EU, order(2,1,2) has smallest AIC = -1597.578

```
# (2). For GBP.train
eacf(diff_GBP, ar.max = 7, ma.max = 7)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7
## 0 x o o o o o o o
## 1 x x o o o o o o
## 2 o x o o o o o o
## 3 x o x o o o o o
## 4 x x x o x o o o
## 5 o x x x o o o o
## 6 x o o o o x o o
## 7 x x o o o x o o
```

```
#ARIMA order
test_modelA <- function(p,d,q){
  mod = arima(GBP.train, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  print(paste(p,d,q,current.aic,sep=" "))
  return(df)
}

orders = data.frame(Inf,Inf,Inf,Inf)
names(orders) <- c("p","d","q","AIC")

for (p in 0:5){
  for (d in 0:1){
    for (q in 0:5) {
        possibleError <- tryCatch(
          orders<-rbind(orders,test_modelA(p,d,q)),
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next
    }
  }
}
```

```
## [1] "0 0 0 -330.301720322643"
## [1] "0 0 1 -650.076776990763"
## [1] "0 0 2 -893.160930747565"
## [1] "0 0 3 -1032.92784668572"
## [1] "0 0 4 -1130.19864791136"
## [1] "0 0 5 -1193.09528310091"
## [1] "0 1 0 -1433.70198442377"
## [1] "0 1 1 -1446.50226421002"
## [1] "0 1 2 -1445.55302814691"
## [1] "0 1 3 -1445.18442233315"
## [1] "0 1 4 -1444.76997716519"
## [1] "0 1 5 -1443.21758522843"
## [1] "1 0 0 -1430.35684037952"
## [1] "1 0 1 -1443.4645809414"
## [1] "1 0 2 -1442.39255415945"
## [1] "1 0 3 -1440.49351258405"
## [1] "1 0 4 -1438.87212276947"
## [1] "1 0 5 -1432.82491486945"
## [1] "1 1 0 -1443.1765144326"
## [1] "1 1 1 -1445.13588532426"
## [1] "1 1 2 -1444.025687351"
## [1] "1 1 3 -1445.03742664363"
## [1] "1 1 4 -1443.25781343478"
## [1] "1 1 5 -1441.23746295156"
## [1] "2 0 0 -1440.15729834176"
## [1] "2 0 1 -1441.99774707329"
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```
## [1] "2 0 2 -1445.20761148731"
## [1] "2 0 3 -1444.09291158828"
## [1] "2 0 4 -1442.68067001211"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "2 0 5 -1438.06141843861"
## [1] "2 1 0 -1446.66976705289"
## [1] "2 1 1 -1444.74438429653"
## [1] "2 1 2 -1444.85347711051"
## [1] "2 1 3 -1443.21861897562"
## [1] "2 1 4 -1441.25234127713"
## [1] "2 1 5 -1439.24795087742"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 1 -1441.31390077184"
## [1] "3 0 2 -1441.7069360608"
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 3 -1438.66065361787"
```

```
## Warning in log(s2): NaNs produced
```

```
## [1] "3 0 4 -1405.66200926066"
```

```
## Warning in log(s2): NaNs produced
```

```
## [1] "3 0 5 -1425.45784013977"
## [1] "3 1 0 -1444.81182290483"
## [1] "3 1 1 -1442.93597408179"
## [1] "3 1 2 -1443.93996559177"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 1 3 -1442.72818405331"
## [1] "3 1 4 -1439.23263323571"
## [1] "3 1 5 -1444.21588059596"
## [1] "4 0 0 -1441.59064899862"
## [1] "4 0 1 -1439.99408697856"
## [1] "4 0 2 -1440.02360447251"
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```
## [1] "4 1 0 -1443.67914760522"
## [1] "4 1 1 -1443.05371295972"
## [1] "4 1 2 -1441.27845006902"
## [1] "4 1 3 -1439.24891960863"
## [1] "4 1 4 -1437.25043941691"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 1 5 -1439.10544807646"
## [1] "5 0 0 -1440.52599551448"
## [1] "5 0 1 -1439.88962318393"
## [1] "5 0 2 -1438.55100073567"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 3 -1440.19081538762"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 4 -1436.60682619326"
## [1] "5 1 0 -1443.48141056784"
## [1] "5 1 1 -1441.50727032089"
## [1] "5 1 2 -1439.54735707939"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 1 3 -1440.45038159771"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 1 4 -1439.26051115593"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 1 5 -1437.02506069357"
```

```
orders <- orders[order(-orders$AIC),]
tail(orders)
```

```
##    p d q       AIC
## 21 1 1 1 -1445.136
## 11 0 1 3 -1445.184
## 28 2 0 2 -1445.208
## 10 0 1 2 -1445.553
## 9  0 1 1 -1446.502
## 32 2 1 0 -1446.670
```

For GBP, order(0,1,1) has smallest AIC = -1446.502

```
# (3). For AU.train
eacf(diff_AU, ar.max = 7, ma.max = 7)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7
## 0 x o o o o o o o
## 1 x x o o o o o o
## 2 o x o o o o o o
## 3 x x o o o o o o
## 4 x x o x o o o o
## 5 o x x x o o o o
## 6 x x o x o x o o
## 7 x o o x x o o o
```

```
#ARIMA order
test_modelA <- function(p,d,q){
  mod = arima(AU.train, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  print(paste(p,d,q,current.aic,sep=" "))
  return(df)
}

orders = data.frame(Inf,Inf,Inf,Inf)
names(orders) <- c("p","d","q","AIC")

for (p in 0:5){
  for (d in 0:1){
    for (q in 0:5) {
        possibleError <- tryCatch(
          orders<-rbind(orders,test_modelA(p,d,q)),
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next
    }
  }
}
```

```
## [1] "0 0 0 -776.186440159815"
## [1] "0 0 1 -1077.36656608779"
## [1] "0 0 2 -1314.73906038303"
## [1] "0 0 3 -1412.49666355877"
## [1] "0 0 4 -1519.21158367723"
## [1] "0 0 5 -1550.97922267678"
## [1] "0 1 0 -1735.38204813289"
## [1] "0 1 1 -1749.48093888563"
## [1] "0 1 2 -1747.75821165781"
## [1] "0 1 3 -1746.75786813242"
## [1] "0 1 4 -1745.34259729591"
## [1] "0 1 5 -1743.93542349334"
## [1] "1 0 0 -1733.24462449917"
## [1] "1 0 1 -1747.69958949025"
## [1] "1 0 2 -1745.90989911716"
## [1] "1 0 3 -1749.40889296213"
## [1] "1 0 4 -1742.08024030314"
## [1] "1 0 5 -1740.96843354744"
## [1] "1 1 0 -1746.89346427009"
## [1] "1 1 1 -1747.66172552659"
## [1] "1 1 2 -1746.0595877755"
## [1] "1 1 3 -1748.93925770485"
## [1] "1 1 4 -1747.28500880066"
## [1] "1 1 5 -1745.85410459788"
```

```
## Warning in log(s2): NaNs produced
```

```
## [1] "2 0 0 -1745.19026742562"
## [1] "2 0 1 -1745.73328691683"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "2 0 2 -1745.89913554186"
## [1] "2 0 3 -1743.38875037881"
## [1] "2 0 4 -1742.3283898904"
## [1] "2 0 5 -1742.61567096906"
## [1] "2 1 0 -1748.58079054399"
## [1] "2 1 1 -1746.58386187862"
## [1] "2 1 2 -1744.67576248667"
## [1] "2 1 3 -1747.96161650338"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "2 1 4 -1746.73269845991"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "2 1 5 -1744.78171227688"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 0 -1651.33990418419"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 1 -1742.08235513057"
## [1] "3 0 2 -1748.00106775101"
## [1] "3 0 3 -1740.53126543325"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 4 -1741.5343991792"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 5 -1747.47800569897"
## [1] "3 1 0 -1746.58561998364"
## [1] "3 1 1 -1744.59960815475"
## [1] "3 1 2 -1742.67544983684"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 1 3 -1747.14070134522"
## [1] "3 1 4 -1743.20503030348"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 1 5 -1743.1719923114"
## [1] "4 0 0 -1744.61626076828"
## [1] "4 0 1 -1744.62947226885"
## [1] "4 0 2 -1751.55478180008"
## [1] "4 0 3 -1749.81431171211"
## [1] "4 0 4 -1741.83991929577"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 0 5 -1742.00730820254"
## [1] "4 1 0 -1744.84889406572"
## [1] "4 1 1 -1745.32079922821"
## [1] "4 1 2 -1743.41704256553"
## [1] "4 1 3 -1750.18248722683"
```

```
## Warning in log(s2): NaNs produced
```

```
## [1] "4 1 4 -1745.66287230022"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 1 5 -1743.98214983119"
## [1] "5 0 0 -1742.95039192448"
## [1] "5 0 1 -1743.42318473465"
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```
## [1] "5 0 2 -1745.10094403679"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 3 -1742.1114051355"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 4 -1736.77052370574"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 0 5 -1745.13830202746"
## [1] "5 1 0 -1742.89832659015"
## [1] "5 1 1 -1740.85020703726"
## [1] "5 1 2 -1742.88662449752"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "5 1 3 -1743.48701051638"
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```
## [1] "5 1 4 -1748.94399911496"
## [1] "5 1 5 -1748.16253875314"
```

```
orders <- orders[order(-orders$AIC),]
tail(orders)
```

```
##    p d q       AIC
## 72 5 1 4 -1748.944
## 17 1 0 3 -1749.409
## 9  0 1 1 -1749.481
## 53 4 0 3 -1749.814
## 59 4 1 3 -1750.182
## 52 4 0 2 -1751.555
```

For AU, order(4,0,2) has smallest -1751.555

```
# (4). For NZ.train
eacf(diff_NZ, ar.max = 7, ma.max = 7)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7
## 0 x o o o o o o x
## 1 x o o o o o o x
## 2 x o o o o o o x
## 3 o x o o o o o x
## 4 o o x o o o o o
## 5 x x o o o o o o
## 6 x o o o o o o o
## 7 x x o o x x x o
```

```
#ARIMA order
test_modelA <- function(p,d,q){
  mod = arima(NZ.train, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  print(paste(p,d,q,current.aic,sep=" "))
  return(df)
}

orders = data.frame(Inf,Inf,Inf,Inf)
names(orders) <- c("p","d","q","AIC")

for (p in 0:5){
  for (d in 0:1){
    for (q in 0:5) {
        possibleError <- tryCatch(
          orders<-rbind(orders,test_modelA(p,d,q)),
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next
    }
  }
}
```

```
## [1] "0 0 0 -835.100712481775"
## [1] "0 0 1 -1121.34493561546"
## [1] "0 0 2 -1318.45442171017"
## [1] "0 0 3 -1438.98623890577"
## [1] "0 0 4 -1502.04777043548"
## [1] "0 0 5 -1539.88035096144"
## [1] "0 1 0 -1711.67777789182"
## [1] "0 1 1 -1714.68023704525"
## [1] "0 1 2 -1712.68026835221"
## [1] "0 1 3 -1710.74141082174"
## [1] "0 1 4 -1708.74146668238"
## [1] "0 1 5 -1706.74320469619"
## [1] "1 0 0 -1710.03918795973"
## [1] "1 0 1 -1713.34776829947"
## [1] "1 0 2 -1711.35543420473"
## [1] "1 0 3 -1704.83727020721"
## [1] "1 0 4 -1696.98682944133"
## [1] "1 0 5 -1695.22299576051"
## [1] "1 1 0 -1714.56483536174"
## [1] "1 1 1 -1712.67705042529"
## [1] "1 1 2 -1710.6803692635"
## [1] "1 1 3 -1708.74066001924"
## [1] "1 1 4 -1706.74149361556"
## [1] "1 1 5 -1704.74219833511"
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```
## [1] "2 0 0 -1713.28690338592"
## [1] "2 0 1 -1711.35370211817"
## [1] "2 0 2 -1715.0456486771"
## [1] "2 0 3 -1713.39422578541"
## [1] "2 0 4 -1711.5835102947"
## [1] "2 0 5 -1701.05403315226"
## [1] "2 1 0 -1712.70224987256"
## [1] "2 1 1 -1710.69247101884"
## [1] "2 1 2 -1708.74310843232"
## [1] "2 1 3 -1706.74066465139"
## [1] "2 1 4 -1704.74151379266"
## [1] "2 1 5 -1702.74224149543"
## [1] "3 0 0 -1711.36538985442"
## [1] "3 0 1 -1714.3781105992"
## [1] "3 0 2 -1713.64518210451"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 3 -1708.7627192545"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 4 -1709.92181508448"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "3 0 5 -1705.79905136643"
## [1] "3 1 0 -1710.73223982841"
## [1] "3 1 1 -1708.73467142881"
## [1] "3 1 2 -1706.73429262947"
## [1] "3 1 3 -1704.74259581007"
## [1] "3 1 4 -1702.74080955939"
## [1] "3 1 5 -1700.7406928594"
## [1] "4 0 0 -1709.37081641311"
## [1] "4 0 1 -1713.09078011263"
## [1] "4 0 2 -1705.36340899276"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 0 3 -1711.55089987359"
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 0 4 -1709.5826469392"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 0 5 -1705.13609747765"
## [1] "4 1 0 -1708.74162436597"
## [1] "4 1 1 -1706.74302532579"
## [1] "4 1 2 -1704.74278161482"
## [1] "4 1 3 -1702.74291339861"
## [1] "4 1 4 -1700.741474583"
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
## [1] "4 1 5 -1705.88783864769"
## [1] "5 0 0 -1707.40393019515"
## [1] "5 0 1 -1711.35827065122"
## [1] "5 0 2 -1711.27580960181"
## [1] "5 0 3 -1711.841942915"
## [1] "5 0 4 -1709.60313781221"
## [1] "5 0 5 -1708.67380448224"
## [1] "5 1 0 -1706.75798709872"
## [1] "5 1 1 -1704.7520883296"
## [1] "5 1 2 -1702.75300799946"
## [1] "5 1 3 -1718.09804300049"
## [1] "5 1 4 -1710.0509400387"
## [1] "5 1 5 -1718.63142133643"
```

```
orders <- orders[order(-orders$AIC),]
tail(orders)
```

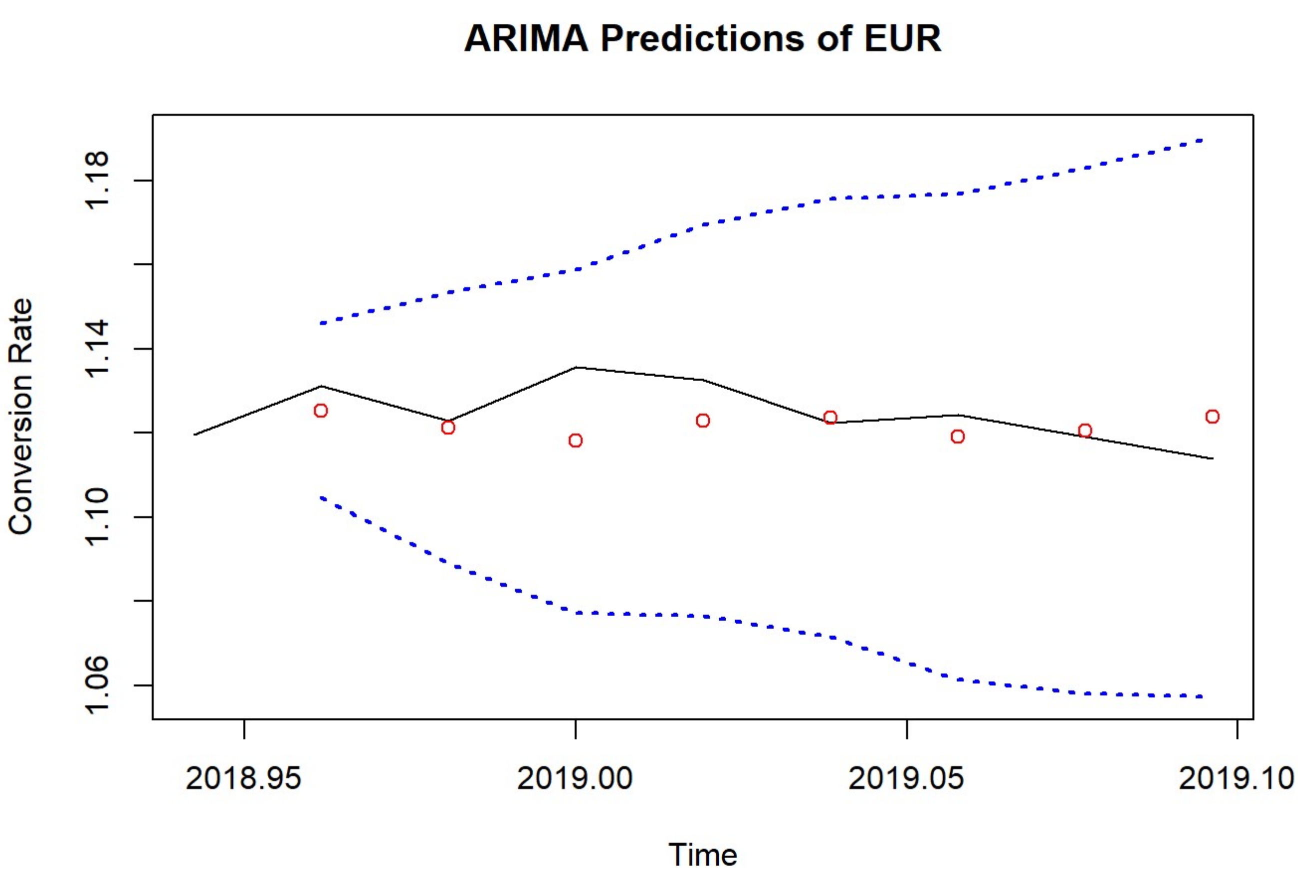
```
##    p d q       AIC
## 39 3 0 1 -1714.378
## 20 1 1 0 -1714.565
## 9  0 1 1 -1714.680
## 28 2 0 2 -1715.046
## 71 5 1 3 -1718.098
## 73 5 1 5 -1718.631
```

For NZ, order(5,1,3) has smallest -1718.098

b. Fit univariate model.

(1). EUR(2,1,3) Prediction

```
EU.data <- ts(data$USD.EU, start=c(2014,1), freq=52)
#time = time(EU.data)
vol=time(EU.data)
n = length(EU.data)
nfit = n - 8

EU_train = arima(EU.data[1:nfit], order = c(2,1,3), method = "ML")
EU_pred = as.vector(predict(EU_train, n.ahead=8))
ubound = EU_pred$pred+1.96*EU_pred$se
lbound = EU_pred$pred-1.96*EU_pred$se
ymin = min(lbound)
ymax = max(ubound)

{plot(vol[nfit:n], EU.data[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab = 'C
onversion Rate', main="ARIMA Predictions of EUR")
points(vol[(nfit+1):n], EU_pred$pred, col="red")
lines(vol[(nfit+1):n], ubound,lty=3,lwd= 2, col="blue")
lines(vol[(nfit+1):n], lbound,lty=3,lwd= 2, col="blue")}
```

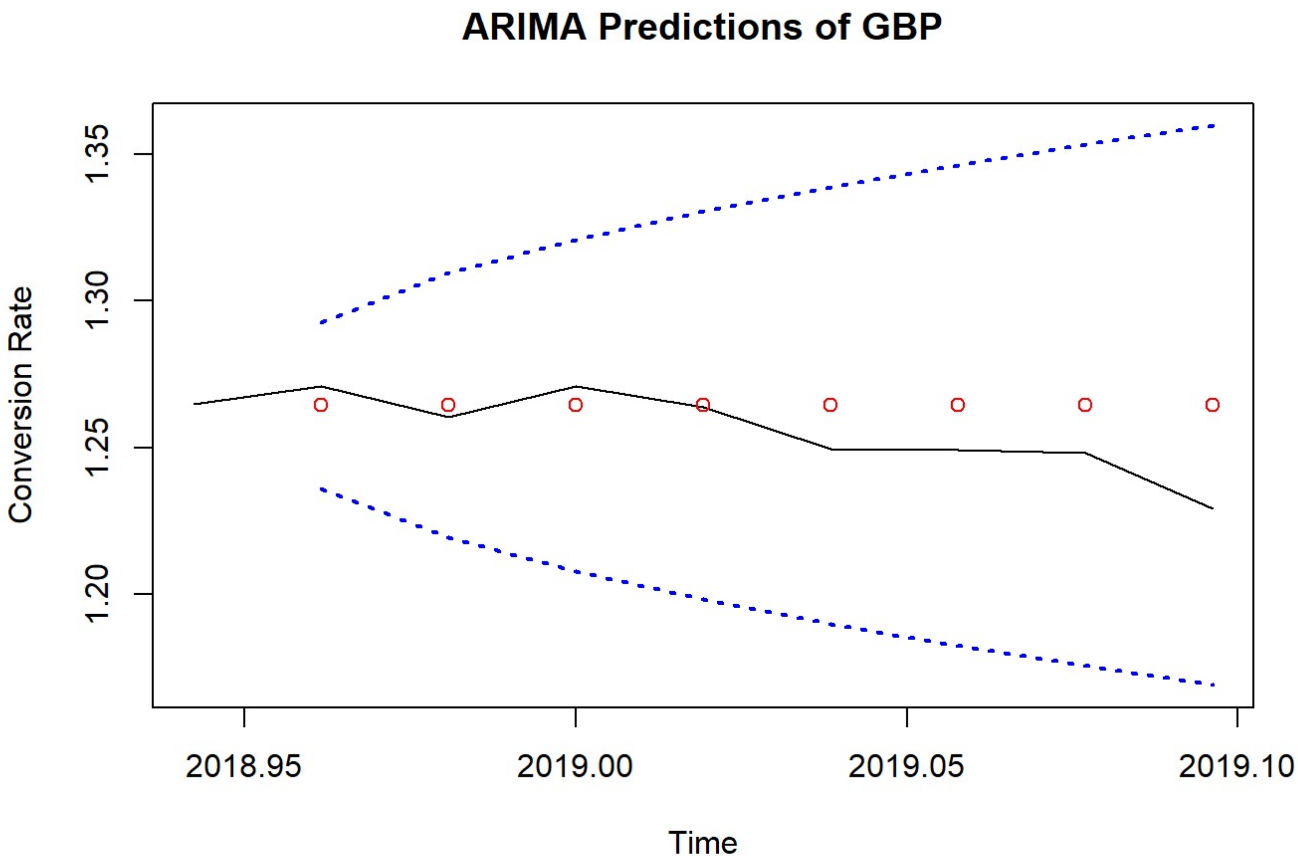


```
preds = EU_pred$pred
#obs = EU.data[(nfit+1):n]
obs = test$USD.EU

#Mean Absolute Percentage Measure (MAPE)
100*mean(abs(preds-obs)/obs)
```

```
## [1] 0.5789397
```

(2). GBP(1,1,1) Prediction

```
GBP.data <- ts(data$USD.GBP, start=c(2014,1), freq=52)
#time = time(GBP.data)
vol=time(GBP.data)
n = length(GBP.data)
nfit = n - 8

GBP_train = arima(GBP.data[1:nfit], order = c(1,1,1), method = "ML")
GBP_pred = as.vector(predict(GBP_train, n.ahead=8))
ubound = GBP_pred$pred+1.96*GBP_pred$se
lbound = GBP_pred$pred-1.96*GBP_pred$se
ymin = min(lbound)
ymax = max(ubound)

{plot(vol[nfit:n], GBP.data[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab = '
Conversion Rate', main="ARIMA Predictions of GBP")
points(vol[(nfit+1):n], GBP_pred$pred, col="red")
lines(vol[(nfit+1):n], ubound,lty=3,lwd= 2, col="blue")
lines(vol[(nfit+1):n], lbound,lty=3,lwd= 2, col="blue")}
```

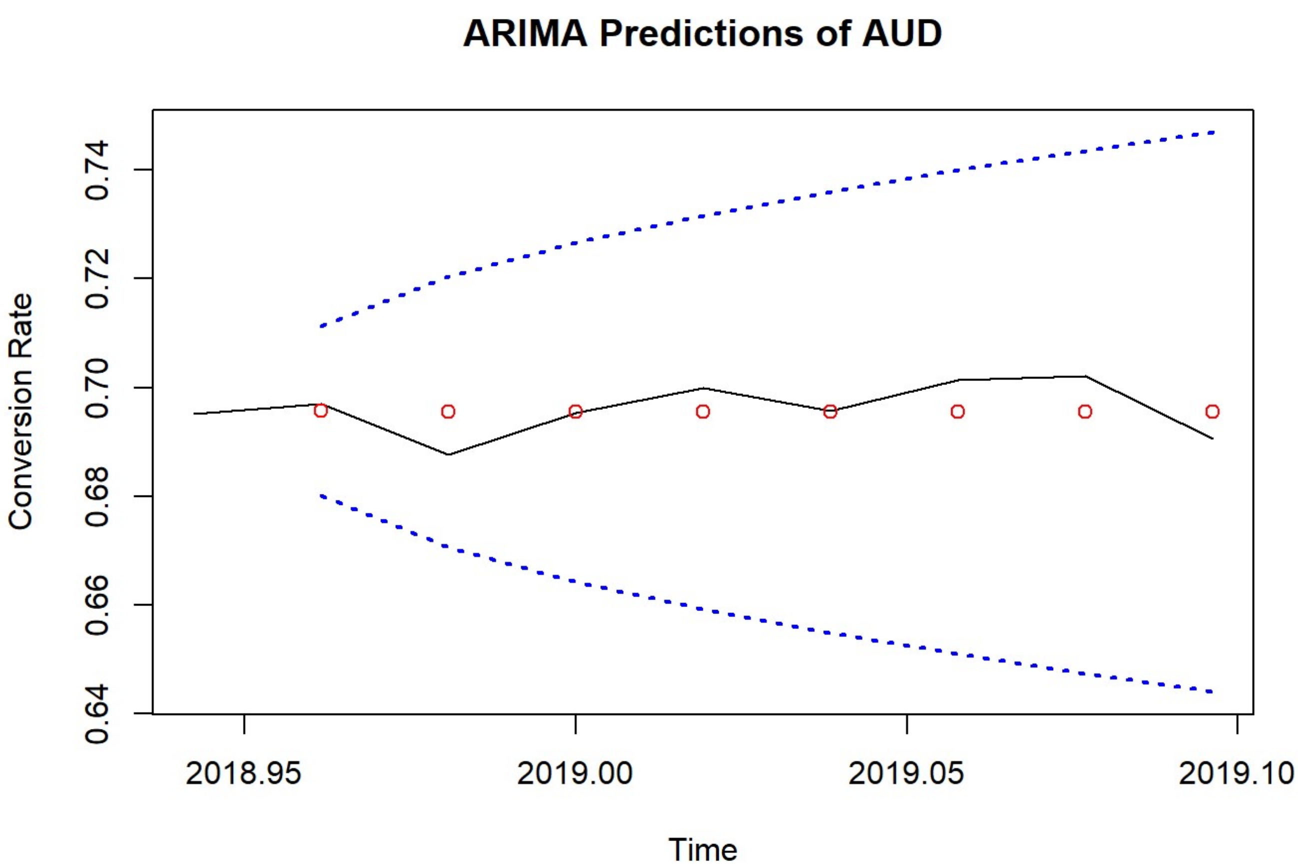



```
preds = GBP_pred$pred
#obs = GBP.data[(nfit+1):n]
obs = test$USD.GBP

#Mean Absolute Percentage Measure (MAPE)
100*mean(abs(preds-obs)/obs)
```

```
## [1] 0.9954838
```

(3). AUD(1,0,3) Prediction

```
AUD.data <- ts(data$USD.AUD, start=c(2014,1), freq=52)
#time = time(EU.data)
vol=time(AUD.data)
n = length(AUD.data)
nfit = n - 8

AUD_train = arima(AUD.data[1:nfit], order = c(1,0,3), method = "ML")
AUD_pred = as.vector(predict(AUD_train, n.ahead=8))
ubound = AUD_pred$pred+1.96*AUD_pred$se
lbound = AUD_pred$pred-1.96*AUD_pred$se
ymin = min(lbound)
ymax = max(ubound)

{plot(vol[nfit:n], AUD.data[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab = '
Conversion Rate', main="ARIMA Predictions of AUD")
points(vol[(nfit+1):n], AUD_pred$pred, col="red")
lines(vol[(nfit+1):n], ubound,lty=3,lwd= 2, col="blue")
lines(vol[(nfit+1):n], lbound,lty=3,lwd= 2, col="blue")}
```

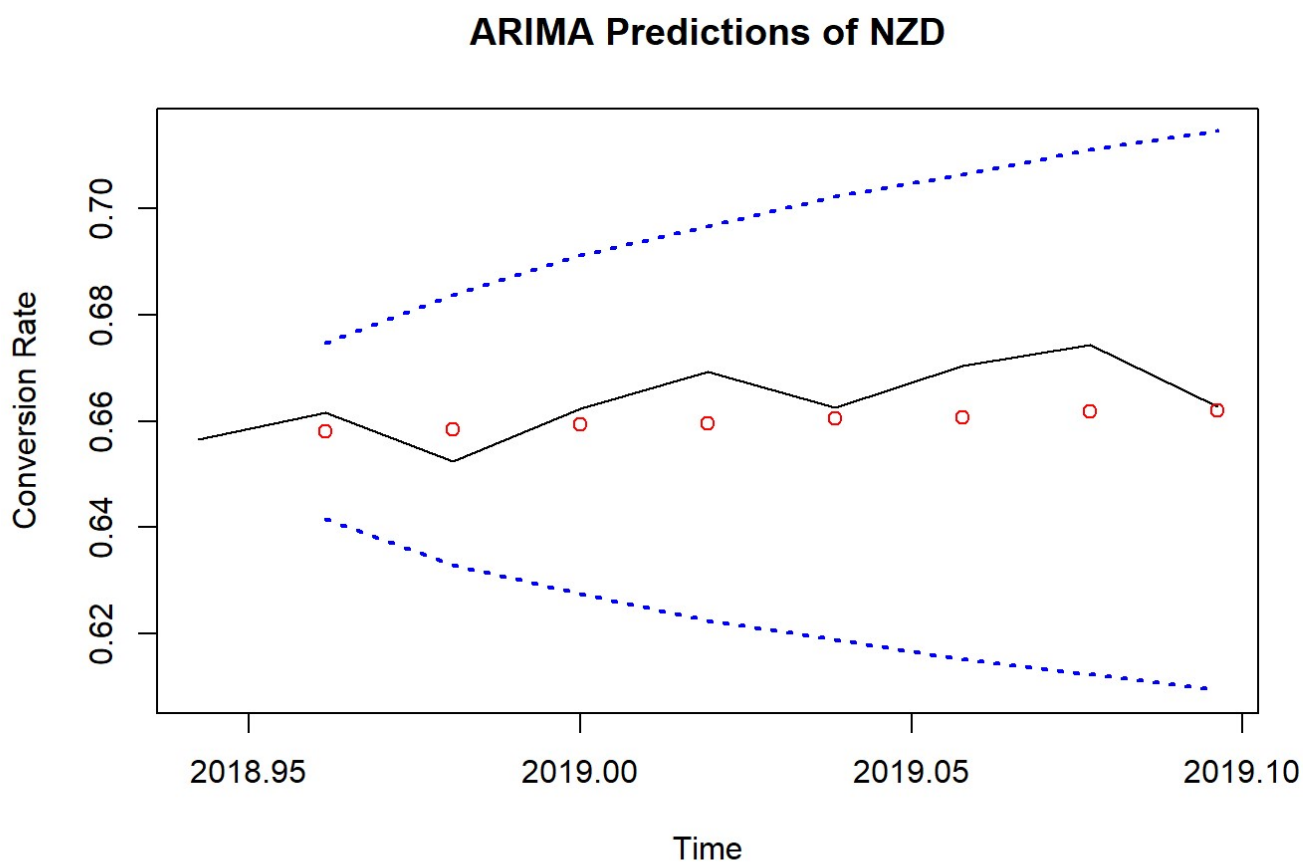


```
preds = AUD_pred$pred
#obs = AUD.data[(nfit+1):n]
obs = test$USD.AUD

#Mean Absolute Percentage Measure (MAPE)
100*mean(abs(preds-obs)/obs)
```

```
## [1] 0.5598311
```

(4). NZD(2,0,2) Prediction

```
NZ.data <- ts(data$USD.NZ, start=c(2014,1), freq=52)
#time = time(NZ.data)
vol=time(NZ.data)
n = length(NZ.data)
nfit = n - 8

NZ_train = arima(NZ.data[1:nfit], order = c(2,0,2), method = "ML")
NZ_pred = as.vector(predict(NZ_train, n.ahead=8))
ubound = NZ_pred$pred+1.96*NZ_pred$se
lbound = NZ_pred$pred-1.96*NZ_pred$se
ymin = min(lbound)
ymax = max(ubound)

{plot(vol[nfit:n], NZ.data[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab = 'C
onversion Rate', main="ARIMA Predictions of NZD")
points(vol[(nfit+1):n], NZ_pred$pred, col="red")
lines(vol[(nfit+1):n], ubound,lty=3,lwd= 2, col="blue")
lines(vol[(nfit+1):n], lbound,lty=3,lwd= 2, col="blue")}
```




```
preds = NZ_pred$pred
#obs = NZ.data[(nfit+1):n]
obs = test$USD.NZ

#Mean Absolute Percentage Measure (MAPE)
100*mean(abs(preds-obs)/obs)
```

```
## [1] 0.884671
```

c. Answer: MAPE is 0.5789 for EU(2,1,3); MAPE is 0.9955 for GBP(1,1,1); MAPE is 0.5598 for AUD(1,0,3); MAPE is 0.8847 for NZD(2,0,2). Among of them, MAPE of GBP is highest, while MAPE of AUD is lowest.

Question 3: Multivariate Analysis

a. Select the best VAR model

```
library(vars)
vs <- VARselect(train.ts)
vs$select
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      1      2
```

```
# = 2
```

The best VAR model is order = 2

b. Fit VAR(2) model.

```
# VAR(2)
mod <- VAR(train.ts, p=2)
mod
```

```
##
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation USD.EU:
## ===========================================
## Call:
## USD.EU = USD.EU.l1 + USD.GBP.l1 + USD.AUD.l1 + USD.NZ.l1 + USD.EU.l2 + USD.GBP.l2 +
USD.AUD.l2 + USD.NZ.l2 + const
##
##   USD.EU.l1  USD.GBP.l1  USD.AUD.l1   USD.NZ.l1   USD.EU.l2  USD.GBP.l2
##  1.12079671 -0.06517562  0.26728742 -0.23080126 -0.16541656  0.05261566
##  USD.AUD.l2   USD.NZ.l2       const
## -0.14101517  0.11343827  0.05481732
##
##
## Estimated coefficients for equation USD.GBP:
## ============================================
## Call:
## USD.GBP = USD.EU.l1 + USD.GBP.l1 + USD.AUD.l1 + USD.NZ.l1 + USD.EU.l2 + USD.GBP.l2
+ USD.AUD.l2 + USD.NZ.l2 + const
##
##   USD.EU.l1  USD.GBP.l1  USD.AUD.l1   USD.NZ.l1   USD.EU.l2  USD.GBP.l2
##  0.02490457  1.14079540  0.09629982 -0.05507429 -0.05968524 -0.16225990
##  USD.AUD.l2   USD.NZ.l2       const
##  0.04234498 -0.05418893  0.04019775
##
##
## Estimated coefficients for equation USD.AUD:
## ============================================
## Call:
## USD.AUD = USD.EU.l1 + USD.GBP.l1 + USD.AUD.l1 + USD.NZ.l1 + USD.EU.l2 + USD.GBP.l2
+ USD.AUD.l2 + USD.NZ.l2 + const
##
##   USD.EU.l1  USD.GBP.l1  USD.AUD.l1   USD.NZ.l1   USD.EU.l2  USD.GBP.l2
##  0.06255827 -0.14445365  1.23728049 -0.01277848 -0.05776768  0.14182986
##  USD.AUD.l2   USD.NZ.l2       const
## -0.30241168  0.05555740  0.01661440
##
##
## Estimated coefficients for equation USD.NZ:
## ===========================================
## Call:
## USD.NZ = USD.EU.l1 + USD.GBP.l1 + USD.AUD.l1 + USD.NZ.l1 + USD.EU.l2 + USD.GBP.l2 +
USD.AUD.l2 + USD.NZ.l2 + const
##
##   USD.EU.l1  USD.GBP.l1  USD.AUD.l1   USD.NZ.l1   USD.EU.l2  USD.GBP.l2
## -0.01882316 -0.10535227  0.07085500  1.10573721  0.01555854  0.09981881
##  USD.AUD.l2   USD.NZ.l2       const
## -0.04146662 -0.16663715  0.03138554
```

```
# ARCH, Residual Analysis:  Constant Variance Assumption (ARCH test)
arch.test(mod)
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object mod
## Chi-squared = 677.05, df = 500, p-value = 2.005e-07
```

Result: p-value is close to 0, which means reject null hypothesis of constant variance.

```
# J-B, Residual Analysis: normality Assumption (Jarque-Bera test)
normality.test(mod)
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object mod
## Chi-squared = 271.66, df = 8, p-value < 2.2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object mod
## Chi-squared = 37.838, df = 4, p-value = 1.21e-07
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object mod
## Chi-squared = 233.83, df = 4, p-value < 2.2e-16
```

Result: p-value is close to 0, which means reject null hypothesis of normality.

```
# Portamantau, Residual Analysis: Uncorrelated Errors Assumption (Portmanteau test)
serial.test(mod)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object mod
## Chi-squared = 266.8, df = 224, p-value = 0.02637
```

Result: p-value is small (level of 0.05), which means that there is probably no obvious uncorrelated residuals.

c. Prediction for next 8 weeks with VAR(2) model.

(a). Predict for EU.

```
preds.all <- as.vector(predict(mod, n.ahead=8))

# EU
preds1_CI <- preds.all$fcst$USD.EU
preds1_CI
```

```
##          fcst    lower    upper         CI
## [1,] 1.119814 1.099147 1.140481 0.02066704
## [2,] 1.119935 1.089488 1.150382 0.03044711
## [3,] 1.120079 1.082556 1.157602 0.03752310
## [4,] 1.120280 1.077118 1.163443 0.04316240
## [5,] 1.120537 1.072644 1.168431 0.04789365
## [6,] 1.120843 1.068860 1.172826 0.05198328
## [7,] 1.121190 1.065605 1.176775 0.05558471
## [8,] 1.121572 1.062776 1.180367 0.05879511
```

```
preds1 <- as.numeric(preds.all$fcst$USD.EU[,1])
obs1 <- as.numeric(test$USD.EU)

#MAPE
100*mean(abs(obs1 - preds1)/abs(obs1))
```

```
## [1] 0.6303623
```

(2). Predict for GBP.

```
# GBP
preds2_CI <- preds.all$fcst$USD.GBP
preds2_CI
```

```
##          fcst    lower    upper         CI
## [1,] 1.263440 1.235221 1.291660 0.02821932
## [2,] 1.262166 1.218923 1.305410 0.04324350
## [3,] 1.261059 1.206847 1.315271 0.05421166
## [4,] 1.260117 1.197193 1.323041 0.06292415
## [5,] 1.259323 1.189087 1.329559 0.07023571
## [6,] 1.258658 1.182070 1.335246 0.07658815
## [7,] 1.258107 1.175874 1.340339 0.08223238
## [8,] 1.257656 1.170333 1.344979 0.08732325
```

```
preds2 <- as.numeric(preds.all$fcst$USD.GBP[,1])
obs2 <- as.numeric(test$USD.GBP)
#MAPE
100*mean(abs(obs2 - preds2)/abs(obs2))
```

```
## [1] 0.8045432
```

(3). Predict for AUD.

```
# AUD
preds3_CI <- preds.all$fcst$USD.AUD
preds3_CI
```

```
##           fcst     lower     upper         CI
## [1,] 0.6979567 0.6828058 0.7131076 0.01515089
## [2,] 0.7002968 0.6769369 0.7236568 0.02335995
## [3,] 0.7024007 0.6729828 0.7318186 0.02941788
## [4,] 0.7043528 0.6702331 0.7384726 0.03411975
## [5,] 0.7061855 0.6682711 0.7440999 0.03791438
## [6,] 0.7079154 0.6668416 0.7489892 0.04107382
## [7,] 0.7095533 0.6657852 0.7533214 0.04376809
## [8,] 0.7111073 0.6650001 0.7572144 0.04610712
```

```
preds3 <- as.numeric(preds.all$fcst$USD.AUD[,1])
obs3 <- as.numeric(test$USD.AUD)
#MAPE
100*mean(abs(obs3 - preds3)/abs(obs3))
```

```
## [1] 1.264681
```

(4). Predict for NZ.

```
# NZ
preds4_CI <- preds.all$fcst$USD.NZ
preds4_CI
```

```
##           fcst     lower     upper         CI
## [1,] 0.6585929 0.6422616 0.6749242 0.01633128
## [2,] 0.6603236 0.6361979 0.6844494 0.02412578
## [3,] 0.6619465 0.6322871 0.6916059 0.02965938
## [4,] 0.6634933 0.6295369 0.6974496 0.03395635
## [5,] 0.6649714 0.6275083 0.7024345 0.03746310
## [6,] 0.6663849 0.6259748 0.7067949 0.04041006
## [7,] 0.6677372 0.6248039 0.7106704 0.04293326
## [8,] 0.6690311 0.6239097 0.7141526 0.04512144
```

```
preds4 <- as.numeric(preds.all$fcst$USD.NZ[,1])
obs4 <- as.numeric(test$USD.EU)
#MAPE
100*mean(abs(obs4 - preds4)/abs(obs4))
```

```
## [1] 40.98134
```

Results: From above results, the MAPE of EU is 0.63, MAPE of GBP is 0.80, MAPE of AUD is 1.26, MAPE os NZ is 40.98. Among of them, the MAPE of EU is the lowest, and MAPE of NZ is the highest.

Question 4. Summarization of Univariate model and VAR model.

Answer: Univariate forecase seems to be better than multivariate forecast because that: For Univerrate model, each univerate model can make one feasible forecast, the forecasting is fast and accurate than multivariate model. On the contrary, multivariate model (VAR) is some complicate and takes longer time to estimate, VAR model sometimes introduces more noise than signal, leading to poorer forecasts and inaccurate results. Therefore, ARIMA may have more uncertainty where the prediction band includes the observed values, but has better predictions.