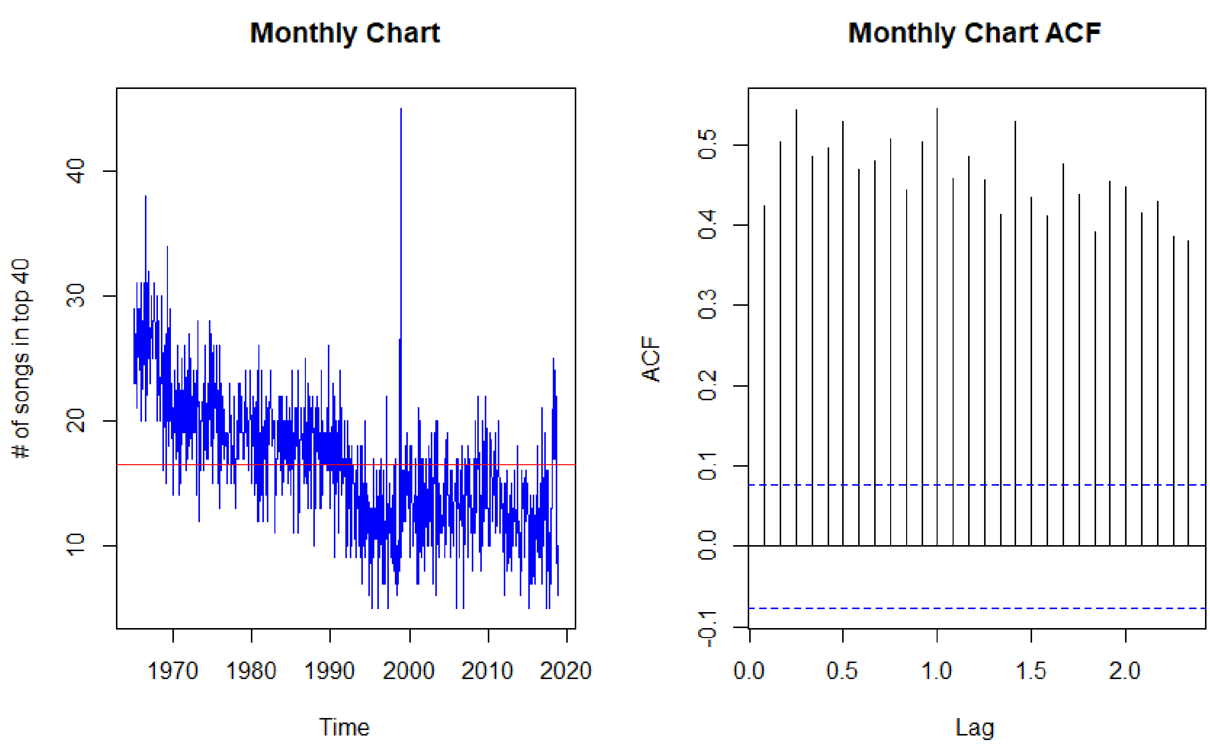
npark62 - ISYE6502 Homework 2 – 2/16/2020

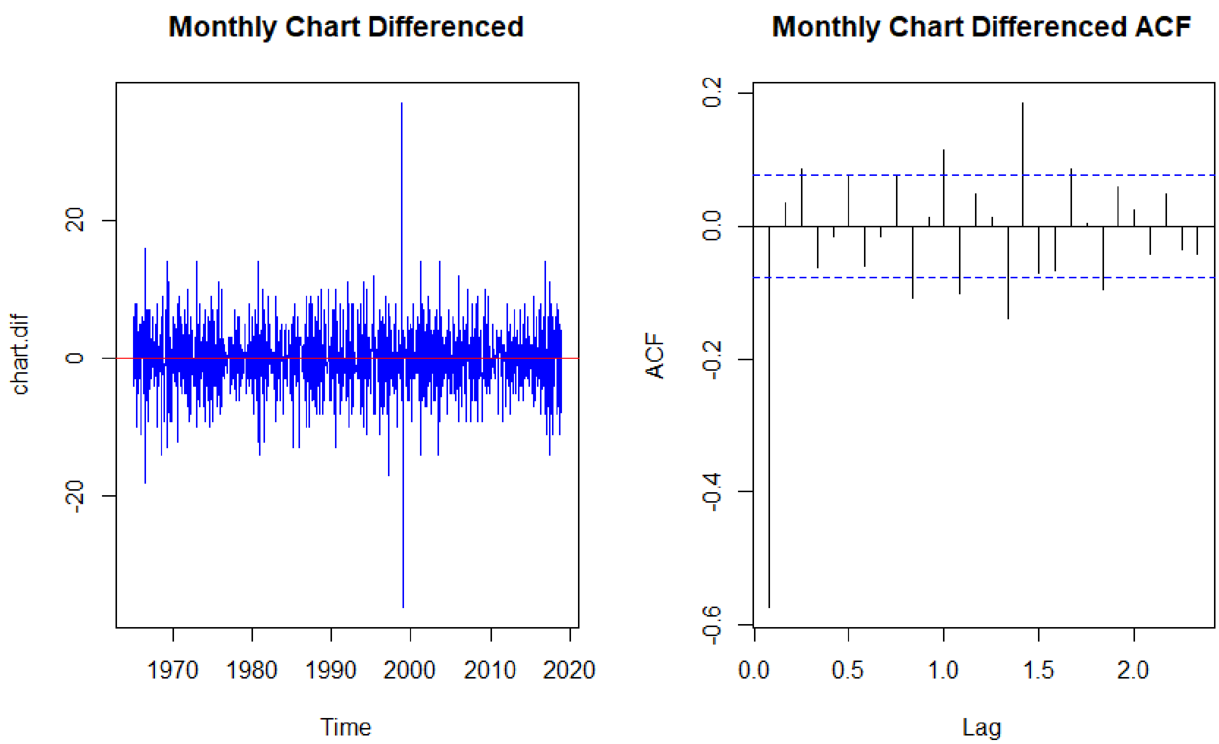
(All discussions are included as part of code as #comments)

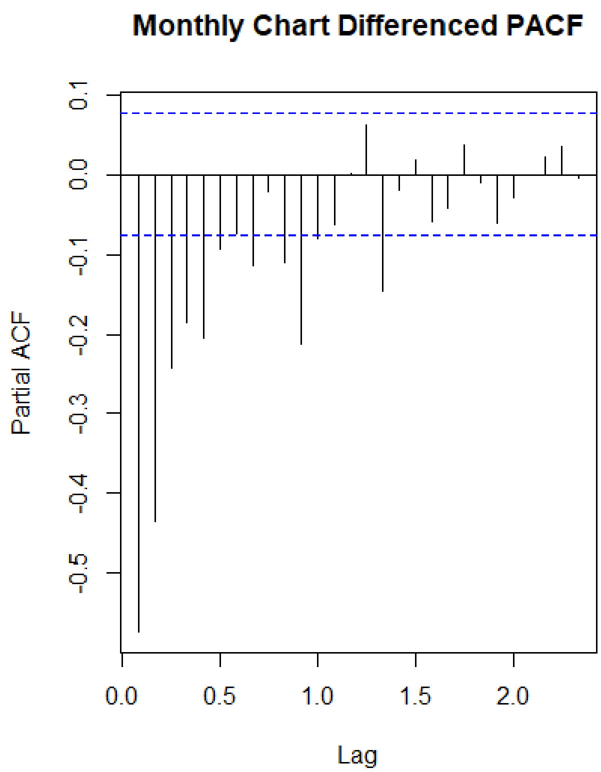
```
#npark62 ISYE6402 HW2

#Load chart_monthly.csv

library(TSA)
library(mgcv)

fname <- file.choose("C://Users//nicholas.park//Downloads//chart_monthly")
data <- read.csv(fname)
data <- data[,2]
chart = ts(data,start=c(1965,1),freq=12)
chart.dif = diff(chart)

#1a
par(mfrow=c(1,2))

ts.plot(chart,main='Monthly Chart',ylab='# of songs in top 40', col='blue')
abline(a=mean(chart),b=0,col='red')
acf(chart,main='Monthly Chart ACF')
```




```
# Unclear/inconsistent downward trend. Needs more quantitative testing to
understand trend.
# 1. Non-constant mean.
```

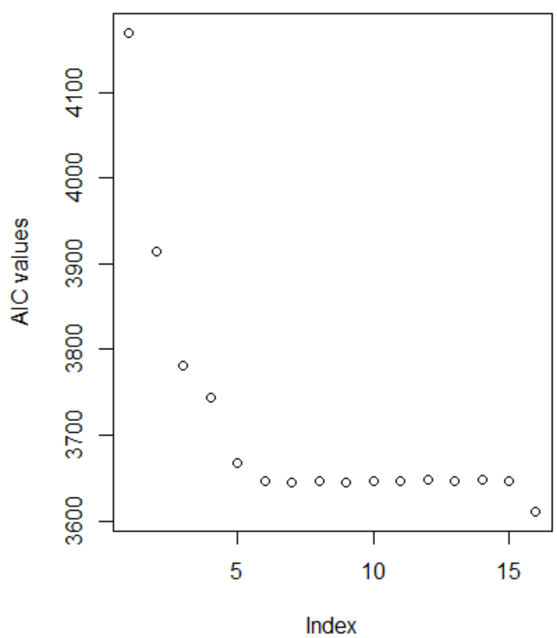
```
# 2. Seemingly constant (& finite) variance except for a suddenly large
variance in 2000 due to the spike in the data.
# 3. Significant autocorrelation is present.
# Thus, not stationary.

par(mfrow=c(1,2))
ts.plot(chart.dif,main='Monthly Chart Differenced', col='blue')
abline(a=mean(chart.dif),b=0,col='red')
acf(chart.dif,main='Monthly Chart Differenced ACF')
pacf(chart.dif,main='Monthly Chart Differenced PACF')
```





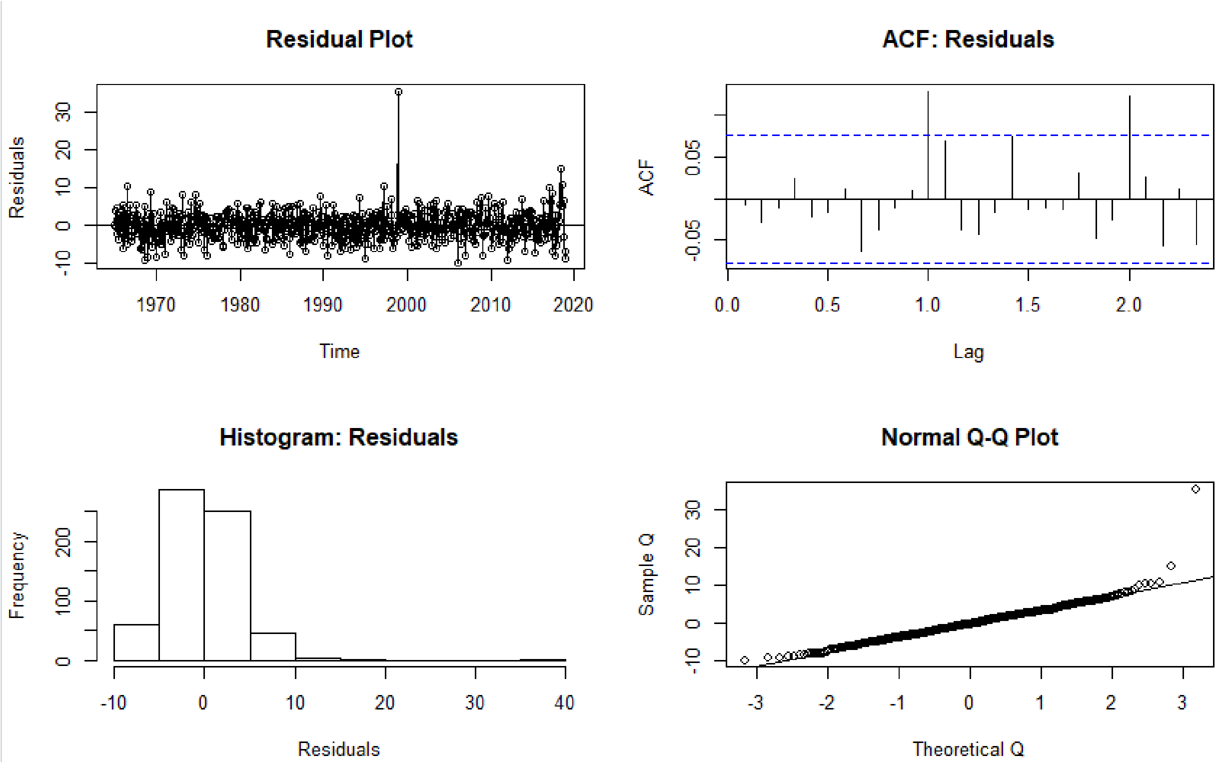

```
# Analysis:
# 1. Constant mean.
# 2. Heteroscedasticity is very minor. Seemingly constant (& finite) variance
except for near year 2000.
# 3. ACF shows autocorrelation at lag 0.1.
# 4. PACF shows significant partial autocorrelation up to lag 1.
# Due to the ACF shown, the different time series is not so stationary and
doesn't quite resemble white noise.
# Additional modeling is likely necessary for further improvement, as we will
prove in the next question.


#1b


n = length(chart)
norder = 4 # since max order is 3.
p = c(1:norder)-1; q = c(1:norder)-1
aic = matrix(0,norder,norder)
for(i in 1:norder){
  for(j in 1:norder){
    modij = arima(chart,order = c(p[i],1,q[j]), method='ML')
    aic[i,j] = modij$aic-2*(p[i]+q[j]+1)+2*(p[i]+q[j]+1)*n/(n-p[i]-q[j]-2)
  }
}

aic2 = matrix(0,norder,norder)
for(i in 1:norder){
  for(j in 1:norder){
    modij = arima(chart,order = c(p[i],0,q[j]), method='ML')
    aic2[i,j] = modij$aic-2*(p[i]+q[j]+1)+2*(p[i]+q[j]+1)*n/(n-p[i]-q[j]-2)
```

```
  }
}

min(aic) #3610.863
min(aic2) #3634.947

# since the minimum occurs when the differencing order d = 1 as opposed to 0.
# We move on to finding the p & q values of ARIMA(p, 1, q)

aicv = as.vector(aic)
plot(aicv,ylab="AIC values")
which(aic==min(aic)) # 16 which is (i,j)=(4,4) where 3610.863 occurs.

# Therefore ARIMA(3,1,3) is suitable due to lowest AIC as shown above.
```



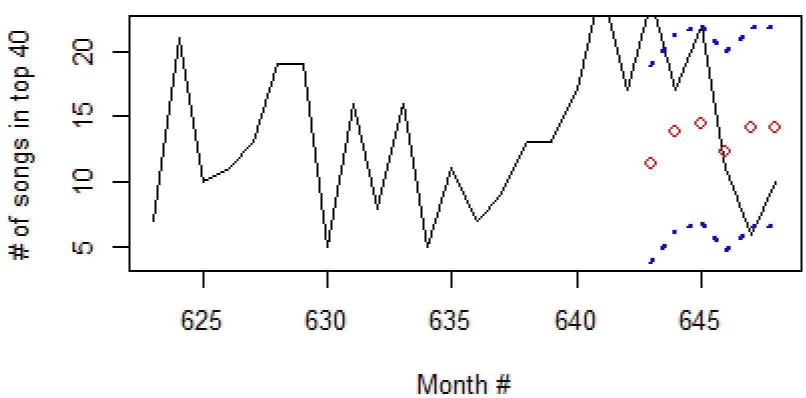

```
final_model = arima(chart, order = c(3,1,3), method = "ML")

# Call:
#   arima(x = chart, order = c(3, 1, 3), method = "ML")
#
# Coefficients:
#   ar1      ar2      ar3     ma1      ma2      ma3
# -1.3082  -1.1759  -0.1525  0.2440  -0.0409  -0.9053
# s.e.   0.0420   0.0486   0.0420  0.0194   0.0212   0.0173
#
# sigma^2 estimated as 15.05:  log likelihood = -1799.34,  aic = 3610.69

par(mfrow=c(2,2))
plot(resid(final_model), ylab='Residuals',type='o',main="Residual Plot")
abline(h=0)
acf(resid(final_model),main="ACF: Residuals")
```

```r
hist(resid(final_model),xlab='Residuals',main='Histogram: Residuals')
qqnorm(resid(final_model),ylab="Sample Q",xlab="Theoretical Q")
qqline(resid(final_model))
```




```r
# Residual plot does not seem to have a pattern.
# Variance is constant. ACF is not too significant beyond the confidence bands for the most part.
# QQ Norm seems to have a tail on both left and right, but for the most part is symmetric.


Box.test(final_model$resid, lag = (3+3+1), type = "Box-Pierce", fitdf = (3+3))
Box.test(final_model$resid, lag = (3+3+1), type = "Ljung-Box", fitdf = (3+3))
# P-values are 0.2027 and 0.2007, which are high enough to accept the null hypthesis that the residuals consist of uncorrelated variables.


# Question 1c



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

# 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 = 3612.78

# ARIMA n full form: y_t where [x_t = y_t - y_(t-1)] &
# x_t + 1.1589*X_(t-1) + 0.9993*x_(t-2) = 0.0912*w_(t-1) - 0.0963*w_(t-2) -
0.8896*w_(t-3) + 0.1598*w_(t-4)

time.pts = c(1:length(chart))
n = length(chart)
nfit = n-6
model = arima(chart[1:nfit], order = c(2,1,4),method = "ML")
outpred = predict(model,n.ahead=6)
ubound = outpred$pred+1.96*outpred$se
lbound = outpred$pred-1.96*outpred$se
ymin = min(lbound)
ymax = max(ubound)
plot(time.pts[(n-25):n],chart[(n-25):n],type="l", ylim=c(ymin,ymax),
xlab="Month #", ylab="# of songs in top 40")
points(time.pts[(nfit+1):n],outpred$pred, col="red")
lines(time.pts[(nfit+1):n],ubound,lty=3,lwd= 2, col="blue")
lines(time.pts[(nfit+1):n],lbound,lty=3,lwd= 2, col="blue")
```



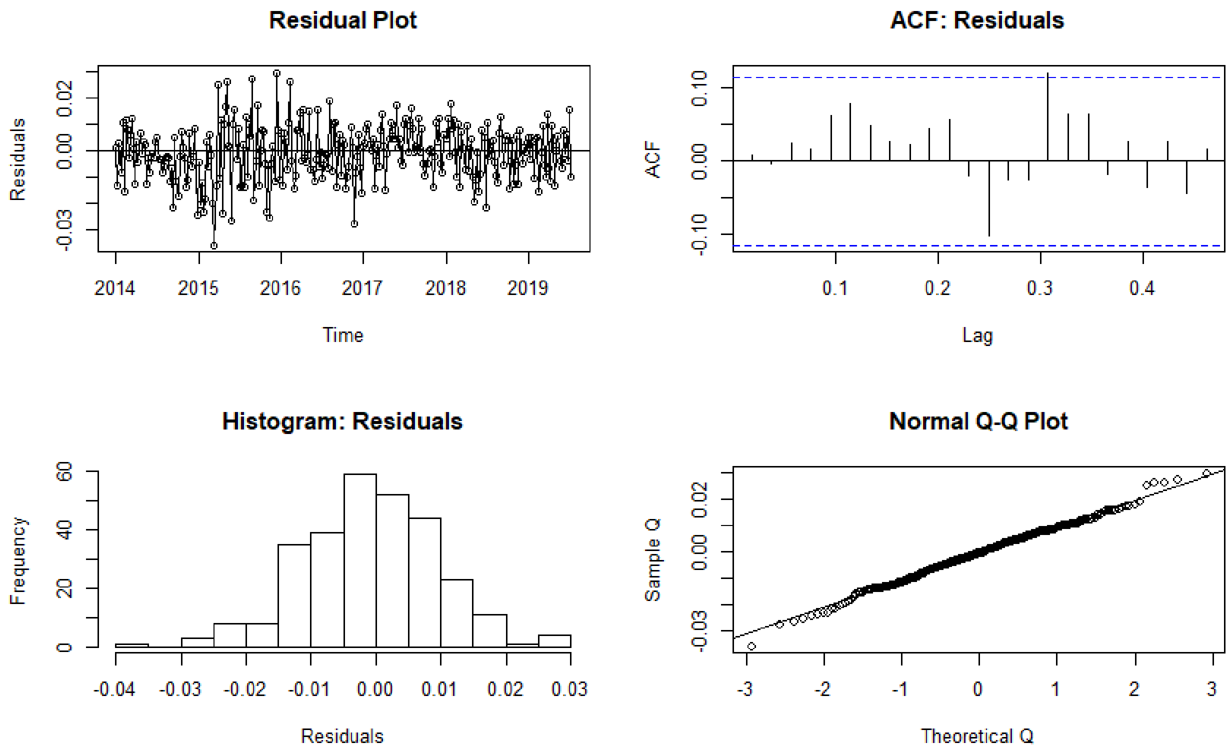

```
# > outpred$se
# Time Series:
#   Start = 643
# End = 648
# Frequency = 1
# [1] 3.826478 3.844869 3.855123 3.870862 3.885505 3.895783

for (i in 1:6){
  cat("The ", i, "th prediction's confidence interval is (" , lbound[i], ","
, ubound[i], ") \n")
}

# The  1 th prediction's confidence interval is ( 3.896225 , 18.89602 )
# The  2 th prediction's confidence interval is ( 6.301965 , 21.37385 )
# The  3 th prediction's confidence interval is ( 6.926165 , 22.03825 )
# The  4 th prediction's confidence interval is ( 4.752241 , 19.92602 )
# The  5 th prediction's confidence interval is ( 6.563145 , 21.79432 )
# The  6 th prediction's confidence interval is ( 6.552773 , 21.82424 )
```

```r
## Compute Accuracy Measures
obsprice = chart[(nfit+1):n]
predprice = outpred$pred

### Mean Squared Prediction Error (MSPE)
mean((predprice-obsprice)^2)
# 51.93373

### Mean Absolute Prediction Error (MAE)
mean(abs(predprice-obsprice))
# 6.165023

### Mean Absolute Percentage Error (MAPE)
mean(abs(predprice-obsprice)/obsprice)
# 0.4927662

### Precision Measure (PM)
sum((predprice-obsprice)^2)/sum((obsprice-mean(obsprice))^2)
# 1.217197

# The 4 accuracy measures: MSPE & MAE & MAPE & PM were not that low, so we
can say that the prediction was not the best using this model.
```

```r
# Question 2a

#Libraries and Data

library(TSA)
library(mgcv)

#USD to EU

fname <- file.choose("C://Users//nicholas.park//Downloads//USD_to_EU")
data1 <- read.csv(fname)
data1 <- data1[,2]
EU = ts(data1,start=c(2014),freq=52)


#USD to GBP

fname2 <- file.choose("C://Users//nicholas.park//Downloads//USD_to_GBP")
data2 <- read.csv(fname2)
data2 <- data2[,2]
GBP = ts(data2,start=c(2014),freq=52)


# Question 2a


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 0:2){
    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 -569.420853736522"
# [1] "0 0 1 -917.969448175781"
# [1] "0 0 2 -1172.65879202353"
# [1] "0 0 3 -1355.38383000796"
# [1] "0 1 0 -1784.85289049491"
```

```
# [1] "0 1 1 -1788.70792245714"
# [1] "0 1 2 -1786.81923283967"
# [1] "0 1 3 -1786.27635631268"
# [1] "0 2 0 -1622.22817773909"
# [1] "0 2 1 -1774.06048056079"
# [1] "0 2 2 -1776.50757790129"
# [1] "0 2 3 -1774.92343233889"
# [1] "1 0 0 -1782.87765210213"
# [1] "1 0 1 -1786.92093245319"
# [1] "1 0 3 -1787.57530512863"
# [1] "1 1 0 -1788.22872828377"
# [1] "1 1 1 -1786.76302109289"
# [1] "1 1 2 -1784.94794213084"
# [1] "1 1 3 -1784.58680033341"
# [1] "1 2 0 -1669.12059630621"
# [1] "1 2 1 -1775.9706223139"
# [1] "1 2 2 -1774.67378352835"
# [1] "1 2 3 -1773.2960304864"
# [1] "2 0 0 -1786.46590069187"
# [1] "2 0 1 -1784.96352201317"
# [1] "2 0 3 -1778.6280106619"
# [1] "2 1 0 -1787.24390722201"
# [1] "2 1 1 -1785.32555803117"
# [1] "2 1 2 -1798.00812015844"
# [1] "2 1 3 -1796.10009662035"
# [1] "2 2 0 -1694.18164066097"
# [1] "2 2 1 -1775.66283339909"
# [1] "2 2 2 -1773.98221697775"
# [1] "2 2 3 -1786.40920002392"
# [1] "3 0 0 -1572.09389517767"
# [1] "3 0 1 -1782.57505604075"
# [1] "3 0 2 -1787.57246385201"
# [1] "3 0 3 -1783.9558167744"
# [1] "3 1 0 -1785.65739135954"
# [1] "3 1 1 -1784.37651976233"
# [1] "3 1 2 -1796.0987647007"
# [1] "3 1 3 -1796.76079749679"
# [1] "3 2 0 -1727.84285678383"
# [1] "3 2 1 -1774.92023437415"
# [1] "3 2 2 -1773.5370075969"
# [1] "3 2 3 -1784.41245565212"

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

# p d q       AIC
# 17 1 1 0 -1788.229
# 7  0 1 1 -1788.708
# 42 3 1 2 -1796.099
# 31 2 1 3 -1796.100
# 43 3 1 3 -1796.761
# 30 2 1 2 -1798.008

# We can observe that the minimum AIC occurs in (p,d,q) = (2,1,2)


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

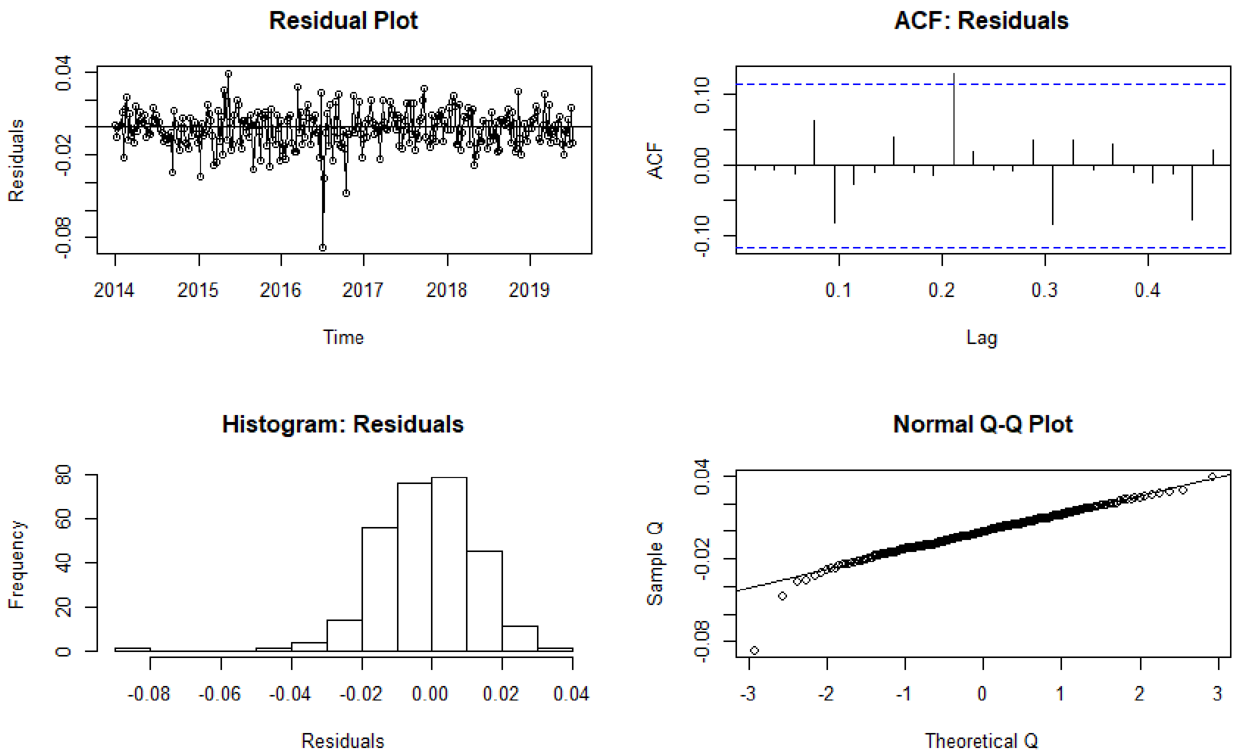
```
# Call:
#   arima(x = EU, order = c(2, 1, 2), method = "ML")
#
# Coefficients:
#     ar1      ar2     ma1     ma2
# -0.4504  -0.8491  0.5958  0.9286
# s.e.   0.0798   0.0582  0.0614  0.0573
#
# sigma^2 estimated as 0.0001072:  log likelihood = 904,  aic = -1800.01

par(mfrow=c(2,2))
plot(resid(EU_final_model), ylab='Residuals',type='o',main="Residual Plot")
abline(h=0)
acf(resid(EU_final_model),main="ACF: Residuals")
hist(resid(EU_final_model),xlab='Residuals',main='Histogram: Residuals')
qqnorm(resid(EU_final_model),ylab="Sample Q",xlab="Theoretical Q")
qqline(resid(EU_final_model))
```




```
# Residual plot does not seem to have a pattern. Heteroscedasticity is minor.
# Variance is somewhat constant. ACF is not significant beyond the confidence
bands.
# Residual plot appears normal from histogram.
# QQ Norm seems to indicate a symmetric normal distribution.

# Same procedure with GBP.


test_modelB <- function(p,d,q){
```

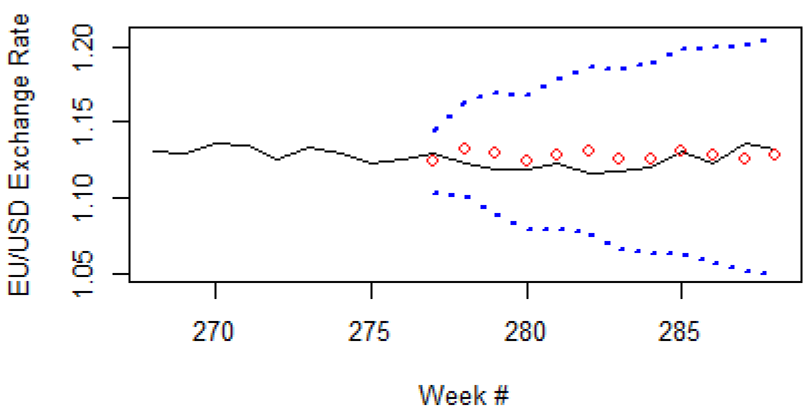
```r
  mod = arima(GBP, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  print(paste(p,d,q,current.aic,sep=" "))
  return(df)
}

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


for (p in 0:3){
  for (d in 0:2){
    for (q in 0:3) {
      possibleError <- tryCatch(
        orders<-rbind(orders,test_modelB(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 0 3 -1111.94375778033"
# [1] "0 1 0 -1619.85724246706"
# [1] "0 1 1 -1631.80095196078"
# [1] "0 1 2 -1631.01849308131"
# [1] "0 1 3 -1630.53287452623"
# [1] "0 2 0 -1476.94141415387"
# [1] "0 2 1 -1608.08134363572"
# [1] "0 2 2 -1619.63606508976"
# [1] "0 2 3 -1618.99324218288"
# [1] "1 0 0 -1616.91059168528"
# [1] "1 0 1 -1629.12875099032"
# [1] "1 0 2 -1628.21555964443"
# [1] "1 0 3 -1625.63262130587"
# [1] "1 1 0 -1628.84298539492"
# [1] "1 1 1 -1630.53419482776"
# [1] "1 1 2 -1629.50656416397"
# [1] "1 1 3 -1630.3381509502"
# [1] "1 2 0 -1506.16260565096"
# [1] "1 2 1 -1616.6637017359"
# [1] "1 2 2 -1618.4320937032"
# [1] "1 2 3 -1617.6129365503"
# [1] "2 0 0 -1626.18347599749"
# [1] "2 0 1 -1627.79422165298"
# [1] "2 0 2 -1630.63340960347"
# [1] "2 0 3 -1627.18007540341"
# [1] "2 1 0 -1632.05834292573"
# [1] "2 1 1 -1630.10642864094"
# [1] "2 1 2 -1629.9632069891"
# [1] "2 1 3 -1628.65142261479"
# [1] "2 2 0 -1542.94155479698"
```

```
# [1] "2 2 1 -1620.15552253138"
# [1] "2 2 2 -1618.18511908651"
# [1] "2 2 3 -1617.95135948199"
# [1] "3 0 0 -1512.31858802444"
# [1] "3 0 1 -1627.07251843573"
# [1] "3 0 3 -1605.79059387389"
# [1] "3 1 0 -1630.14796794164"
# [1] "3 1 1 -1628.22885253255"
# [1] "3 1 2 -1629.68453359146"
# [1] "3 1 3 -1628.14650991635"
# [1] "3 2 0 -1565.58861769817"
# [1] "3 2 1 -1618.20782422363"
# [1] "3 2 2 -1616.78165254184"
# [1] "3 2 3 -1616.67399964506"

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

# p d q       AIC
# 9  0 1 3 -1630.533
# 19 1 1 1 -1630.534
# 28 2 0 2 -1630.633
# 8  0 1 2 -1631.018
# 7  0 1 1 -1631.801
# 30 2 1 0 -1632.058

# We can observe that the minimum AIC occurs in (p,d,q) = (2,1,0)

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


par(mfrow=c(2,2))
plot(resid(GBP_final_model), ylab='Residuals',type='o',main="Residual Plot")
abline(h=0)
acf(resid(GBP_final_model),main="ACF: Residuals")
hist(resid(GBP_final_model),xlab='Residuals',main='Histogram: Residuals')
qqnorm(resid(GBP_final_model),ylab="Sample Q",xlab="Theoretical Q")
qqline(resid(GBP_final_model))
```



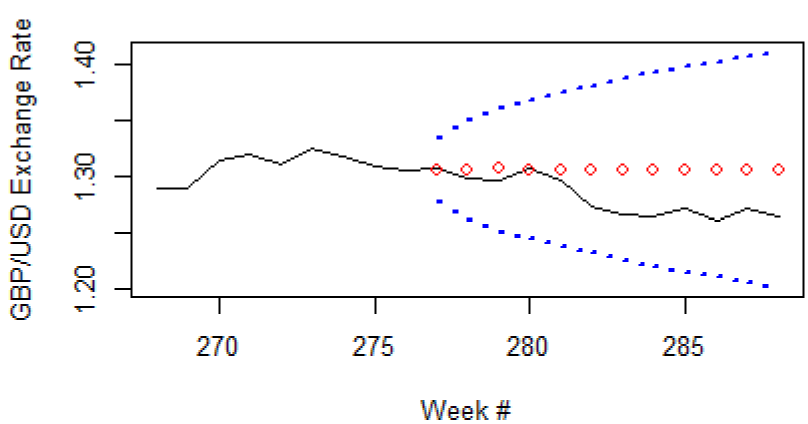

```
# Residual plot does not seem to have a pattern.
# Variance is mostly constant. ACF is not significant beyond the confidence
bands.
# Residual plot appears approximately normal from histogram.
# QQ Norm seems to indicate a symmetric normal distribution, except for a
slight left tail.



# Question 2b


EU_final_model
# Call:
#   arima(x = EU, order = c(2, 1, 2), method = "ML")
#
# Coefficients:
#   ar1       ar2      ma1      ma2
# -0.4504  -0.8491  0.5958  0.9286
# s.e.   0.0798   0.0582  0.0614  0.0573
#
# sigma^2 estimated as 0.0001072:  log likelihood = 904,  aic = -1800.01



GBP_final_model
# Call:
#   arima(x = GBP, order = c(2, 1, 0), method = "ML")
#
# Coefficients:
```

```
#   ar1      ar2
# 0.2199  -0.1340
# s.e.  0.0585   0.0584
#
# sigma^2 estimated as 0.0001944:  log likelihood = 819.03,  aic = -1634.06

eu.time.pts = c(1:length(EU))
n = length(EU)
nfit = n-12
EU_model = arima(EU[1:nfit], order = c(2,1,2),method = "ML")
EU_outpred = predict(EU_model,n.ahead=12)
eu_ubound = EU_outpred$pred+1.96*EU_outpred$se
eu_lbound = EU_outpred$pred-1.96*EU_outpred$se
ymin = min(eu_lbound)
ymax = max(eu_ubound)
plot(eu.time.pts[(n-20):n],EU[(n-20):n],type="l", ylim=c(ymin,ymax),
xlab="Week #", ylab= "EU/USD Exchange Rate")
points(eu.time.pts[(nfit+1):n],EU_outpred$pred,col="red")
lines(eu.time.pts[(nfit+1):n],eu_ubound,lty=3,lwd= 2, col="blue")
lines(eu.time.pts[(nfit+1):n],eu_lbound,lty=3,lwd= 2, col="blue")
```




```
for (i in 1:12){
  cat("The ", i, "th prediction's confidence interval is (" , eu_lbound[i],
"," , eu_ubound[i], ") \n")
}
# The  1 th prediction's confidence interval is ( 1.103374 , 1.144223 )
# The  2 th prediction's confidence interval is ( 1.100951 , 1.163255 )
# The  3 th prediction's confidence interval is ( 1.08962 , 1.168672 )
# The  4 th prediction's confidence interval is ( 1.078527 , 1.168353 )
# The  5 th prediction's confidence interval is ( 1.07854 , 1.178575 )
# The  6 th prediction's confidence interval is ( 1.075484 , 1.186656 )
# The  7 th prediction's confidence interval is ( 1.065552 , 1.185596 )
# The  8 th prediction's confidence interval is ( 1.062199 , 1.189697 )
# The  9 th prediction's confidence interval is ( 1.062548 , 1.198346 )
# The  10 th prediction's confidence interval is ( 1.056225 , 1.199927 )
# The  11 th prediction's confidence interval is ( 1.050241 , 1.20043 )
# The  12 th prediction's confidence interval is ( 1.050213 , 1.20699 )
```

```r
## Compute Accuracy Measures
EU_obsprice = EU[(nfit+1):n]
EU_predprice = EU_outpred$pred

### Mean Squared Prediction Error (MSPE)
mean((EU_predprice-EU_obsprice)^2)
# 6.395014e-05

### Mean Absolute Prediction Error (MAE)
mean(abs(EU_predprice-EU_obsprice))
# 0.007161764

### Mean Absolute Percentage Error (MAPE)
mean(abs(EU_predprice-EU_obsprice)/EU_obsprice)
# 0.006380015

### Precision Measure (PM)
sum((EU_predprice-EU_obsprice)^2)/sum((EU_obsprice-mean(EU_obsprice))^2)
# 1.616309

# The 4 accuracy measures: MSPE & MAE & MAPE & PM were very low. Prediction
was quite accurate.

# Same procedure for GBP

gbp.time.pts = c(1:length(GBP))
n = length(GBP)
nfit = n-12
GBP_model = arima(GBP[1:nfit], order = c(2,1,0),method = "ML")
GBP_outpred = predict(GBP_model,n.ahead=12)
gbp_ubound = GBP_outpred$pred+1.96*GBP_outpred$se
gbp_lbound = GBP_outpred$pred-1.96*GBP_outpred$se
ymin = min(gbp_lbound)
ymax = max(gbp_ubound)
plot(gbp.time.pts[(n-20):n],GBP[(n-20):n],type="l", ylim=c(ymin,ymax),
xlab="Week #", ylab= "GBP/USD Exchange Rate")
points(gbp.time.pts[(nfit+1):n],GBP_outpred$pred,col="red")
lines(gbp.time.pts[(nfit+1):n],gbp_ubound,lty=3,lwd= 2, col="blue")
lines(gbp.time.pts[(nfit+1):n],gbp_lbound,lty=3,lwd= 2, col="blue")
```



```r
for (i in 1:12){
  cat("The ", i, "th prediction's confidence interval is (" , gbp_lbound[i],
"," , gbp_ubound[i], ") \n")
}

# The  1 th prediction's confidence interval is ( 1.278195 , 1.333416 )
# The  2 th prediction's confidence interval is ( 1.262772 , 1.350049 )
# The  3 th prediction's confidence interval is ( 1.252742 , 1.360283 )
# The  4 th prediction's confidence interval is ( 1.244859 , 1.368047 )
# The  5 th prediction's confidence interval is ( 1.237889 , 1.374962 )
# The  6 th prediction's confidence interval is ( 1.231512 , 1.381343 )
# The  7 th prediction's confidence interval is ( 1.225625 , 1.387239 )
# The  8 th prediction's confidence interval is ( 1.220141 , 1.392724 )
# The  9 th prediction's confidence interval is ( 1.214988 , 1.397876 )
# The  10 th prediction's confidence interval is ( 1.21011 , 1.402754 )
# The  11 th prediction's confidence interval is ( 1.205467 , 1.407396 )
# The  12 th prediction's confidence interval is ( 1.201029 , 1.411835 )


## Compute Accuracy Measures
GBP_obsprice = GBP[(nfit+1):n]
GBP_predprice = GBP_outpred$pred

### Mean Squared Prediction Error (MSPE)
mean((GBP_predprice-GBP_obsprice)^2)
# 0.0009270949

### Mean Absolute Prediction Error (MAE)
mean(abs(GBP_predprice-GBP_obsprice))
# 0.02527872

### Mean Absolute Percentage Error (MAPE)
mean(abs(GBP_predprice-GBP_obsprice)/GBP_obsprice)
# 0.01990941

### Precision Measure (PM)
sum((GBP_predprice-GBP_obsprice)^2)/sum((GBP_obsprice-mean(GBP_obsprice))^2)
# 3.133436

# The 4 accuracy measures: MSPE & MAE & MAPE & PM are quite low, but not as
low as prediction done by EU_model.
# Prediction was still quite accurate but less accurate toward later points.
```

```
# Question 3

# We can see that ARIMA modeling sometimes performs well with AIC model
selection criteria, sometimes not so much. Overall the effectiveness of it is
better than just differencing or using AR or MA models.
# For ARIMA to work well, there must be a time series that contains a
significant number of datapoints to forecast values using its data’s inertia.
It would be appropriate to use for contexts with a lot of datapoints such as
a 20+ year old stock’s price. I would recommend against it for newer stock’s
with less datapoints.
# It appears models with the most amount of coefficients do not necessarily
perform better (i.e., models with p & Q values close to the max value).
Perhaps it has to do with the possibility of overfitting the data that can
turn out to predict rather poorly on test data. The EU model which uses
ARIMA(2,1,2) performs slightly better than GBP model which uses ARIMA(2,1,0),
by taking into account MA coefficients which happened due to model selection
via low resulting AIC. This makes me think the context and structure of the
data is also important in determining the criteria for model selection +
interpretation. Currently, the model selection simply relies on searching for
low values of AIC on different combinations of (p,d,q), but there are better
models that can better select the model to increase performance even more. I
could've further taken into consideration seasonal ARIMA instead of normal
ARIMA. I also want to note that the residual analysis after ARIMA was not
always a perfect white noise, due to non-ideal aspects of real-world data.
```