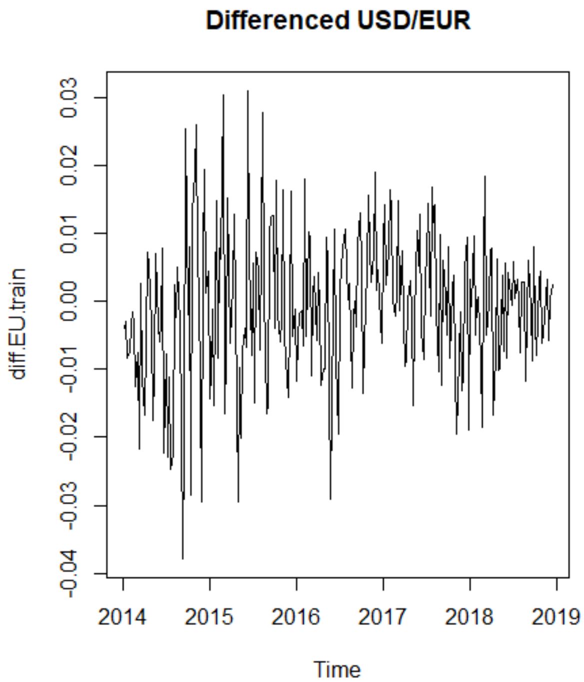
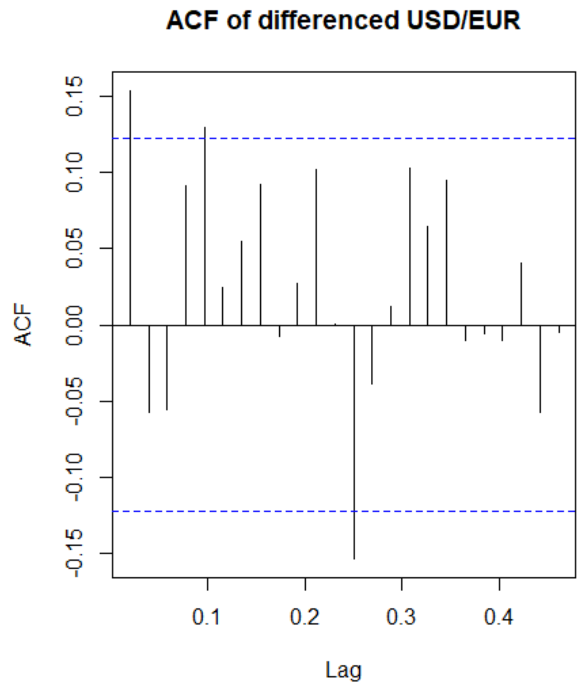
npark62 – 3/22/20 – ISYE 6402 – HW3 peer response

Question 1: Data Exploration (8 pts)



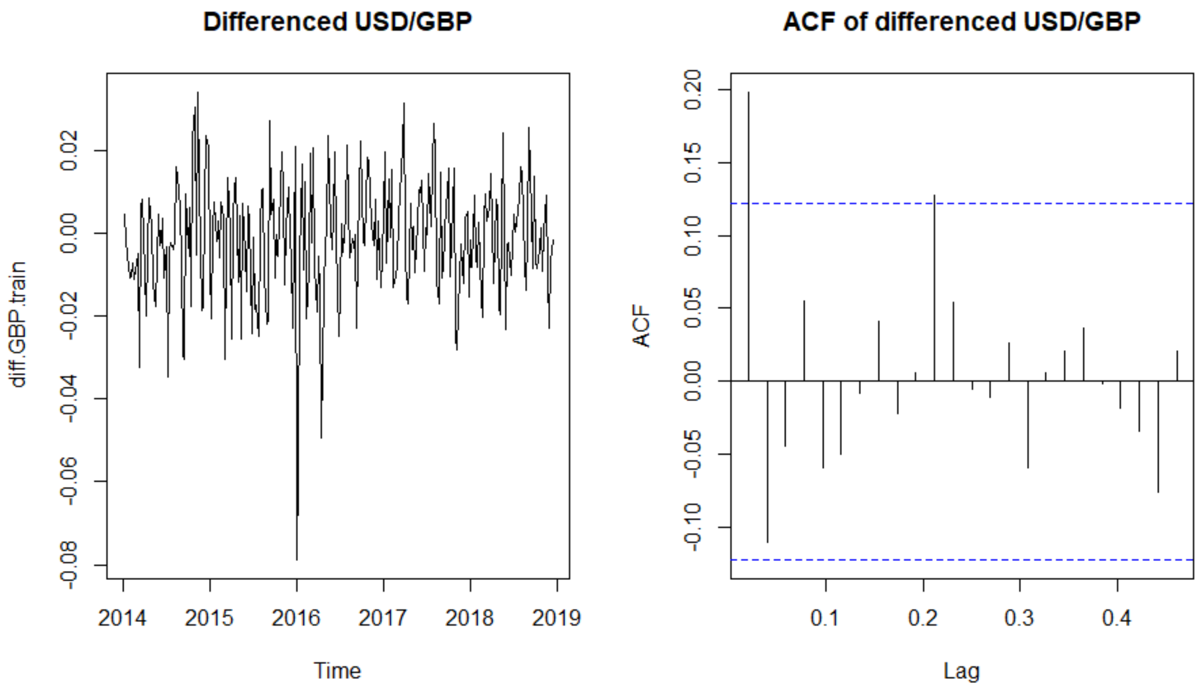


The differenced time series is more constant than the original, but still seems to have some inconsistencies.

There is clear non-constant variance (heteroskedasticity) observed

The autocorrelation shows some significant lags

In summary, not stationary.

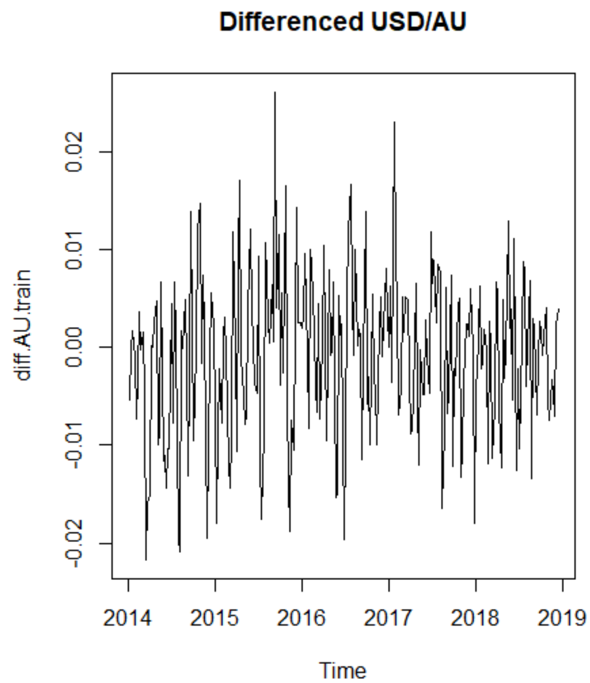


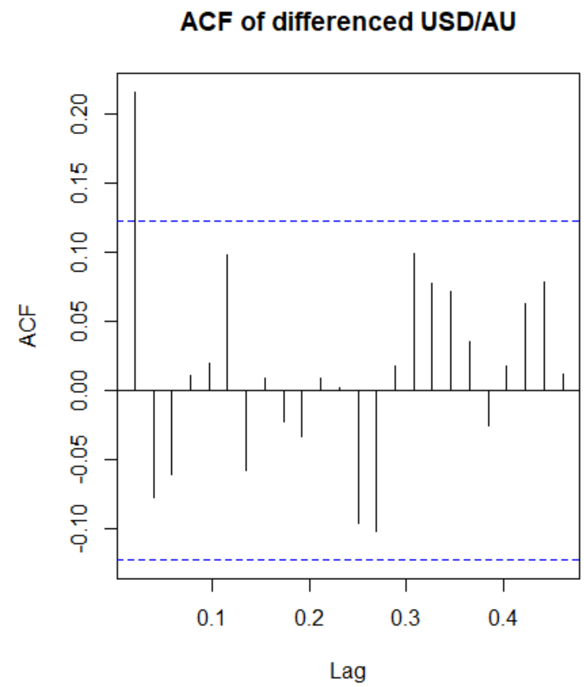
The mean of the differenced time series is quite constant

The constnat seems finite and constant except for a spike at 2016.

No siginificant correlation is observed from the ACF

In summary, not perfectly statonary, but has some characteristics.



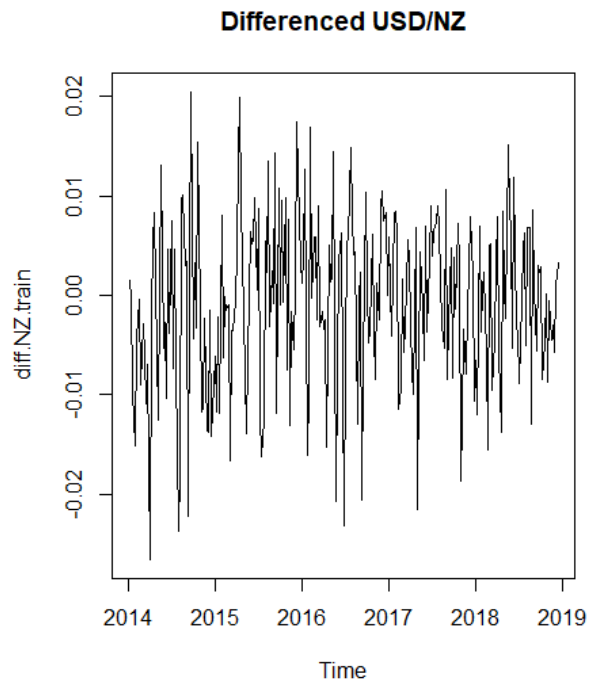


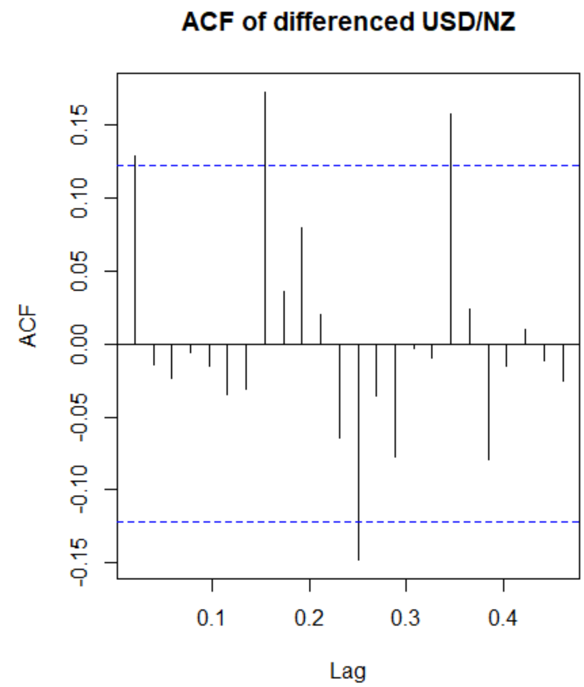
The mean of the differenced time series is not strictly constant.

The variance seems to be non-constant.

There is no significant autocorrelation.

In summary, not stationary.





The mean of the differenced time series still is not strictly constant.

The variance is non-constant

The autocorrelation shows some significant lags

In summary, not stationary.

Question 2: Univariate Analysis (28 pts)

a. (12 pts)

For this question, we used lower tail AIC to grab minimum AIC (using threshold 2 preferring simplicity to avoid overfitting) for all four cases.

Below shows the EACF table and the orders of ARIMA chosen from lowest tail AIC for EU.

eacf(diff.EU.train, 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

# p d q       AIC
# 2 1 2 -1597.578
```

The EACF includes the lowest AIC selection with threshold 2 (p,d,q) = (2,1,2) as a viable option but still has other simpler order selection. Note that eacf was used on already differenced data, d = 1.

Below shows the EACF table and the orders of ARIMA chosen from lowest tail AIC for GBP.

eacf(diff.GBP.train, 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

p d q   AIC

0 1 1 -1446.502

The EACF includes the lowest AIC selection with threshold 2 (p,d,q) = (0,1,1) as a viable option but still has other simpler order selection.

Below shows the EACF table and the orders of ARIMA chosen from lowest tail AIC for AU.

eacf(diff.AU.train, 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

```
# p d q       AIC
# 0 1 1 -1749.481
# The EACF reflects the lowest AIC selection with threshold 2 (p,d,q) = (0,1,1) as a viable option but still has other simpler order selection.
```

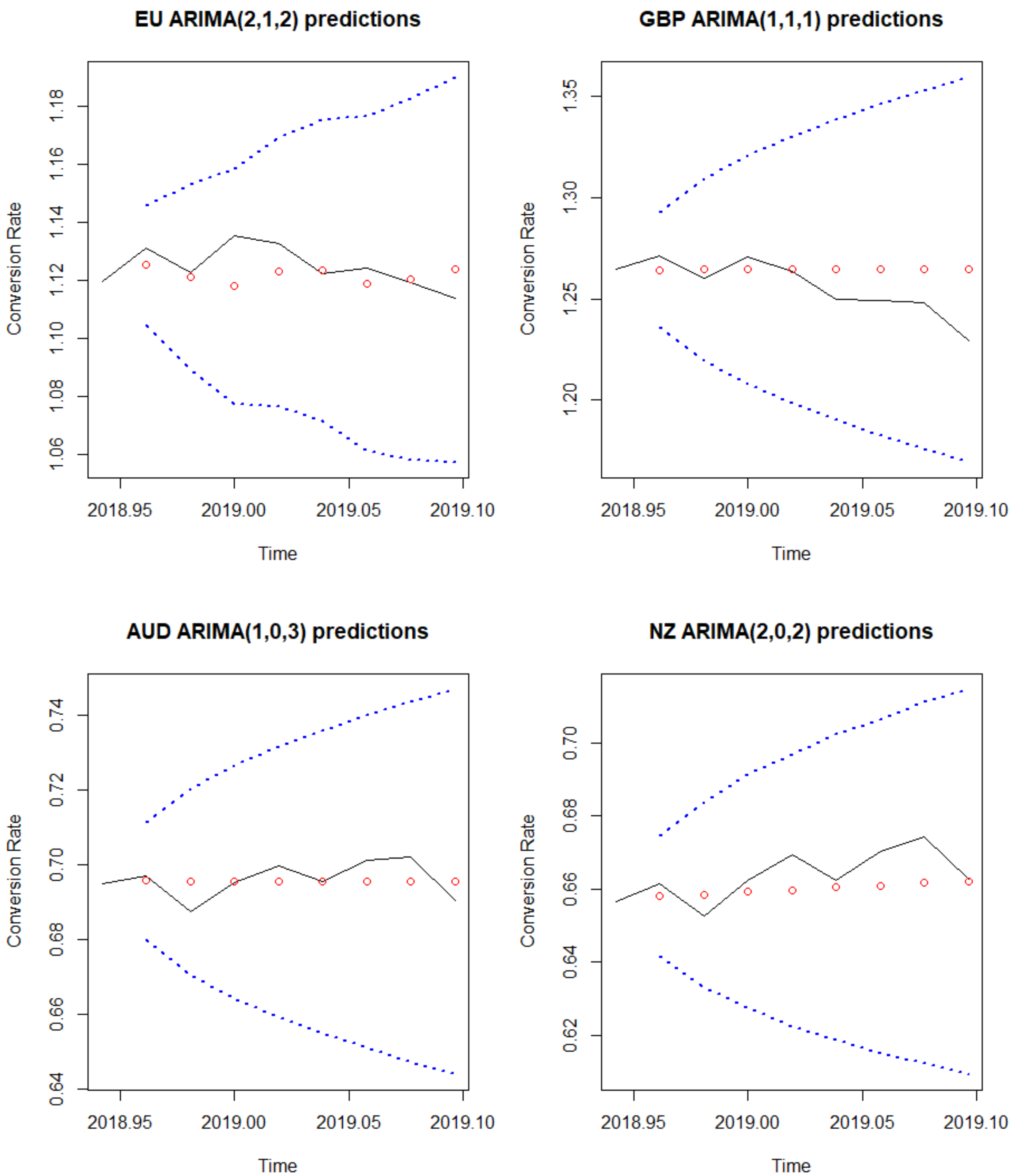
Below shows the EACF table and the orders of ARIMA chosen from lowest tail AIC for NZ.

```
eacf(diff.NZ.train, 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
```

```
# p d q       AIC
# 5 1 3 -1718.098
# The EACF reflects the lowest AIC selection with threshold 2 (p,d,q) = (5,1,3) as a viable option but still has other simpler order selection.
```

b. (12 pts)


EU ARIMA(2,1,2) predictions
GBP ARIMA(1,1,1) predictions
AUD ARIMA(1,0,3) predictions
NZ ARIMA(2,0,2) predictions
Conversion Rate
Time
1.06
1.08
1.10
1.12
1.14
1.16
1.18
1.20
1.25
1.30
1.35
0.64
0.66
0.68
0.70
0.72
0.74
0.62
2018.95
2019.00
2019.05
2019.10

c. (4 pts)

- EU ARIMA Prediction MAPE:

mean(abs(preds_EU-obs_EU)/obs_EU)

0.005789397

- GBP ARIMA Prediction MAPE:

mean(abs(preds_GBP-obs_GBP)/obs_GBP)

0.009954838

- AU ARIMA Prediction MAPE:

mean(abs(preds_AU-obs_AU)/obs_AU)

0.005598311

- NZ ARIMA Prediction MAPE:

mean(abs(preds_NZ-obs_NZ)/obs_NZ)

0.00884671

EU and AU conversion rates are the most accurate among the 4, having the lowest MAPE values compared to GBP and NZ. All are quite low, also considering that the predictions in all cases are within the confidence interval boundary.

Question 3: Multivariate Analysis (28 pts)

a. (4 pts)

data.train = ts.union(EU.train, GBP.train, AU.train, NZ.train) # or train.ts

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

AIC(n) HQ(n) SC(n) FPE(n)

2   2   1   2

We can recognize that p = 2 is the best model to use according to ACF.

model.var= VAR(data.train, p = 2)

b. (12 pts)

ARCH (multivariate) test

data: Residuals of VAR object model.var2

Chi-squared = 677.05, df = 500, p-value = 2.005e-07

P-value is almost 0, so we reject the null of constant variance. We can recognize that the constant variance assumption is violated.

JB-Test (multivariate)

data: Residuals of VAR object model.var2

Chi-squared = 271.66, df = 8, p-value < 2.2e-16

$Skewness

Skewness only (multivariate)

data: Residuals of VAR object model.var2

Chi-squared = 37.838, df = 4, p-value = 1.21e-07

$Kurtosis

Kurtosis only (multivariate)

data: Residuals of VAR object model.var2

Chi-squared = 233.83, df = 4, p-value < 2.2e-16

The resulting p-values of JB-test, skewness and kurtosis shows that all p-values are close to 0. This means we reject the null hypothesis of normality. We therefore realize that the normality assumption is violated.

Portmanteau Test (asymptotic)

data: Residuals of VAR object model.var2

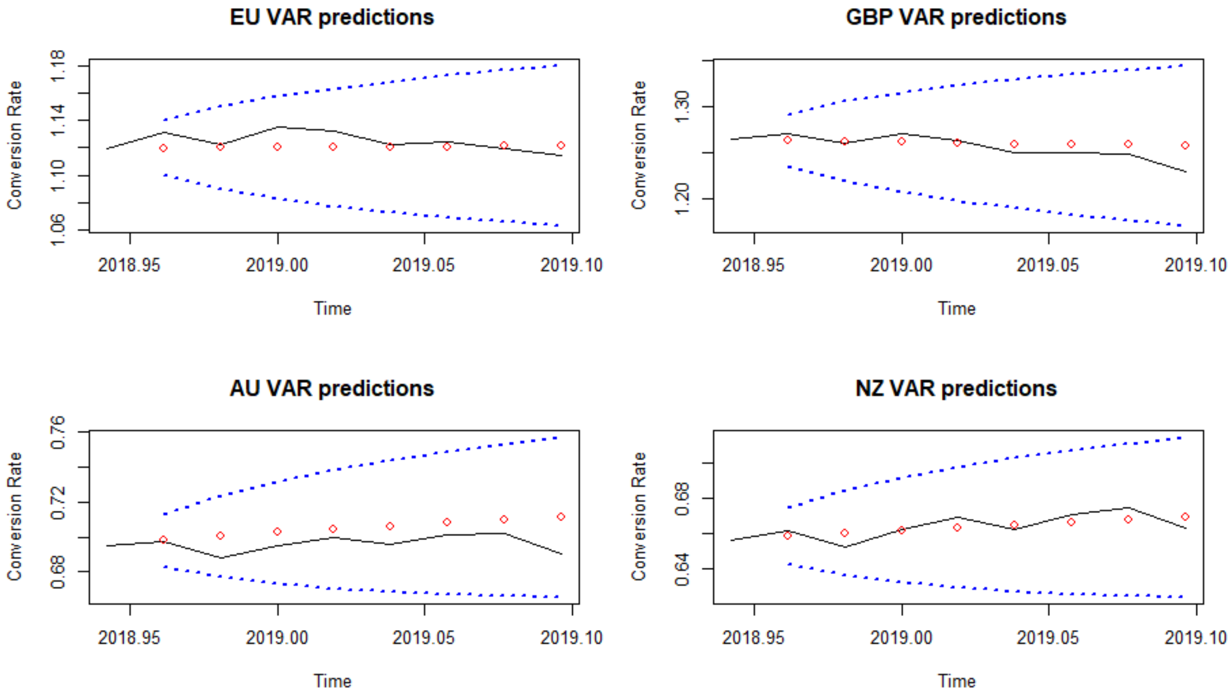
Chi-squared = 266.8, df = 224, p-value = 0.02637

Given significance of 0.05, this p-value indicates we reject the null hypothesis of uncorrelated variables and conclude that they are somewhat correlated. This means that the residuals violate the assumption of uncorrelation.

Will skip checking for roots being outside the unit circle for stability as it was not asked by the question.

c. (12 pts)

```
model.var2.predict = predict(model.var2, n.ahead = 8)
```




```
# $EU.train
# 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

# $GBP.train
```

```
# 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

# $AU.train
# 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

# $NZ.train
# 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

EU Var Prediction MAPE:

mean(abs(EU.train.var.predict-obs_EU)/obs_EU)

0.006303623 - VAR

0.005789397 - ARIMA

GBPVar Prediction MAPE:

mean(abs(GBP.train.var.predict-obs_GBP)/obs_GBP)

0.008045432 - VAR

0.009954838 - ARIMA

AU Var Prediction MAPE:

mean(abs(AU.train.var.predict-obs_AU)/obs_AU)

0.01264681 - VAR

0.005598311 - ARIMA

NZ Var Prediction MAPE:

mean(abs(NZ.train.var.predict-obs_NZ)/obs_NZ)

0.006890983 - VAR

0.00884671 - ARIMA

It can be shown from the mean absolute percentage error that for GBP and NZ the VAR outperforms the univariate model (ARIMA) prediction. (Since the MAPE is smaller in this case). Again, all are quite low considering that the predictions in all cases are within the confidence interval boundary.

Question 4: Reflection (6 pts)

It is often the case that more than one time series can be cross-correlated, in which case it provides an advantage in prediction power compared to just univariate modeling that relies on serial correlation only. In the case that a time series has contextual relationships with other time series, these other time series data are available, and we have enough computing power -- VAR is recommended for superior accuracy and analyzing the complex relationships between them. Sometimes you are only able to gather data from one time series (unavailable data), then univariate modeling is the only feasible forecasting method. More importantly, if there are little or no relationships between the time series (little influence on each other hence more noise than signal), the VAR modeling may only introduce noise and weaken prediction power. Also, VAR requires more computational power compared to univariate modeling, which also leads to the argument that it may suffer in speed when forecasting needs to be done fast (usually the case). Finally, one may be interested in understanding/visualizing a time series based on itself with no consideration to other processes, especially when we want to look at the time series from its own historical perspective.

Code:

```
# npark62 HW3



#Initialization
library(TSA)
library(data.table)
library(vars)

# fname <- file.choose()
data <-
read.csv("C:\\Users\\nicholas.park\\Downloads\\Currency_Conversion_Data.csv")
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
```

```r
#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


diff.EU.train = diff(EU.train)

diff.GBP.train = diff(GBP.train)

diff.AU.train = diff(AU.train)

diff.NZ.train = diff(NZ.train)


par(mfrow=c(1,2))

ts.plot(diff.EU.train, main = "Differenced USD/EUR")

acf(diff.EU.train, main = "ACF of differenced USD/EUR")


# The differenced time series is more constant than the original, but still
seems to have some inconsistencies.

# There is clear non-constant variance (heteroskedasticity) observed

# The autocorrelation shows some significant lags

# In summary, not stationary.



par(mfrow=c(1,2))

ts.plot(diff.GBP.train, main = "Differenced USD/GBP")

acf(diff.GBP.train, main = "ACF of differenced USD/GBP")


# The mean of the differenced time series is quite constant
```

```r
# The constnat seems finite and constant except for a spike at 2016.
# No siginificant correlation is observed from the ACF
# In summary, not perfectly statonary, but has some characteristics.

par(mfrow=c(1,2))
ts.plot(diff.AU.train, main = "Differenced USD/AU")
acf(diff.AU.train, main = "ACF of differenced USD/AU")

# The mean of the differenced time series is not strictly constant.
# The variance seems to be non-constant.
# There is no significant autocorrelation.
# In summary, not stationary.


par(mfrow=c(1,2))
ts.plot(diff.NZ.train, main = "Differenced USD/NZ")
acf(diff.NZ.train, main = "ACF of differenced USD/NZ")

# The mean of the differenced time series still is not strictly constant.
# The variance is non-constant
# The autocorrelation shows some significant lags
# In summary, not stationary.


# Question 2

# a

eacf(diff.EU.train, ar.max=7, ma.max = 7)

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

```r
# 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

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

```r
# p d q       AIC

# 2 1 2 -1597.578

# The EACF includes the (p,d,q) = (2,1,2) as a viable option. Note that eacf
was used on already differenced data, d = 1.

eacf(diff.GBP.train, 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

test_modelB <- 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")
```

```r
for (p in 0:5){
  for (d in 0:1){
    for (q in 0:5) {
      possibleError <- tryCatch(
        orders<-rbind(orders,test_modelB(p,d,q)),
        error=function(e) e
      )
      if(inherits(possibleError, "error")) next}}}
orders <- orders[order(-orders$AIC),]
tail(orders)


# p d q       AIC

# 0 1 1 -1446.502

# The EACF includes the (p,d,q) = (0,1,1) as a viable option.


eacf(diff.AU.train, 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
```

```r
# 7 x o o x x o o o

test_modelC <- 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_modelC(p,d,q)),
        error=function(e) e
      )
      if(inherits(possibleError, "error")) next}}}
orders <- orders[order(-orders$AIC),]
tail(orders)

# p d q       AIC

# 0 1 1 -1749.481

# The EACF reflects that the order (p,d,q) = (0,1,1) is a viable option.
```

```
eacf(diff.NZ.train, 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

test_modelD <- 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_modelD(p,d,q)),
        error=function(e) e
```

```r
      )
      if(inherits(possibleError, "error")) next}}}
orders <- orders[order(-orders$AIC),]
tail(orders)

# p d q       AIC

# 5 1 3 -1718.098

# The EACF reflects that the order (p,d,q) = (5,1,3) is a viable option.



# B

whole_EU <- ts(data$USD.EU,start=c(2014,1),freq=52)
whole_GBP <- ts(data$USD.GBP,start=c(2014,1),freq=52)
whole_AU <- ts(data$USD.AU,start=c(2014,1),freq=52)
whole_NZ <- ts(data$USD.NZ,start=c(2014,1),freq=52)


vol=time(whole_EU)
n = length(whole_EU)
nfit = n - 8

par(mfrow(c(2,2)))

EU_train = arima(EU.train, order = c(2,1,3),method = "ML")
arima_pred_EU = as.vector(predict(EU_train,n.ahead=8))
ubound = arima_pred_EU$pred+1.96*arima_pred_EU$se
lbound = arima_pred_EU$pred-1.96*arima_pred_EU$se
ymin = min(lbound)
ymax = max(ubound)
```

```
plot(vol[nfit:n],whole_EU[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time",
ylab = 'Conversion Rate', main="EU ARIMA(2,1,2) predictions")

points(vol[(nfit+1):n],arima_pred_EU$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")



GBP_train = arima(GBP.train, order = c(1,1,1),method = "ML")

arima_pred_GBP = as.vector(predict(GBP_train,n.ahead=8))

ubound = arima_pred_GBP$pred+1.96*arima_pred_GBP$se

lbound = arima_pred_GBP$pred-1.96*arima_pred_GBP$se

ymin = min(lbound)

ymax = max(ubound)


plot(vol[nfit:n],whole_GBP[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time",
ylab = 'Conversion Rate', main="GBP ARIMA(1,1,1) predictions")

points(vol[(nfit+1):n],arima_pred_GBP$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")



AU_train = arima(AU.train, order = c(1,0,3),method = "ML")

arima_pred_AU = as.vector(predict(AU_train,n.ahead=8))

ubound = arima_pred_AU$pred+1.96*arima_pred_AU$se

lbound = arima_pred_AU$pred-1.96*arima_pred_AU$se

ymin = min(lbound)

ymax = max(ubound)


plot(vol[nfit:n],whole_AU[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time",
ylab = 'Conversion Rate', main="AUD ARIMA(1,0,3) predictions")

points(vol[(nfit+1):n],arima_pred_AU$pred,col="red")

lines(vol[(nfit+1):n],ubound,lty=3,lwd= 2, col="blue")
```

```r
lines(vol[(nfit+1):n],lbound,lty=3,lwd= 2, col="blue")


NZ_train = arima(NZ.train, order = c(2,0,2),method = "ML")

arima_pred_NZ = as.vector(predict(NZ_train,n.ahead=8))

ubound = arima_pred_NZ$pred+1.96*arima_pred_NZ$se

lbound = arima_pred_NZ$pred-1.96*arima_pred_NZ$se

ymin = min(lbound)

ymax = max(ubound)


plot(vol[nfit:n],whole_NZ[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time",
ylab = 'Conversion Rate', main="NZ ARIMA(2,0,2) predictions")

points(vol[(nfit+1):n],arima_pred_NZ$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")


# C


preds_EU = arima_pred_EU$pred

obs_EU = EU.test


preds_GBP = arima_pred_GBP$pred

obs_GBP = GBP.test


preds_AU = arima_pred_AU$pred

obs_AU = AU.test


preds_NZ = arima_pred_NZ$pred

obs_NZ = NZ.test


mean(abs(preds_EU-obs_EU)/obs_EU)

#  0.005789397
```

```
mean(abs(preds_GBP-obs_GBP)/obs_GBP)

#  0.009954838


mean(abs(preds_AU-obs_AU)/obs_AU)

#  0.005598311


mean(abs(preds_NZ-obs_NZ)/obs_NZ)

#  0.00884671


# Question 3


# a


data.train = ts.union(EU.train, GBP.train, AU.train, NZ.train) # or train.ts

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


# AIC(n)  HQ(n)  SC(n) FPE(n)

# 2      2      1      2


# We can recognize that p = 2 is the best model to use.


model.var= VAR(data.train, p = 2)


# Estimation results for equation EU.train:

#   ==========================================

#   EU.train = EU.train.l1 + GBP.train.l1 + AU.train.l1 + NZ.train.l1 +
EU.train.l2 + GBP.train.l2 + AU.train.l2 + NZ.train.l2 + const

#

# Estimation results for equation GBP.train:

#   ===========================================

#   GBP.train = EU.train.l1 + GBP.train.l1 + AU.train.l1 + NZ.train.l1 +
EU.train.l2 + GBP.train.l2 + AU.train.l2 + NZ.train.l2 + const
```

```r
#
# Estimation results for equation AU.train:
#   ==========================================
#   AU.train = EU.train.l1 + GBP.train.l1 + AU.train.l1 + NZ.train.l1 +
EU.train.l2 + GBP.train.l2 + AU.train.l2 + NZ.train.l2 + const
#
# Estimation results for equation NZ.train:
#   ==========================================
#   NZ.train = EU.train.l1 + GBP.train.l1 + AU.train.l1 + NZ.train.l1 +
EU.train.l2 + GBP.train.l2 + AU.train.l2 + NZ.train.l2 + const


# b


model.var2 = VAR(data.train, p = 2)


arch.test(model.var2)


# ARCH (multivariate)
#
# data:  Residuals of VAR object model.var2
# Chi-squared = 677.05, df = 500, p-value = 2.005e-07


# P-value is almost 0, so we reject the null of constant variance.
# We can recognize that the constant variance assumption is violated.


normality.test(model.var2)


# $JB (This is jarque bera btw)
#
# JB-Test (multivariate)
#
# data:  Residuals of VAR object model.var2
# Chi-squared = 271.66, df = 8, p-value < 2.2e-16
```

```
#
#
# $Skewness
#
# Skewness only (multivariate)
#
# data:  Residuals of VAR object model.var2
# Chi-squared = 37.838, df = 4, p-value = 1.21e-07
#
#
# $Kurtosis
#
# Kurtosis only (multivariate)
#
# data:  Residuals of VAR object model.var2
# Chi-squared = 233.83, df = 4, p-value < 2.2e-16



# The resulting p-values of skewness and kurtosis shows that both p-values are close to 0.
# This means we reject the null hypothesis of normality.
# We therefore realize that the normality assumption is violated.

serial.test(model.var2)

# Portmanteau Test (asymptotic)
#
# data:  Residuals of VAR object model.var2
# Chi-squared = 266.8, df = 224, p-value = 0.02637

# Given significance of 0.05, this p-value indicates we reject the null hypothesis of uncorrelated variables and conclude that they are somewhat correlated.
```

```r
# This means that the residuals violate the assumption of uncorrelation. See
4 sets of graphs below

serialtest = serial.test(model.var2)
plot(serialtest)

# c

model.var2.predict = predict(model.var2, n.ahead = 8)

# $EU.train
# 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
#
# $GBP.train
# 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
```

```
#
# $AU.train
# 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
#
# $NZ.train
# 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

EU.train.var.predict = model.var2.predict[[1]]$EU.train[,1]
GBP.train.var.predict = model.var2.predict[[1]]$GBP.train[,1]
AU.train.var.predict = model.var2.predict[[1]]$AU.train[,1]
NZ.train.var.predict = model.var2.predict[[1]]$NZ.train[,1]

EU.train.var.predict.lower = model.var2.predict[[1]]$EU.train[,2]
GBP.train.var.predict.lower = model.var2.predict[[1]]$GBP.train[,2]
AU.train.var.predict.lower = model.var2.predict[[1]]$AU.train[,2]
NZ.train.var.predict.lower = model.var2.predict[[1]]$NZ.train[,2]
```

```
EU.train.var.predict.upper = model.var2.predict[[1]]$EU.train[,3]

GBP.train.var.predict.upper = model.var2.predict[[1]]$GBP.train[,3]

AU.train.var.predict.upper = model.var2.predict[[1]]$AU.train[,3]

NZ.train.var.predict.upper = model.var2.predict[[1]]$NZ.train[,3]


par(mfrow=c(2,2))


ymin = min(EU.train.var.predict.lower)

ymax = max(EU.train.var.predict.upper)


plot(vol[nfit:n],whole_EU[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time",
ylab = 'Conversion Rate', main="EU VAR predictions")

points(vol[(nfit+1):n],EU.train.var.predict,col="red")

lines(vol[(nfit+1):n],EU.train.var.predict.upper,lty=3,lwd= 2, col="blue")

lines(vol[(nfit+1):n],EU.train.var.predict.lower,lty=3,lwd= 2, col="blue")


ymin = min(GBP.train.var.predict.lower)

ymax = max(GBP.train.var.predict.upper)


plot(vol[nfit:n],whole_GBP[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time",
ylab = 'Conversion Rate', main="GBP VAR predictions")

points(vol[(nfit+1):n],GBP.train.var.predict,col="red")

lines(vol[(nfit+1):n],GBP.train.var.predict.upper,lty=3,lwd= 2, col="blue")

lines(vol[(nfit+1):n],GBP.train.var.predict.lower,lty=3,lwd= 2, col="blue")


ymin = min(AU.train.var.predict.lower)

ymax = max(AU.train.var.predict.upper)


plot(vol[nfit:n],whole_AU[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time",
ylab = 'Conversion Rate', main="AU VAR predictions")

points(vol[(nfit+1):n],AU.train.var.predict,col="red")

lines(vol[(nfit+1):n],AU.train.var.predict.upper,lty=3,lwd= 2, col="blue")
```

```r
lines(vol[(nfit+1):n],AU.train.var.predict.lower,lty=3,lwd= 2, col="blue")

ymin = min(NZ.train.var.predict.lower)
ymax = max(NZ.train.var.predict.upper)

plot(vol[nfit:n],whole_NZ[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time",
ylab = 'Conversion Rate', main="NZ VAR predictions")
points(vol[(nfit+1):n],NZ.train.var.predict,col="red")
lines(vol[(nfit+1):n],NZ.train.var.predict.upper,lty=3,lwd= 2, col="blue")
lines(vol[(nfit+1):n],NZ.train.var.predict.lower,lty=3,lwd= 2, col="blue")


mean(abs(EU.train.var.predict-obs_EU)/obs_EU)
#  0.006303623

#  0.005789397 - ARIMA

mean(abs(GBP.train.var.predict-obs_GBP)/obs_GBP)
#  0.008045432

#  0.009954838 - ARIMA


mean(abs(AU.train.var.predict-obs_AU)/obs_AU)
#  0.01264681

#  0.005598311 - ARIMA


mean(abs(NZ.train.var.predict-obs_NZ)/obs_NZ)
#  0.006890983
```

```
#  0.00884671 - ARIMA

# It can be shown from the mean absolute percentage error that for GBP and NZ the VAR outperforms the univariate model prediction. (Since the MAPE is smaller in this case)

# Question 4

# It is often the case that more than one time series can be cross-correlated, in which case it provides an advantage in prediction power compared to just univariate modeling that relies on serial correlation only.

# In the case that a time series has  contextual relationships with other time series, VAR is recommended.

# But if you are only able to gather data from one time series (you don't have a choice), then univariate modeling is the only feasible forecasting method.

# And if there are little or no relationships between the time series (little influence on each other hence more noise than signal), the VAR modeling may only introduce noise and weaken prediction power.

# Also, VAR requires more computational power compared to univariate modeling, which also leads to the argument that it may suffer in speed when forecasting needs to be done fast (usually the case).

# Finally, one may be interested in analyzing/visualizing a time series based on itself with no consideration to other processes, especially when we want to look at the time series from its own historical persepective.
```