ISyE6402 HW3

Jim Liu

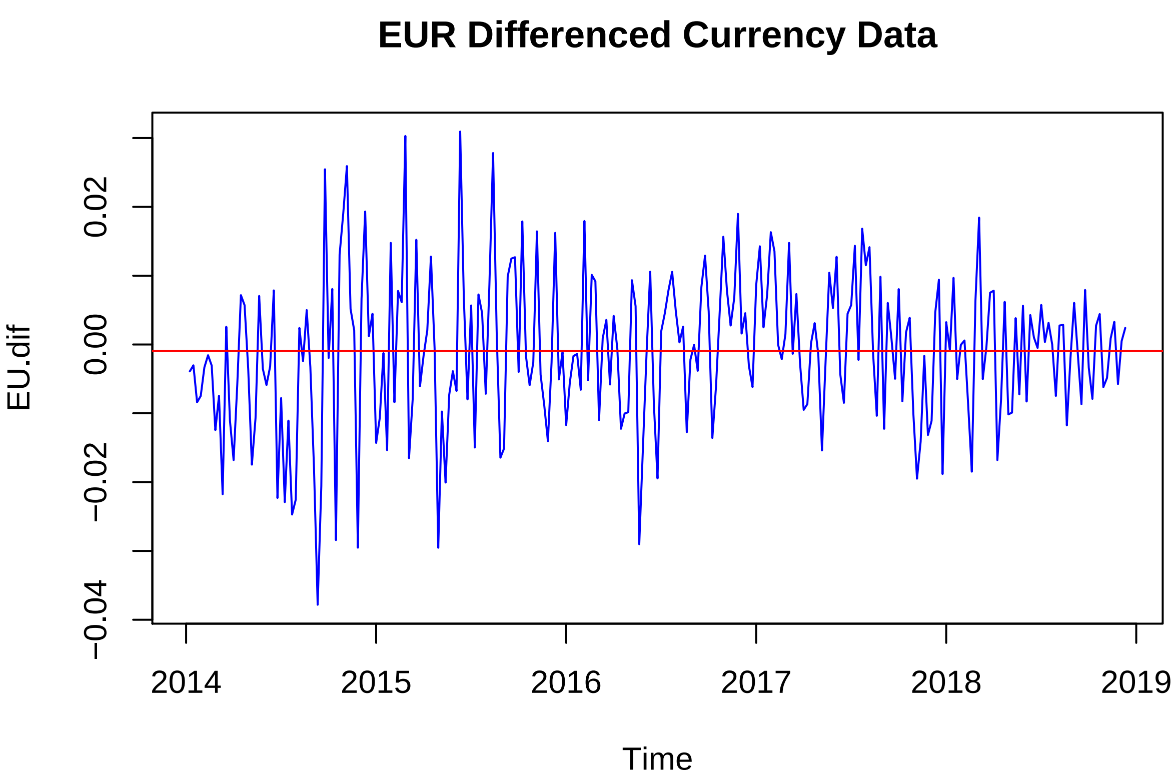
10/30/2020

Question 1: Data Exploration (8 pts)

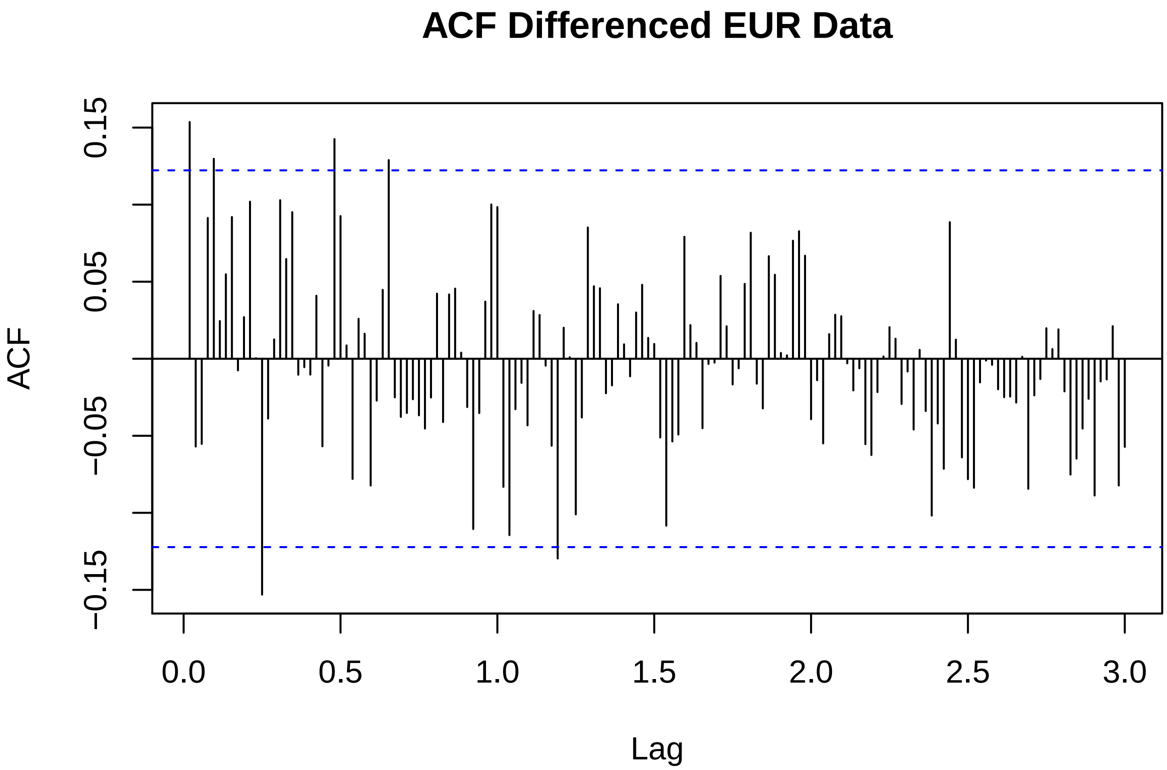
We already identified in past assignments that differenced currency conversion data tends to more closely resemble a stationary process. Knowing this, skip analyzing the original series, and for each series take the 1st order difference of the data. Use relevant plots to evaluate assumptions of stationarity.

USD/EUR

```
EU.dif = diff(EU.train)
ts.plot(EU.dif,main='EUR Differenced Currency Data',col='blue')
abline(a=mean(EU.dif),b=0,col='red')
```



```r
acf(EU.dif,main='ACF Differenced EUR Data', lag.max = 3 * 52)
```



Assumptions of stationarity for EUR:

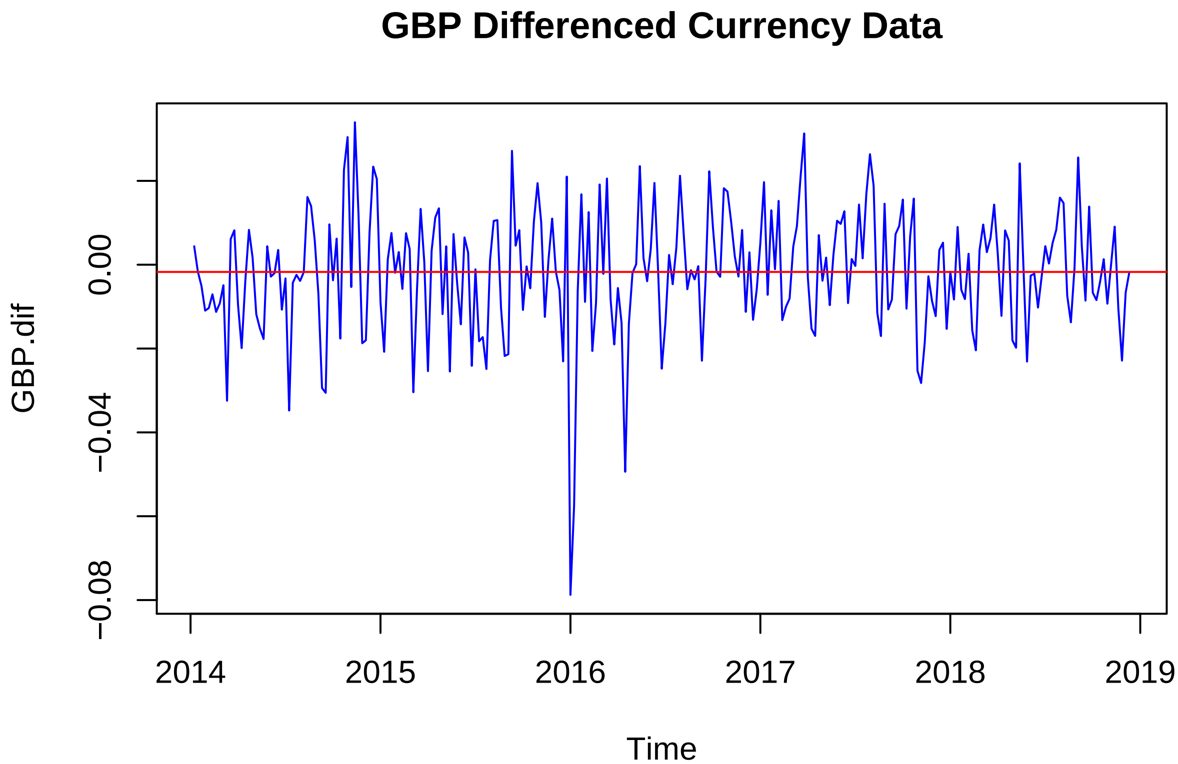
(1) Constant Mean: From the time series plot, we can see there is no trend. So this plot follows the assumption of the constant mean.

(2) Constant Variance: based on the time series plot, there is a non-constant variance. We could see in the beginning there is bigger variance, but around 2017, the variance becomes smaller.

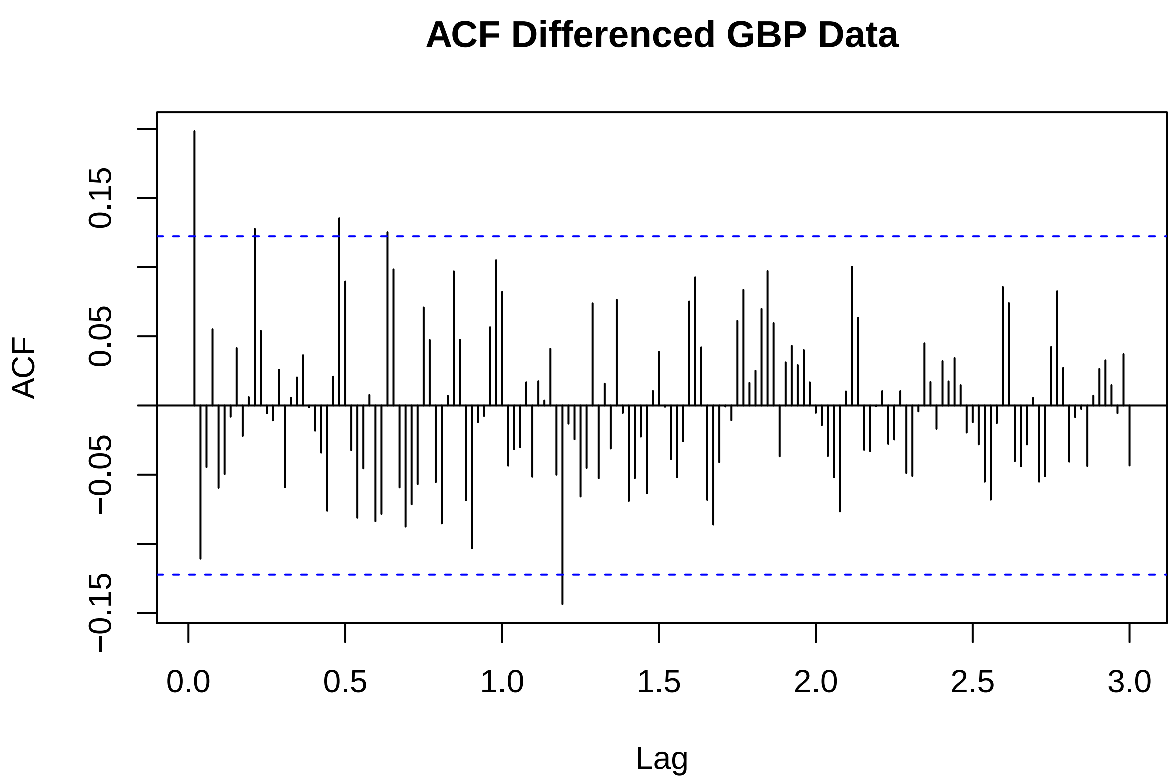
(3) Covariance Independent of Time: based on the ACF plot, we don't see any seasonality. So the assumption of covariance independent of time holds.

USD/GBP

```r
GBP.dif = diff(GBP.train)
ts.plot(GBP.dif,main='GBP Differenced Currency Data',col='blue')
abline(a=mean(GBP.dif),b=0,col='red')
```




```r
acf(GBP.dif,main='ACF Differenced GBP Data', lag.max = 3 * 52)
```



Assumptions of stationarity for GBP:

(1) Constant Mean: From the time series plot, we can see there is no trend. So this plot follows the

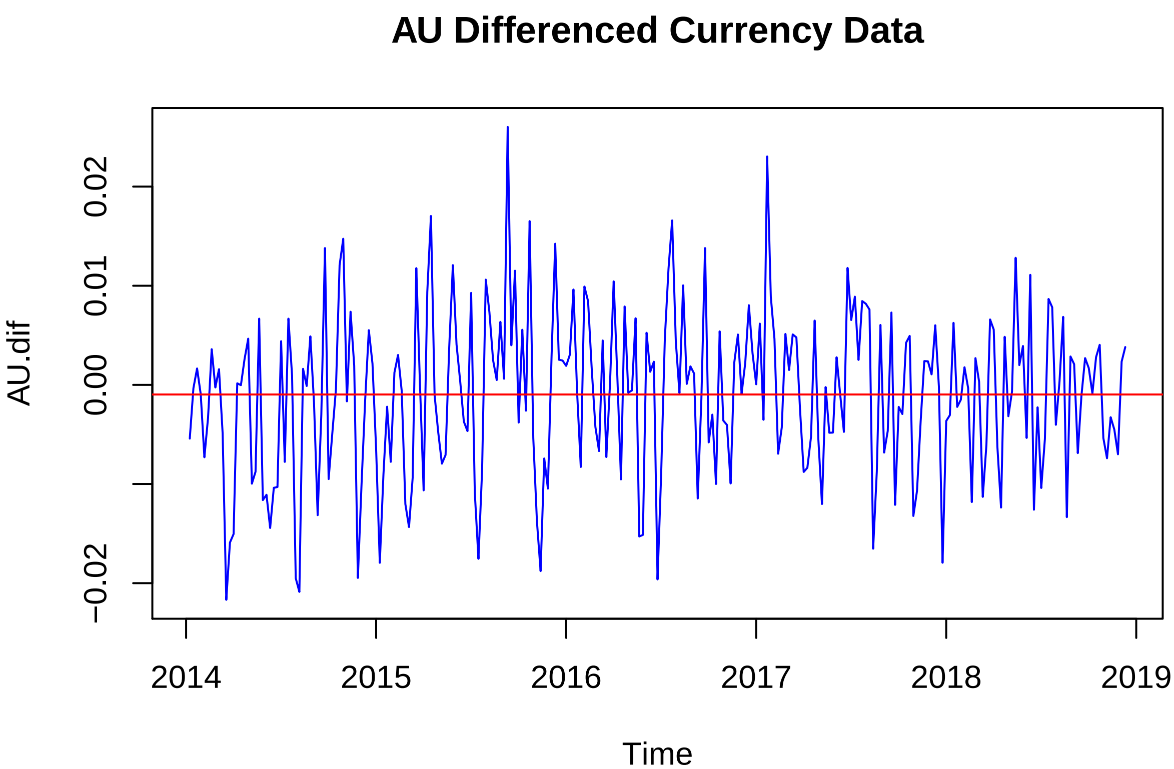
assumption of the constant mean.

(2) Constant Variance: based on the time series plot, although there is a large spike in 2016, other than that the variance is constant throughout the time.

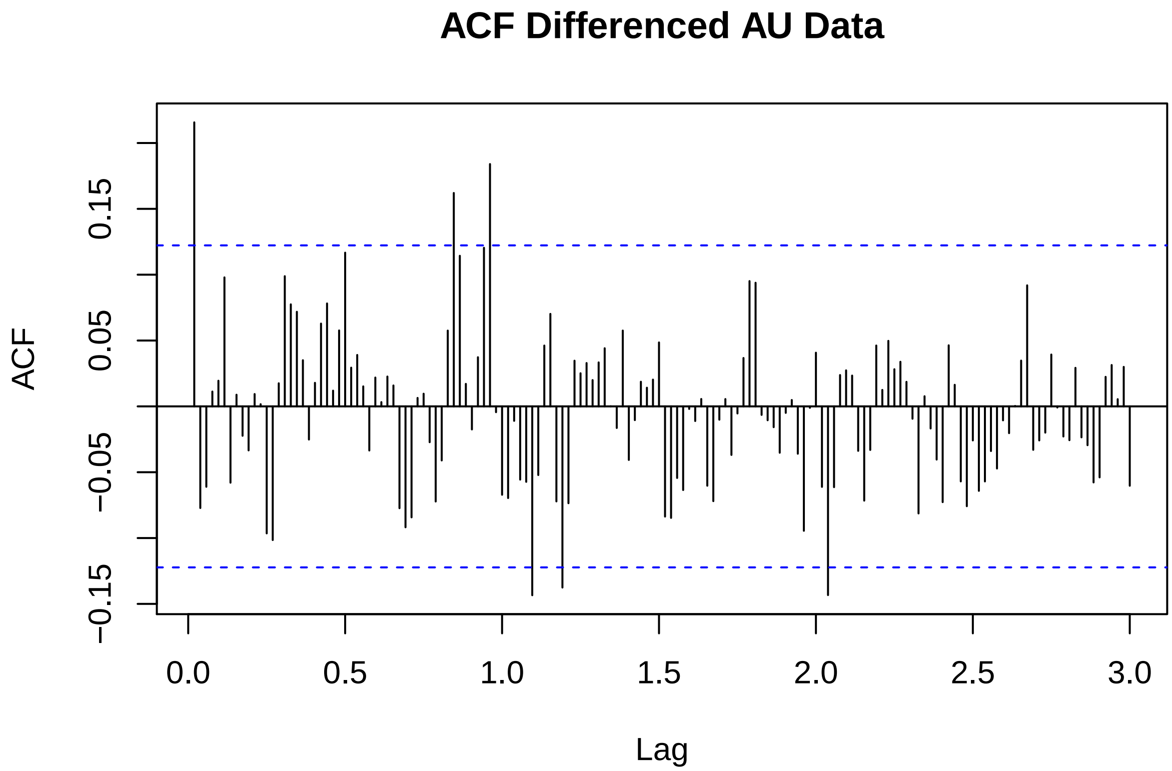
(3) Covariance Independent of Time: based on the ACF plot, we don't see any seasonality. So the assumption of covariance independent of time holds.

USD/AU

```
AU.dif = diff(AU.train)
ts.plot(AU.dif,main='AU Differenced Currency Data',col='blue')
abline(a=mean(AU.dif),b=0,col='red')
```




```
acf(AU.dif,main='ACF Differenced AU Data', lag.max = 3 * 52)
```



Assumptions of stationarity for AU:

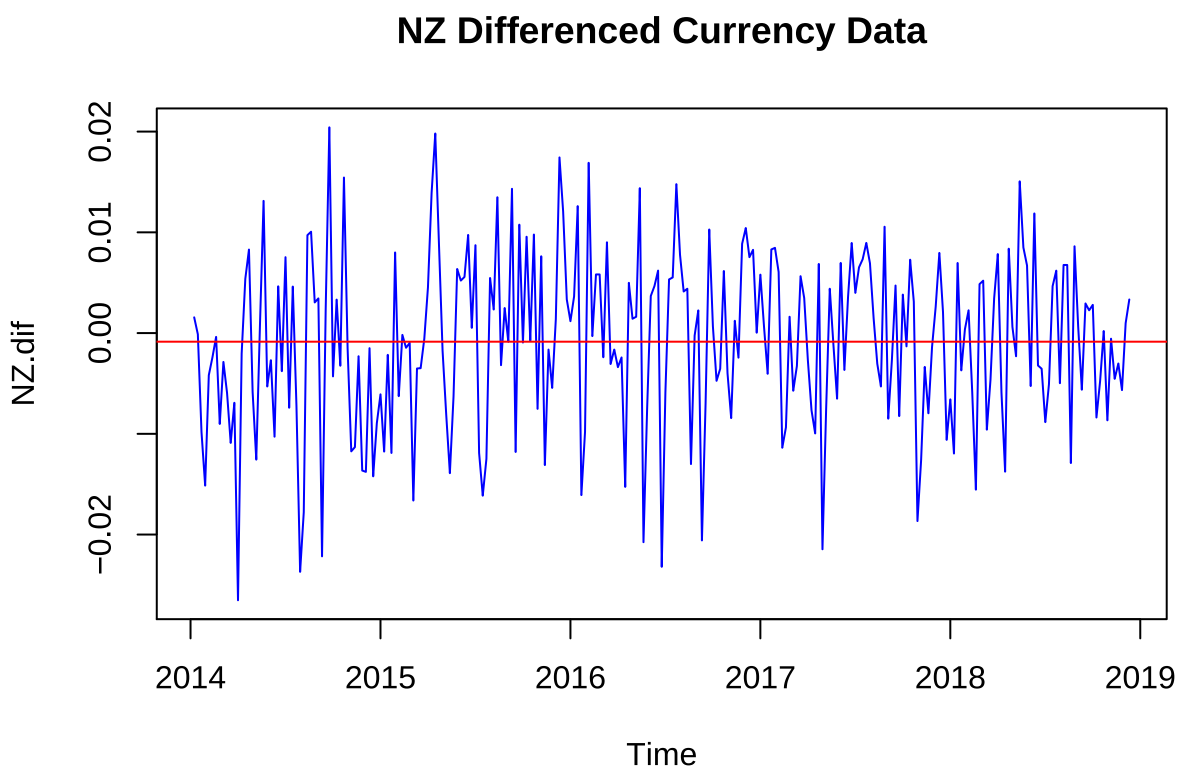
(1) Constant Mean: From the time series plot, we can see there is no trend. So this plot follows the assumption of the constant mean.

(2) Constant Variance: based on the time series plot, the assumption of constant variance does not hold. The variance before 2017 is larger than one after 2017.

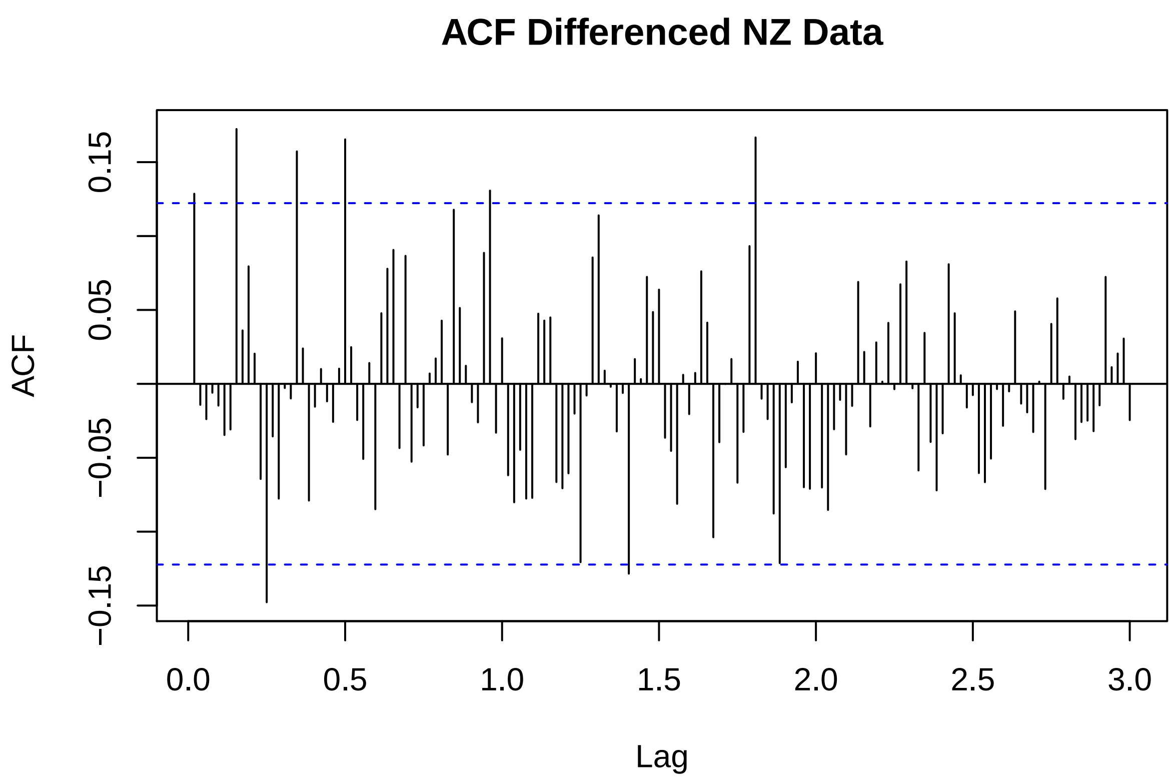
(3) Covariance Independent of Time: based on the ACF plot, we don't see any seasonality. So the assumption of covariance independent of time holds.

USD/NZ

```
NZ.dif = diff(NZ.train)
ts.plot(NZ.dif,main='NZ Differenced Currency Data',col='blue')
abline(a=mean(NZ.dif),b=0,col='red')
```




```r
acf(NZ.dif,main='ACF Differenced NZ Data', lag.max = 3 * 52)
```



Assumptions of stationarity for NZ:

(1) Constant Mean: From the time series plot, we can see there is no trend. So this plot follows the

assumption of the constant mean.

(2) Constant Variance: based on the time series plot, the assumption of constant variance does not hold. The variance before 2017 is larger than one after 2018.

(3) Covariance Independent of Time: based on the ACF plot, we don't see any seasonality. So the assumption of covariance independent of time holds.

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?

We know EACF is a quick way to select orders for ARMA models. The EACF table for an ARMA(p, q) process should theoretically have a triangular pattern of zeroes with the top-left zero occurring in the p-th row and q-th column (with the row and column labels both starting from 0).

USD/EUR

```
## Order selection -- EACF
eacf(EU.dif,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
```

- EACF Order Selection: We could see probably the suitable order via EACF is to ARMA(p=3, q=3)

```
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
```

- AIC Order Selection: Our threshold is 2. So we need to look over five options since they all are within the threshold. The last two option is ARIMA(3, 0, 2) and ARIMA(2, 1, 2). We select ARIMA(2, 1, 2) because it has a lower AIC value than ARIMA(3, 0 , 2).

USD/GBP

```
## Order selection -- EACF
eacf(GBP.dif,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
```

- EACF Order Selection: We could see probably the suitable order via EACF is to ARMA(p=1, q=2)

```
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
```

- AIC Order Selection: Our threshold is 2. So we need to look over five options since they all are within the threshold. We select the ARIMA(0 , 1, 1) since the number of this parameter is lower than ARIMA(2, 1, 0).

USD/AU

```
## Order selection -- EACF
eacf(AU.dif,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
```

- EACF Order Selection: We could see probably the suitable order via EACF is to ARMA(p=0, q=1)

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

```
##    p d q       AIC
## 71 5 1 4 -1748.944
## 17 1 0 3 -1749.409
## 9  0 1 1 -1749.481
## 52 4 0 3 -1749.814
## 58 4 1 3 -1750.183
## 51 4 0 2 -1751.555
```

USD/NZ

```
## Order selection -- EACF
eacf(NZ.dif,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
```

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

```
##    p d q       AIC
## 28 2 0 2 -1713.021
## 14 1 0 0 -1714.497
## 3  0 0 1 -1714.672
## 65 5 0 3 -1718.531
## 43 3 0 5 -1718.848
## 67 5 0 5 -1719.165
```

- AIC Order Selection: Our threshold is 2. So we need to look over three options since they all are within the threshold. We select the ARIMA(3, 0, 5) since the number of the parameters is lower than ARIMA(5, 0, 5). Plus, the AIC value is lower than ARIMA(5, 0, 3) which has the same number of paramters as ARIMA(3, 0, 5).

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.

USD/EUR

```
par(mfrow=c(1,1))
arima_model.EU = arima(EU.train, order = c(2,1,3),method = "ML")
arima_model.EU
```

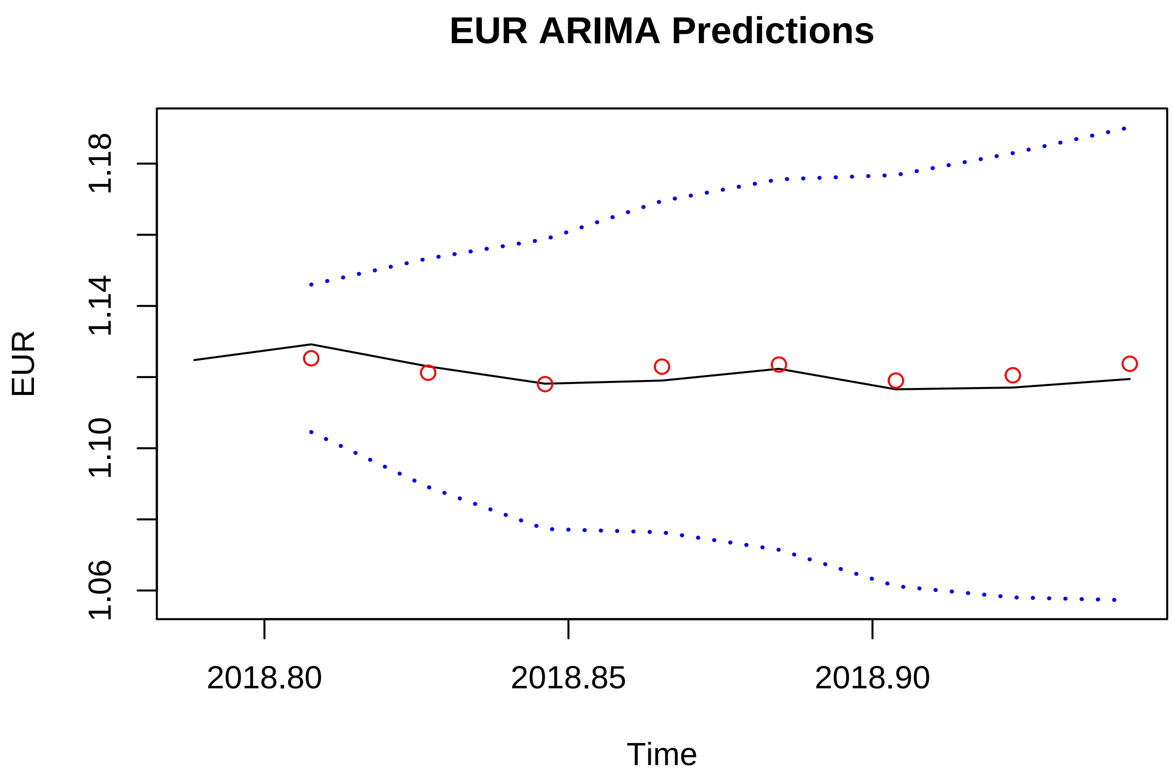
```
##
## Call:
## arima(x = EU.train, order = c(2, 1, 3), method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1     ma2     ma3
##       -0.4442  -0.8627  0.6287  0.9667  0.0530
## s.e.   0.0848   0.0675  0.1077  0.0714  0.0753
##
## sigma^2 estimated as 0.0001119:  log likelihood = 804.04,  aic = -1598.07
```

```
time = time(EU.train)
n = length(EU.train)
nfit = n - length(EU.test)

arima_model.EU = arima(EU.train, order = c(2,1,3),method = "ML")
arima_pred.EU = as.vector(predict(arima_model.EU,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(times[nfit:n],c(EU.train, EU.test)[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab = 'EUR',
points(times[(nfit+1):n],arima_pred.EU$pred,col="red")
lines(times[(nfit+1):n],ubound,lty=3,lwd= 2, col="blue")
lines(times[(nfit+1):n],lbound,lty=3,lwd= 2, col="blue")
```



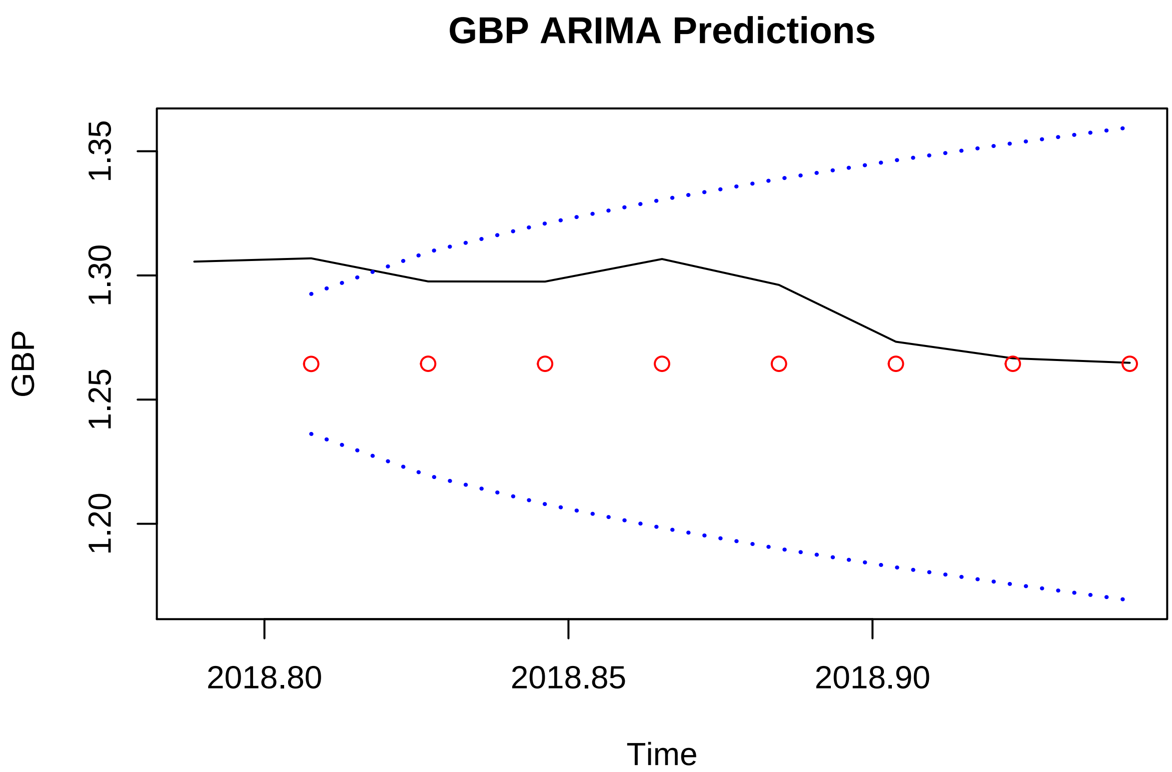
USD/GBP

```
n = length(GBP.train)
nfit = n - length(GBP.test)

arima_model.GBP = arima(GBP.train, order = c(1,1,1),method = "ML")
arima_pred.GBP = as.vector(predict(arima_model.GBP,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(times[nfit:n],c(GBP.train, GBP.test)[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab = 'GBP
points(times[(nfit+1):n],arima_pred.GBP$pred,col="red")
lines(times[(nfit+1):n],ubound,lty=3,lwd= 2, col="blue")
lines(times[(nfit+1):n],lbound,lty=3,lwd= 2, col="blue")
```



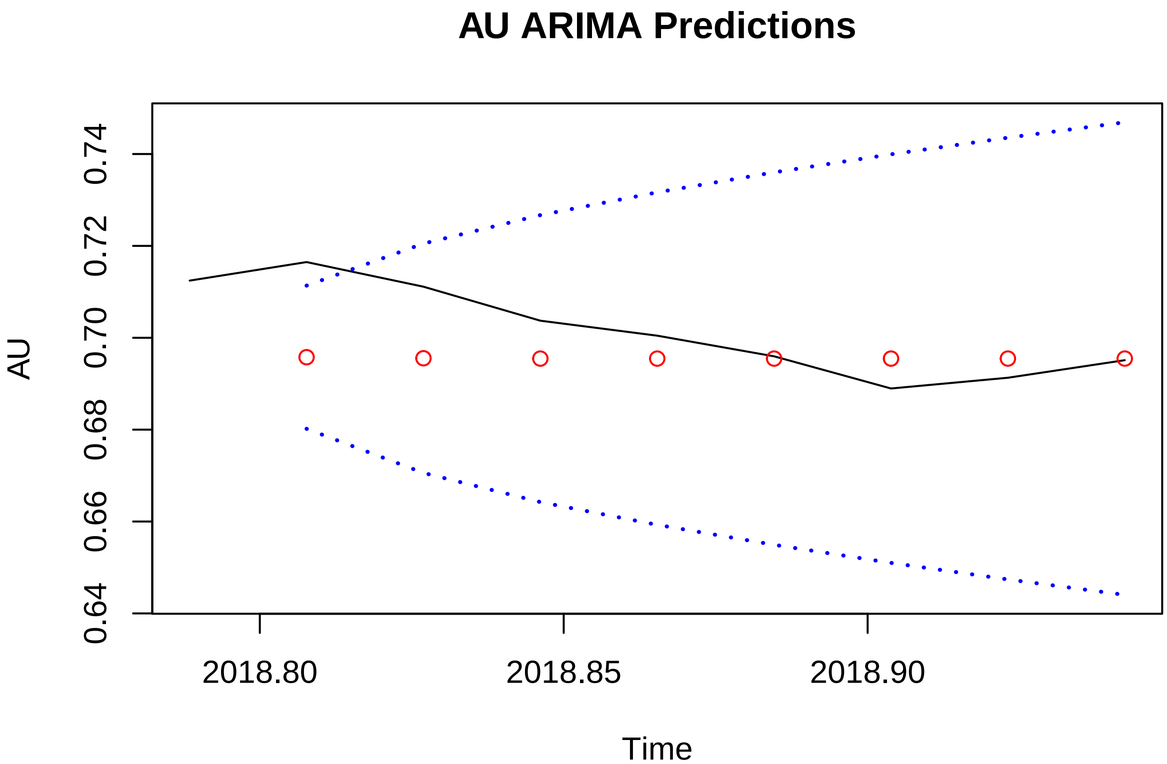
USD/AU

```
n = length(AU.train)
nfit = n - length(AU.test)

arima_model.AU = arima(AU.train, order = c(1,0,3),method = "ML")
arima_pred.AU = as.vector(predict(arima_model.AU,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(times[nfit:n],c(AU.train, AU.test)[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab = 'AU', m
points(times[(nfit+1):n],arima_pred.AU$pred,col="red")
lines(times[(nfit+1):n],ubound,lty=3,lwd= 2, col="blue")
lines(times[(nfit+1):n],lbound,lty=3,lwd= 2, col="blue")
```



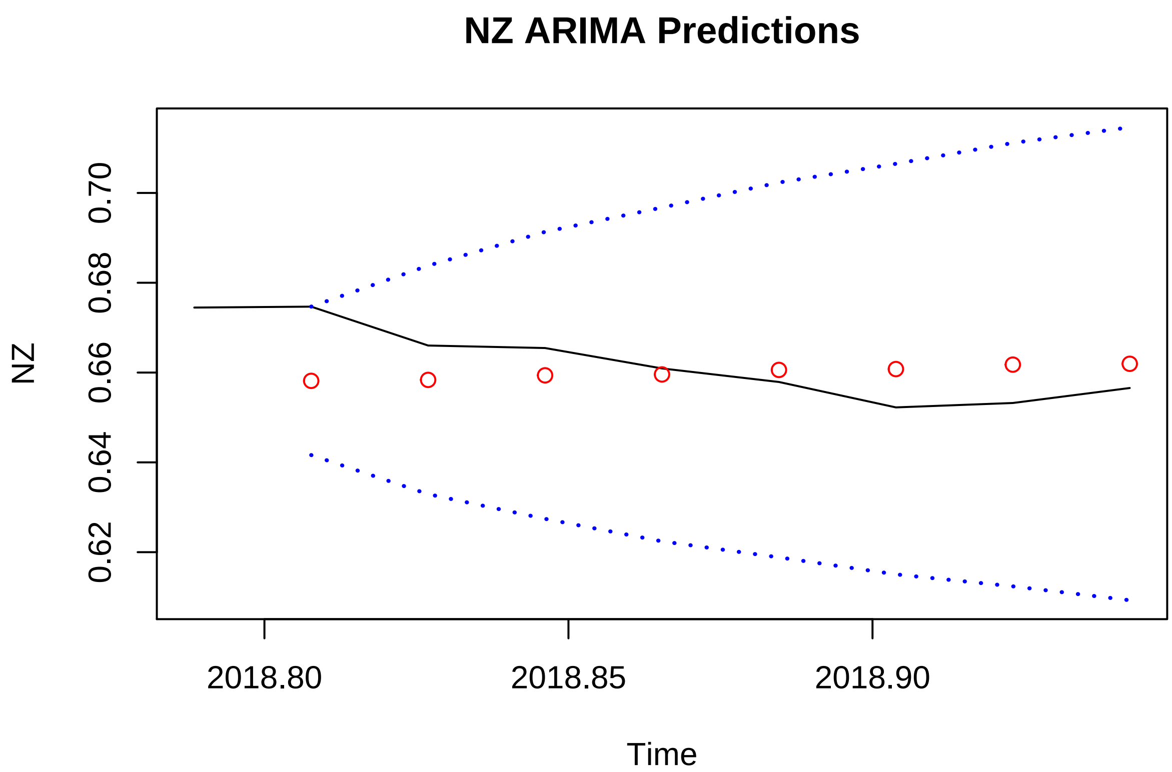
USD/NZ

```
n = length(NZ.train)
nfit = n - length(NZ.test)

arima_model.NZ = arima(NZ.train, order = c(2,0,2),method = "ML")
arima_pred.NZ = as.vector(predict(arima_model.NZ,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(times[nfit:n],c(NZ.train, NZ.test)[nfit:n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab = 'NZ', 
points(times[(nfit+1):n],arima_pred.NZ$pred,col="red")
lines(times[(nfit+1):n],ubound,lty=3,lwd= 2, col="blue")
lines(times[(nfit+1):n],lbound,lty=3,lwd= 2, col="blue")
```



c. (4 pts)

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

```
100*mean(abs(arima_pred.EU$pred-EU.test)/EU.test)
```

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

```
100*mean(abs(arima_pred.GBP$pred-GBP.test)/GBP.test)
```

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

```
100*mean(abs(arima_pred.AU$pred-AU.test)/AU.test)
```

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

```
100*mean(abs(arima_pred.NZ$pred-NZ.test)/NZ.test)
```

```
## [1] 0.8846701
```

Based on 4 ARIMA models, we could see USD/EUR and USD/AU has lower MAPE compared to USD/GBP and USD/NZ. But, those four values does not perform well since we hope to get smaller MAPE if possible.

Question 3: Multivariate Analysis (28 pts)

a. (4 pts)

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

```
data.ts = ts.union(EU.train,GBP.train, AU.train, NZ.train)
VARselect(data.ts, lag.max = 20)$selection
```

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

```
## Model Fitting: Unrestricted VAR
mod_aic = VAR(data.ts,lag.max=20,ic="AIC", type="none")
pord_1 = mod_aic$p
pord_1
```

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

```
## Fit VAR Model with Selected Order
mod = VAR(data.ts,pord_1, type="none")
```

The best VAR model is VAR(2) based on AIC criteria.

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

ARCH test

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

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object mod
## Chi-squared = 686.73, df = 500, p-value = 5.245e-08
```

The p-value of the ARCH test is 5.245e-08 which smaller than 0.05. A hypothesis testing procedure is to make sure whether the residuals have constant variance. Based on p-value, we could not reject null hypothesis so that the assumption of the constant variance holds.

Jarque-Bera test

```r
## Residual Analysis: Normality Assumption
normality.test(mod)
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object mod
## Chi-squared = 271.95, df = 8, p-value < 2.2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object mod
## Chi-squared = 37.581, df = 4, p-value = 1.367e-07
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object mod
## Chi-squared = 234.37, df = 4, p-value < 2.2e-16
```

The p-value of the normality test is 2.2e-16 which smaller than 0.05. A hypothesis testing procedure is to test whether the data residuals in this case are normally distributed. Based on p-value, we could not reject null hypothesis so that the assumption that data residuals are normally distributed.

the Portmanteau test

```r
## Residual Analysis: Uncorrelated Errors Assumption
serial.test(mod)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object mod
## Chi-squared = 262.94, df = 224, p-value = 0.03805
```

The p-value of the Portmanteau test is 0.03805 which smaller than 0.05. A hypothesis testing procedure is to test whether the residuals are uncorrelated residuals. Based on p-value, we could not reject null hypothesis so that the assumption that data residuals are uncorrelated.

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.

```r
data.predict = predict(mod,n.ahead=8)
```

```r
100*mean(abs(data.predict[[1]]$EU.train[,1]-EU.test)/EU.test)
```

```
## [1] 0.9192391
```

```r
100*mean(abs(data.predict[[1]]$GBP.train[,1]-GBP.test)/GBP.test)
```

```
## [1] 0.588012
```

```r
100*mean(abs(data.predict[[1]]$AU.train[,1]-AU.test)/AU.test)
```

```
## [1] 0.9759582
```

```r
100*mean(abs(data.predict[[1]]$NZ.train[,1]-NZ.test)/NZ.test)
```

```
## [1] 0.8015219
```

VAR model seems to perform worse than univariate ARIMA models. The result of univariate ARIMA models is :

[1] USD/EUR 0.5789397 [1] USD/GBP 0.9954838 [1] USD/AU 0.5598311 [1] USD/NZ 0.8846701

Originally in univariate ARIMA models, USD/EU and USD/AU has around 0.55 MAPE, but in this case they all have higher MAPE(0.919 and 0.975 respectively). However, USD/GBP and USD/NZ prediction accuracy are improved via VAR model(0.588 and 0.80 respectively).

Question 4: Reflection (6 pts)

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

Reflection: From the slides, the strengths are (1) easy to estimate (2) good forecasting (3) all variables are endogenous. And based on my understanding of VAR, the VAR model is useful for describing the dynamic behavior of economic and financial time series and for forecasting. It often provides superior forecasts to those from univariate time series models and elaborate theory-based simultaneous equations models. Forecasts from VAR models are quite flexible because they can be made conditional on the potential future paths of specified variables in the model.

However, the weaknesses are (1) too many parameters and (2) VARs are atheoretical. Since in this model we assume every variable influences every other variable in the system, which makes direct interpretation of the estimated coefficients very difficult.

When do we use VAR? The advantage of VAR is that it incorporates other information into time series which enhances forecasting capacities. So if we believe other variable affects another variable, I would use it over other methods.