VAR(p)

```
#Initialization
library(TSA)
```

```
## Warning: package 'TSA' was built under R version 4.0.3
```

```
##
## 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)
library(vars)
```

```
## Warning: package 'vars' was built under R version 4.0.3
```

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

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

```
## Warning: package 'strucchange' was built under R version 4.0.3
```

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

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

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

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

```
## Warning: package 'sandwich' was built under R version 4.0.3
```

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

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

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.4
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
setwd("C:/Users/reshed001/Desktop/OMSA_7.26.2020/ISYE 6402 - Time Series/Module 4/HW")
data = read.csv("Currency Conversion Data.csv",header=T)
head(data)
```

```
##        DATE  USD.EU  USD.GBP  USD.AUD  USD.NZ
## 1  7/2/2014 1.36536 1.709020 0.943540 0.87654
## 2  7/9/2014 1.36145 1.713425 0.938125 0.87810
## 3 7/16/2014 1.35842 1.711780 0.937800 0.87798
## 4 7/23/2014 1.35002 1.706680 0.939460 0.86816
## 5 7/30/2014 1.34254 1.695720 0.938520 0.85302
## 6  8/6/2014 1.33922 1.685380 0.931220 0.84884
```

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


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

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

#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: Data Exploration (8 pts)

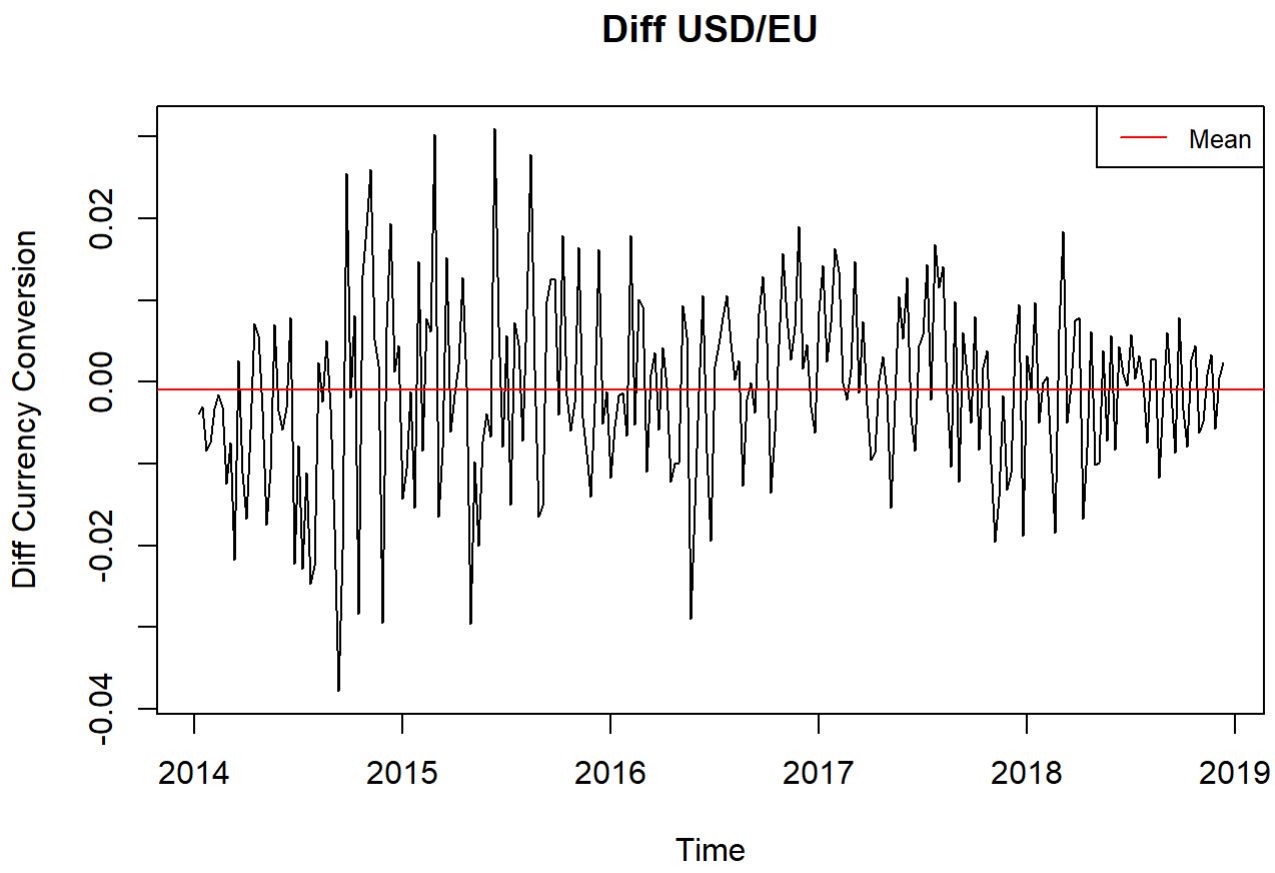
We already identified in past assignments that differenced currency conversion data tends to more closely resemble a stationary process. Knowing this, skip analyzing the original series, and for each series take the 1st order difference of the data.

Use relevant plots to evaluate assumptions of stationarity.

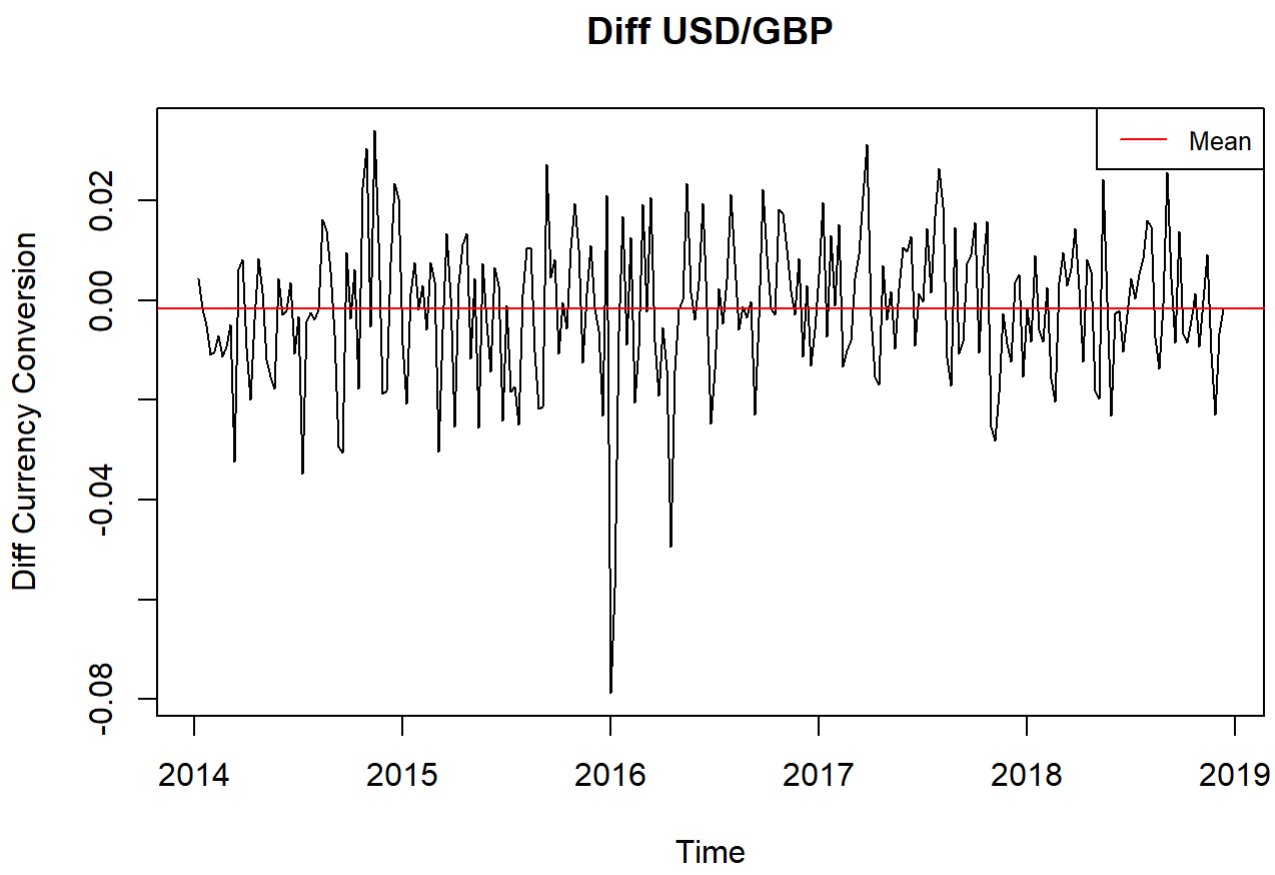
```
# Differencing Train data
USD.EU.diff = diff(EU.train)
USD.GBP.diff = diff(GBP.train)
USD.AU.diff = diff(AU.train)
USD.NZ.diff = diff(NZ.train)
```

Plot Diff Time Series Processes

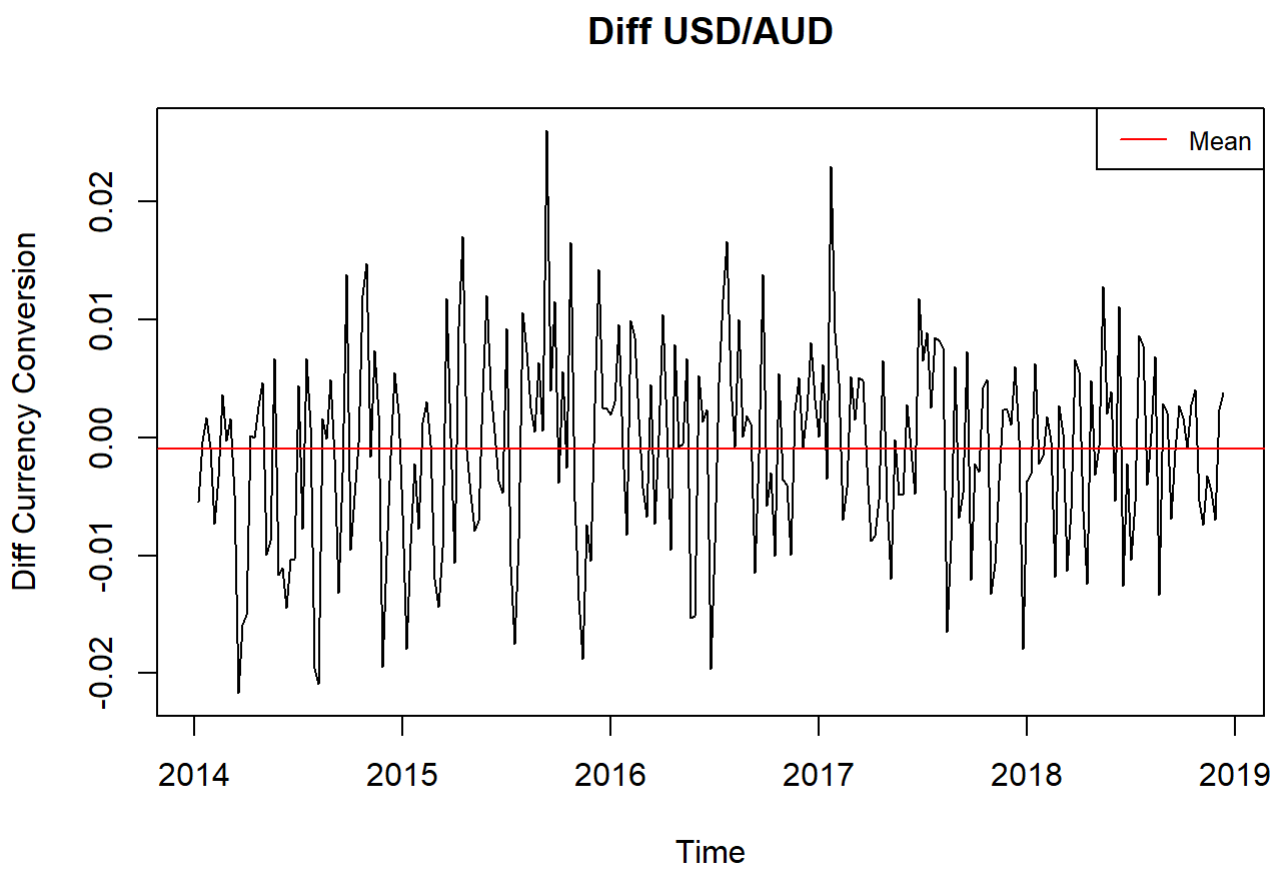
```
# Plot diff time series data
ts.plot(USD.EU.diff, ylab = "Diff Currency Conversion", main = "Diff USD/EU")
abline(h=mean(USD.EU.diff), col="red")
legend("topright", legend="Mean",
       col="red", lty=1:2, cex=0.8)
```



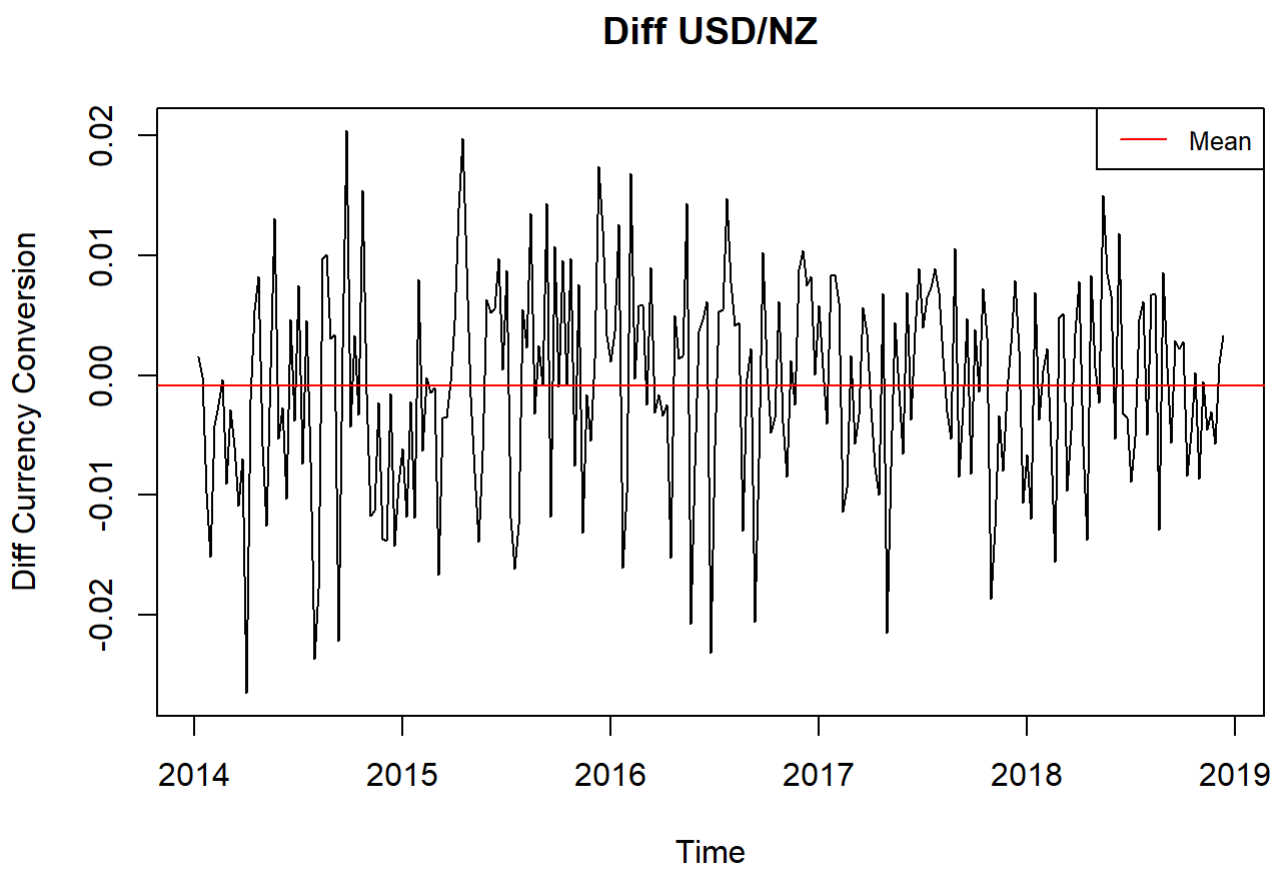

```
ts.plot(USD.GBP.diff, ylab = "Diff Currency Conversion", main = "Diff USD/GBP")
abline(h=mean(USD.GBP.diff), col="red")
legend("topright", legend="Mean",
       col="red", lty=1:2, cex=0.8)
```




```
ts.plot(USD.AU.diff, ylab = "Diff Currency Conversion", main = "Diff USD/AUD")
abline(h=mean(USD.AU.diff), col="red")
legend("topright", legend="Mean",
       col="red", lty=1:2, cex=0.8)
```

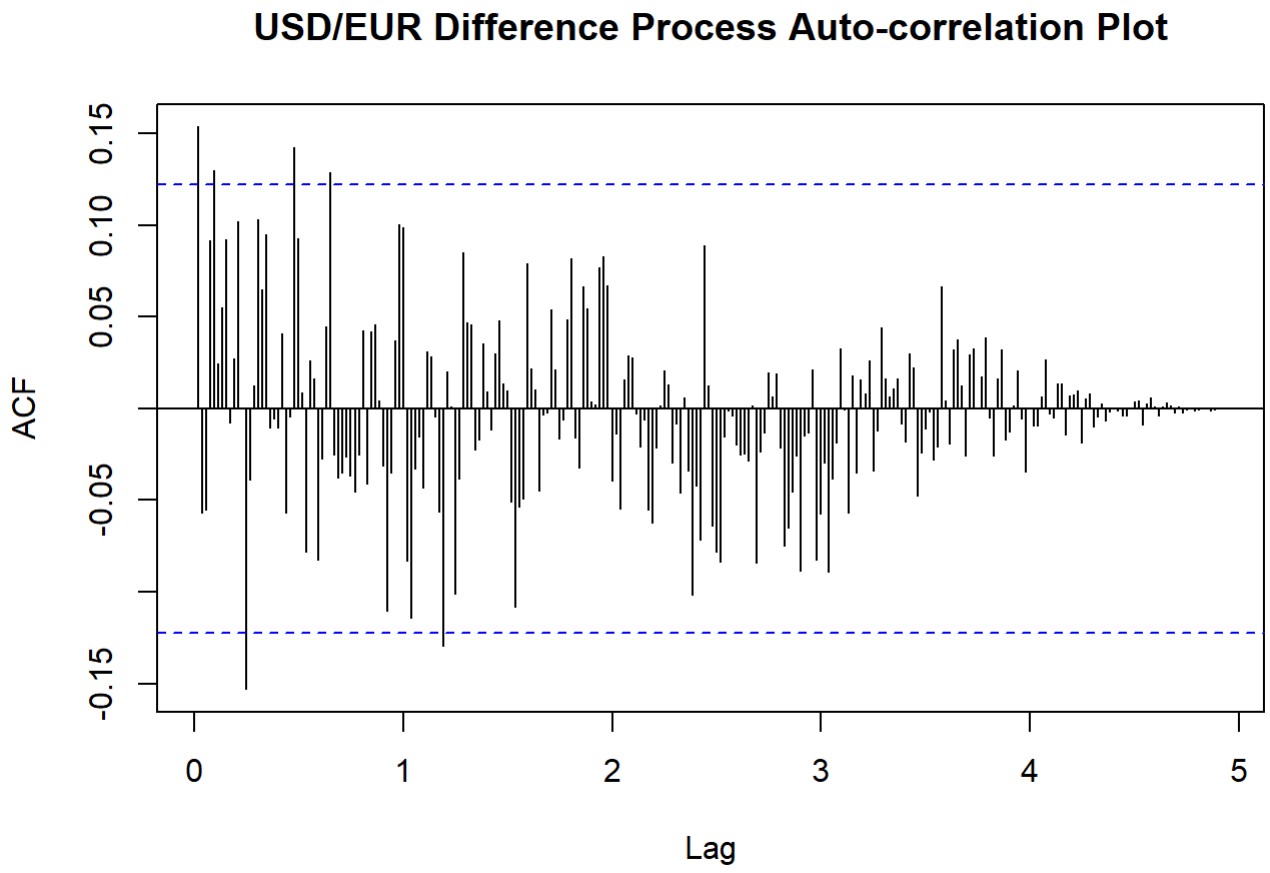



```
ts.plot(USD.NZ.diff, ylab = "Diff Currency Conversion", main = "Diff USD/NZ")
abline(h=mean(USD.NZ.diff), col="red")
legend("topright", legend="Mean",
       col="red", lty=1:2, cex=0.8)
```

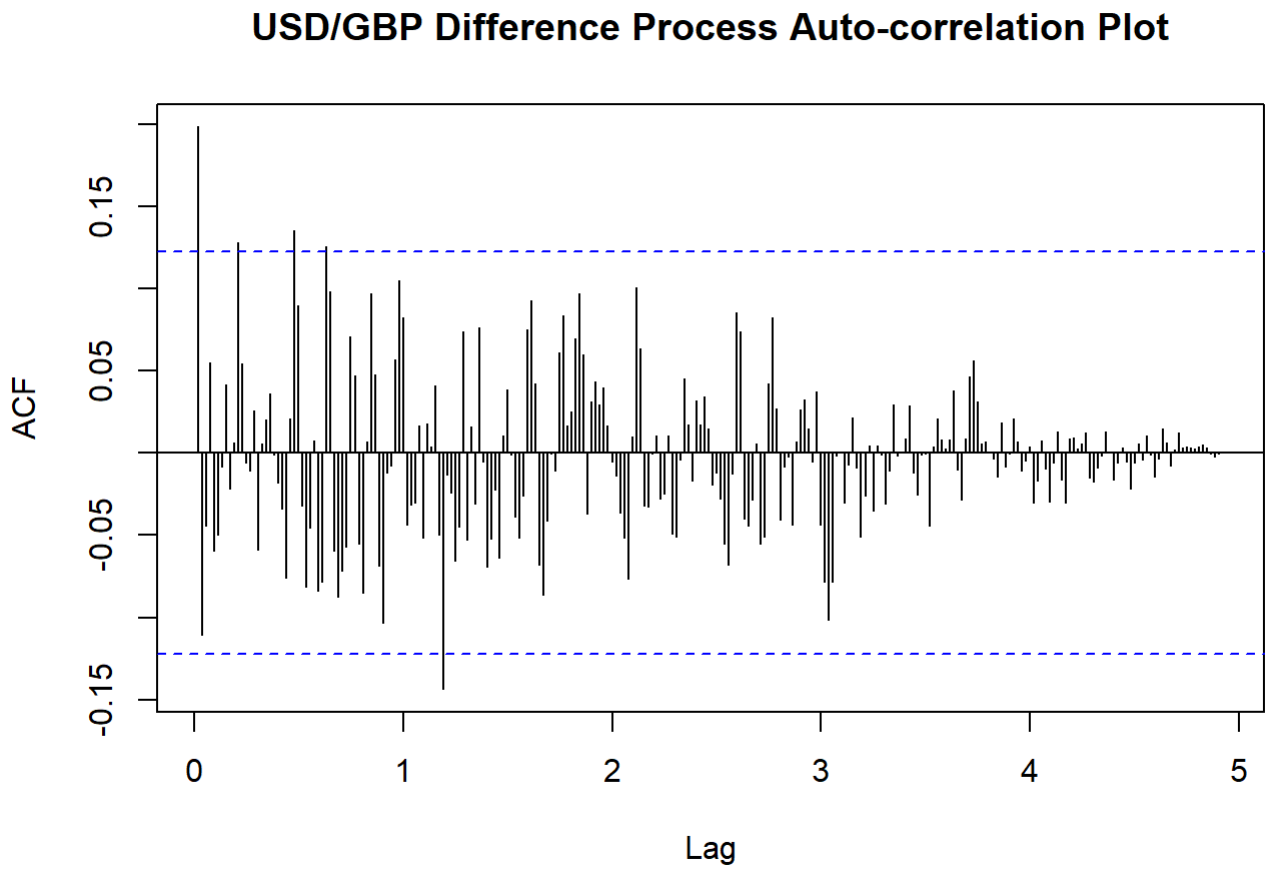


Plot Diff Time Series ACFs

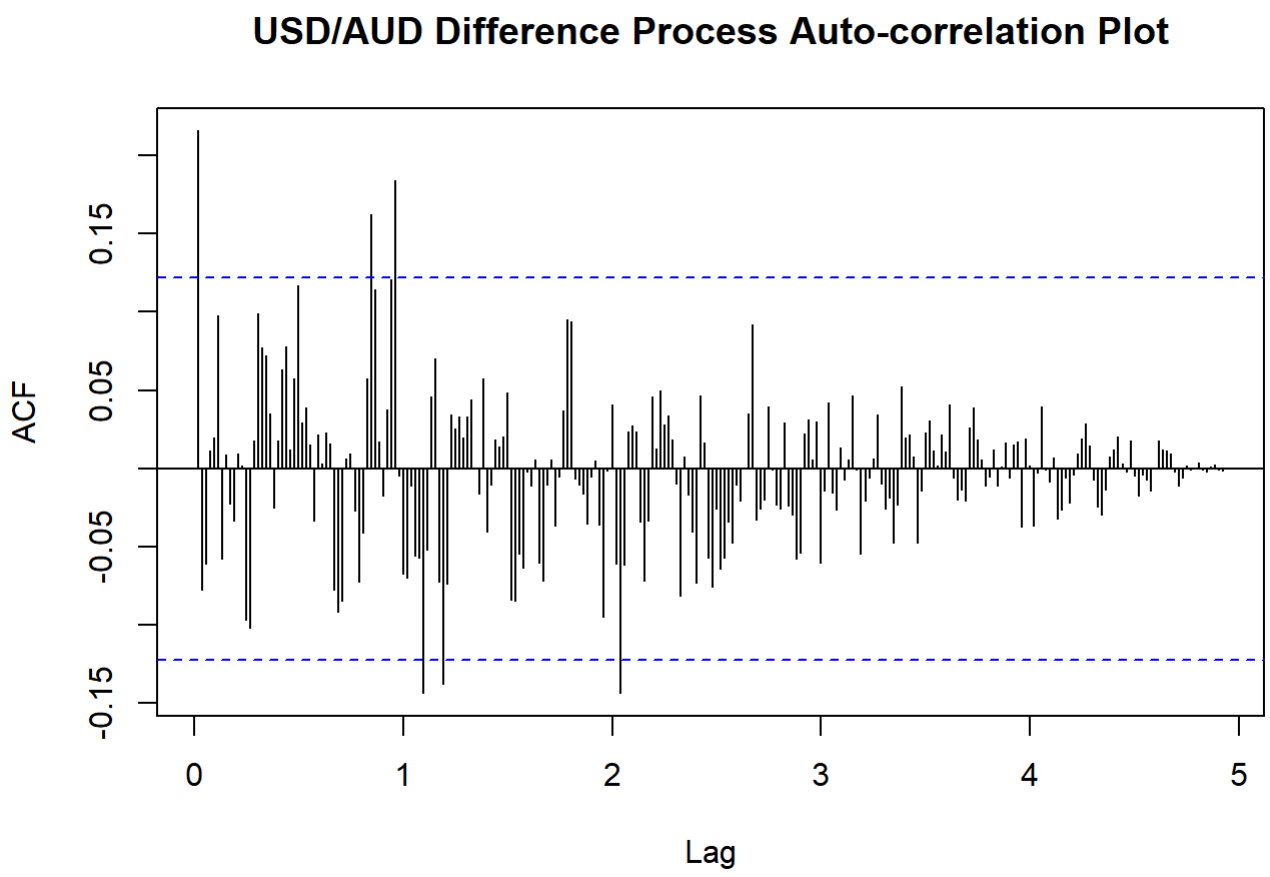
```
# Plot ACF plots
acf(USD.EU.diff,lag.max=52*6,main="USD/EUR Difference Process Auto-correlation Plot")
```



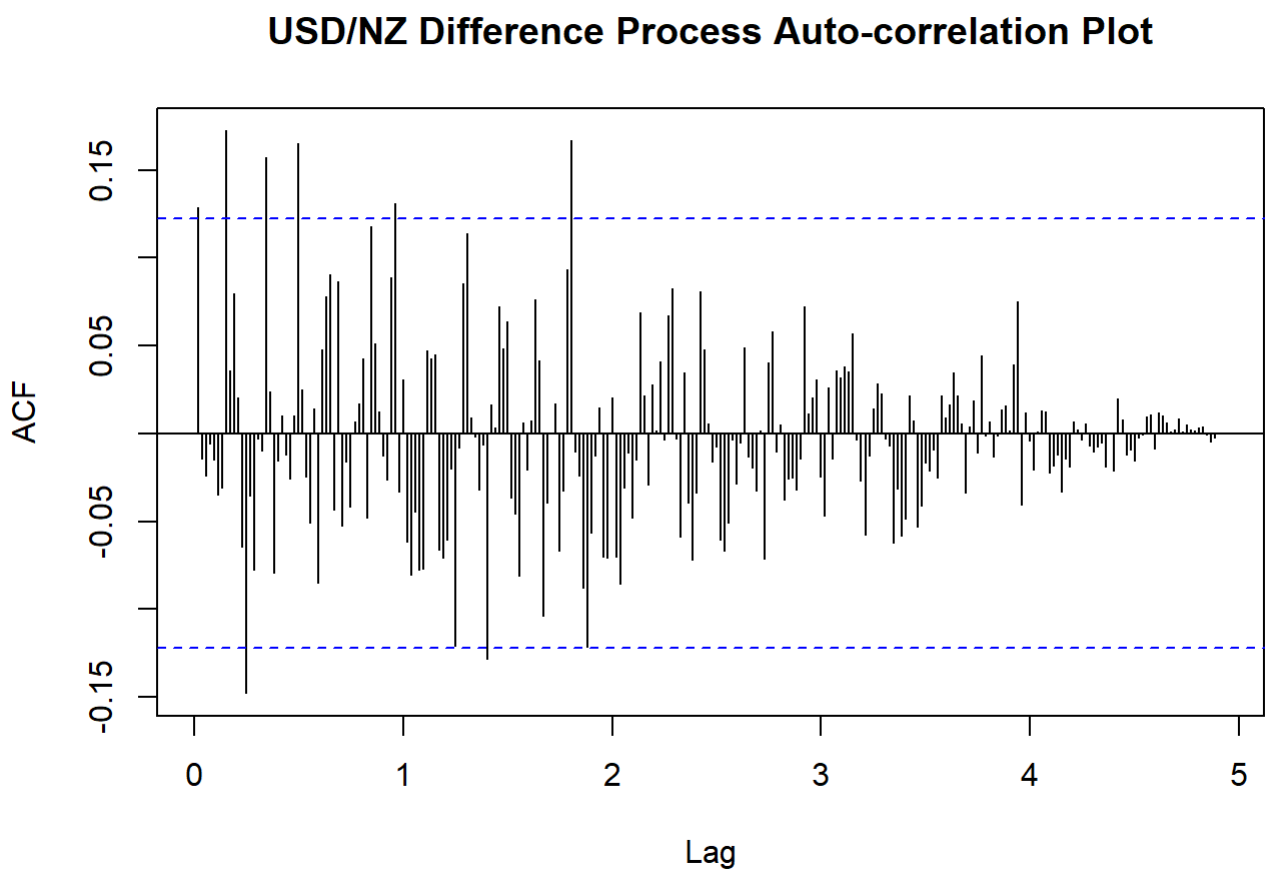

```
acf(USD.GBP.diff,lag.max=52*6,main="USD/GBP Difference Process Auto-correlation Plot")
```



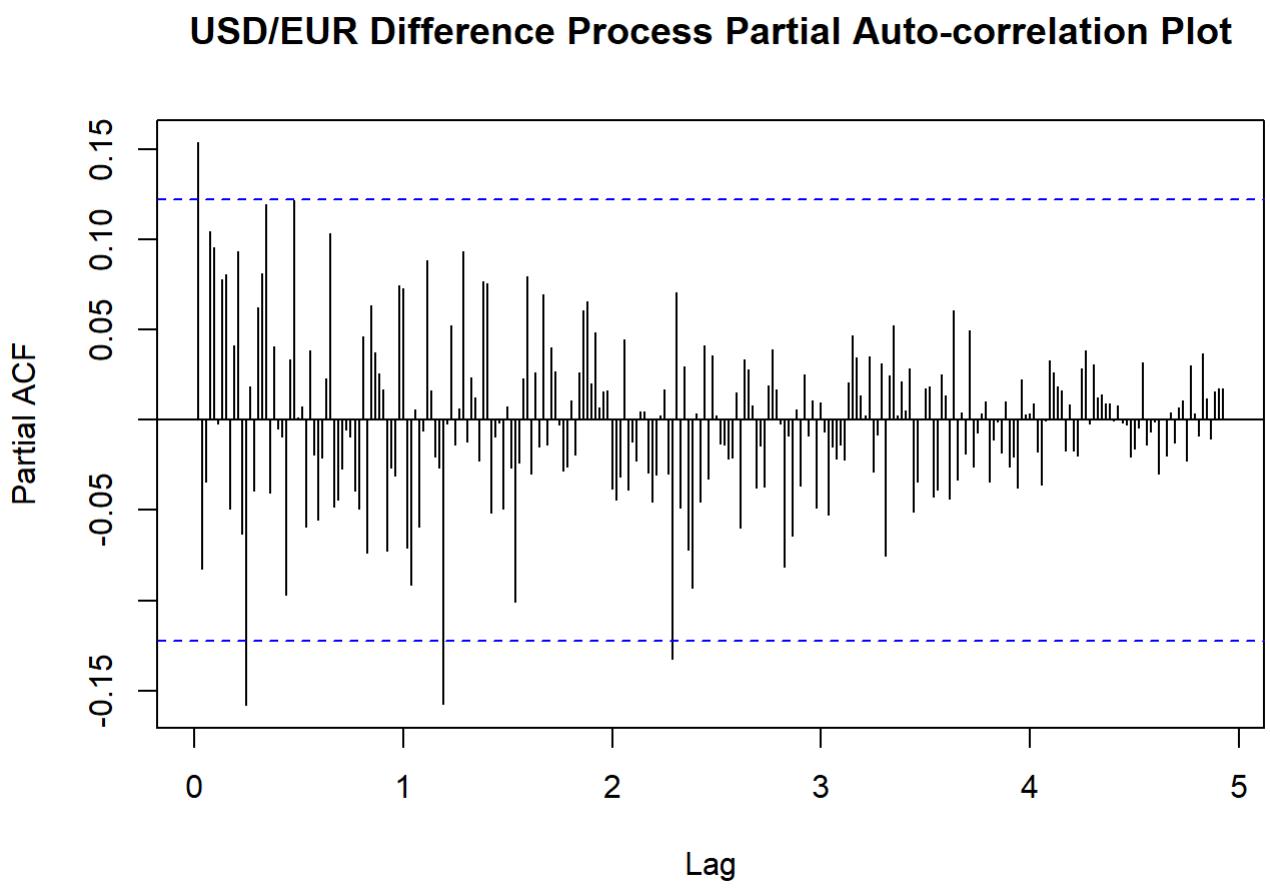

```
acf(USD.AU.diff,lag.max=52*6,main="USD/AUD Difference Process Auto-correlation Plot")
```



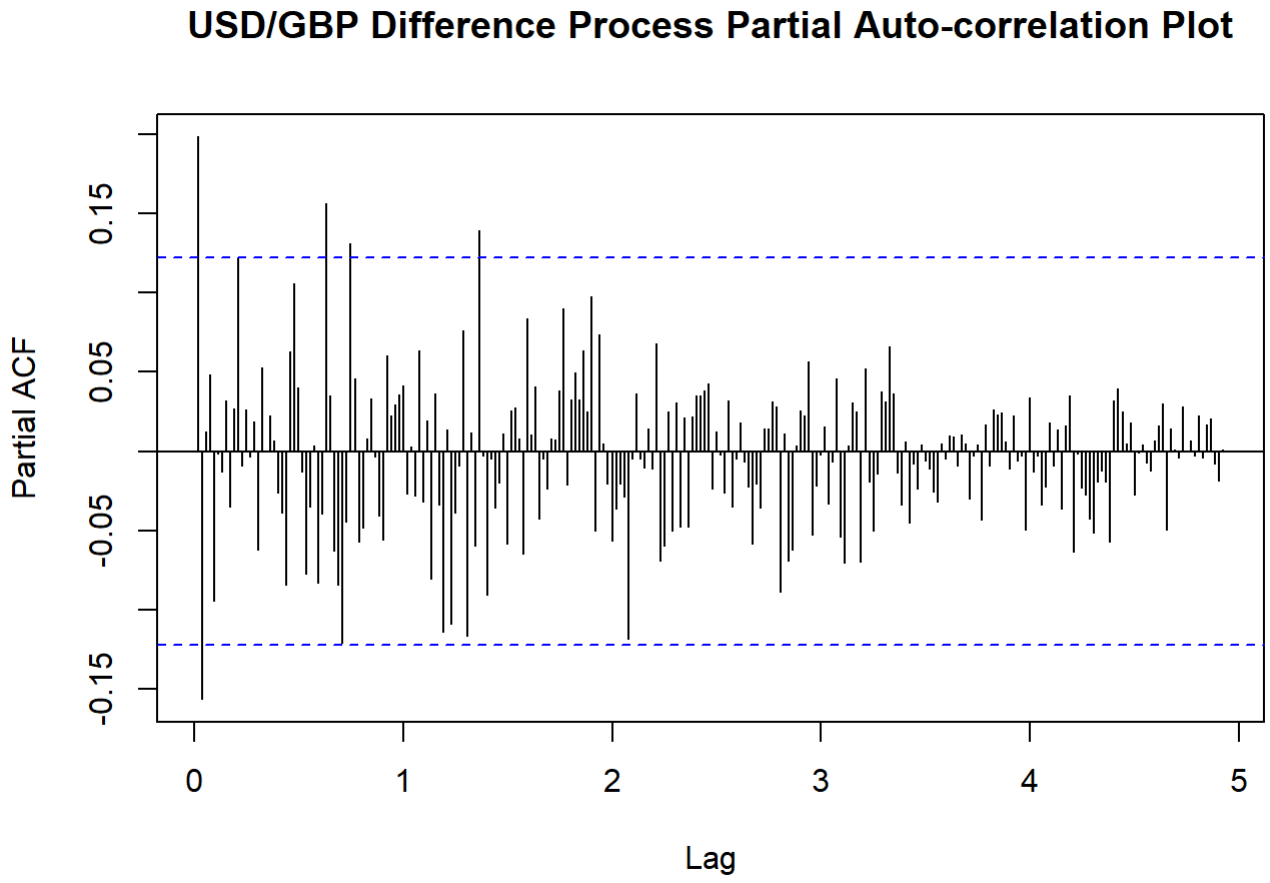

```
acf(USD.NZ.diff,lag.max=52*6,main="USD/NZ Difference Process Auto-correlation Plot")
```



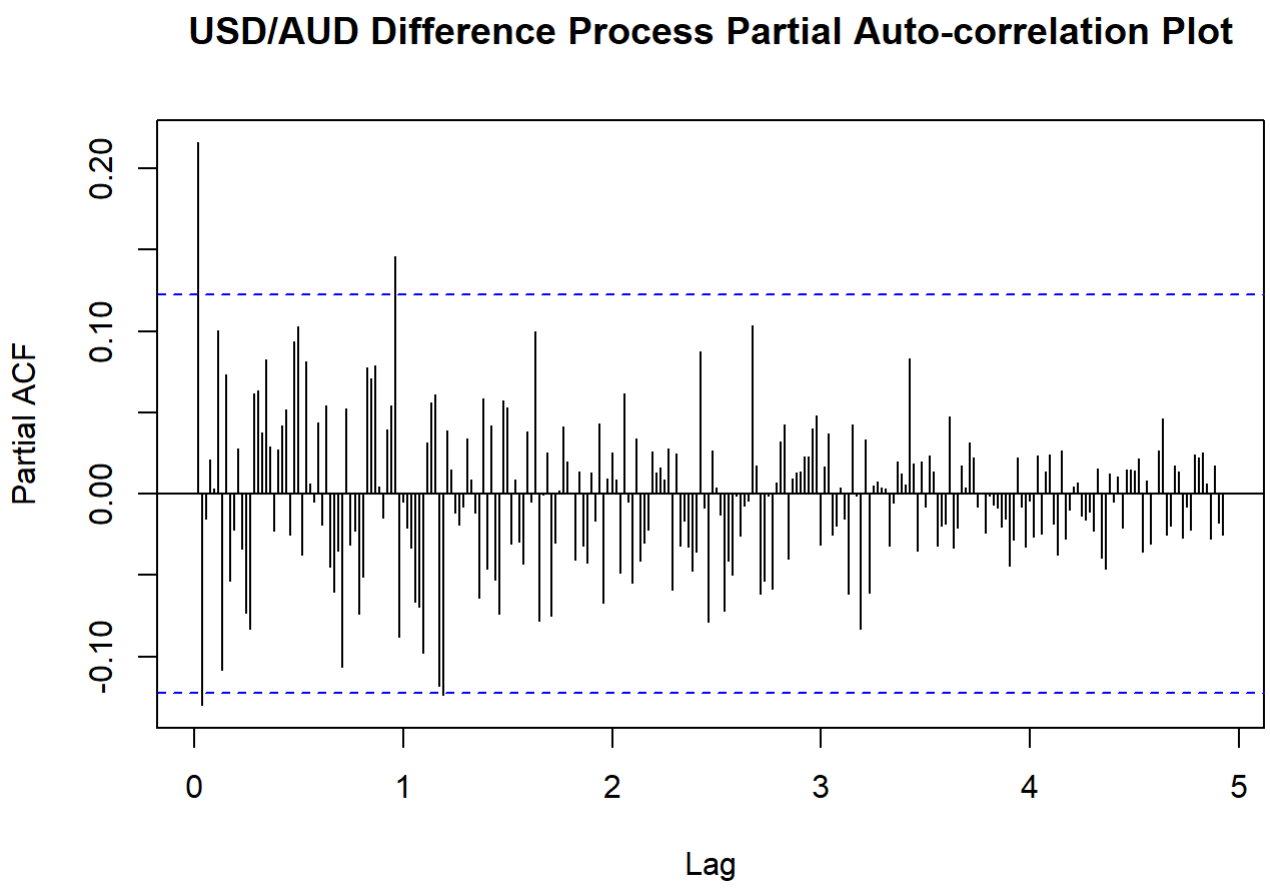

```
# Plot PACF plots
pacf(USD.EU.diff,lag.max=52*6,main="USD/EUR Difference Process Partial Auto-correlation Plot")
```



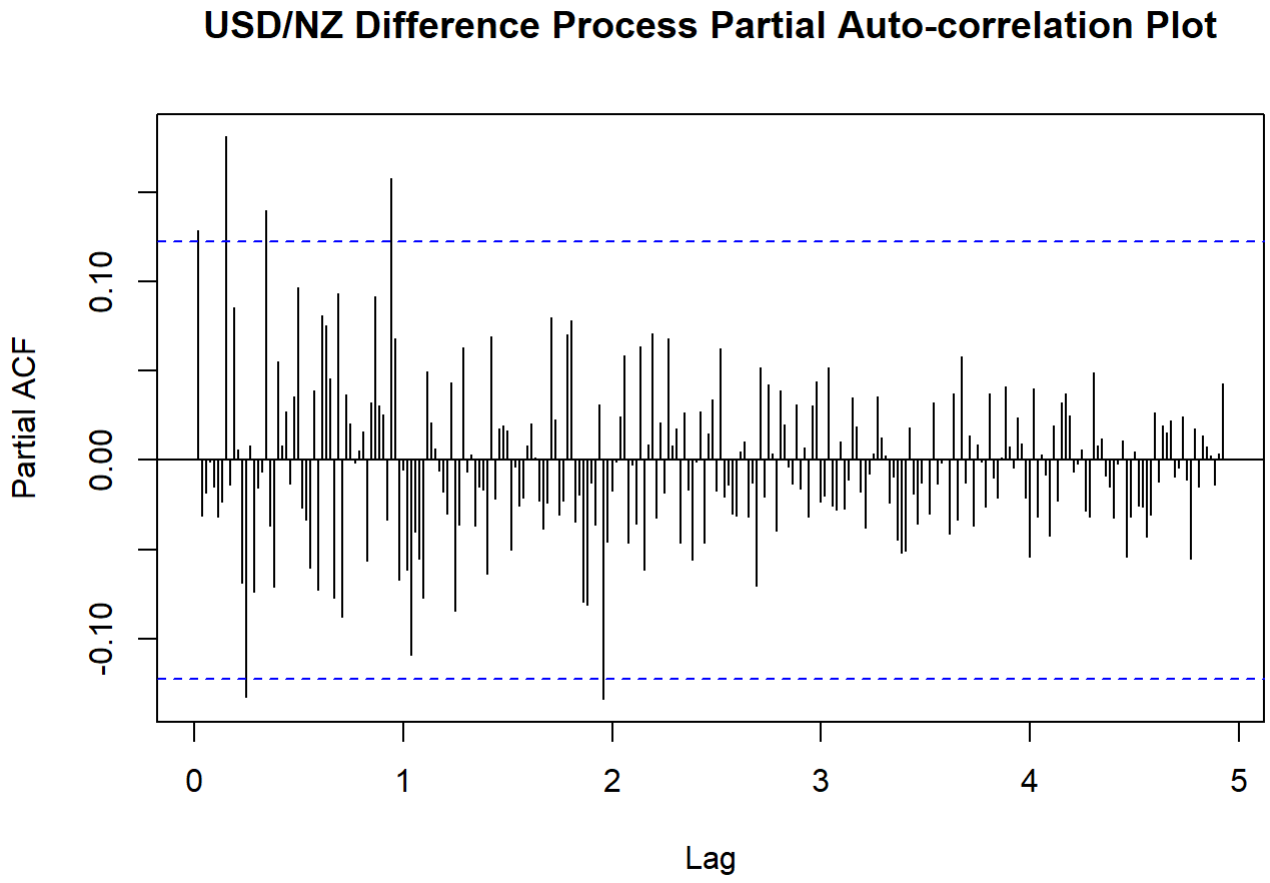

```
pacf(USD.GBP.diff,lag.max=52*6,main="USD/GBP Difference Process Partial Auto-correlation Plot")
```




```
pacf(USD.AU.diff,lag.max=52*6,main="USD/AUD Difference Process Partial Auto-correlation Plot")
```




```
pacf(USD.NZ.diff,lag.max=52*6,main="USD/NZ Difference Process Partial Auto-correlation Plot")
```



ADF Hypothesis Testing for Stationarity. Null Hypothesis: Time Series is Not Stationary

```
adf.test(USD.EU.diff)
```

```
## Warning in adf.test(USD.EU.diff): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  USD.EU.diff
## Dickey-Fuller = -5.0163, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(USD.GBP.diff)
```

```
## Warning in adf.test(USD.GBP.diff): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  USD.GBP.diff
## Dickey-Fuller = -6.377, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(USD.AU.diff)
```

```
## Warning in adf.test(USD.AU.diff): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  USD.AU.diff
## Dickey-Fuller = -6.1306, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(USD.NZ.diff)
```

```
## Warning in adf.test(USD.NZ.diff): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  USD.NZ.diff
## Dickey-Fuller = -6.4038, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

Answer

Looking at the differenced processes, we can observe a constant mean across all time series processes, which is ~ 0. However, based on these plots, the constant variance assumption does not seem to hold for the diff USD/EU as the variance decreases starting around year 2016. The diff USD/GBP variance seems constant other than a couple of downward spikes in year 2016. Diff USD/AUD process seems to have constant variance for the most part other than a couple of increasing variance around year 2016. Diff USD/NZ seems to have, for the most part, constant variance. The ACF and PACF plots of all 4 time series show few values outside the confidence bands, an indication for potential auto-correlation which would suggest that the time series are not stationary. However, based on the Augmented Dickey Fuller (ADF) test, which is a hypothesis testing for stationarity with a null hypothesis that the process is not stationary, we can argue that the processes are stationary. The ADF test's p-value is smaller than 0.01 which suggest to reject the null hypothesis of non-stationarity, and hence they are stationary.

Question 2: Univariate Analysis (28 pts)

a. (12 pts)

Use EACF plots (max order = 7) on the differenced data and iteration by AIC (original data, max order = 5, max differencing = 1, significance threshold of 2) to identify the best order for each series. How do the orders selected through each method compare?

EACF plots of differenced data

```
# EACF plots of differenced data
print("USD.EU.diff EACF")
```

```
## [1] "USD.EU.diff EACF"
```

```
eacf(USD.EU.diff, 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
```

```
print("USD.GBP.diff EACF")
```

```
## [1] "USD.GBP.diff EACF"
```

```
eacf(USD.GBP.diff, 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
```

```
print("USD.AU.diff EACF")
```

```
## [1] "USD.AU.diff EACF"
```

```
eacf(USD.AU.diff, 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
```

```
print("USD.NZ.diff EACF")
```

```
## [1] "USD.NZ.diff EACF"
```

```
eacf(USD.NZ.diff, 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
```

Iteration by AIC (original data, max order = 5, max differencing = 1, significance threshold of 2)

```
# Iteration by AIC (original data, max order = 5, max differencing = 1, significance threshold o
f 2)

# USD.EU ORIGINAL TRAIN data
order_arma = function(ts_data, max_p, max_d, max_q){
  orders = data.frame()
  for (p in 0:max_p){
    for (d in 0:max_d){
      for (q in 0:max_q) {
        possibleError <- tryCatch({
          mod = arima(ts_data, order=c(p,d,q), method="ML")
          current.aic = AIC(mod)
          #current.aic = current.aic-2*(p+q+1)+2*(p+q+1)*n/(n-p-q-2)
          orders<-rbind(orders,c(p,d,q,current.aic))
        },
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next

      }
    }
  }
  names(orders) <- c("p","d","q","AIC")
  return(orders[order(orders$AIC),])
}

max_p = 5
max_q = 5
max_d = 1
ts_data = EU.train

# Observe first best models based on smallest AICs
orders = order_arma(ts_data, max_p, max_d, max_q)
```

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

```
## 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
```

```
## 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

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

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

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

```
## 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

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

```
## 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

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

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

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

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

```
head(orders, 8)
```

```
##    p d q       AIC
## 33 2 1 2 -1597.578
## 46 3 1 3 -1596.823
## 34 2 1 3 -1596.075
## 45 3 1 2 -1596.049
## 72 5 1 5 -1595.922
## 39 3 0 2 -1595.806
## 60 4 1 5 -1595.657
## 71 5 1 4 -1595.548
```

```
# USD.GBP ORIGINAL TRAIN data
order_arma = function(ts_data, max_p, max_d, max_q){
  orders = data.frame()
  for (p in 0:max_p){
    for (d in 0:max_d){
      for (q in 0:max_q) {
        possibleError <- tryCatch({
          mod = arima(ts_data, order=c(p,d,q), method="ML")
          current.aic = AIC(mod)
          #current.aic = current.aic-2*(p+q+1)+2*(p+q+1)*n/(n-p-q-2)
          orders<-rbind(orders,c(p,d,q,current.aic))
        },
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next

      }
    }
  }
  names(orders) <- c("p","d","q","AIC")
  return(orders[order(orders$AIC),])
}

max_p = 5
max_q = 5
max_d = 1
ts_data = GBP.train

# Observe first best models based on smallest AICs
orders = order_arma(ts_data, max_p, max_d, max_q)
```

```
## 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

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

```
## 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
```

```
## 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
```

```
## 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
```

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

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

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

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

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

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

```
head(orders, 8)
```

```
##    p d q       AIC
## 31 2 1 0 -1446.670
## 8  0 1 1 -1446.502
## 9  0 1 2 -1445.553
## 27 2 0 2 -1445.208
## 10 0 1 3 -1445.184
## 20 1 1 1 -1445.136
## 22 1 1 3 -1445.037
## 33 2 1 2 -1444.853
```

```
# USD.AUD ORIGINAL TRAIN data
order_arma = function(ts_data, max_p, max_d, max_q){
  orders = data.frame()
  for (p in 0:max_p){
    for (d in 0:max_d){
      for (q in 0:max_q) {
        possibleError <- tryCatch({
          mod = arima(ts_data, order=c(p,d,q), method="ML")
          current.aic = AIC(mod)
          #current.aic = current.aic-2*(p+q+1)+2*(p+q+1)*n/(n-p-q-2)
          orders<-rbind(orders,c(p,d,q,current.aic))
        },
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next

      }
    }
  }
  names(orders) <- c("p","d","q","AIC")
  return(orders[order(orders$AIC),])
}

max_p = 5
max_q = 5
max_d = 1
ts_data = AU.train

# Observe first best models based on smallest AICs
orders = order_arma(ts_data, max_p, max_d, max_q)
```

```
## 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

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

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

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

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

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

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

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

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

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

```
## 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
```

```
## 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

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

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

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

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

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

```
head(orders, 8)
```

```
##    p d q       AIC
## 51 4 0 2 -1751.555
## 58 4 1 3 -1750.182
## 52 4 0 3 -1749.814
## 8  0 1 1 -1749.481
## 16 1 0 3 -1749.409
## 71 5 1 4 -1748.944
## 22 1 1 3 -1748.939
## 31 2 1 0 -1748.581
```

```
# USD.NZ ORIGINAL TRAIN data
order_arma = function(ts_data, max_p, max_d, max_q){
  orders = data.frame()
  for (p in 0:max_p){
    for (d in 0:max_d){
      for (q in 0:max_q) {
        possibleError <- tryCatch({
          mod = arima(ts_data, order=c(p,d,q), method="ML")
          current.aic = AIC(mod)
          #current.aic = current.aic-2*(p+q+1)+2*(p+q+1)*n/(n-p-q-2)
          orders<-rbind(orders,c(p,d,q,current.aic))
        },
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next

      }
    }
  }
  names(orders) <- c("p","d","q","AIC")
  return(orders[order(orders$AIC),])
}

max_p = 5
max_q = 5
max_d = 1
ts_data = NZ.train

# Observe first best models based on smallest AICs
orders = order_arma(ts_data, max_p, max_d, max_q)
```

```
## 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

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

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

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

```
## 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

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

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

```
head(orders, 8)
```

```
##    p d q       AIC
## 72 5 1 5 -1718.631
## 70 5 1 3 -1718.098
## 27 2 0 2 -1715.046
## 8  0 1 1 -1714.680
## 19 1 1 0 -1714.565
## 38 3 0 1 -1714.378
## 39 3 0 2 -1713.645
## 28 2 0 3 -1713.394
```

Answer

Based on the EACF plots, the order selection is ARMA(0,1) for all ***differenced*** processes, which is equivalent to ARIMA(0,1,1) with respect to the original train data.

Based on the iteration approach, the order selection is:

EU.train - ***ARIMA(2,1,2)***

GBP.train - ***ARIMA(0,1,1)*** Note that the smallest AIC is -1446.670 for ARMA(2,1,0), but due to the significance threshold of 2 defined in the question, we select AMRA(0,1,1) with AIC of -1446.502

AU.train - ***ARIMA(4,0,2)*** with AIC of -1751.555. Note that the order of (0,1,1) has an AIC of -1749.481. While the difference between the two is 2.074, it is still greater than the 2 threshold hence selected ARIMA(4,0,2).

NZ.train - ***ARIMA(5,1,3)*** Note that the smallest AIC is -1718.631 for ARMA(5,1,5), but due to the significance threshold of 2 defined in the question, we select AMRA(5,1,3) with AIC of -1718.098

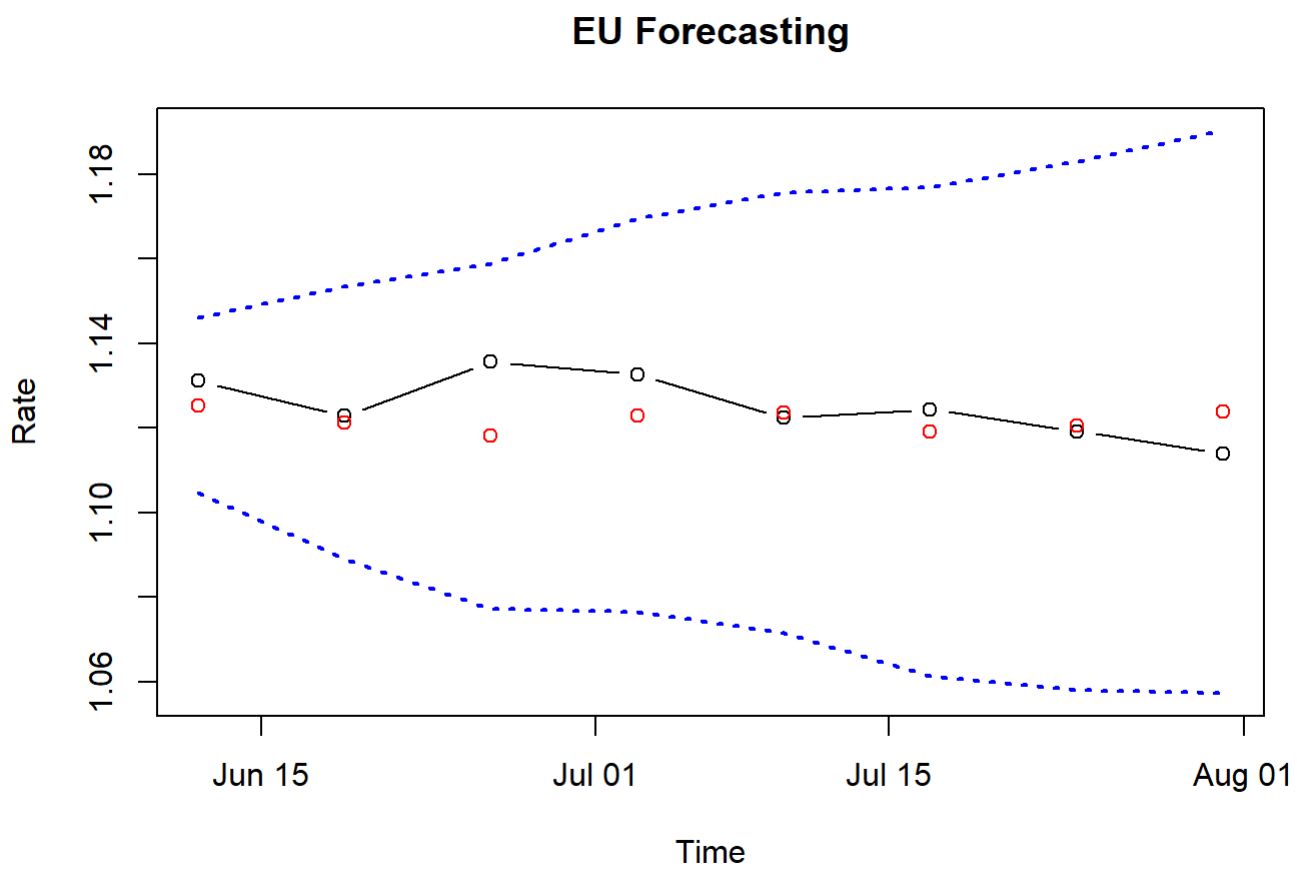
b. (12 pts)

Regardless of what you observed above, fit the following ARIMA orders: EUR (2,1,3), GBP (1,1,1), AUD (1,0,3), NZD (2,0,2). Using these models, predict ahead 8 points. Plot these points with their 95% confidence intervals against the test data for each series.

```
# Fit models
model_EU = arima(EU.train,order=c(2,1,3), method="ML")
model_GBP = arima(GBP.train,order=c(1,1,1), method="ML")
model_AUD = arima(AU.train,order=c(1,0,3), method="ML")
model_NZ = arima(NZ.train,order=c(2,0,2), method="ML")
```

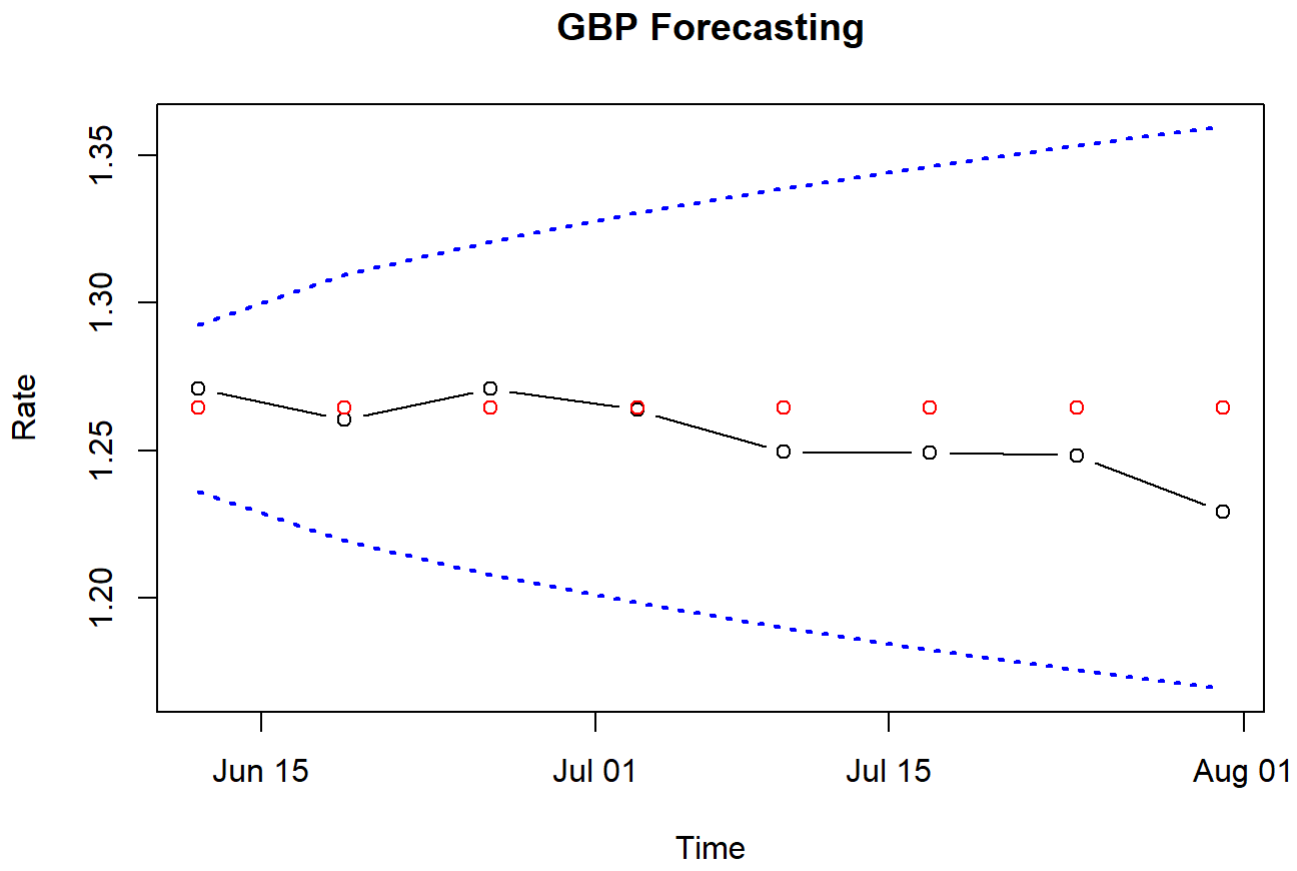
EU Prediction

```
d = test$DATE
d = as.Date(d, format = "%m/%d/%Y")


pred_EU = predict(model_EU,n.ahead=8)
ubound = pred_EU$pred+1.96*pred_EU$se
lbound = pred_EU$pred-1.96*pred_EU$se
ymin = min(lbound)
ymax = max(ubound)

plot(d, test$USD.EU, type="b", ylim=c(ymin,ymax), xlab="Time", ylab="Rate", main="EU Forecastin
g")
points(d, pred_EU$pred,col="red")
lines(d, ubound,lty=3,lwd= 2, col="blue")
lines(d, lbound,lty=3,lwd= 2, col="blue")
```



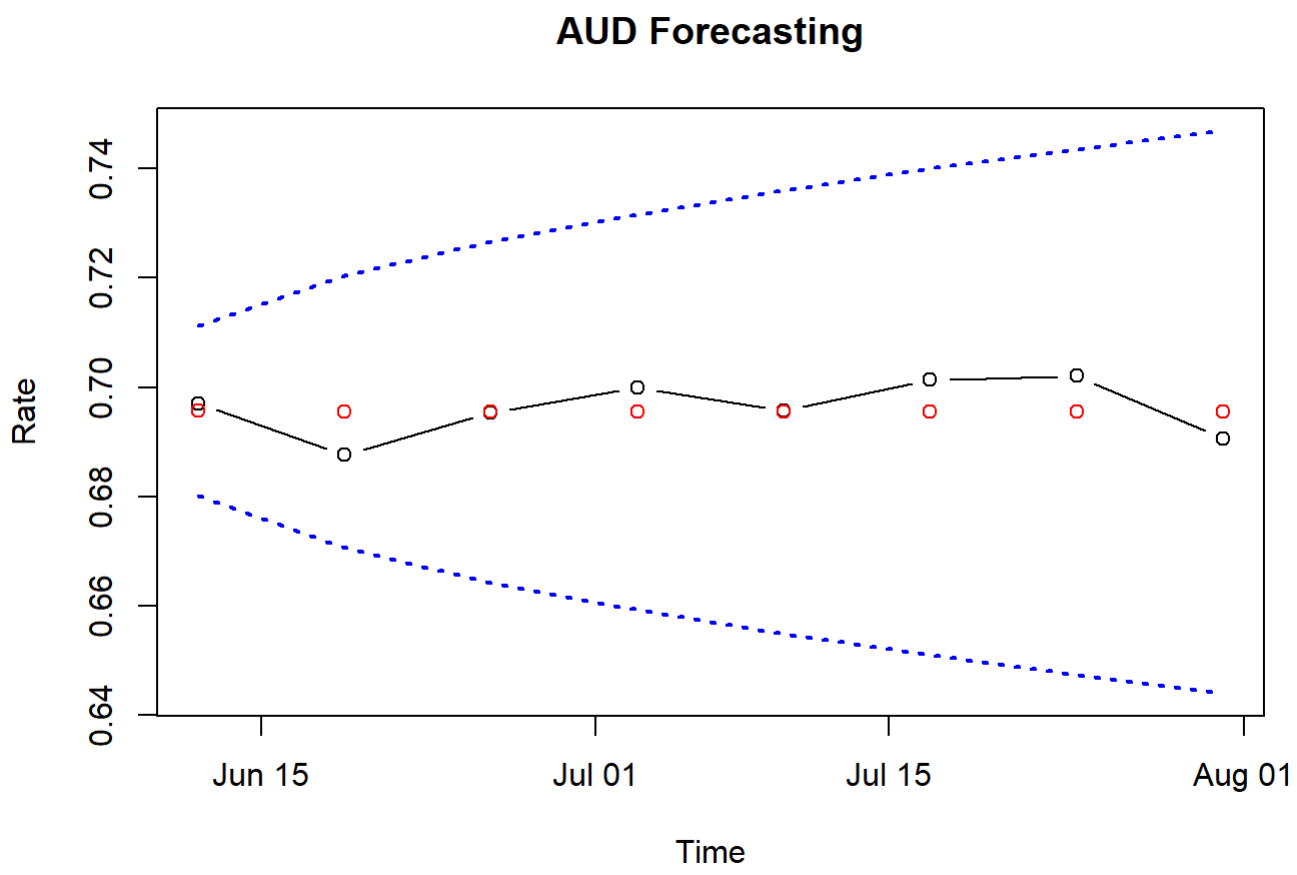
GBP Prediction

```
pred_GBP = predict(model_GBP,n.ahead=8)
ubound = pred_GBP$pred+1.96*pred_GBP$se
lbound = pred_GBP$pred-1.96*pred_GBP$se
ymin = min(lbound)
ymax = max(ubound)

plot(d, test$USD.GBP, type="b", ylim=c(ymin,ymax), xlab="Time", ylab="Rate", main="GBP Forecasti
ng")
points(d, pred_GBP$pred,col="red")
lines(d, ubound,lty=3,lwd= 2, col="blue")
lines(d, lbound,lty=3,lwd= 2, col="blue")
```



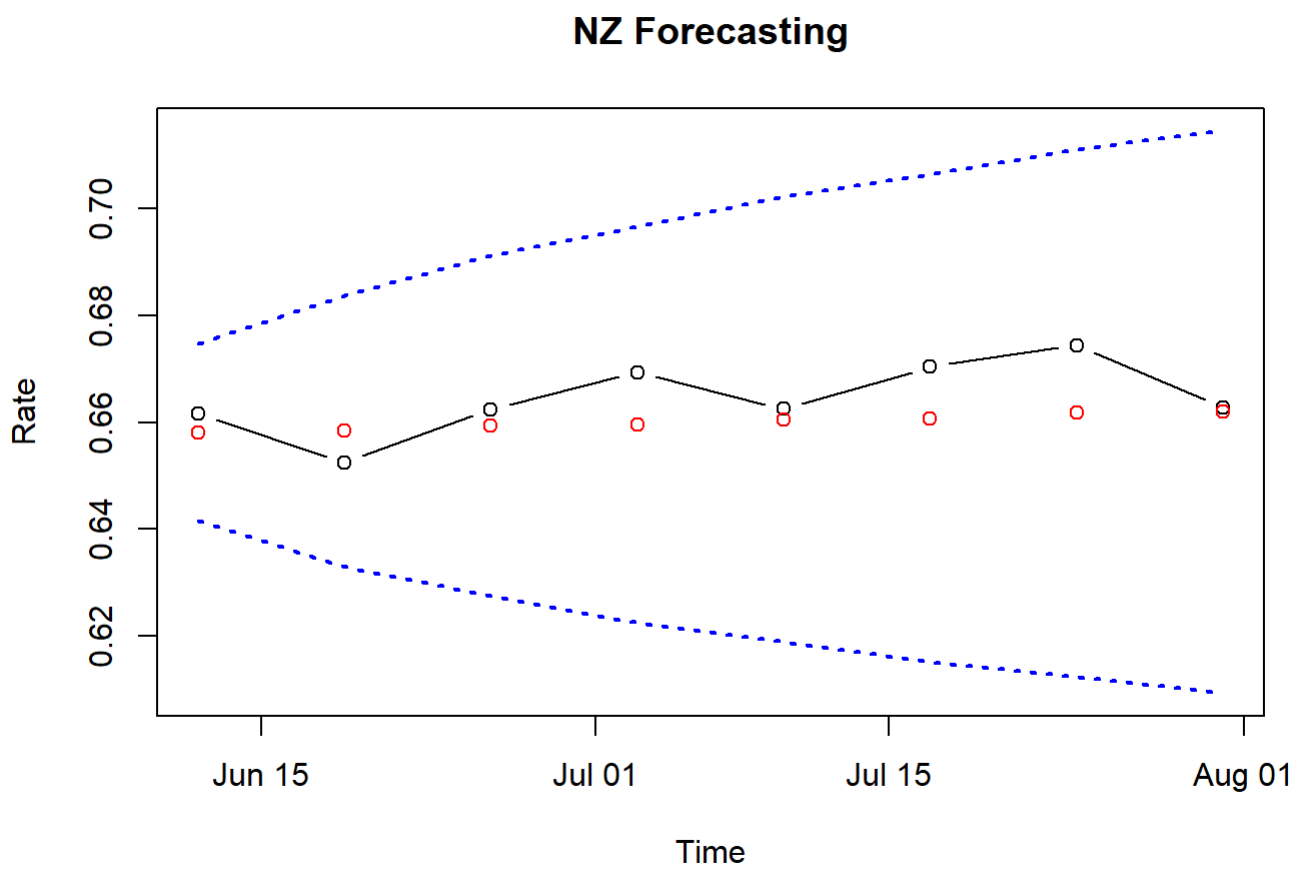
AUD Prediction

```
pred_AUD = predict(model_AUD,n.ahead=8)
ubound = pred_AUD$pred+1.96*pred_AUD$se
lbound = pred_AUD$pred-1.96*pred_AUD$se
ymin = min(lbound)
ymax = max(ubound)

plot(d, test$USD.AUD, type="b", ylim=c(ymin,ymax), xlab="Time", ylab="Rate", main="AUD Forecasti
ng")
points(d, pred_AUD$pred,col="red")
lines(d, ubound,lty=3,lwd= 2, col="blue")
lines(d, lbound,lty=3,lwd= 2, col="blue")
```



NZ Prediction

```
pred_NZ = predict(model_NZ,n.ahead=8)
ubound = pred_NZ$pred+1.96*pred_NZ$se
lbound = pred_NZ$pred-1.96*pred_NZ$se
ymin = min(lbound)
ymax = max(ubound)

plot(d, test$USD.NZ, type="b", ylim=c(ymin,ymax), xlab="Time", ylab="Rate", main="NZ Forecastin
g")
points(d, pred_NZ$pred,col="red")
lines(d, ubound,lty=3,lwd= 2, col="blue")
lines(d, lbound,lty=3,lwd= 2, col="blue")
```



Answer

See above the predicted against the observed values with 95% confidence intervals.

c. (4 pts)

For each set of predictions, compute mean absolute percentage error (MAPE). How do they compare in accuracy?

```
#Mean Absolute Percentage Error (MAPE)
MAPE_EU = mean(abs(pred_EU$pred - test$USD.EU)/abs(test$USD.EU))
MAPE_GBP = mean(abs(pred_GBP$pred - test$USD.GBP)/abs(test$USD.GBP))
MAPE_AUD = mean(abs(pred_AUD$pred - test$USD.AUD)/abs(test$USD.AUD))
MAPE_NZ = mean(abs(pred_NZ$pred - test$USD.NZ)/abs(test$USD.NZ))

print(paste("EU MAPE:", MAPE_EU))
```

```
## [1] "EU MAPE: 0.00578939671665407"
```

```
print(paste("GBP MAPE:", MAPE_GBP))
```

```
## [1] "GBP MAPE: 0.00995483842353229"
```

```
print(paste("AUD MAPE:", MAPE_AUD))
```

```
## [1] "AUD MAPE: 0.00559831107126138"
```

```
print(paste("NZ MAPE:", MAPE_NZ))
```

```
## [1] "NZ MAPE: 0.00884671033615939"
```

Answer

MAPE, calculated as the mean of the absolute values of the differences between the foretasted and actual values, scaled by the actual responses is 0.005789, 0.009954, 0.005598 and 0.008846 for EU, GBP, AUD and NZ, respectively. Based on these results, seems like the models predicted quite well with the AUD model most accurate prediction and smallest error.

Question 3: Multivariate Analysis (28 pts)

a. (4 pts)

Identify the best VAR model that fits the four time series simultaneously according to AIC.

```
var_order = VAR(train.ts,lag.max=20,ic="AIC", type="const")
var_order
```

```
##
## 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.l
2 + 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.l
2 + 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
```

```
VARselect(train.ts, lag.max=20)$selection
```

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

```
var_order$p
```

```
## AIC(n)
##      2
```

Answer

According to AIC, the VAR model is VAR(2)

b. (12 pts)

Fit a VAR(2) model and check the model fitting using the multivariate ARCH test, the Jarque-Bera test and the Portmanteau test. State which assumptions are satisfied and which are violated

```
var = VAR(train.ts,p=2)
var_residuals=residuals(var)

arch.test(var)
```

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

```
normality.test(var)
```

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

```
serial.test(var)
```

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

Answer

The ARCH test is used to evaluate whether the residuals have constant variance. Since the p-value is approximately zero, we reject the null hypothesis of constant variance.

The Jarque-Bera test is used to evaluate whether the residuals are normally distributed. The null hypothesis is that the residuals are normally distributed and with a p-value of approximately 0, we reject the null and conclude that the residuals are not normally distributed.

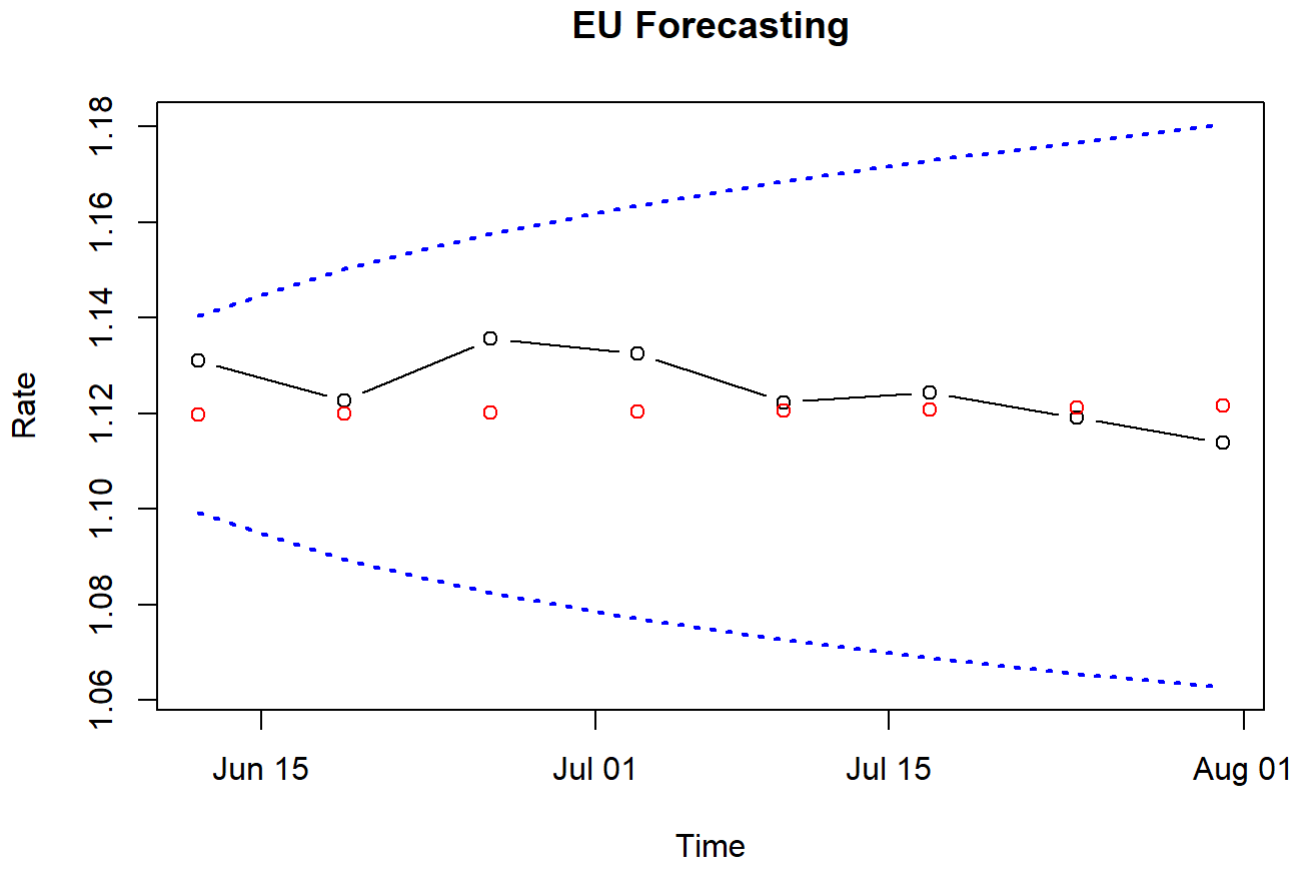
The Portmanteau Test is used to evaluate whether the residuals are uncorrelated. Since the p-value is smaller than 0.05, we reject the null hypothesis of uncorrelated residuals at α = 0.05 (but not α = 0.01).

c. (12 pts)

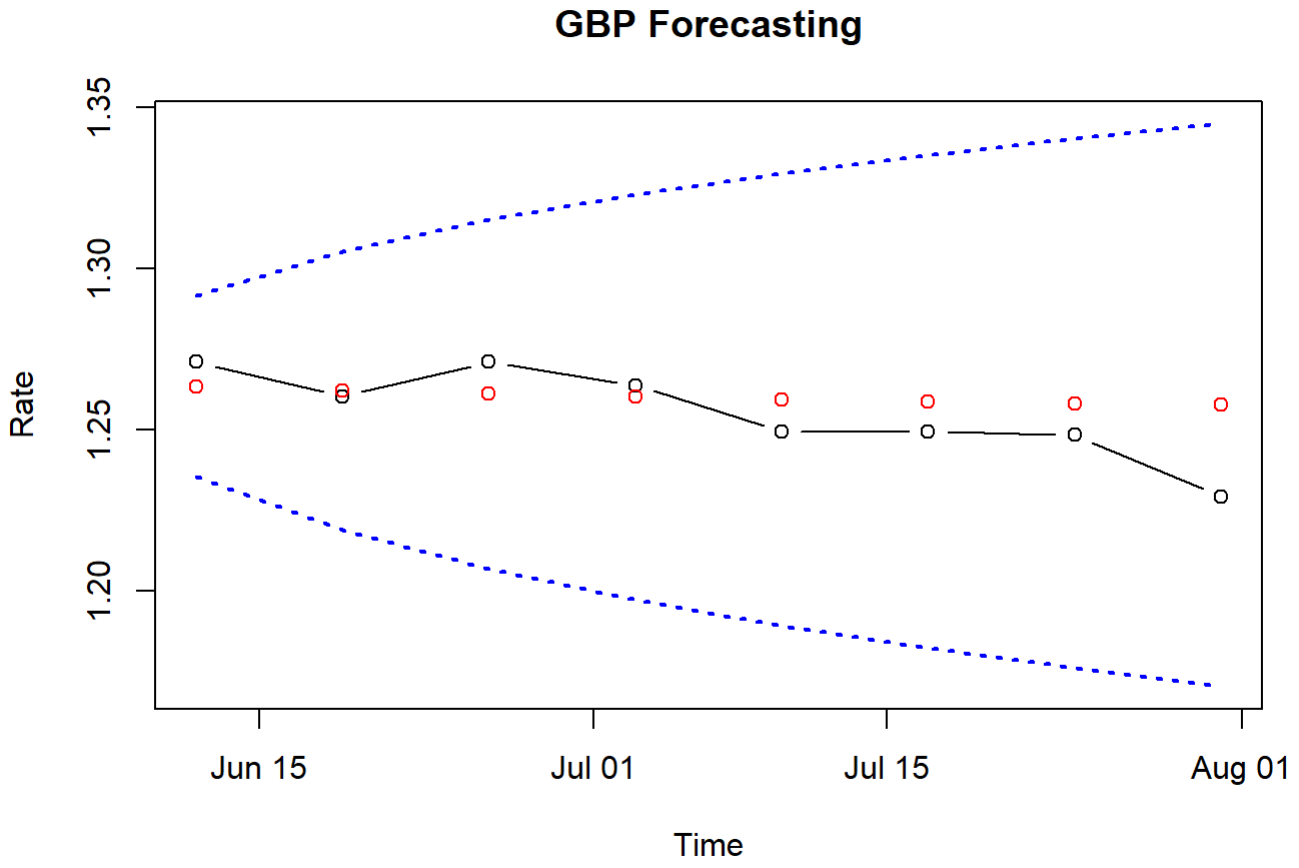
Give the forecasts for the next 8 weeks of the series using a VAR(2) model. You don't need to plot them (but can if you'd like). Include confidence intervals.

Using mean absolute percentage error, compare the predictions for the four time series derived from the univariate analysis and multivariate analysis.

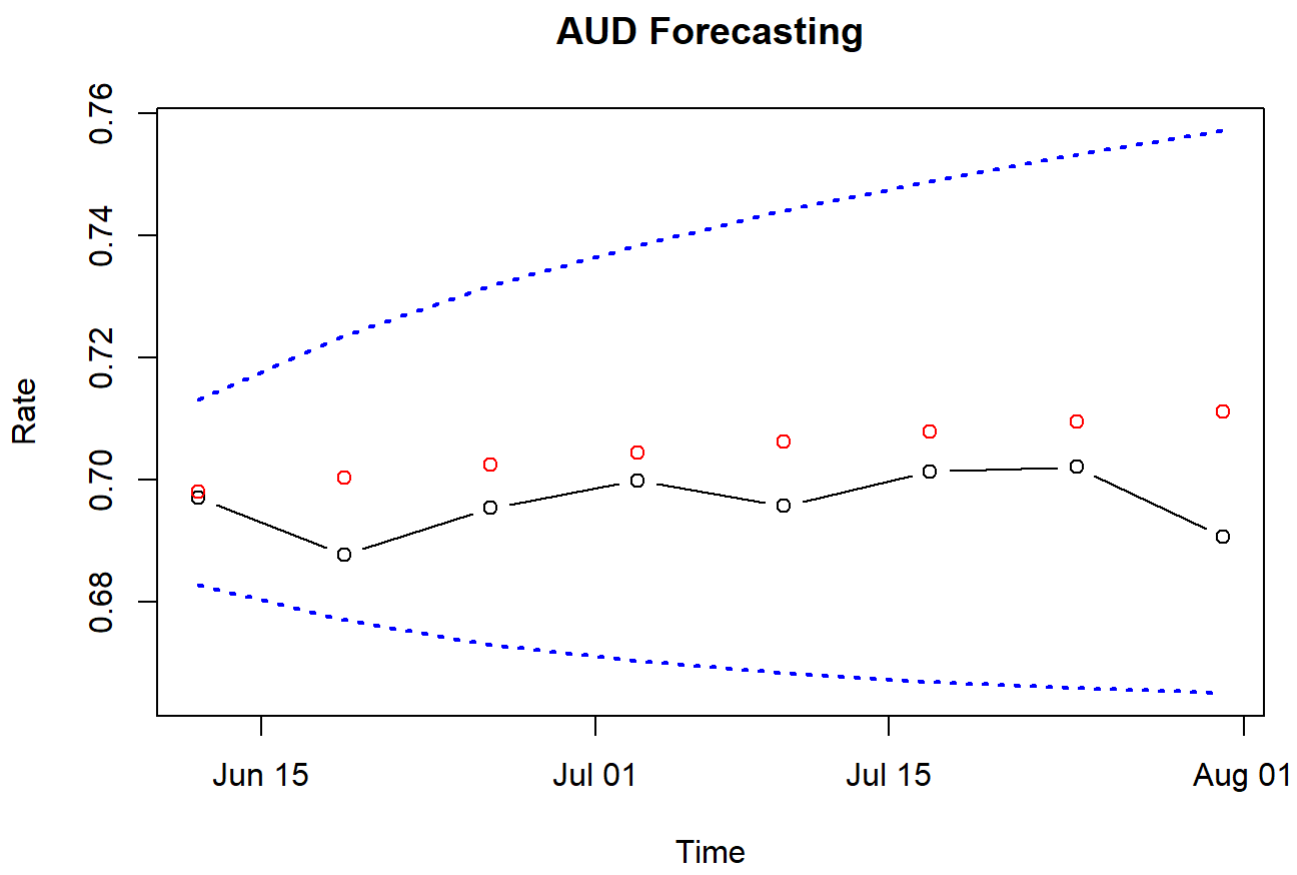
```
# Prediction
var_pred = predict(var,n.ahead=8)

# Plot
# EU
ubound = var_pred$fcst$USD.EU[,3]
lbound = var_pred$fcst$USD.EU[,2]
ymin = min(lbound)
ymax = max(ubound)

plot(d, test$USD.EU, type="b", ylim=c(ymin,ymax), xlab="Time", ylab="Rate", main="EU Forecastin
g")
points(d, var_pred$fcst$USD.EU[,1],col="red")
lines(d, ubound,lty=3,lwd= 2, col="blue")
lines(d, lbound,lty=3,lwd= 2, col="blue")
```



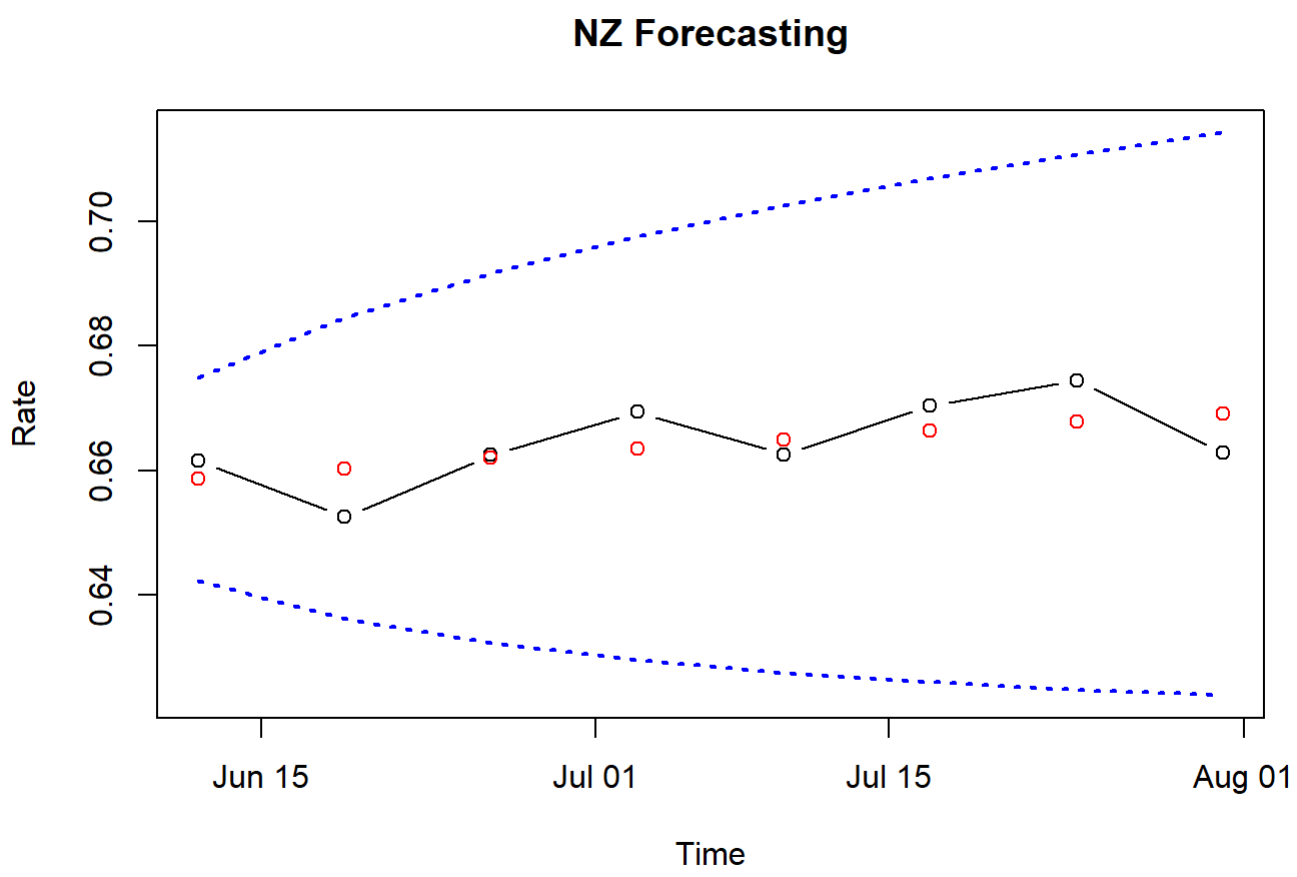
```
##############################################################################
# GBP
ubound = var_pred$fcst$USD.GBP[,3]
lbound = var_pred$fcst$USD.GBP[,2]
ymin = min(lbound)
ymax = max(ubound)

plot(d, test$USD.GBP, type="b", ylim=c(ymin,ymax), xlab="Time", ylab="Rate", main="GBP Forecasti
ng")
points(d, var_pred$fcst$USD.GBP[,1],col="red")
lines(d, ubound,lty=3,lwd= 2, col="blue")
lines(d, lbound,lty=3,lwd= 2, col="blue")
```




```
##############################################################################
# AUD
ubound = var_pred$fcst$USD.AUD[,3]
lbound = var_pred$fcst$USD.AUD[,2]
ymin = min(lbound)
ymax = max(ubound)

plot(d, test$USD.AUD, type="b", ylim=c(ymin,ymax), xlab="Time", ylab="Rate", main="AUD Forecasti
ng")
points(d, var_pred$fcst$USD.AUD[,1],col="red")
lines(d, ubound,lty=3,lwd= 2, col="blue")
lines(d, lbound,lty=3,lwd= 2, col="blue")
```




```
################################################################################
# NZ
ubound = var_pred$fcst$USD.NZ[,3]
lbound = var_pred$fcst$USD.NZ[,2]
ymin = min(lbound)
ymax = max(ubound)

plot(d, test$USD.NZ, type="b", ylim=c(ymin,ymax), xlab="Time", ylab="Rate", main="NZ Forecastin
g")
points(d, var_pred$fcst$USD.NZ[,1],col="red")
lines(d, ubound,lty=3,lwd= 2, col="blue")
lines(d, lbound,lty=3,lwd= 2, col="blue")
```




```
print("USD.EU prediction, including confidence interval:")
```

```
## [1] "USD.EU prediction, including confidence interval:"
```

```
var_pred$fcst$USD.EU
```

```
##          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
```

```
print("USD.GBP prediction, including confidence interval:")
```

```
## [1] "USD.GBP prediction, including confidence interval:"
```

```
var_pred$fcst$USD.GBP
```

```
##          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
```

```
print("USD.AUD prediction, including confidence interval:")
```

```
## [1] "USD.AUD prediction, including confidence interval:"
```

```
var_pred$fcst$USD.AUD
```

```
##           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
```

```
print("USD.NZ prediction, including confidence interval:")
```

```
## [1] "USD.NZ prediction, including confidence interval:"
```

```
var_pred$fcst$USD.NZ
```

```
##           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
```

```
MAPE_EU_VAR = mean(abs(var_pred$fcst$USD.EU[,1] - test$USD.EU)/abs(test$USD.EU))
MAPE_GBP_VAR = mean(abs(var_pred$fcst$USD.GBP[,1] - test$USD.GBP)/abs(test$USD.GBP))
MAPE_AUD_VAR = mean(abs(var_pred$fcst$USD.AUD[,1] - test$USD.AUD)/abs(test$USD.AUD))
MAPE_NZ_VAR = mean(abs(var_pred$fcst$USD.NZ[,1] - test$USD.NZ)/abs(test$USD.NZ))

print(paste("EU MAPE:", MAPE_EU_VAR))
```

```
## [1] "EU MAPE: 0.00630362330751923"
```

```
print(paste("GBP MAPE:", MAPE_GBP_VAR))
```

```
## [1] "GBP MAPE: 0.00804543198111515"
```

```
print(paste("AUD MAPE:", MAPE_AUD_VAR))
```

```
## [1] "AUD MAPE: 0.012646812433796"
```

```
print(paste("NZ MAPE:", MAPE_NZ_VAR))
```

```
## [1] "NZ MAPE: 0.00689098299319095"
```

Answer

See above the plots of the predicted values against the observed values for each pairs of currency using the multivariate VAR(2) model.

The MAPE results predicting with the univariate ARIMA model are:

“EU MAPE: 0.00578939671665407”

“GBP MAPE: 0.00995483842353229”

“AUD MAPE: 0.00559831107126138”

“NZ MAPE: 0.00884671033615939”

The MAPE results predicting with the multivariate VAR model are:

“EU MAPE: 0.00630362330751923”

“GBP MAPE: 0.00804543198111515”

“AUD MAPE: 0.012646812433796”

“NZ MAPE: 0.00689098299319095”

Comparing between the MAPEs, we conclude that the univariate ARIMA model predicted better for USD/EU and USD/AUD, but the multivariate predicted better for USD/GBP and USD/NZ.

Question 4: Reflection (6 pts)

From what you encountered above and your conceptual understanding of VAR modeling, reflect on the relative strengths and weaknesses of each method of modeling. When would you suggest one method over the other?

Answer

The strength of the VAR model is that it allows to include other time series processes that might affect the change in a target time series process. However, it is a little harder to interpret than the univariate ARIMA model. With multivariate VAR model, one could also better understand the relationship between time series processes via co-integration analysis, which is obviously impossible to do with the univariate modeling. As we see above however, using the more complex model does not necessarily give better predictions (as seen for USD/EU and USD/AUD). I would definitely suggest to use or at least explore the multivariate modeling approach when we believe that a target time series could be affected or predicted using other time series processes.

Thank You!