# HW2_Q1

## **Question 1:** Forecasting Music Charts

## **1a.** Exploratory Analysis

```
rm(list=ls())
#Load chart_monthly.csv

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(mgcv)
```

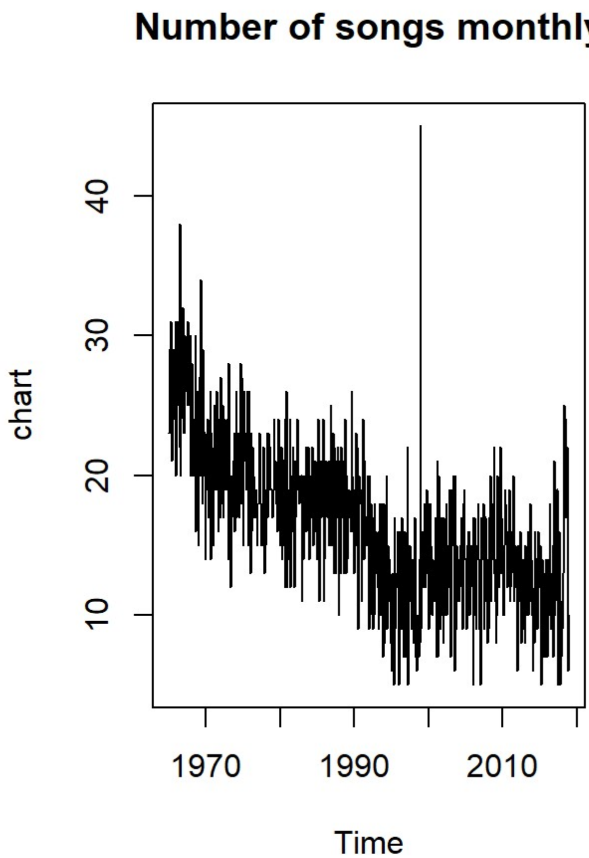
```
## Loading required package: nlme
```

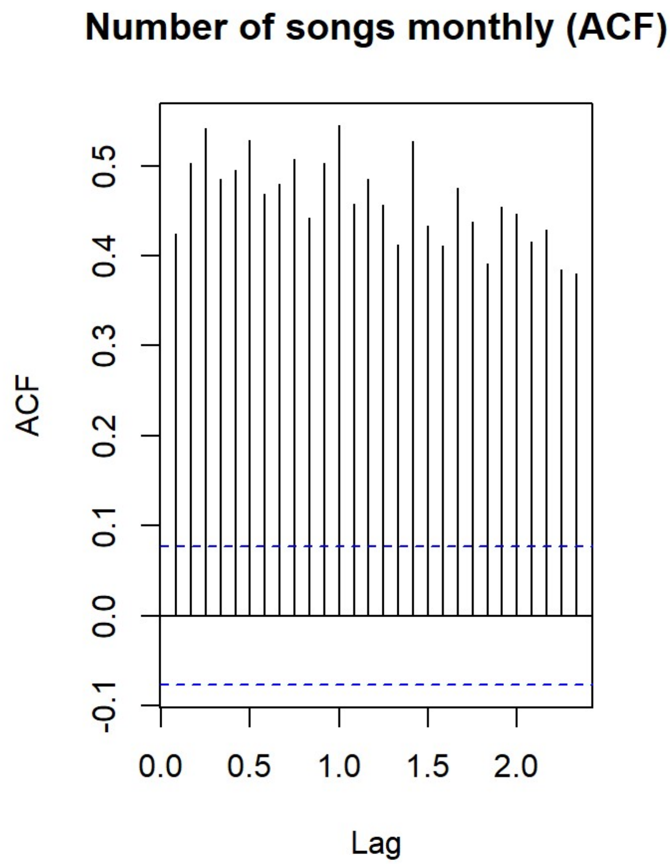
```
## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.
```

```
fname <- file.choose()
data <- read.csv(fname)
data <- data[,2]
chart = ts(data,start=c(1965,1),freq=12)
chart.dif = diff(chart)
```

### (i) Plot the Time Series and ACF plots.

```
par(mfrow=c(1,2))
plot(chart, main = "Number of songs monthly")
acf(chart, main = "Number of songs monthly (ACF)")
```

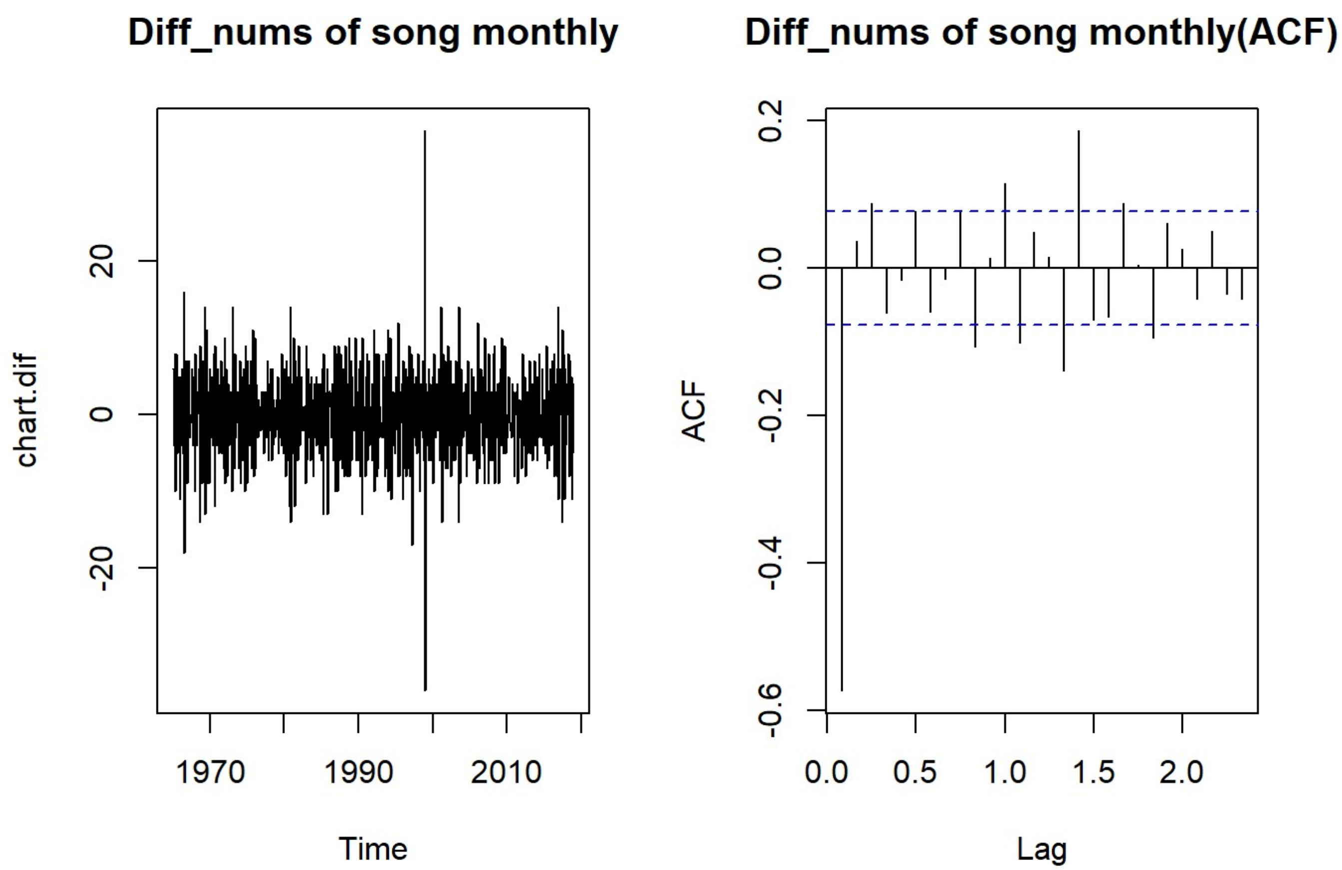



**Answer:** Three assumptions of statoinary: (a) Mean (trend). In the TS plot, it appears that the trend of songs numbers per month reduces gradually, which violents the assumption of stationary (constant mean). (b).Variance. Variance of the data applears finite and constant. (c). Independence of data. The data also violates the assumption of non-correlation with respect to time. In addition, there may have an outlier in 2000.

(ii) Plot tht Time Series and ACF plots for the differencing analysis.

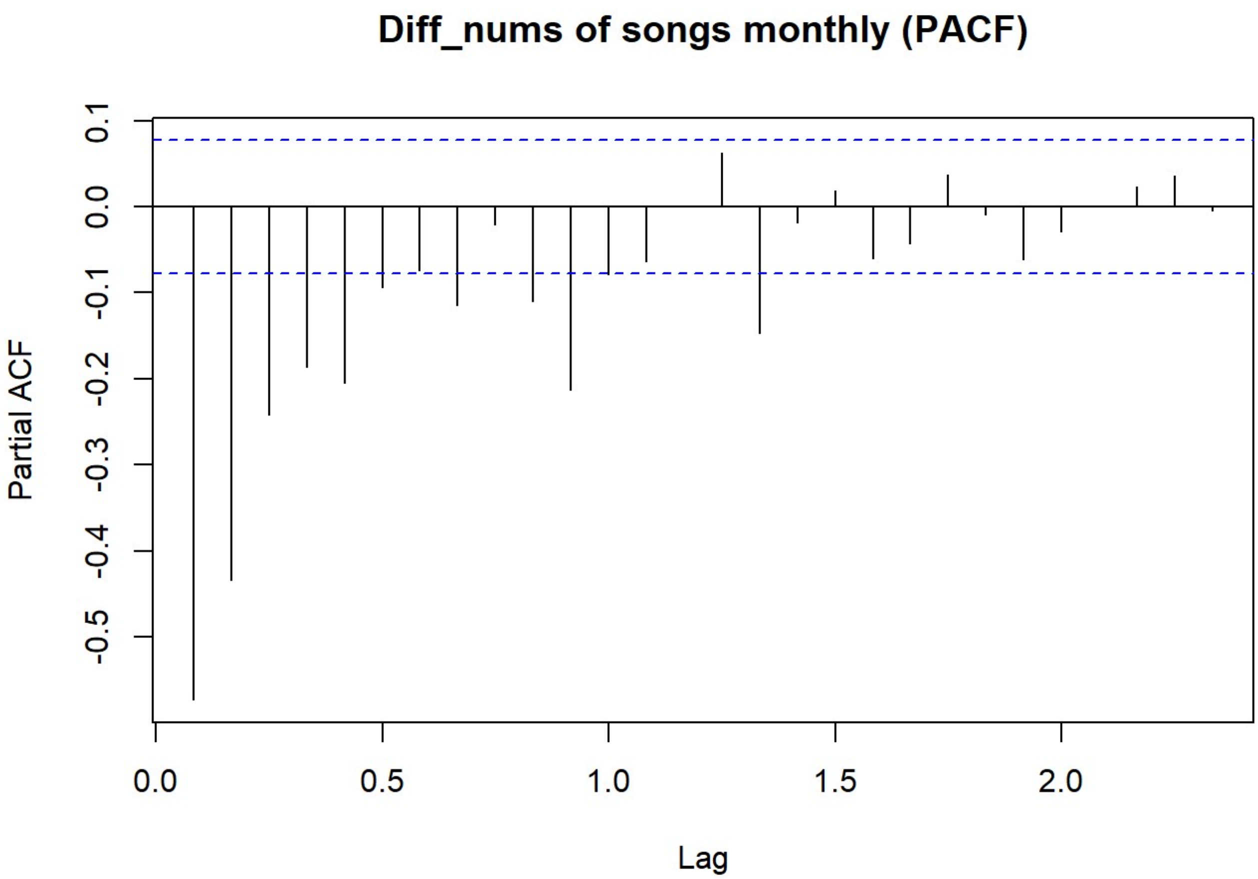
```
par(mfrow=c(1,2))
plot(chart.dif, main = "Diff_nums of song monthly")
acf(chart.dif, main = "Diff_nums of song monthly(ACF)")
```




**Answer:** In the plot, differencing data seems to have constant mean (no trend), and the variance seems finite and constant. In the plot of ACF, the data seems to have some auto-correlation with respect to time, these features are still hard to be ideal stationary because some lags are out of confident line. In addition, there may have an outlier in 2000. Therefore, the differencing data may be suitable for ARMA forecasting, which need to be further analyzed.

## **1b.** ARIMA Modelling

```
pacf(chart.dif, main = "Diff_nums of songs monthly (PACF)")
```


Diff_nums of songs monthly (PACF)
Partial ACF
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
0.0
0.5
1.0
1.5
2.0
Lag

Apply ARIMA

```r
## Order selection -- AIC
# ARIMA order
test_modelA <- function(p,d,q){
  mod = arima(chart, 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 0:3){
  for (d in 0:3){
    for (q in 0:3) {
        possibleError <- tryCatch(
          orders<-rbind(orders,test_modelA(p,d,q)),
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next
    }
  }
}
```

```
## [1] "0 0 0 4089.36281937988"
## [1] "0 0 1 4017.33723577697"
## [1] "0 0 2 3960.04080339343"
## [1] "0 0 3 3895.86480112833"
## [1] "0 1 0 4171.28191077876"
## [1] "0 1 1 3668.82010032562"
## [1] "0 1 2 3646.28341178167"
## [1] "0 1 3 3648.24830092855"
## [1] "0 2 0 4905.76017906079"
## [1] "0 2 1 4174.30239257392"
## [1] "0 2 2 3675.47403092406"
## [1] "0 2 3 3652.68942381487"
## [1] "0 3 0 5685.34958483471"
## [1] "0 3 1 4907.63911907078"
## [1] "0 3 2 4183.49691374881"
## [1] "0 3 3 3725.3601734341"
## [1] "1 0 0 3962.54137338297"
## [1] "1 0 1 3678.85678368979"
```

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

```
## [1] "1 0 2 3656.16105256723"
## [1] "1 0 3 3658.19956394165"
## [1] "1 1 0 3915.57181841383"
## [1] "1 1 1 3648.18845579501"
## [1] "1 1 2 3648.26076371727"
## [1] "1 1 3 3649.38001130163"
## [1] "1 2 0 4484.36978004299"
## [1] "1 2 1 3919.86042605723"
## [1] "1 2 2 3654.23864028266"
## [1] "1 2 3 3667.63545225831"
## [1] "1 3 0 5138.40261855915"
## [1] "1 3 1 4487.95696061233"
## [1] "1 3 2 3930.98722110641"
## [1] "1 3 3 3695.11074903703"
## [1] "2 0 0 3852.30039469936"
## [1] "2 0 1 3658.06348316059"
```

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

```
## [1] "2 0 2 3658.06138552475"
```

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

```
## [1] "2 0 3 3659.30944070717"
## [1] "2 1 0 3782.35816510705"
## [1] "2 1 1 3647.23653552014"
## [1] "2 1 2 3648.00232876075"
## [1] "2 1 3 3648.57211068919"
## [1] "2 2 0 4213.57535192067"
## [1] "2 2 1 3787.50739024648"
```

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

```
## [1] "2 2 2 3653.69729795209"
## [1] "2 2 3 3654.58784125323"
## [1] "2 3 0 4768.62627990239"
## [1] "2 3 1 4218.50778399488"
## [1] "2 3 2 3800.44121080946"
## [1] "2 3 3 3727.04113113617"
## [1] "3 0 0 3762.32278184094"
## [1] "3 0 1 3657.06181561004"
```

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

```
## [1] "3 0 2 3657.83218582684"
```

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

```
## [1] "3 0 3 3636.77202933775"
## [1] "3 1 0 3745.46604670527"
## [1] "3 1 1 3647.68656207913"
## [1] "3 1 2 3649.53918106813"
```

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

```
## [1] "3 1 3 3612.68797033985"
## [1] "3 2 0 4081.7495859725"
## [1] "3 2 1 3751.06624614169"
## [1] "3 2 2 3653.89439037076"
## [1] "3 2 3 3656.64218603347"
```

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

```
## [1] "3 3 0 4538.41569151487"
## [1] "3 3 1 4087.60059121374"
## [1] "3 3 2 3765.58336929049"
## [1] "3 3 3 3714.59648579878"
```

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

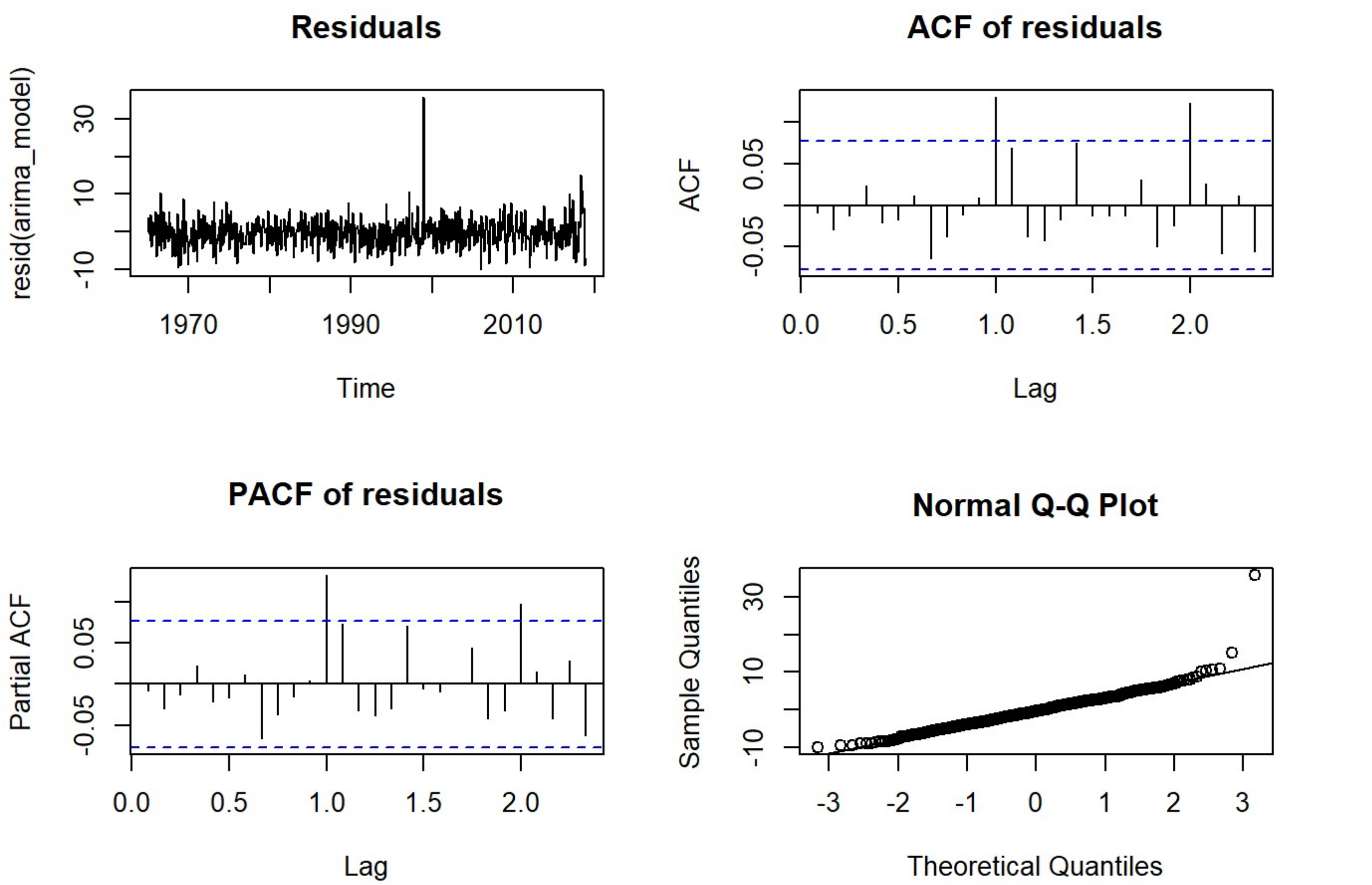
```
##    p d q      AIC
## 40 2 1 2 3648.002
## 55 3 1 1 3647.687
## 39 2 1 1 3647.237
## 8  0 1 2 3646.283
## 53 3 0 3 3636.772
## 57 3 1 3 3612.688
```

## After minimum AIC selection, We found the lowest AIC when p = 3, d = 1, q = 3, order(3,1,3), AIC = 3612.68.

```
arima_model = arima(chart, order=c(3, 1, 3), method="ML")
```

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

```
par(mfrow=c(2,2))
plot(resid(arima_model), main = "Residuals")
acf(resid(arima_model), main = "ACF of residuals")
pacf(resid(arima_model), main = "PACF of residuals")
qqnorm(resid(arima_model))
qqline(resid(arima_model))
```




In the plot the residuals, it seems to envenly distributed, just one outlier maybe at 2000. For plots of ACF and PACF, the seems to have several autocorrelations in the middle position, which indicates that it is not stationary.

Test for independence for final model

```
# p = 3, d = 1, q = 3, lag = 3+1+3 = 7, fitdf = 3+3 = 6
Box.test(resid(arima_model), lag=7, type ="Box-Pierce", fitdf = 6)
```

```
##
##  Box-Pierce test
##
## data:  resid(arima_model)
## X-squared = 1.623, df = 1, p-value = 0.2027
```

```
Box.test(resid(arima_model), lag=7, type ="Ljung-Box", fitdf = 6)
```

```
##
##  Box-Ljung test
##
## data:  resid(arima_model)
## X-squared = 1.6375, df = 1, p-value = 0.2007
```

For the Box-Pierce and Box-Ljung, the null hypothesis that there is uncorrelation for the model, but p-values of them are a little high, which means that we can not reject the assumption of uncorrelated residuals, so it is not stationary.

# **1c.** Forecasting

For arima(2,1,4) model

```
arima_mod6 = arima(chart, order = c(2,1,4), method = "ML")
arima_mod6
```

```
##
## Call:
## arima(x = chart, order = c(2, 1, 4), method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1      ma2      ma3     ma4
##       -1.1589  -0.9993  0.0912  -0.0963  -0.8896  0.1598
## s.e.   0.0017   0.0011  0.0397   0.0186   0.0187  0.0387
##
## sigma^2 estimated as 15.05:  log likelihood = -1799.39,  aic = 3610.78
```

The coefficence is produced by the ARIMA(2,1,4) model, which is significant. The model formular is

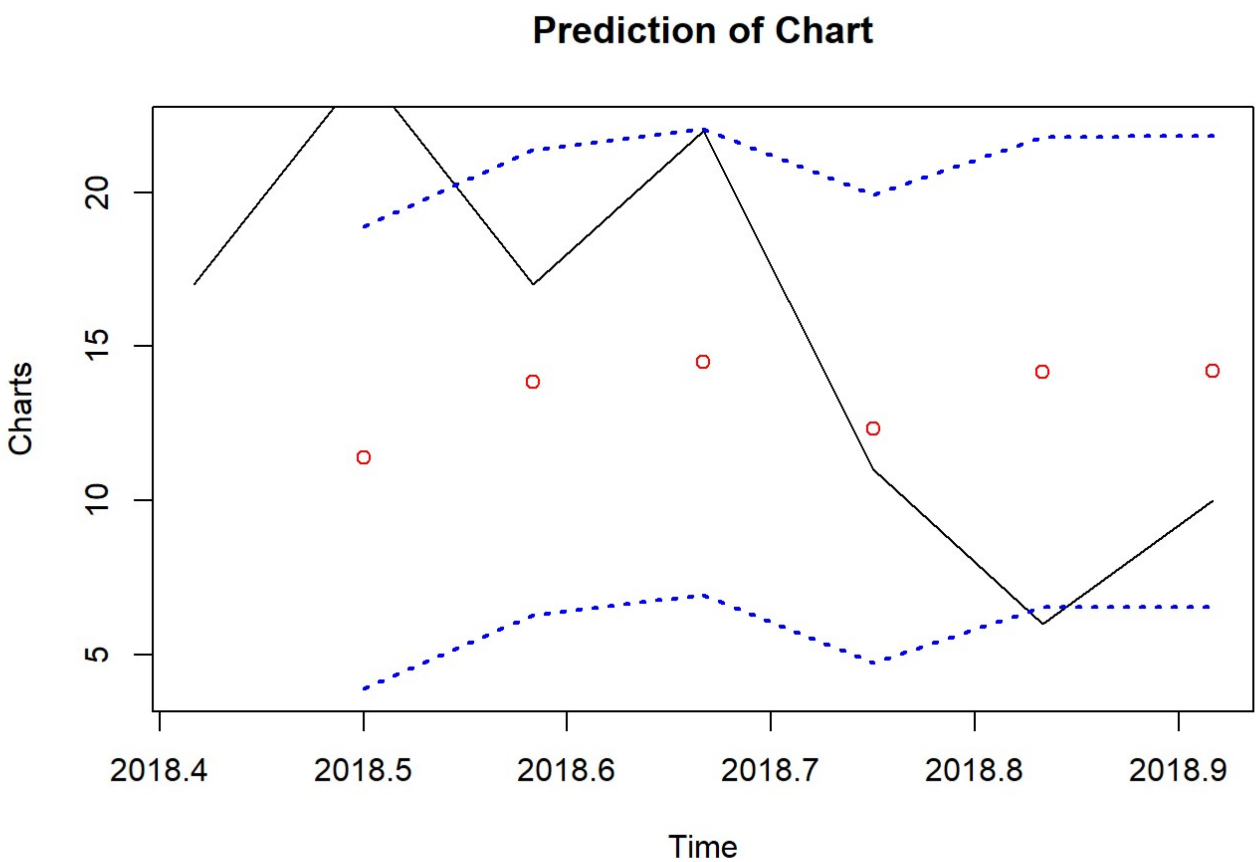
Xt = -1.1598Xt-1 - 0.9993Xt-2 + 0.0912Zt-1 - 0.0963Zt-2 - 0.8896Zt-3 + 0.1598Zt-4 + Zt

Forecasting with ARIMA: 6 months ahead

```
n = length(chart)
nfit = n - 6
new_chart = arima(chart[1:nfit], order = c(2,1,4), method = "ML")
pred_chart = as.vector(predict(new_chart, n.ahead=6))
timevol = time(chart)

ubound = pred_chart$pred + 1.96*pred_chart$se
lbound = pred_chart$pred - 1.96*pred_chart$se
ymin = min(lbound)
ymax = max(ubound)

# plot only the last 6 months of data
plot(timevol[(n-6):n], chart[(n-6):n], type="l", ylim=c(ymin,ymax), xlab="Time", ylab
="Charts", main = "Prediction of Chart")
points(timevol[(nfit+1):n], pred_chart$pred, col="red")
lines(timevol[(nfit+1):n], ubound, lty=3,lwd= 2, col="blue")
lines(timevol[(nfit+1):n], lbound, lty=3,lwd= 2, col="blue")
```




The blue line is confidence intervals, the red dots are predicted results and black line is the actual observed data. From the predicting result, it seems only one predicting result (4th dot) is closed to the actual observation, which means the predicting results are not very good.

Compute Accuracy Measures (Precisions)

```
## Compute Accuracy Measures (Precisions)
obs_data = chart[(nfit + 1) : n]
pred_data = pred_chart$pred

### Mean Absolute Prediction Error (MAE)
mean(abs(pred_data - obs_data))
```

```
## [1] 6.165023
```

```
### Mean Absolute Percentage Error (MAPE)
mean(abs(pred_data - obs_data)/obs_data)
```

```
## [1] 0.4927662
```

```
### Precision Measure (PM)
sum((pred_data-obs_data)^2) / sum((obs_data-mean(obs_data))^2)
```

```
## [1] 1.217197
```

The MAE is 6.165, MAPE = 49.27%, PM = 1.217. Because MAPE is 50%, which is very high and means that the predicting accuracy is not high, probably ARIMA model could not predict well at this situation.

# **Question 2:** ARIMA Model Comparisons for Currency Conversion Rates

```
#Libraries and Data
rm(list=ls())
library(TSA)
library(mgcv)

#USD to EU
fname <- file.choose()
data1 <- read.csv(fname)
data1 <- data1[,2]
EU = ts(data1,start=c(2014),freq=52)

#USD to GBP

fname <- file.choose()
data2 <- read.csv(fname)
data2 <- data2[,2]
GBP = ts(data2,start=c(2014),freq=52)
```

## **2a.** ARIMA fitting

```
#ARIMA order (EU)
test_modelA <- function(p,d,q){
  mod = arima(EU, 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 0:3){
  for (d in 0:3){
    for (q in 0:2) {
        possibleError <- tryCatch(
          orders<-rbind(orders,test_modelA(p,d,q)),
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next
    }
  }
}
```

```
## [1] "0 0 0 -569.420853736522"
## [1] "0 0 1 -917.969448175781"
## [1] "0 0 2 -1172.65879202353"
## [1] "0 1 0 -1784.85289049491"
## [1] "0 1 1 -1788.70792245714"
## [1] "0 1 2 -1786.81923283967"
## [1] "0 2 0 -1622.22817773909"
## [1] "0 2 1 -1774.06048056079"
## [1] "0 2 2 -1776.50757790129"
## [1] "0 3 0 -1323.93615177463"
## [1] "0 3 1 -1607.89914996422"
## [1] "0 3 2 -1752.82789604421"
## [1] "1 0 0 -1782.87765210213"
## [1] "1 0 1 -1786.92093245319"
## [1] "1 1 0 -1788.22872828377"
## [1] "1 1 1 -1786.76302109289"
## [1] "1 1 2 -1784.94794213084"
## [1] "1 2 0 -1669.12059630621"
## [1] "1 2 1 -1775.9706223139"
## [1] "1 2 2 -1774.67378352835"
## [1] "1 3 0 -1451.99200155784"
## [1] "1 3 1 -1653.95788331857"
## [1] "1 3 2 -1753.94478910937"
## [1] "2 0 0 -1786.46590069187"
## [1] "2 0 1 -1784.96352201317"
## [1] "2 1 0 -1787.24390722201"
## [1] "2 1 1 -1785.32555803117"
```

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

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

```
## [1] "2 1 2 -1798.00812015844"
## [1] "2 2 0 -1694.18164066097"
## [1] "2 2 1 -1775.66283339909"
## [1] "2 2 2 -1773.98221697775"
## [1] "2 3 0 -1504.21000382379"
## [1] "2 3 1 -1678.40367928103"
## [1] "2 3 2 -1754.05552267041"
```

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

```
## [1] "3 0 0 -1572.09389517767"
```

```
## 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 1 -1782.57505604075"
## [1] "3 0 2 -1787.57246385201"
## [1] "3 1 0 -1785.65739135954"
## [1] "3 1 1 -1784.37651976233"
## [1] "3 1 2 -1796.0987647007"
## [1] "3 2 0 -1727.84285678383"
## [1] "3 2 1 -1774.92023437415"
## [1] "3 2 2 -1773.5370075969"
## [1] "3 3 0 -1558.10997463237"
## [1] "3 3 1 -1711.35372142911"
## [1] "3 3 2 -1754.19850107986"
```

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

```
##    p d q       AIC
## 27 2 1 0 -1787.244
## 38 3 0 2 -1787.572
## 16 1 1 0 -1788.229
## 6  0 1 1 -1788.708
## 41 3 1 2 -1796.099
## 29 2 1 2 -1798.008
```

We found arima(2,1,2) has lowest AIC (-1798.008) when using EU.

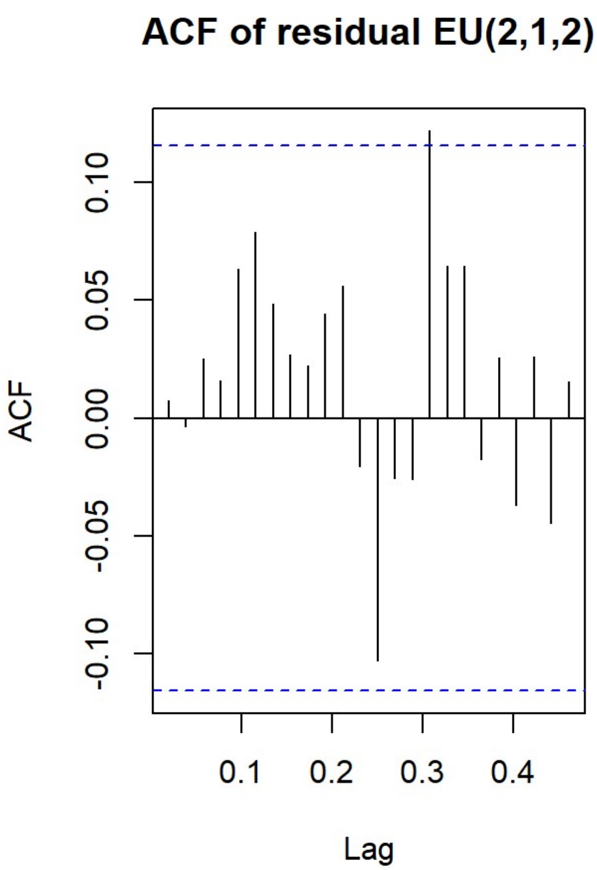
## ARIMA model fitting and test (EU)

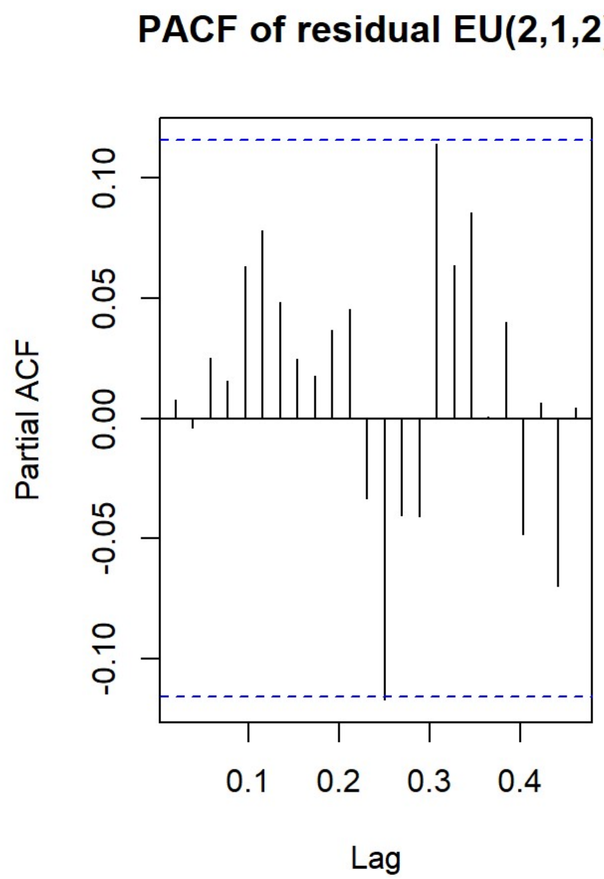
```
EU_ARIMA = arima(EU, order=c(2,1,2), method = "ML")
```

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

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

```
par(mfrow=c(1,2))
acf(resid(EU_ARIMA), main = "ACF of residual EU(2,1,2)")
pacf(resid(EU_ARIMA), main = "PACF of residual EU(2,1,2)")
```




```
## Box test
Box.test(resid(EU_ARIMA), lag = 5, type = "Box-Pierce", fitdf = 4)
```

```
##
##  Box-Pierce test
##
## data:  resid(EU_ARIMA)
## X-squared = 1.4178, df = 1, p-value = 0.2338
```

```
Box.test(resid(EU_ARIMA), lag = 5, type = "Ljung-Box", fitdf = 4)
```

```
##
##  Box-Ljung test
##
## data:  resid(EU_ARIMA)
## X-squared = 1.4511, df = 1, p-value = 0.2284
```

For EU arima model, we have null hypothesis that there is no autocorrelation, since p-value is high, which means that we fail to reject null hypothesis, so there have no signicant autocorrelation in arima(2,1,2) model.

```
#ARIMA order
test_modelA <- function(p,d,q){
  mod = arima(GBP, 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 0:3){
  for (d in 0:3){
    for (q in 0:2) {
        possibleError <- tryCatch(
          orders<-rbind(orders,test_modelA(p,d,q)),
          error=function(e) e
        )
        if(inherits(possibleError, "error")) next

    }
  }
}
```

```
## [1] "0 0 0 -294.663964544714"
## [1] "0 0 1 -658.997635196966"
## [1] "0 0 2 -939.93637211017"
## [1] "0 1 0 -1619.85724246706"
## [1] "0 1 1 -1631.80095196078"
## [1] "0 1 2 -1631.01849308131"
## [1] "0 2 0 -1476.94141415387"
## [1] "0 2 1 -1608.08134363572"
## [1] "0 2 2 -1619.63606508976"
## [1] "0 3 0 -1194.18362282052"
## [1] "0 3 1 -1463.11675986701"
## [1] "0 3 2 -1584.98478390266"
## [1] "1 0 0 -1616.91059168528"
## [1] "1 0 1 -1629.12875099032"
## [1] "1 0 2 -1628.21555964443"
## [1] "1 1 0 -1628.84298539492"
## [1] "1 1 1 -1630.53419482776"
## [1] "1 1 2 -1629.50656416397"
## [1] "1 2 0 -1506.16260565096"
## [1] "1 2 1 -1616.6637017359"
## [1] "1 2 2 -1618.4320937032"
## [1] "1 3 0 -1289.78087905122"
## [1] "1 3 1 -1491.67451052975"
```

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

```
## [1] "1 3 2 -1593.44217963653"
## [1] "2 0 0 -1626.18347599749"
## [1] "2 0 1 -1627.79422165298"
## [1] "2 0 2 -1630.63340960347"
## [1] "2 1 0 -1632.05834292573"
## [1] "2 1 1 -1630.10642864094"
## [1] "2 1 2 -1629.9632069891"
## [1] "2 2 0 -1542.94155479698"
## [1] "2 2 1 -1620.15552253138"
## [1] "2 2 2 -1618.18511908651"
## [1] "2 3 0 -1355.95668239991"
## [1] "2 3 1 -1527.71203946614"
## [1] "2 3 2 -1596.71710867782"
## [1] "3 0 0 -1512.31858802444"
```

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

```
## [1] "3 0 1 -1627.07251843573"
## [1] "3 1 0 -1630.14796794164"
## [1] "3 1 1 -1628.22885253255"
## [1] "3 1 2 -1629.68453359146"
## [1] "3 2 0 -1565.58861769817"
## [1] "3 2 1 -1618.20782422363"
## [1] "3 2 2 -1616.78165254184"
## [1] "3 3 0 -1420.58563869323"
## [1] "3 3 1 -1549.76959159329"
```

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

## Warning in log(s2): NaNs produced

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

```
## [1] "3 3 2 -1594.71625520281"
```

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

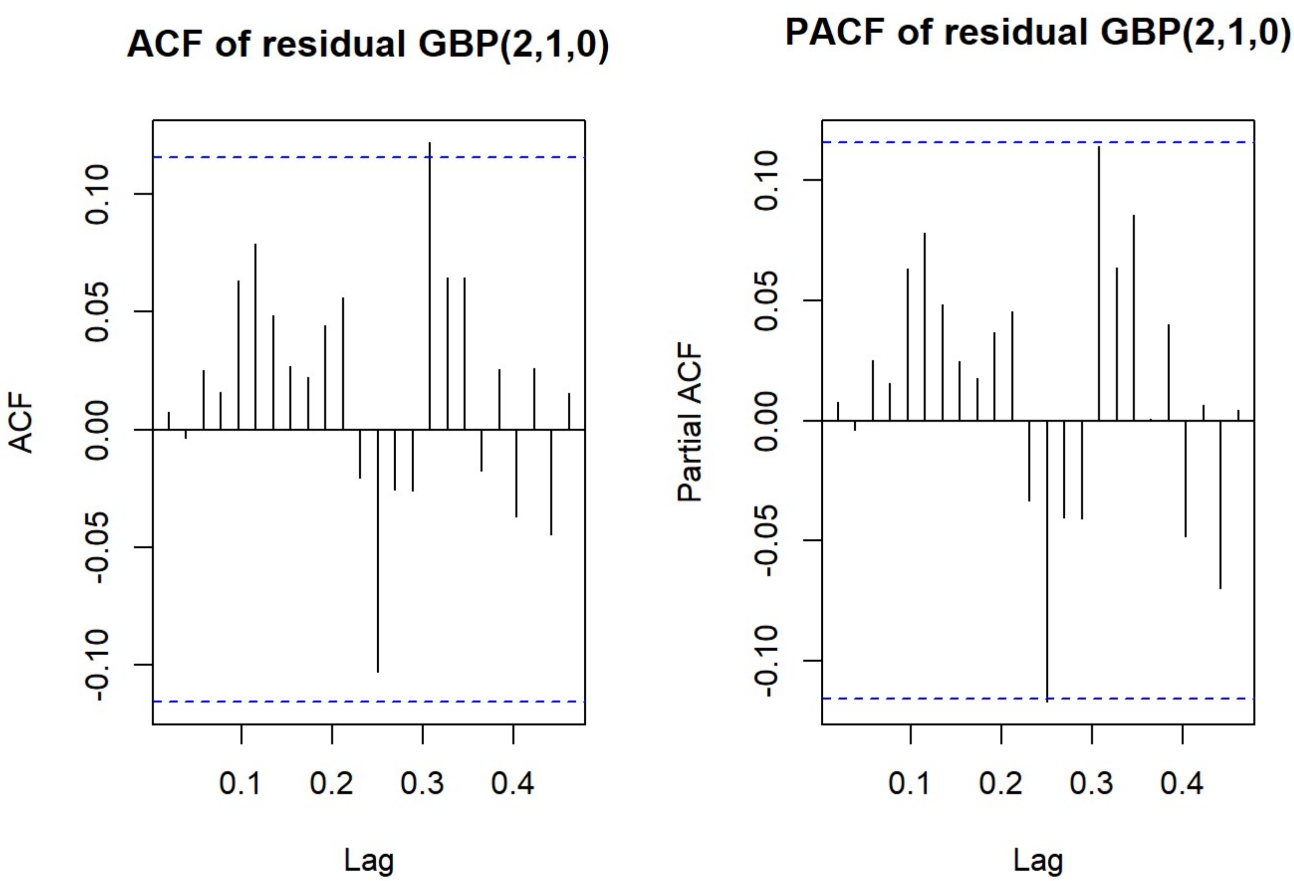
```
##    p d q       AIC
## 40 3 1 0 -1630.148
## 18 1 1 1 -1630.534
## 28 2 0 2 -1630.633
## 7  0 1 2 -1631.018
## 6  0 1 1 -1631.801
## 29 2 1 0 -1632.058
```

We found arima(2,1,0) has lowest AIC(-1632.058) when using GBP.

ARIMA model fitting and test (GBP)

```
GBP_ARIMA = arima(GBP, order=c(2,1,0), method = "ML")

par(mfrow=c(1,2))
acf(resid(EU_ARIMA), main = "ACF of residual GBP(2,1,0)")
pacf(resid(EU_ARIMA), main = "PACF of residual GBP(2,1,0)")
```




```
## Box test
Box.test(resid(GBP_ARIMA), lag = 3, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  resid(GBP_ARIMA)
## X-squared = 0.06064, df = 1, p-value = 0.8055
```

```
Box.test(resid(GBP_ARIMA), lag = 3, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  resid(GBP_ARIMA)
## X-squared = 0.06163, df = 1, p-value = 0.8039
```

For GBP arima model, We have null hypothesis that there is no autocorrelation, since p-value is very high, which means that we could not reject null hypothesis, so there have no signicant autocorrelation in arima (2,1,0) model.

# **2b.** Forecasting

### Forecasting with ARIMA, EU(2,1,2): 12 weeks ahead

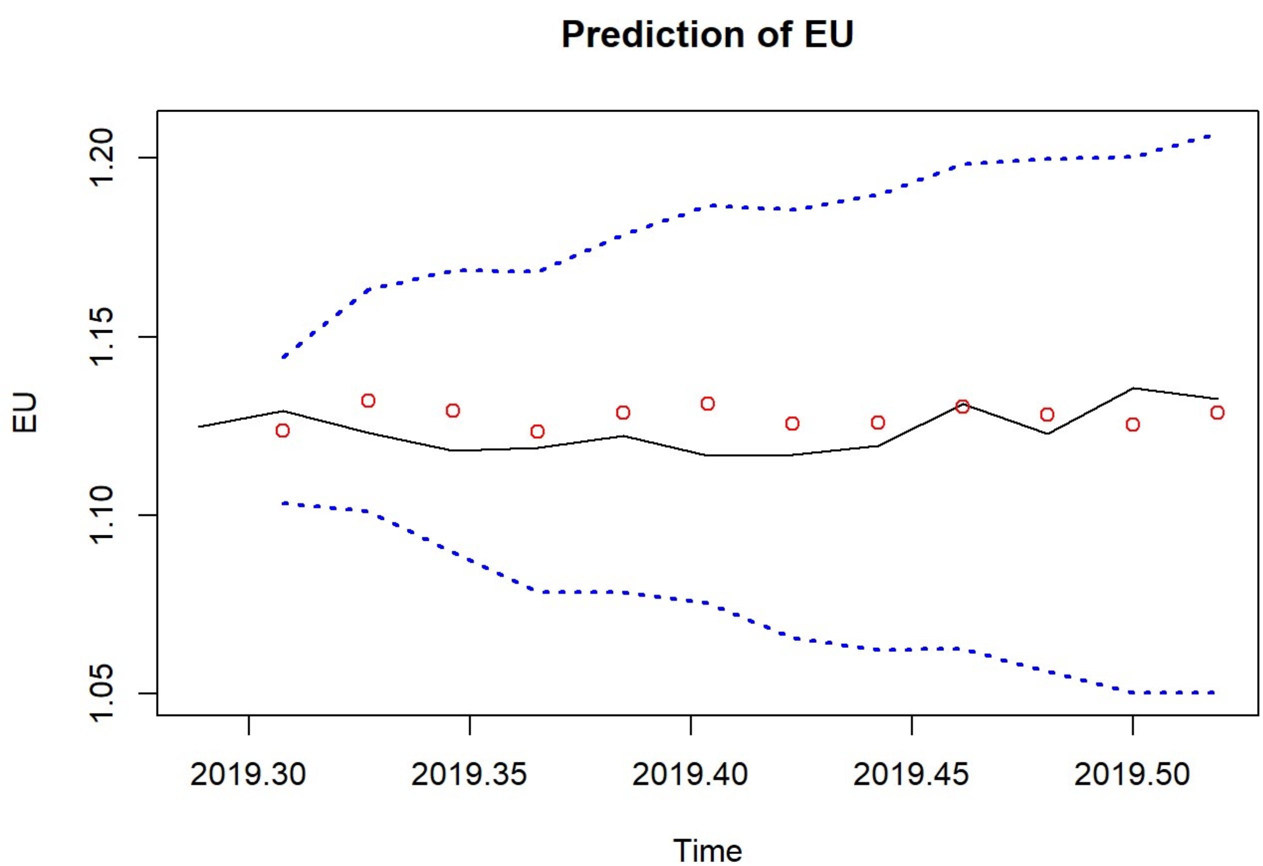
```
n = length(EU)
nfit = n - 12
new_EU = arima(EU[1:nfit], order = c(2,1,2), method = "ML")
new_EU
```

```
##
## Call:
## arima(x = EU[1:nfit], order = c(2, 1, 2), method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1     ma2
##       -0.4565  -0.8497  0.6082  0.9584
## s.e.   0.0598   0.0435  0.0412  0.0357
##
## sigma^2 estimated as 0.0001086:  log likelihood = 864.12,  aic = -1720.24
```

```
pred_EU = as.vector(predict(new_EU, n.ahead=12))
timevol = time(EU)

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 only the last 6 months of data
plot(timevol[(n-12):n], EU[(n-12):n], type="l", ylim=c(ymin,ymax), xlab="Time", ylab="
EU", main = "Prediction of EU")
points(timevol[(nfit+1):n], pred_EU$pred, col="red")
lines(timevol[(nfit+1):n], ubound, lty=3,lwd= 2, col="blue")
lines(timevol[(nfit+1):n], lbound, lty=3,lwd= 2, col="blue")
```




The blue is confidence intervals, the red dots are predicted results and black line is the actuall observed data. From the predicting result, it seems the predicting results are very closed to the actual observation, which means the predicting results are very perfect with high accuracy.

Compute Accuracy Measures (EU(2,1,2))

```
obs_data = EU[(nfit + 1) : n]
pred_data = pred_EU$pred

### Mean Absolute Percentage Error (MAPE)
mean(abs(pred_data - obs_data)/obs_data)
```

```
## [1] 0.006380015
```

```
### Precision Measure (PM)
sum((pred_data-obs_data)^2) / sum((obs_data-mean(obs_data))^2)
```

```
## [1] 1.616309
```

Above results indicate that MAPE is 0.638% and PM is 1.616. Since MAPE is very low, so the arima model could predict very well.

### Forecasting with ARIMA, GBP(2,1,0): 12 weeks ahead

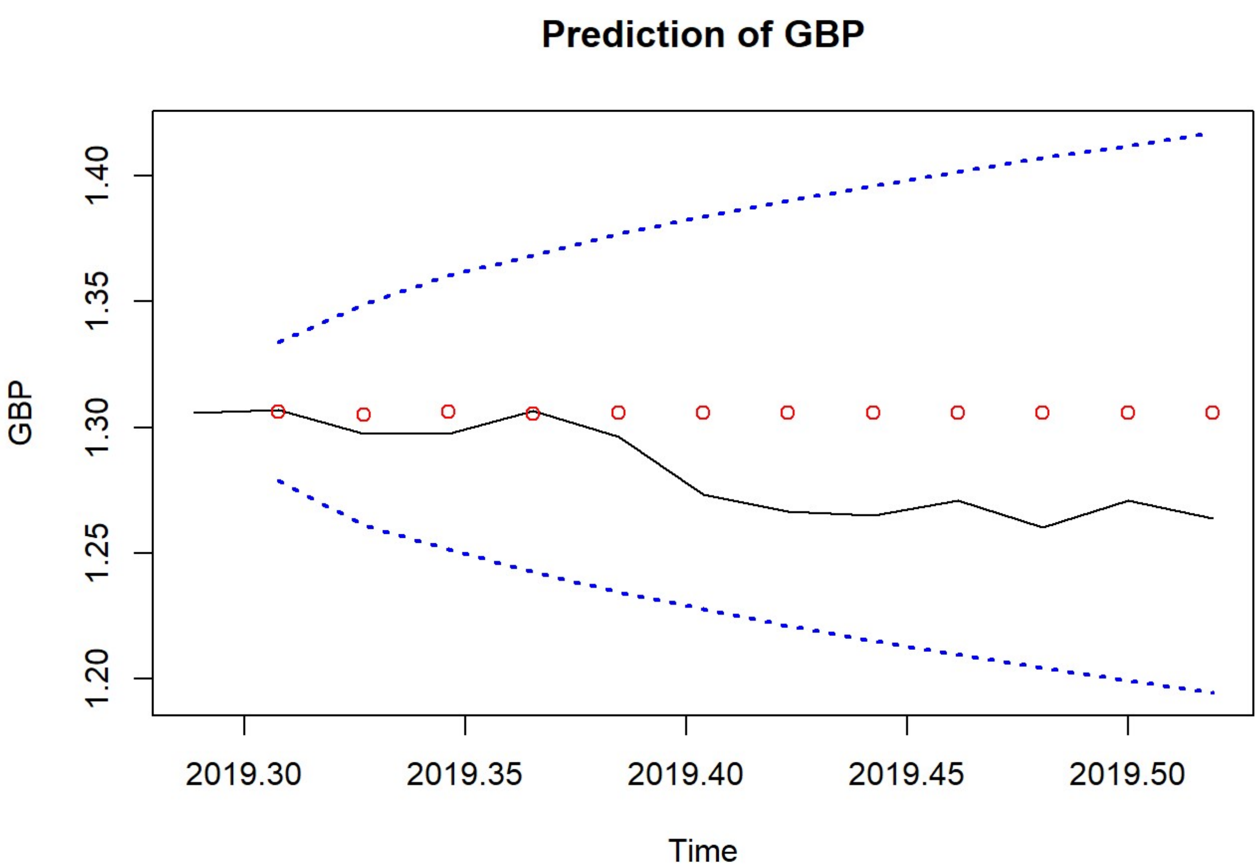
```
n = length(GBP)
nfit = n - 12
new_GBP = arima(GBP[1:nfit], order = c(2,1,2), method = "ML")
new_GBP
```

```
##
## Call:
## arima(x = GBP[1:nfit], order = c(2, 1, 2), method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1     ma2
##       -1.0516  -0.3997  1.2907  0.5898
## s.e.   0.2898   0.1653  0.2758  0.1559
##
## sigma^2 estimated as 0.0001969:  log likelihood = 782.99,  aic = -1557.99
```

```
pred_GBP = as.vector(predict(new_GBP, n.ahead=12))
timevol = time(GBP)

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 only the last 6 months of data
plot(timevol[(n-12):n], GBP[(n-12):n], type="l", ylim=c(ymin,ymax), xlab="Time", ylab
="GBP", main = "Prediction of GBP")
points(timevol[(nfit+1):n], pred_GBP$pred, col="red")
lines(timevol[(nfit+1):n], ubound, lty=3,lwd= 2, col="blue")
lines(timevol[(nfit+1):n], lbound, lty=3,lwd= 2, col="blue")
```




The blue is confidence intervals, the red dots are predicted results and black line is the actuall observed data. From the predicting result, it seems the first four points of predicting results are very closed to the actual observation, which means the some of predicting results are good.

Compute Accuracy Measures (GBP(2,1,0))

```
obs_data = GBP[(nfit + 1) : n]
pred_data = pred_GBP$pred

### Mean Absolute Percentage Error (MAPE)
mean(abs(pred_data - obs_data)/obs_data)
```

```
## [1] 0.01940747
```

```
### Precision Measure (PM)
sum((pred_data-obs_data)^2) / sum((obs_data-mean(obs_data))^2)
```

```
## [1] 2.999087
```

Above results indicate that MAPE is 1.9% and PM is 2.999. MAPE indicates that the prediction of this arima model is acceptable.

Comparison and summary: For the arima models of EU and GBP, the plots of predicting result could give us the first impression that EU model is better than GBP, because its predicting result looks closer to actual observation. Moreover, MAPE of EU (0.638%) and GBP(1.9%), PM of EU(1.616) and GBP(2.99), all those measurements indicates that EU model is better for accurate prediction than GBP.

# **Question 3:** Reflection on ARIMA

**Answer:** Firstly, we need to analyze data that whether it matches the assumption of stationary, including constant mean, finite and constant variance, and no autocorrelation. Secondly, we need to make plots of ACF and PACF, also box test for further analysis whether there is autocorrelation or not. Furthermore, we need to optmize the ARIMA model with minimum AIC, once we find the ideal ARIMA model with optimal parameter (p,d,q), we could perform precision measurement of ARIMA model. Therefore, the good ARIMA model is ideal for prediction.