Non-Technical Summary

The data is the daily candlestick information of the price data on the bid side for the EUR/USD from the dates January 1st 2004 to December 31st 2013. Besides the candlestick data it also includes the given time for the candlestick and the volume. This data was provided by Dukascopy.

Definitions:

-Daily: The time frame that was used to produce a candlestick
-Candlestick: The open, high, low and close for a given time frame
-Bid: The price that the buyers are willing to pay for a security
-EUR/USD: The security of the exchange from Euro to US Dollar
-Volume: The total number of units bought and sold in a given time frame
- Dukascopy: Swiss bank which acts as one of the largest brokerage firms

With this data a prediction model was built using times series models. This models use just the previous prices to predict future models. Only the price values for the high,low and close where investigated for the models. The close data was not suitable for the time series models. However, the data on the high and the low were suitable to look into model building. Models were indeed found. Furthermore, their residuals were satisfactory for a time series models. Residuals are the difference between the predictions and the actual values. It is interesting to note that the two models for both the high and low are very similar. However, this similarity is acceptable since both models use different inputs and therefore will also should have different outputs. Thus, the models themselves may prove to be useful.

The main value of a prediction model for the EUR/USD exchange would be to see if a profitable trading system can be developed from it. Thus, with this research done further research is recommend to see if any profitable trading system can be developed to make these models truly useful.

Technical Summary

1. Data Background Information

The data is the daily candlestick information of the price data on the bid side for the EUR/USD from the dates January 1st 2004 to December 31st 2013. Besides the candlestick data it also includes the given time for the candlestick and the volume. This data was provided by Dukascopy.

Definitions:

-Daily: The time frame that was used to produce a candlestick
-Candlestick: The open, high, low and close for a given time frame
-Bid: The price that the buyers are willing to pay for a security
-EUR/USD: The security of the exchange from Euro to US Dollar
-Volume: The total number of units bought and sold in a given time frame
- Dukascopy: Swiss bank which acts as one of the largest brokerage firms

2. Exploratory Analysis of the Data

Since only the high, low and close of the price will be any real interest to traders this report will focus on creating a time series model for each of them individually. Furthermore, days where the market was close and there is no volume where removed from the dataset. Lastly, all the data it was preprocessed by the difference of the previous value. Before differencing the Dickey-Fuller Test went was accepting the null hypothesis and the Kwiatkowski-Phillips-Schmidt-Shin(KPSS) test rejecting the null hypothesis. While after differencing the Dickey-Fuller Test went was rejecting the null hypothesis and the Kwiatkowski-Phillips-Schmidt-Shin(KPSS) test accepting the null hypothesis. Thus, differencing was required since the data needs to be stationary for time series models to be suitable.

2_Close

The Close data appears to be normally distributed by looking at both the histogram and the normal Q-Q plot(B1,B2). This is further supported by the Jarque – Bera Normalality Test with a p-value of less than 2.2e-16 (A4).

Moving forward, the ACF and PACF graph seem to show that the data is not autocorrelated since no values seem to be significant (B3, B4). This is further supported by the Box-Ljung test p-value of .6134 (A5). This high p-value indicates that the test did not detect autocorrelation in the time series. Thus, no time series model of autoregressive model(AR) , a moving average model(MA), or their combination as an autoregressive moving average model (ARMA) is suitable. Therefore, the close values were no longer investigated for model building. Thus, research for this data stop at this point.

2_High

The High data appears to be normally distributed by looking at both the histogram and the normal Q-Q plot(B5,B6). This is further confirmed by the Jarque – Bera Normalality Test with a p-value of less than 2.2e-16 (A6).

Moving forward, the ACF and PACF graph seem to show that the data is autocorrelated since many values seem to be significant (B7, B8). This is further supported by the Box-Ljung test p-value of less than 2.2e-16 (A7). Thus, autocorrelation seems to be present and a model should try to be developed from an autoregressive model(AR) , a moving average model(MA), or their combination as an autoregressive moving average model (ARMA) or ARIMA with seasonality added SARIMA .

2_Low

The High data appears to be normally distributed by looking at both the histogram and the normal Q-Q plot(B9,B10). This is further confirmed by the Jarque – Bera Normalality Test with a p-value of less than 2.2e-16 (A8).

Moving forward, the ACF and PACF graph seem to show that the data is autocorrelated since many values seem to be significant (B11, B12). This is further supported by the Box-Ljung test p-value of less than 2.2e-16 (A9). Thus, autocorrelation seems to be present and a model should try to be developed from an autoregressive model(AR) , a moving average model(MA), or their combination as an autoregressive moving average model (ARMA) or ARIMA with seasonality added SARIMA .

3. Model Fitting

3_High

The model selected is a SARIMA(0,0,1)(1,0,1)[6] (see A10). This gives an expression of the fitted model for the data as:
$(1-.1727B^6)=(1+.9812B^6)(1-.9988B^6)a_t$

3_Low
The model selected is a SARIMA(0,0,1)(1,0,1)[6] (see A11). This gives an expression of the fitted model for the data as:
$(1-.1930B^6)=(1+.985B^6)(1-.9993B^6)a_t$

4. Residual Analysis and Model Diagnostics

Test to see if the models are acceptable.

4_High
Looking at the models coefficients they all test to be significant with p-values of less than .001 (A12). Looking at the distribution of the residuals they seem to be normal from the QQ-plot(B15). This is further supported by the Jarque – Bera Normality Test with a p-value less than 2.2e-16 . Thus, the residuals appear to be white noise. Furthermore, there seems to be no autocorrelation in the residuals looking at the ACF (B14) and a plot of the residuals themselves (B13). This is further supported by the Box-Ljung test greater than .01 with a value of . 04106 . So there might be a little autocorrelation but it is not very strong relation.

4_Low
Looking at the models coefficients they all test to be significant with p-values of less than .001 (A15). Looking at the distribution of the residuals they seem to be normal from the QQ-plot(B18). This is further supported by the Jarque – Bera Normality Test with a p-value less than 2.2e-16 . Thus, the residuals appear to be white noise. Furthermore, there seems to be no autocorrelation in the residuals looking at the ACF (B17) and a plot of the residuals themselves (B16). This is further supported by the Box-Ljung test greater than .01 with a value of . 2442 .

5. Forecast Analysis

5_High

The model produced the following forecast for a prediction of 6 days ahead.

```
> f_High=forecast.Arima(m1,h=6)
> f_High
      Point Forecast        Lo 80       Hi 80       Lo 95      Hi 95
3131   5.724390e-04 -0.008343974 0.009488851 -0.01306404 0.01420892
3132  -1.036474e-06 -0.009049398 0.009047325 -0.01383931 0.01383724
3133  -6.453665e-05 -0.009112899 0.008983825 -0.01390281 0.01377374
3134  -3.648209e-03 -0.012696571 0.005400153 -0.01748648 0.01019007
3135   2.735481e-03 -0.006312881 0.011783843 -0.01110279 0.01657376
3136   8.858542e-04 -0.008162508 0.009934216 -0.01295242 0.01472413
```

See B19 for graph. The red line is the 95% Hi and the blue line is the Lo 95 with the black line being the mean.

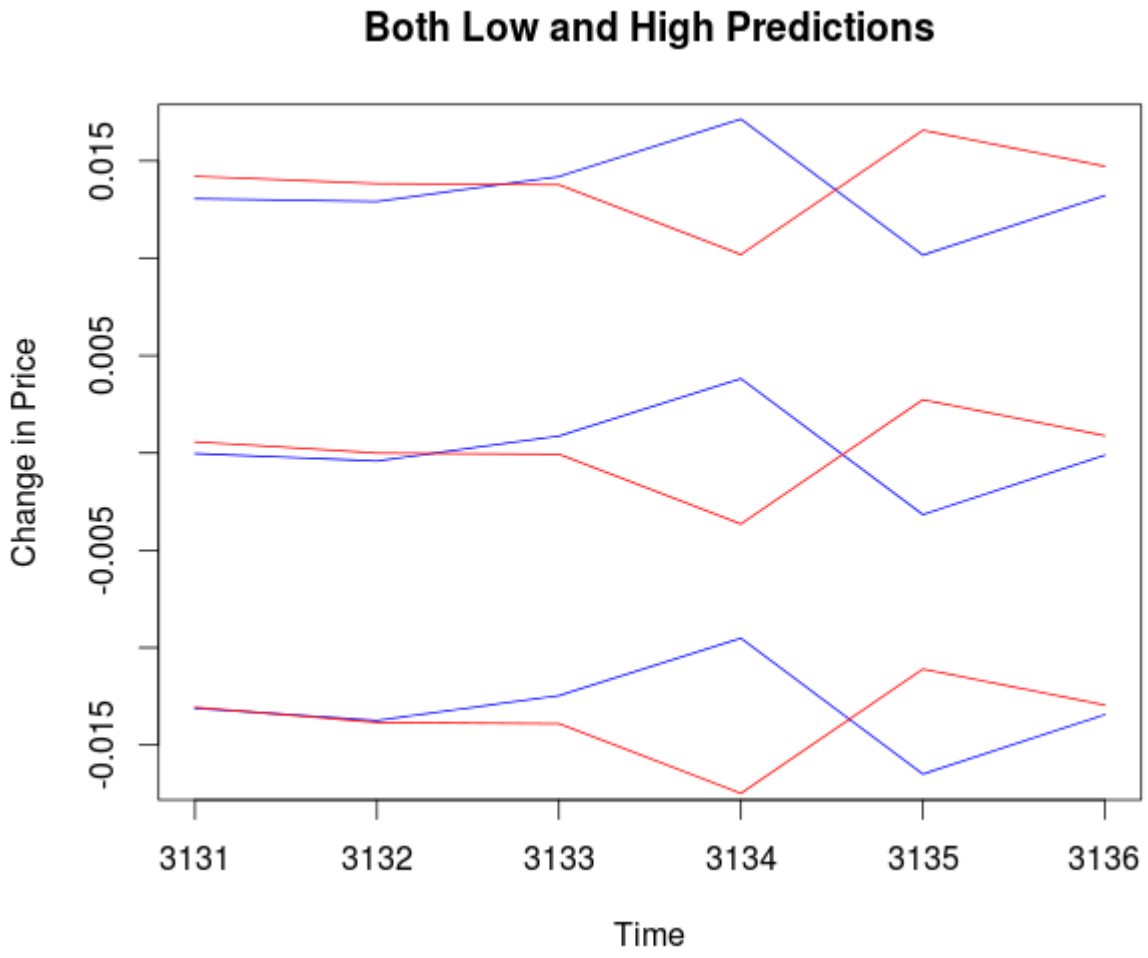
5_Low

The model produced the following forecast for a prediction of 6 days ahead.

```
> f_Low=forecast.Arima(m1,h=6)
> f_Low
      Point Forecast        Lo 80       Hi 80        Lo 95      Hi 95
3131  -2.613017e-05 -0.008584651 0.008532391 -0.013115258 0.01306300
3132  -4.118589e-04 -0.009128368 0.008304650 -0.013742610 0.01291889
3133   8.656230e-04 -0.007850886 0.009582132 -0.012465128 0.01419637
3134   3.815570e-03 -0.004900940 0.012532079 -0.009515181 0.01714632
3135  -3.166332e-03 -0.011882842 0.005550177 -0.016497083 0.01016442
3136  -1.103729e-04 -0.008826882 0.008606136 -0.013441124 0.01322038
```

See B20 for graph. The red line is the 95% Hi and the blue line is the Lo 95 with the black line being the mean.

6. Analysis of the Results and Discussion



Above is a graph of both the high and low predictions from there respected models. The red line is the high values at top 95% middle 50% bottom 95%. The blue line is the high values at top 95% middle 50% bottom 95%. It is interesting to see that on 3134 the high prediction is less than the low prediction. Further interesting is that the two models themselves are very similar. This similarity is fine since both have different inputs going into them the outputs should also be slightly different. This difference is show from the graph above. Moving forward in this project idea beyond the scope of the class. I would recommend the investigation of the model SARIMA(0,0,1)(1,0,1)[6] being trained on a moving window just predicting one value ahead. With these predictions I would look into subsets and see if any trading idea could be profitable from these predictions. Furthermore, looking into different time frames and other currency pairs may be useful.

APPENDIX A

1)
```
> adf.test(dataSet$Close)

        Augmented Dickey-Fuller Test

data:  dataSet$Close
Dickey-Fuller = -2.58, Lag order = 14, p-value = 0.3327
alternative hypothesis: stationary

> kpss.test(dataSet$Close)

        KPSS Test for Level Stationarity

data:  dataSet$Close
KPSS Level = 4.3982, Truncation lag parameter = 12, p-value = 0.01

Warning message:
In kpss.test(dataSet$Close) : p-value smaller than printed p-value
> #just look at the close data
> data <- diff(dataSet$Close, lag=1)
> adf.test(data)

        Augmented Dickey-Fuller Test

data:  data
Dickey-Fuller = -14.3384, Lag order = 14, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(data) : p-value smaller than printed p-value
> kpss.test(data)

        KPSS Test for Level Stationarity

data:  data
KPSS Level = 0.0413, Truncation lag parameter = 12, p-value = 0.1

Warning message:
In kpss.test(data) : p-value greater than printed p-value
```

2)
```
> adf.test(dataSet$High)

        Augmented Dickey-Fuller Test

data:  dataSet$High
Dickey-Fuller = -2.5568, Lag order = 14, p-value = 0.3426
alternative hypothesis: stationary

> kpss.test(dataSet$High)

        KPSS Test for Level Stationarity
```

```
data:  dataSet$High
KPSS Level = 4.482, Truncation lag parameter = 12, p-value = 0.01

Warning message:
In kpss.test(dataSet$High) : p-value smaller than printed p-value
> #just look at the High data
> data <- diff(dataSet$High, lag=1)
> adf.test(data)

         Augmented Dickey-Fuller Test

data:  data
Dickey-Fuller = -14.2194, Lag order = 14, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(data) : p-value smaller than printed p-value
> kpss.test(data)

         KPSS Test for Level Stationarity

data:  data
KPSS Level = 0.0422, Truncation lag parameter = 12, p-value = 0.1

Warning message:
In kpss.test(data) : p-value greater than printed p-value
```

3)

```
> adf.test(dataSet$Low)

         Augmented Dickey-Fuller Test

data:  dataSet$Low
Dickey-Fuller = -2.5589, Lag order = 14, p-value = 0.3417
alternative hypothesis: stationary

> kpss.test(dataSet$Low)

         KPSS Test for Level Stationarity

data:  dataSet$Low
KPSS Level = 4.3373, Truncation lag parameter = 12, p-value = 0.01

Warning message:
In kpss.test(dataSet$Low) : p-value smaller than printed p-value
> #just look at the Low data
> data <- diff(dataSet$Low, lag=1)
> adf.test(data)

         Augmented Dickey-Fuller Test

data:  data
Dickey-Fuller = -14.0977, Lag order = 14, p-value = 0.01
```

```
alternative hypothesis: stationary

Warning message:
In adf.test(data) : p-value smaller than printed p-value
> kpss.test(data)

        KPSS Test for Level Stationarity

data:  data
KPSS Level = 0.0414, Truncation lag parameter = 12, p-value = 0.1

Warning message:
In kpss.test(data) : p-value greater than printed p-value
```

4)
```
> normalTest(data,method=c("jb"))

Title:
 Jarque - Bera Normalality Test

Test Results:
  STATISTIC:
    X-squared: 665.6569
  P VALUE:
    Asymptotic p Value: < 2.2e-16
```

5)
```
> Box.test(data,lag=35,type="Ljung")

        Box-Ljung test

data:  data
X-squared = 32.0067, df = 35, p-value = 0.6134
```

6)
```
> normalTest(data,method=c("jb"))

Title:
 Jarque - Bera Normalality Test

Test Results:
  STATISTIC:
    X-squared: 1567.9312
  P VALUE:
    Asymptotic p Value: < 2.2e-16
```

7)
```
> Box.test(data,lag=35,type="Ljung")

        Box-Ljung test

data:  data
X-squared = 382.9094, df = 35, p-value < 2.2e-16
```

8)
```
> normalTest(data,method=c("jb"))

Title:
 Jarque - Bera Normalality Test

Test Results:
  STATISTIC:
    X-squared: 1567.9312
  P VALUE:
    Asymptotic p Value: < 2.2e-16
```

9)
```
> Box.test(data,lag=35,type="Ljung")

        Box-Ljung test

data:  data
X-squared = 382.9094, df = 35, p-value < 2.2e-16
```

10)
```
> m1=arima(data, order = c(0,0,1), seasonal = list(order = c(1, 0, 1), period = 6))
> m1
Series: data
ARIMA(0,0,1)(1,0,1)[6] with non-zero mean

Coefficients:
          ma1    sar1     sma1  intercept
      0.1727  0.9988  -0.9812     0.0000
s.e.  0.0182  0.0012   0.0084     0.0011

sigma^2 estimated as 4.841e-05:  log likelihood=11103.91
AIC=-22197.81   AICc=-22197.79   BIC=-22167.57
```

11)
```
> m1=arima(data, order = c(0,0,1), seasonal = list(order = c(1, 0, 1), period = 6))
> m1
Series: data
ARIMA(0,0,1)(1,0,1)[6] with non-zero mean

Coefficients:
          ma1    sar1    sma1  intercept
      0.1930  0.9993  -0.985     0.0000
s.e.  0.0172  0.0006   0.005     0.0012

sigma^2 estimated as 4.46e-05:  log likelihood=11231.28
AIC=-22452.56   AICc=-22452.55   BIC=-22422.32
```

12)
```
> coeftest(m1)

z test of coefficients:

               Estimate  Std. Error   z value Pr(>|z|)
```

```
ma1        1.7267e-01  1.8212e-02    9.4811   <2e-16 ***
sar1       9.9877e-01  1.1572e-03  863.0845   <2e-16 ***
sma1      -9.8116e-01  8.4381e-03 -116.2765   <2e-16 ***
intercept  4.4008e-05  1.1299e-03    0.0389   0.9689
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

13)
```
> normalTest(m1$residuals,method=c("jb"))

Title:
 Jarque - Bera Normalality Test

Test Results:
  STATISTIC:
    X-squared: 1779.9295
  P VALUE:
    Asymptotic p Value: < 2.2e-16
```

14)
```
> Box.test(m1$residuals,lag=35, type="Ljung",fitdf=1)

        Box-Ljung test

data:  m1$residuals
X-squared = 49.5872, df = 34, p-value = 0.04106
```

15)
```
> coeftest(m1)

z test of coefficients:

              Estimate  Std. Error   z value Pr(>|z|)
ma1        1.9303e-01  1.7216e-02   11.2121   <2e-16 ***
sar1       9.9929e-01  5.7644e-04 1733.5563   <2e-16 ***
sma1      -9.8498e-01  5.0059e-03 -196.7636   <2e-16 ***
intercept  4.7017e-05  1.2113e-03    0.0388    0.969
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

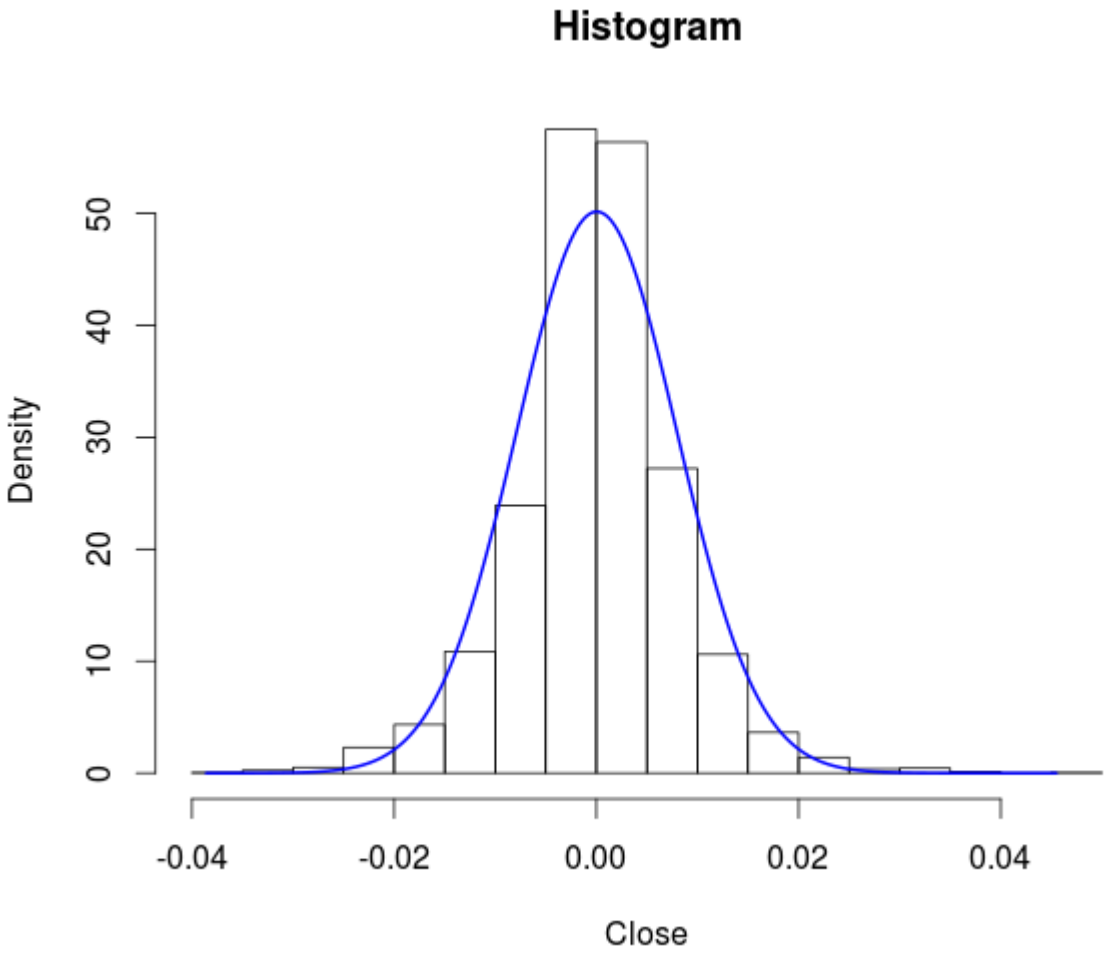
16)
```
> normalTest(m1$residuals,method=c("jb"))

Title:
 Jarque - Bera Normalality Test

Test Results:
  STATISTIC:
    X-squared: 984.6364
  P VALUE:
    Asymptotic p Value: < 2.2e-16
```

17)
```
> Box.test(m1$residuals,lag=35, type="Ljung",fitdf=1)

        Box-Ljung test

data:  m1$residuals
```

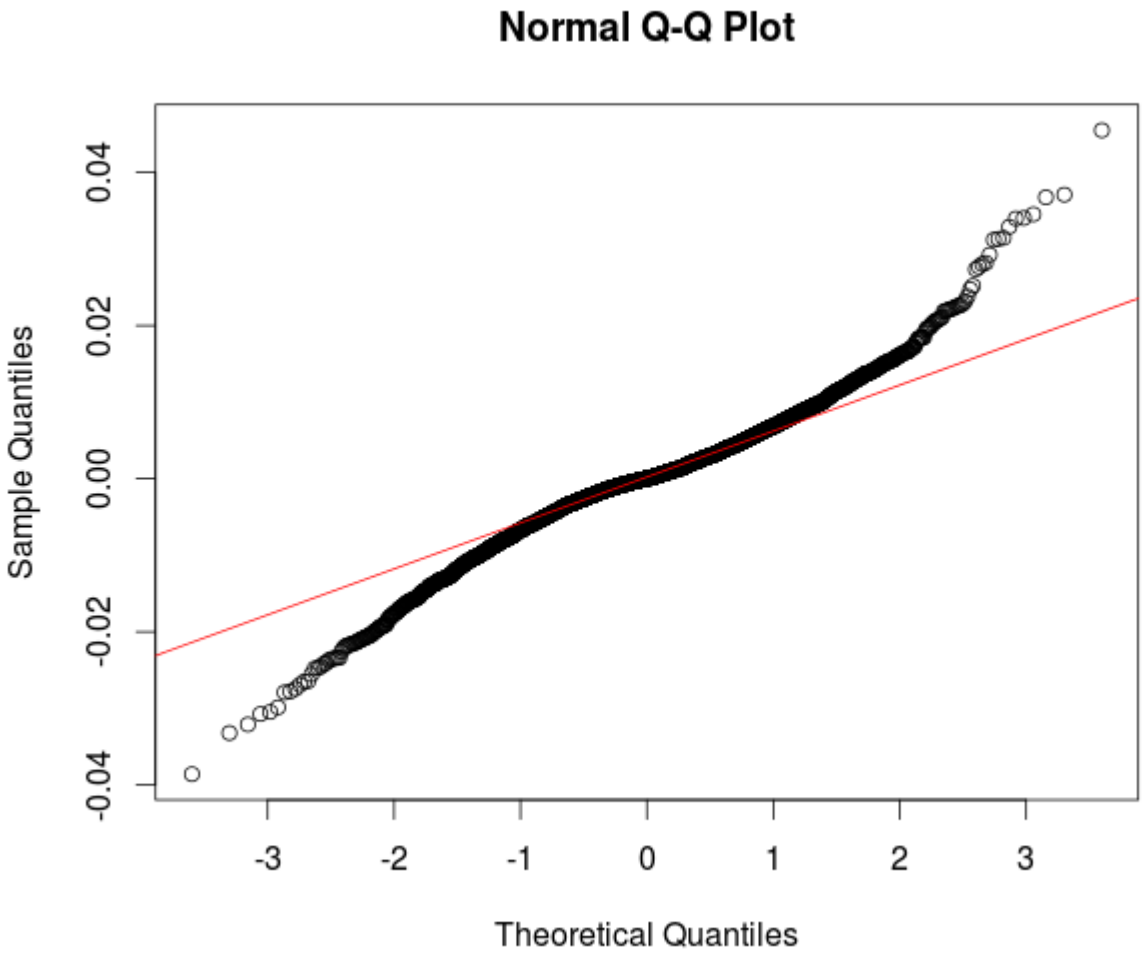
```
X-squared = 39.3067, df = 34, p-value = 0.2442
```

APPENDIX B

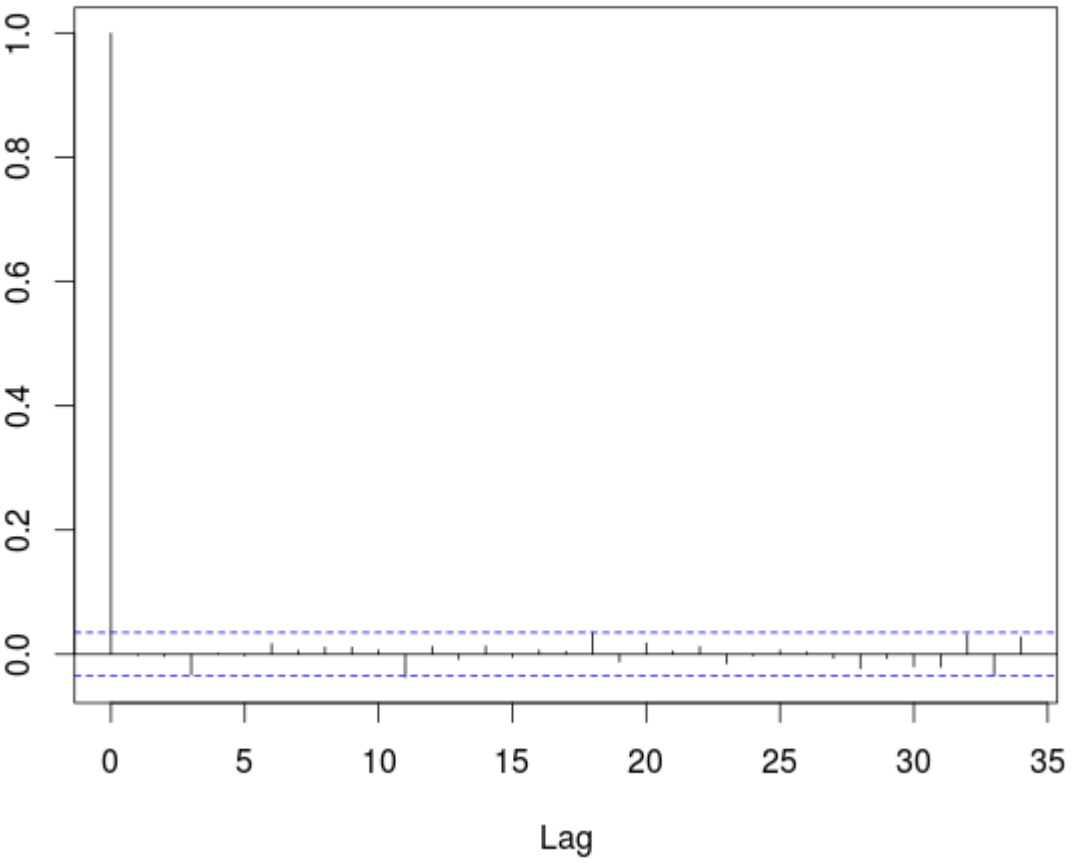
1)



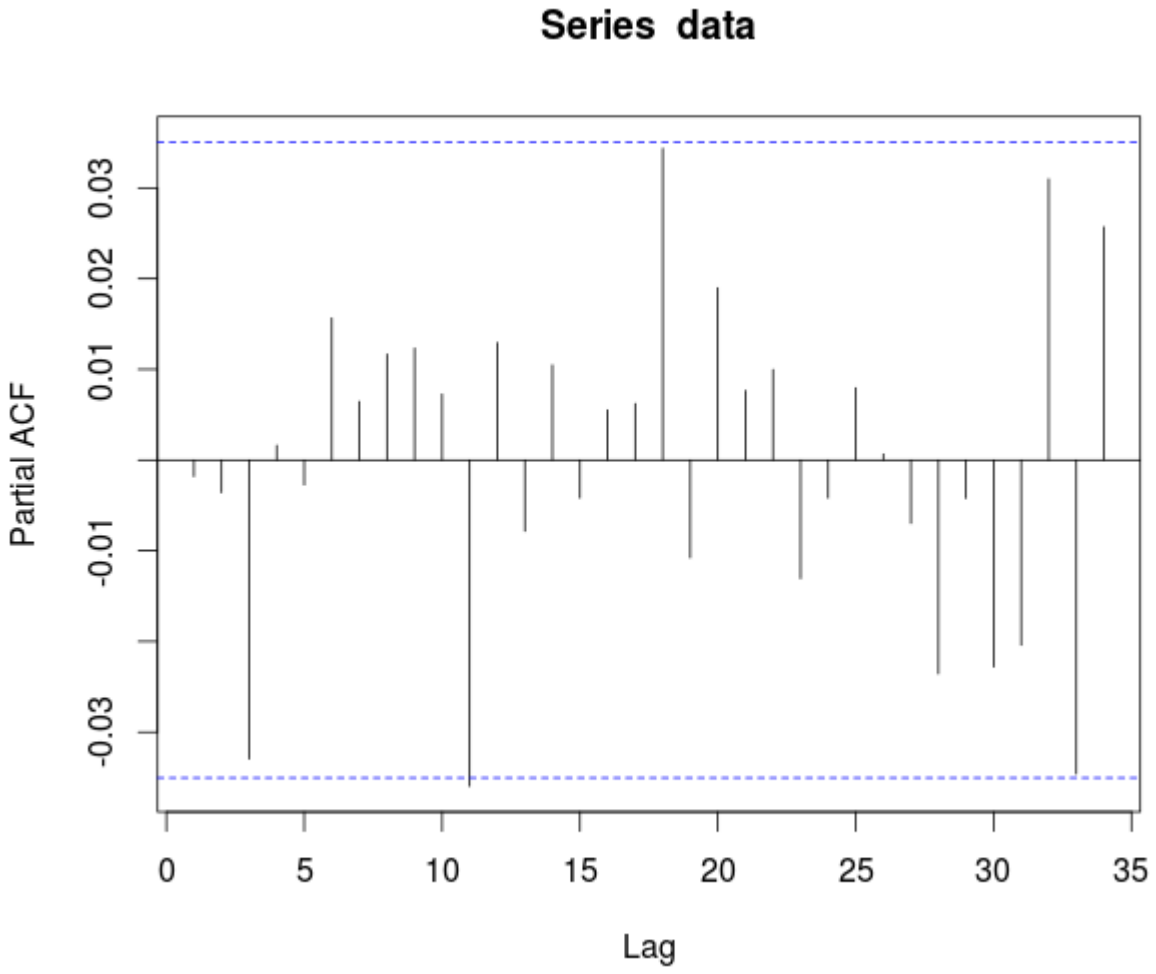
2)


Normal Q-Q Plot
Sample Quantiles
0.04
0.02
0.00
-0.02
-0.04
-3
-2
-1
0
1
2
3
Theoretical Quantiles

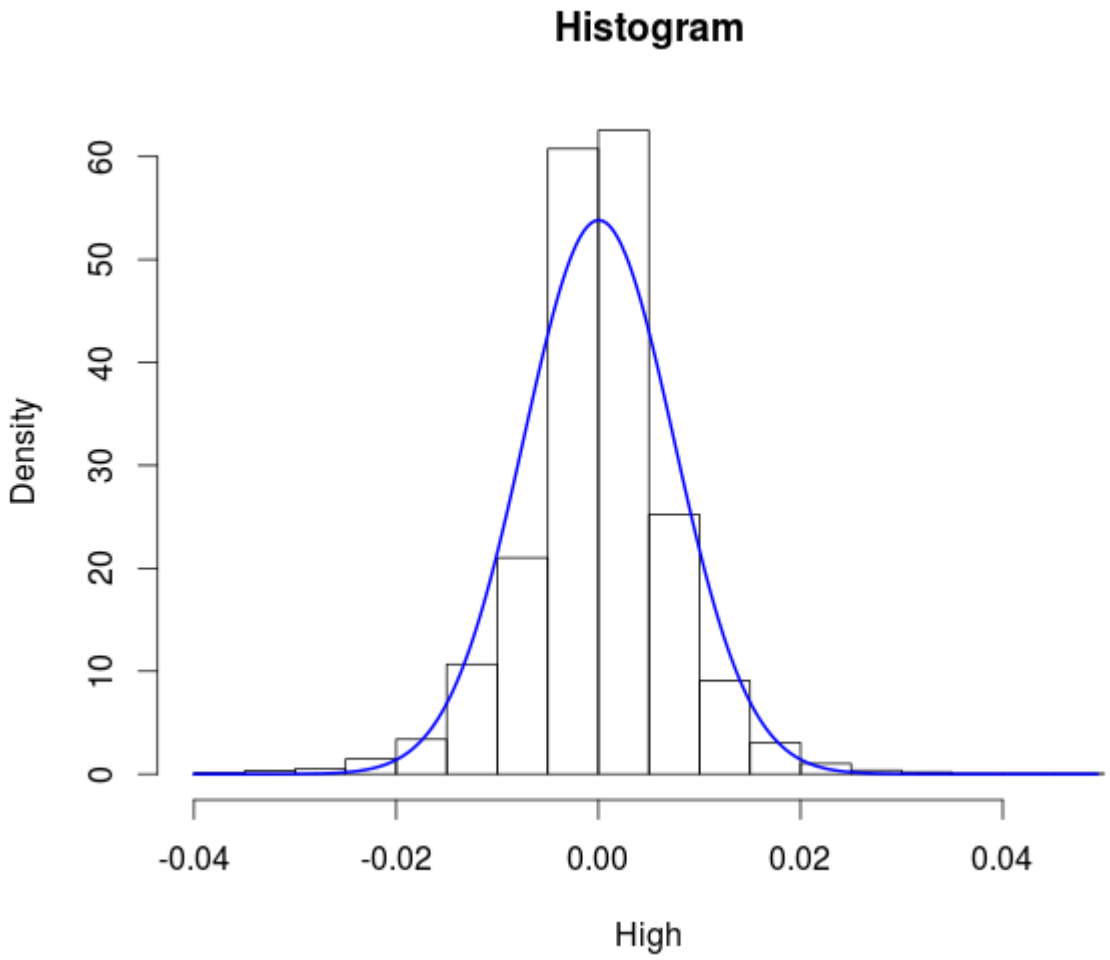
3)



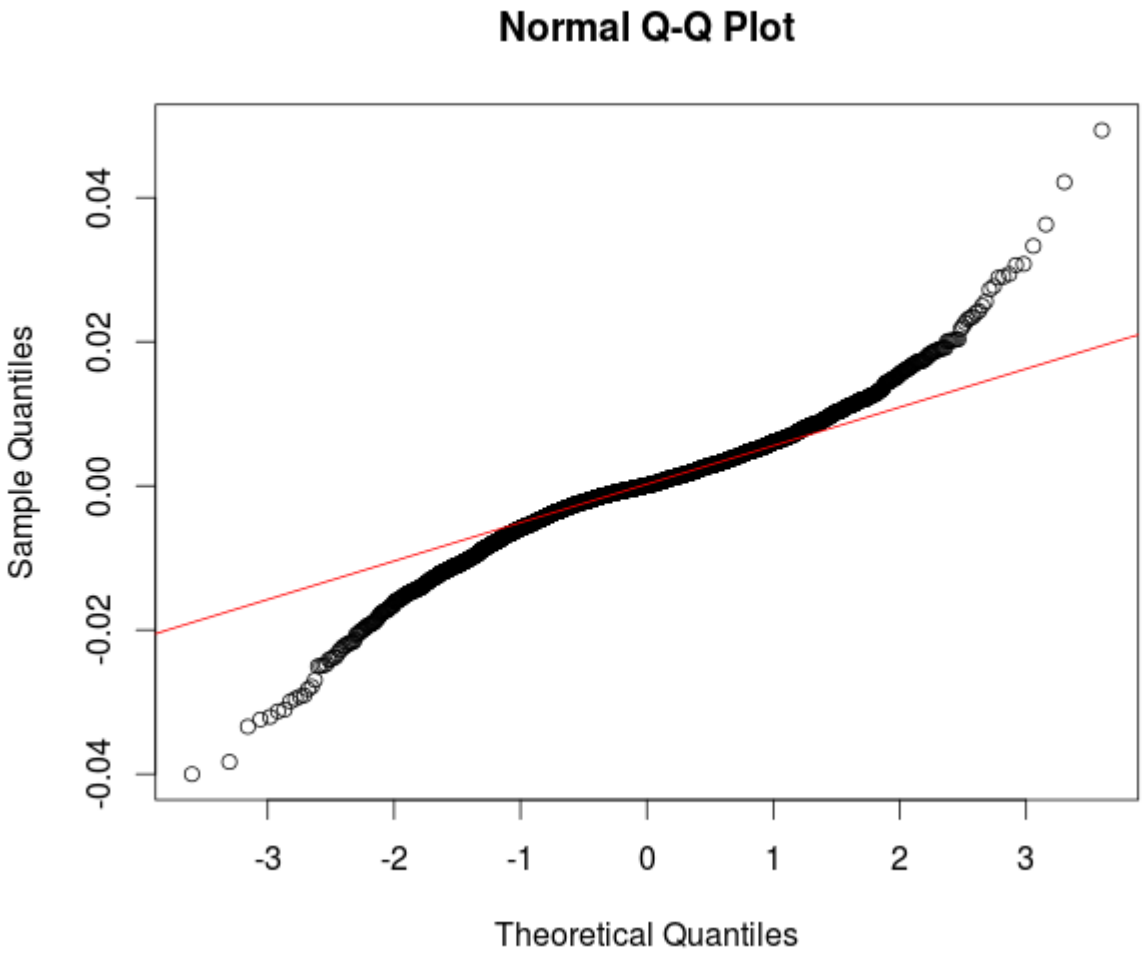
4)


Series data
Partial ACF
0.03
0.02
0.01
-0.01
-0.03
0
5
10
15
20
25
30
35
Lag

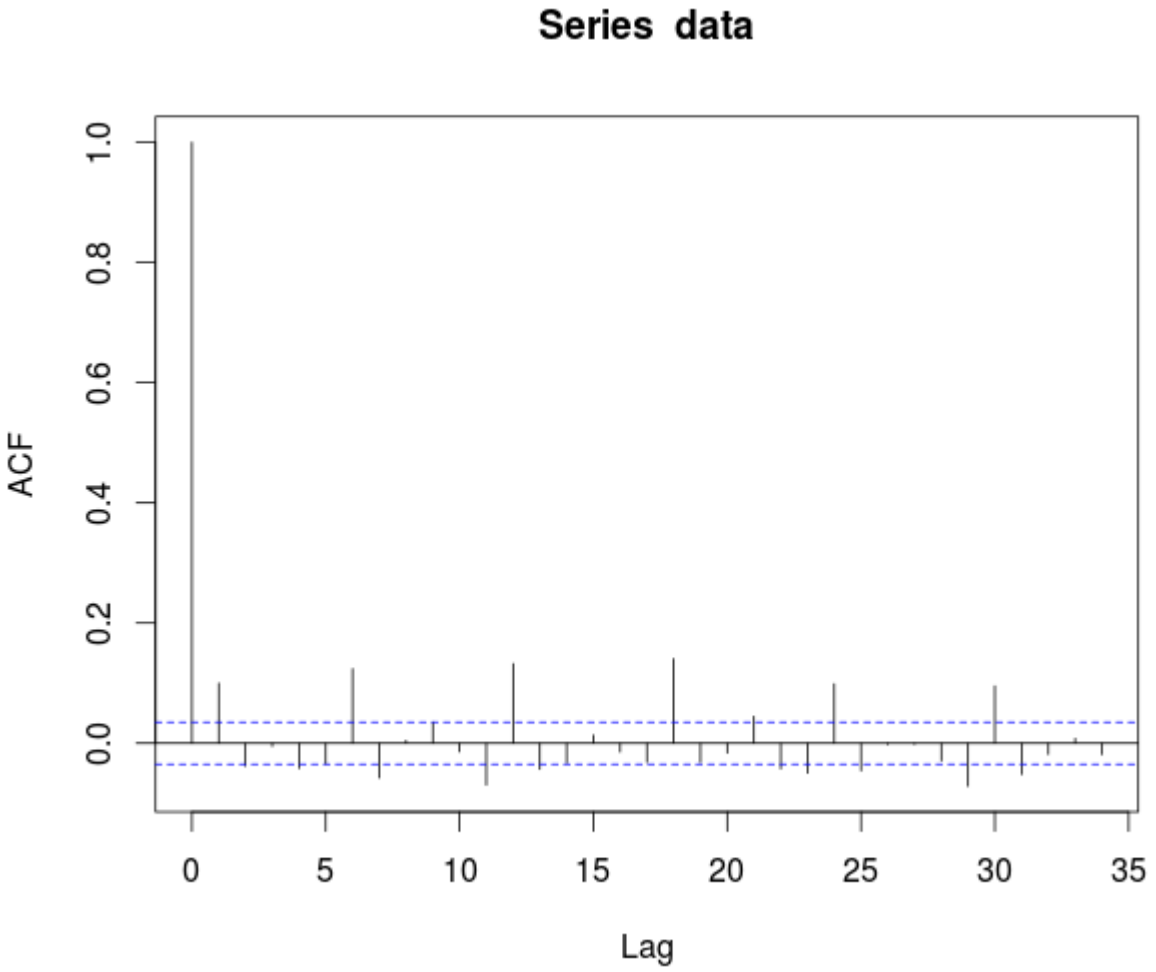
5)



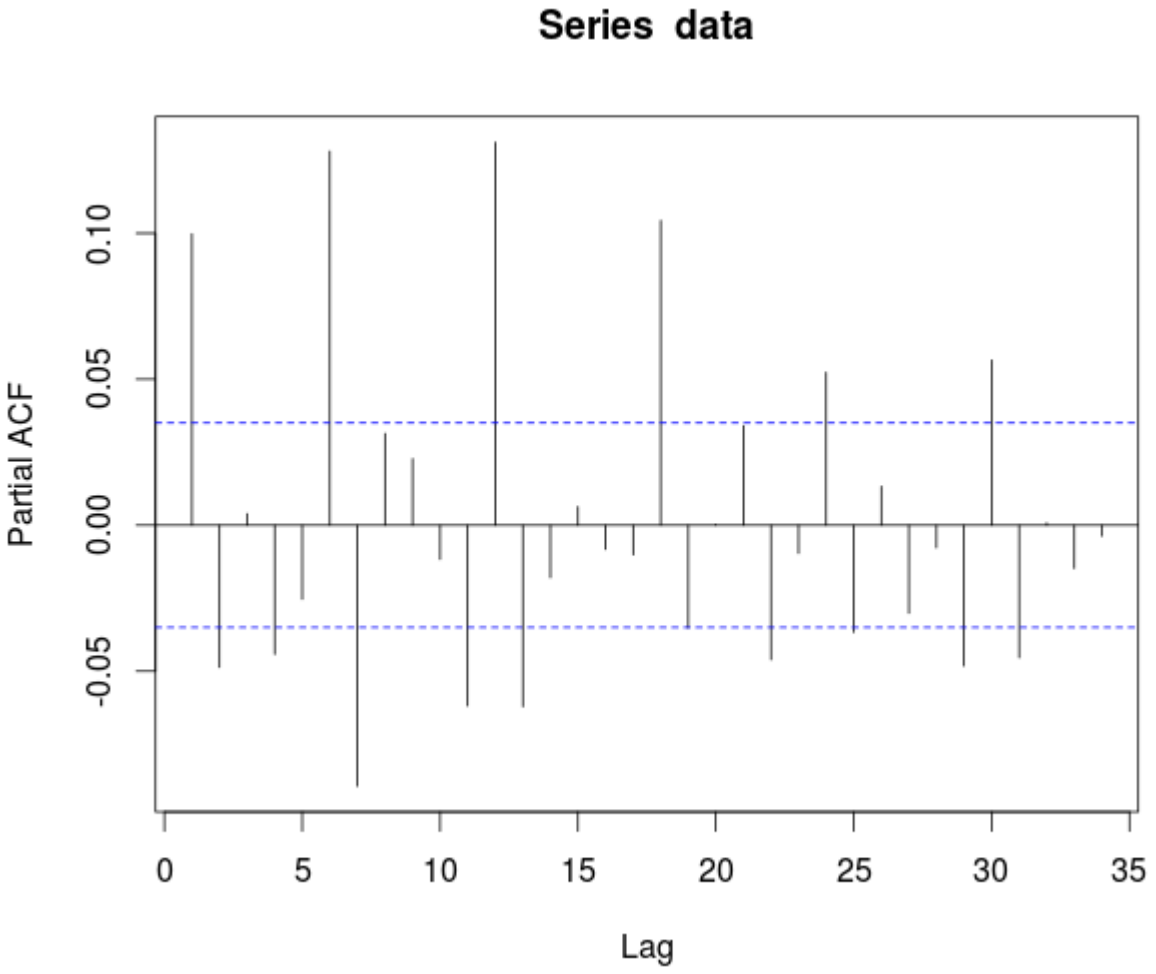
6)


Normal Q-Q Plot
Sample Quantiles
0.04
0.02
0.00
-0.02
-0.04
-3
-2
-1
0
1
2
3
Theoretical Quantiles

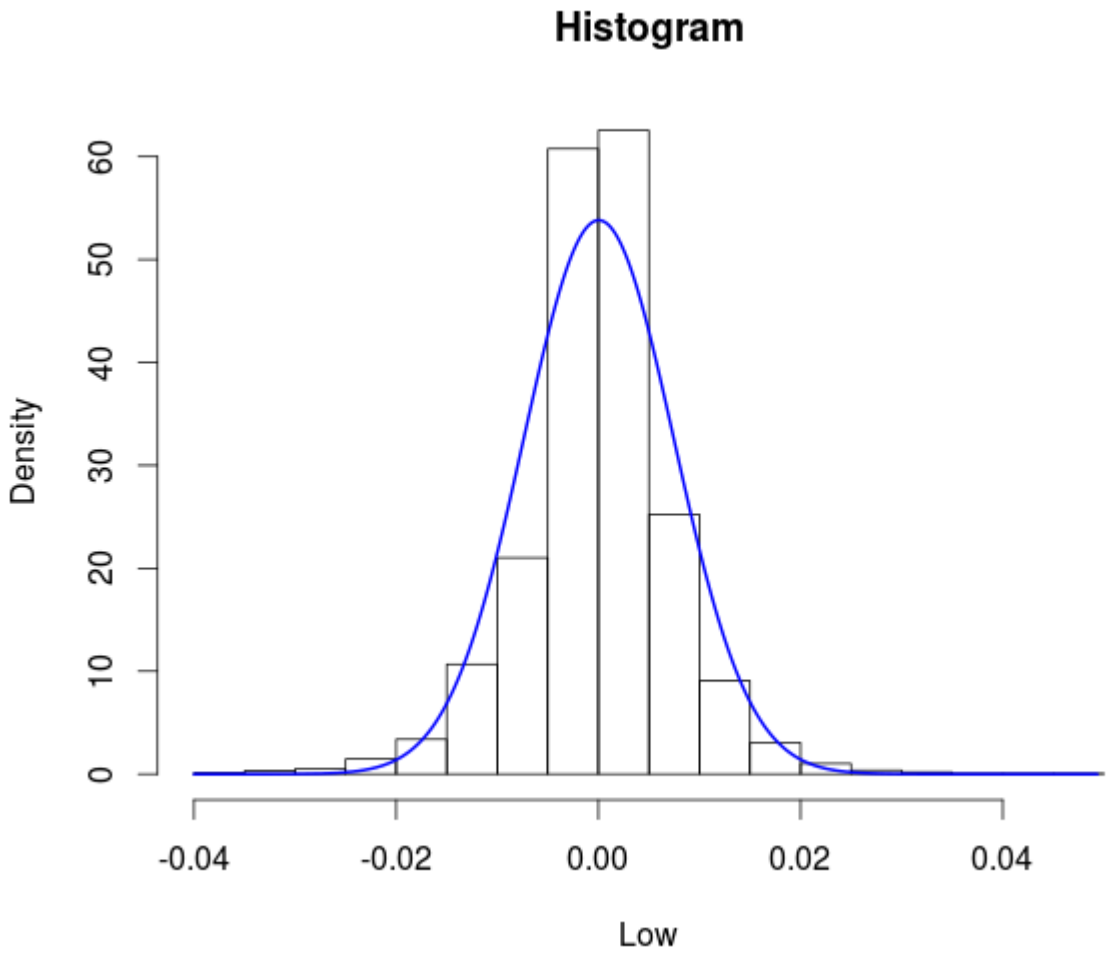
7)



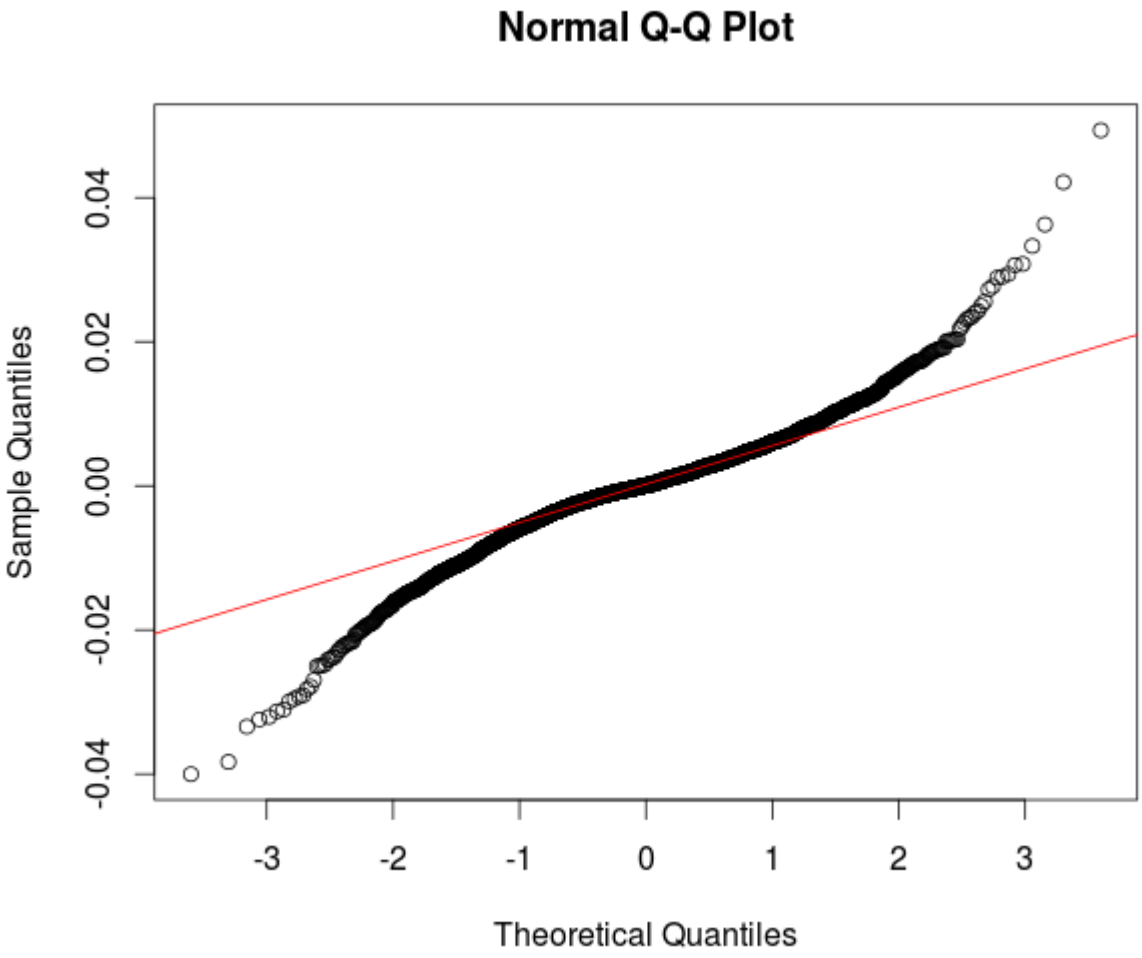
8)



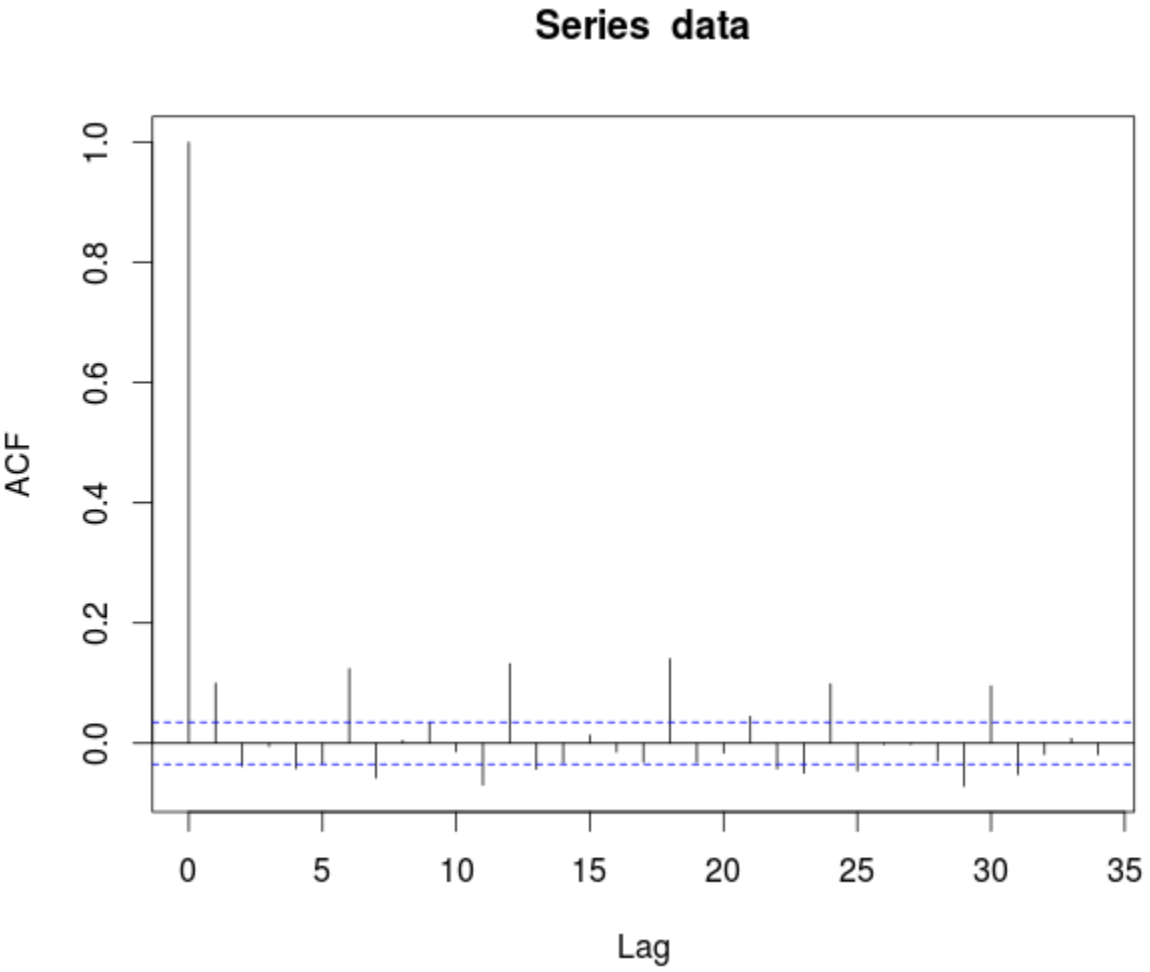
9)



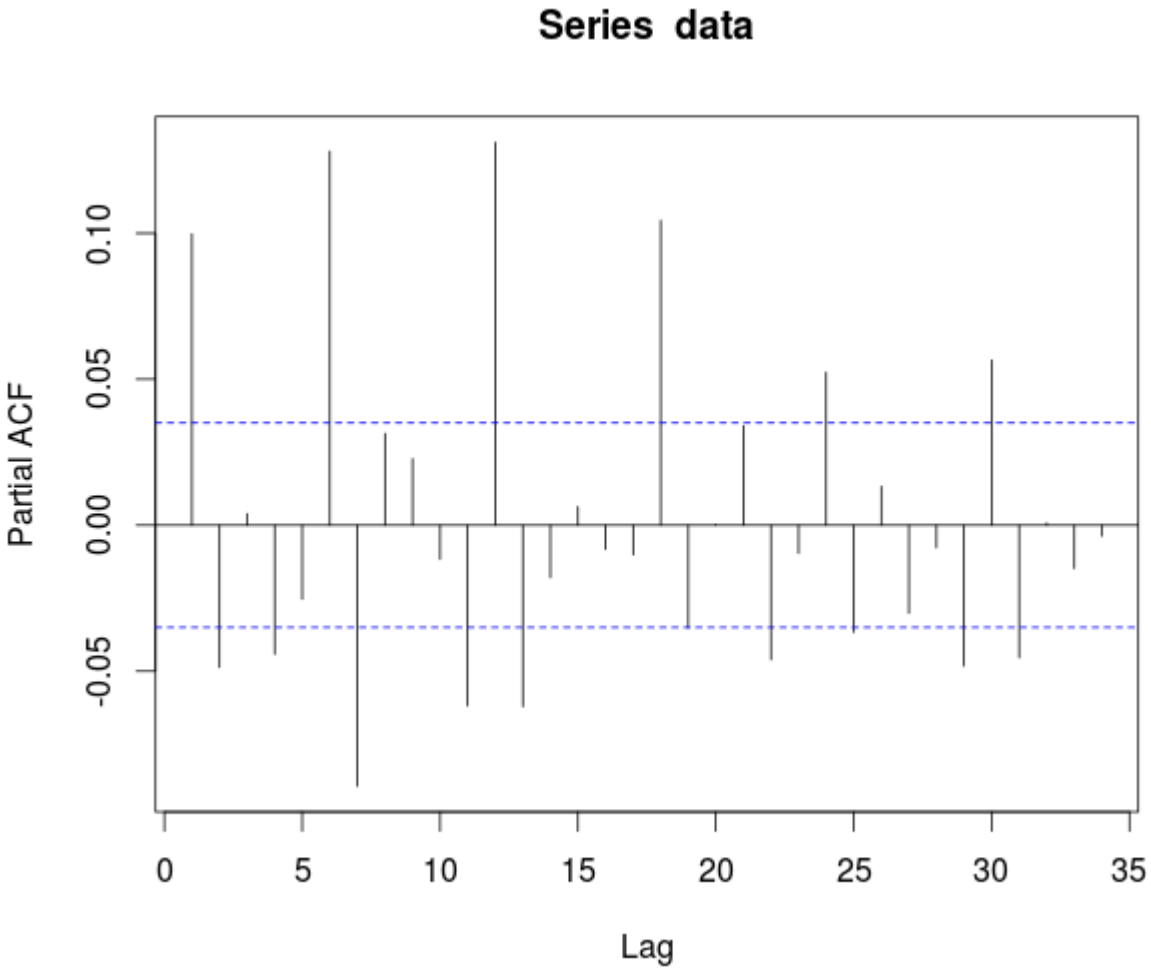
10)


Normal Q-Q Plot
Sample Quantiles
0.04
0.02
0.00
-0.02
-0.04
-3
-2
-1
0
1
2
3
Theoretical Quantiles

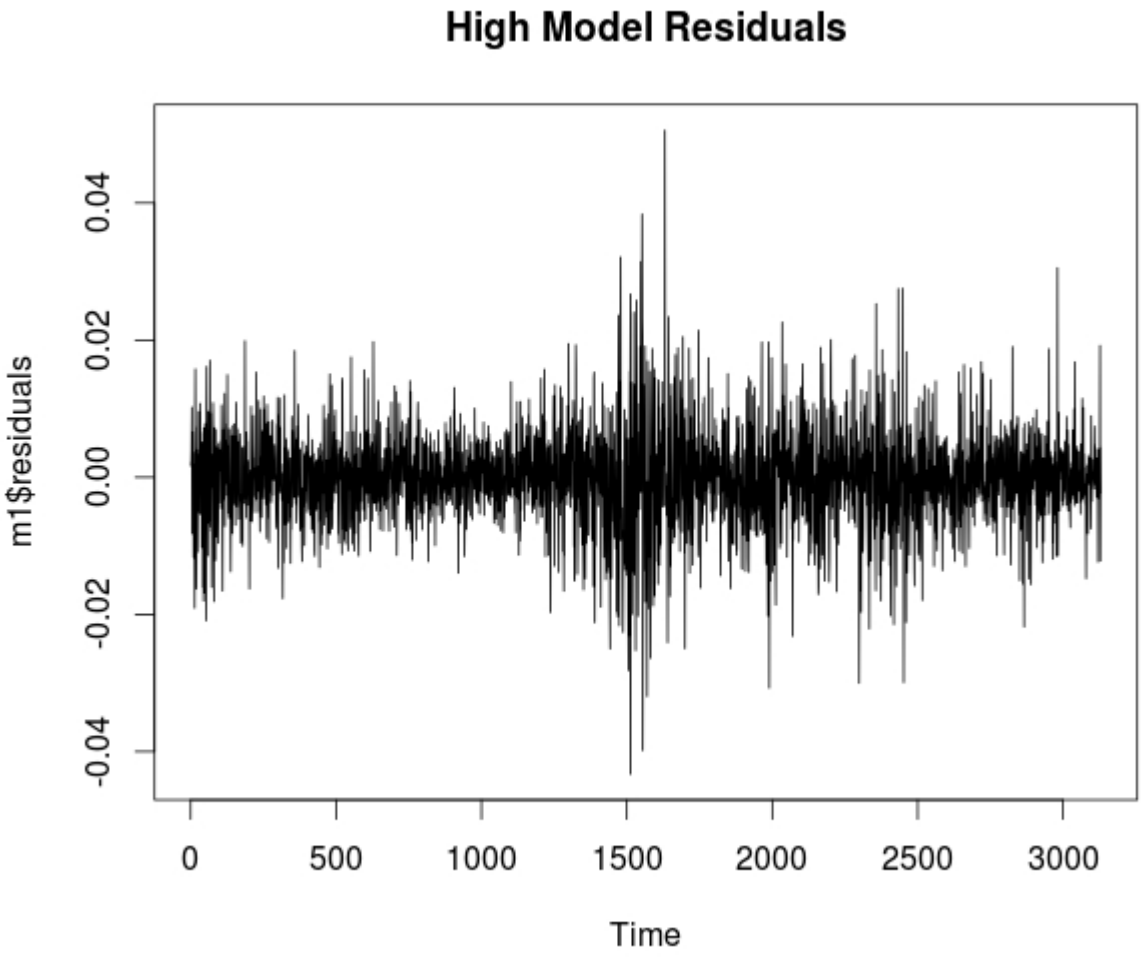
11)



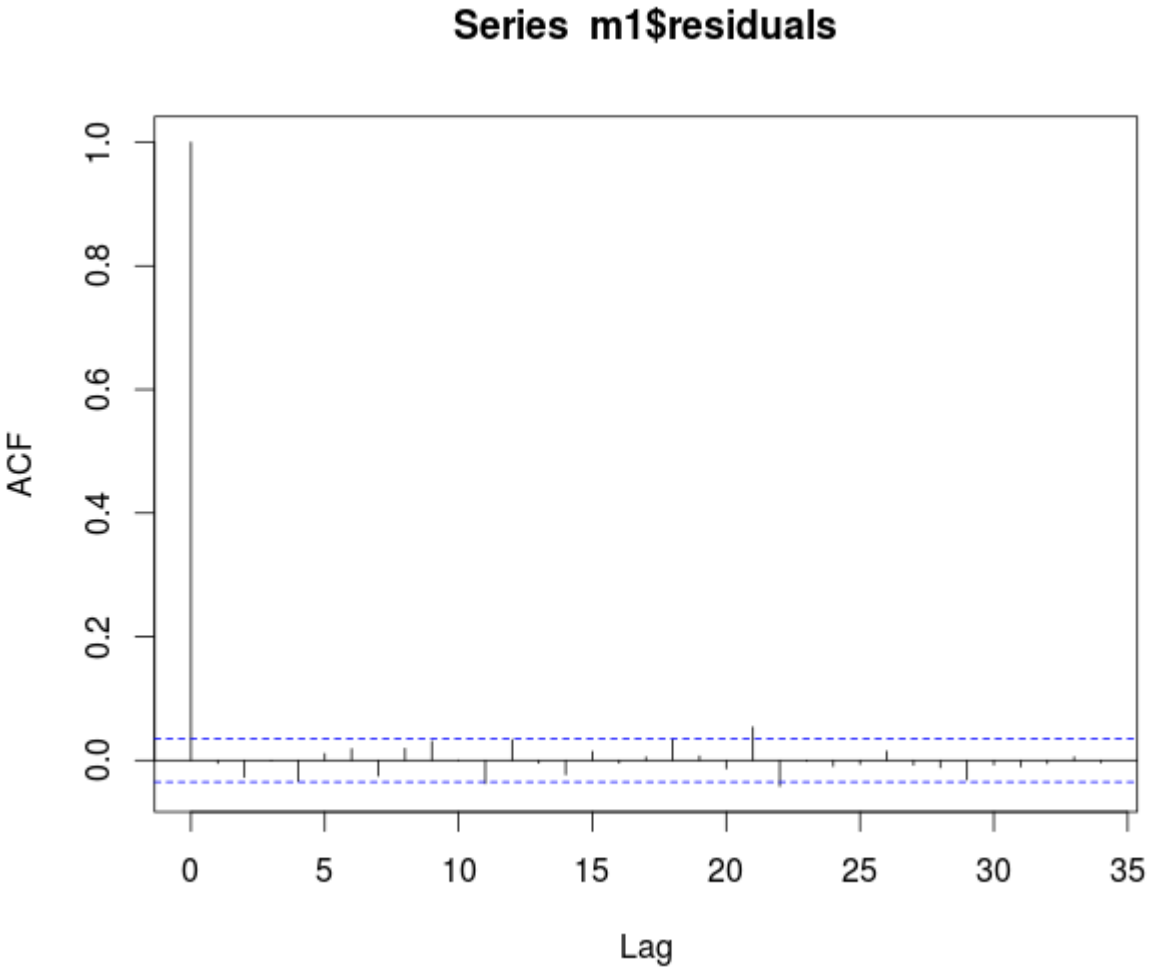
12)



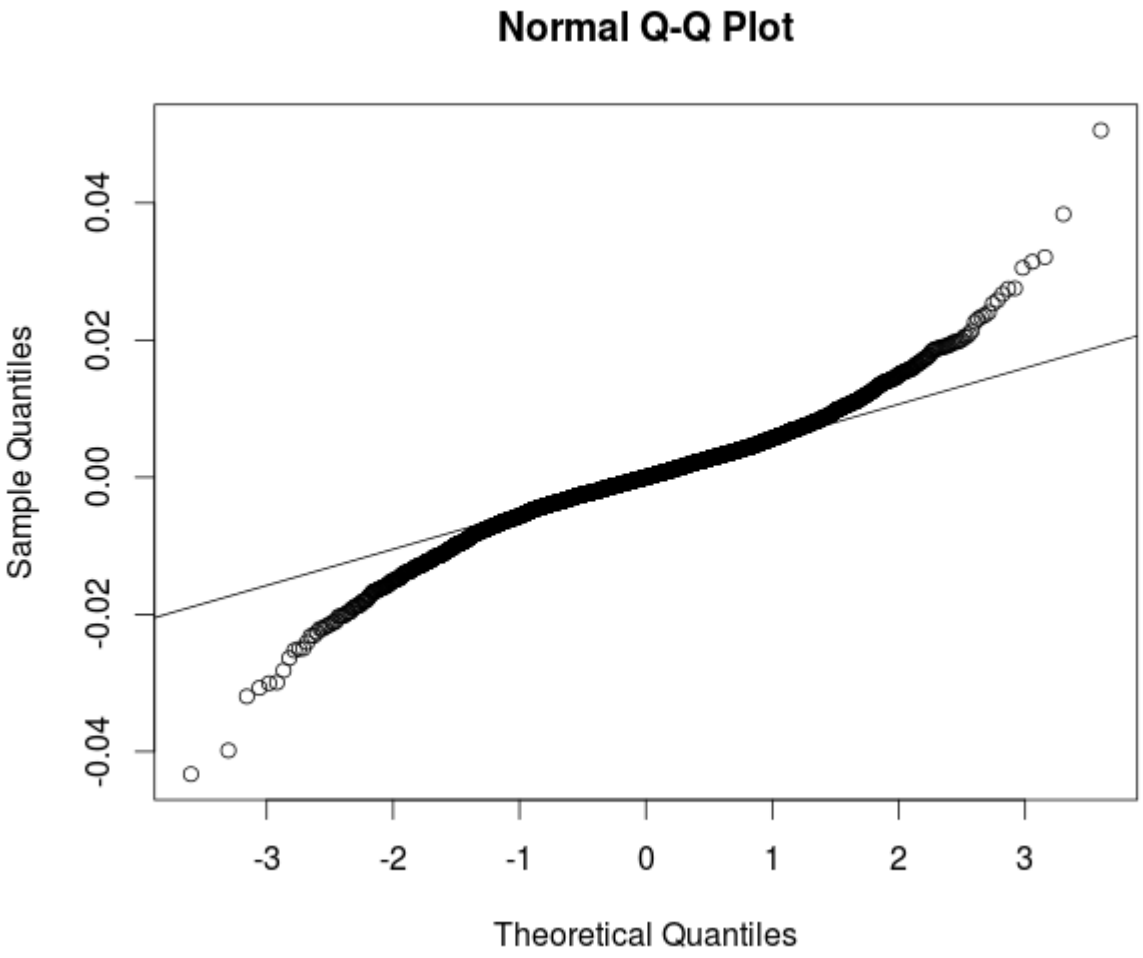
13)


High Model Residuals
m1$residuals
0.04
0.02
0.00
-0.02
-0.04
0
500
1000
1500
2000
2500
3000
Time

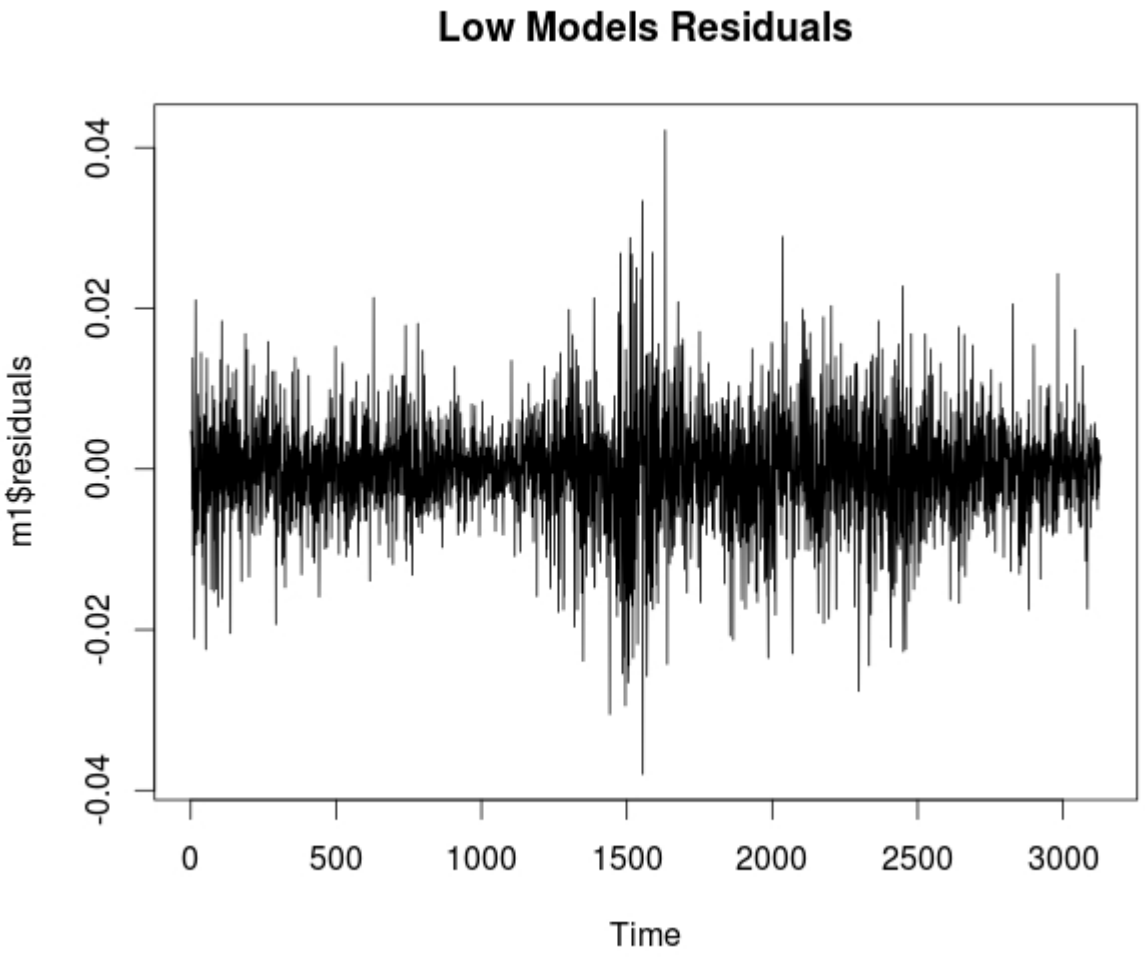
14)



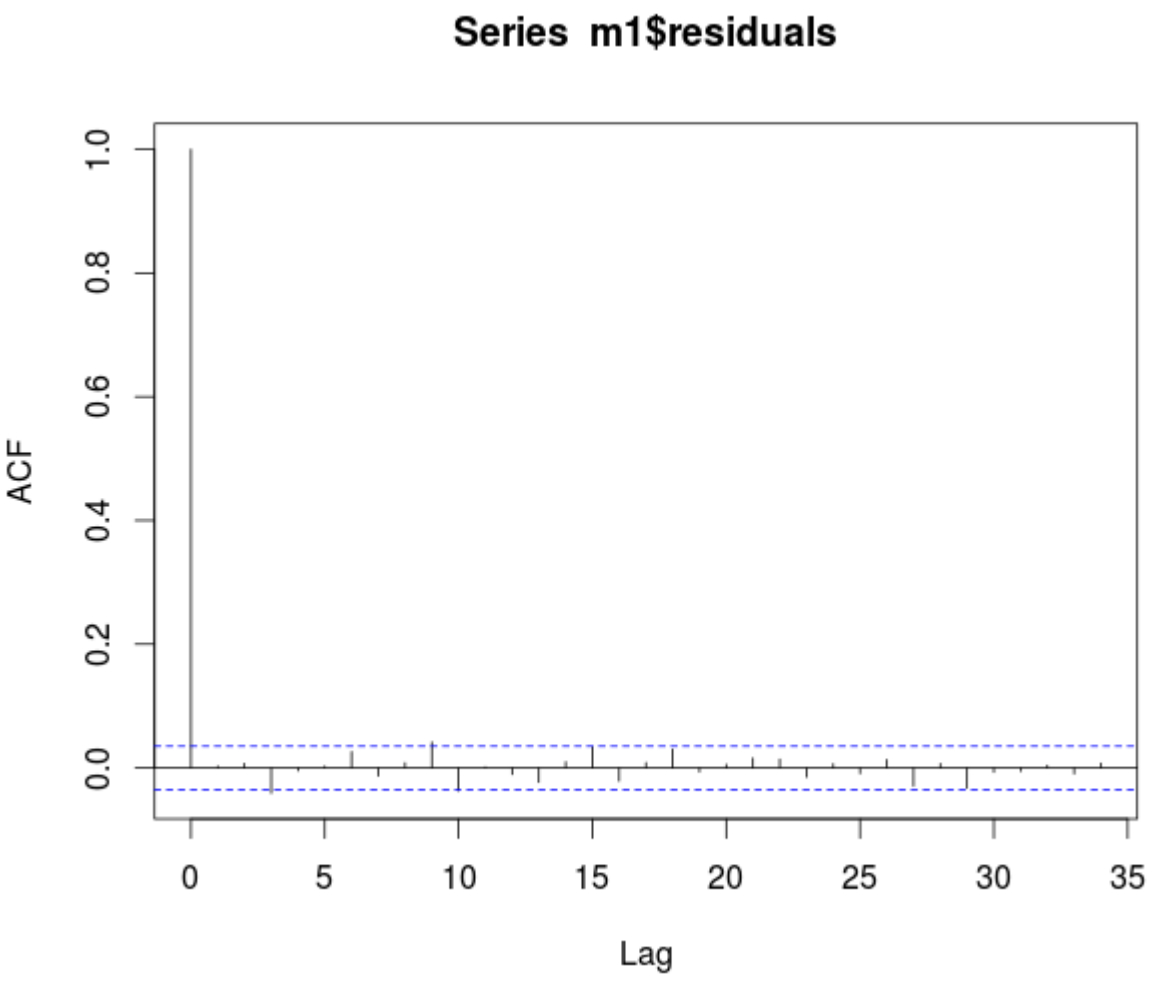
15)


Normal Q-Q Plot
Sample Quantiles
0.04
0.02
0.00
-0.02
-0.04
-3
-2
-1
0
1
2
3
Theoretical Quantiles

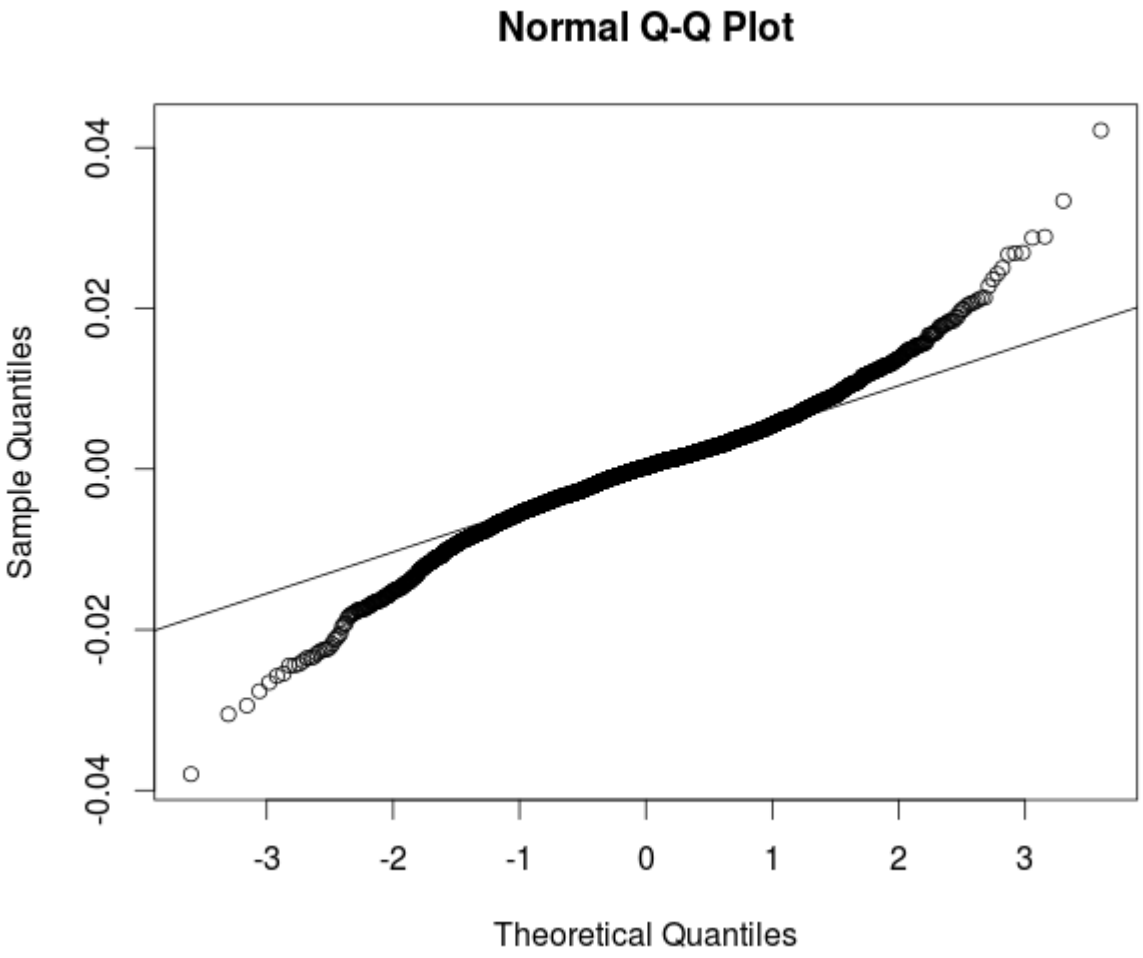
16)


Low Models Residuals
m1$residuals
0.04
0.02
0.00
-0.02
-0.04
0
500
1000
1500
2000
2500
3000
Time

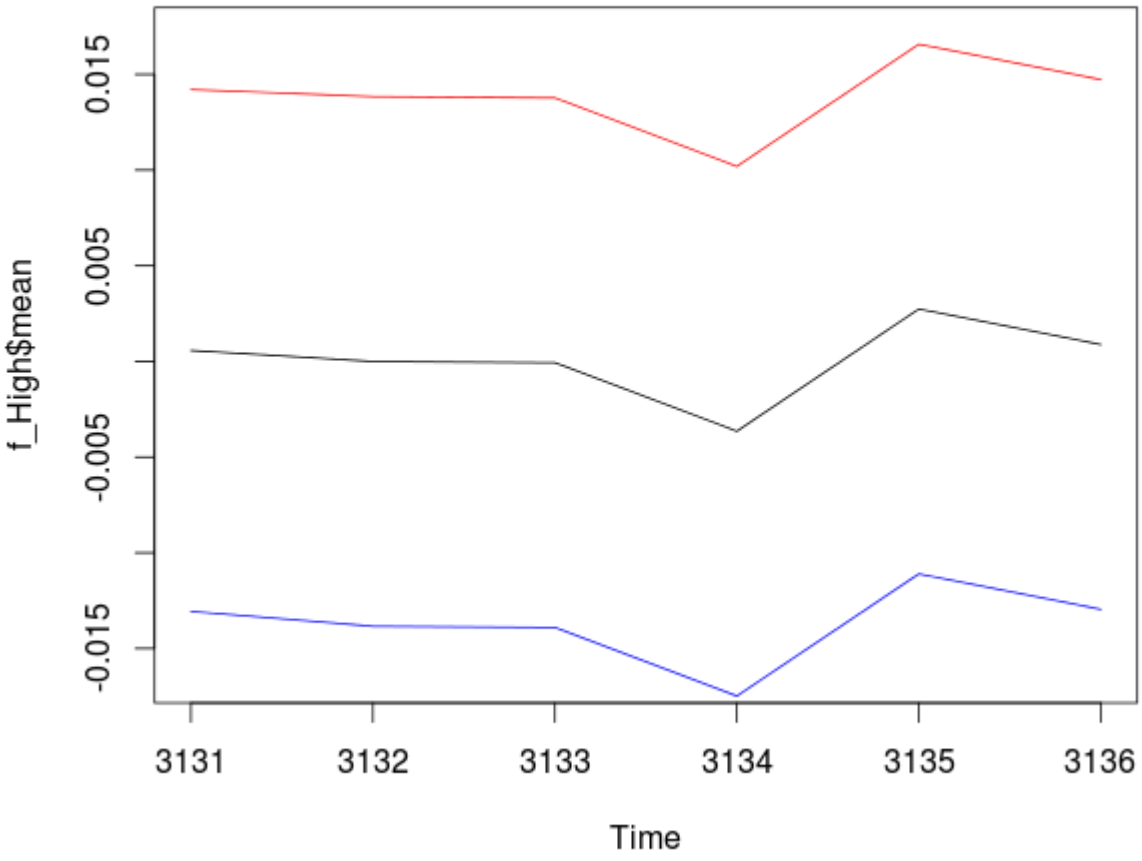
17)



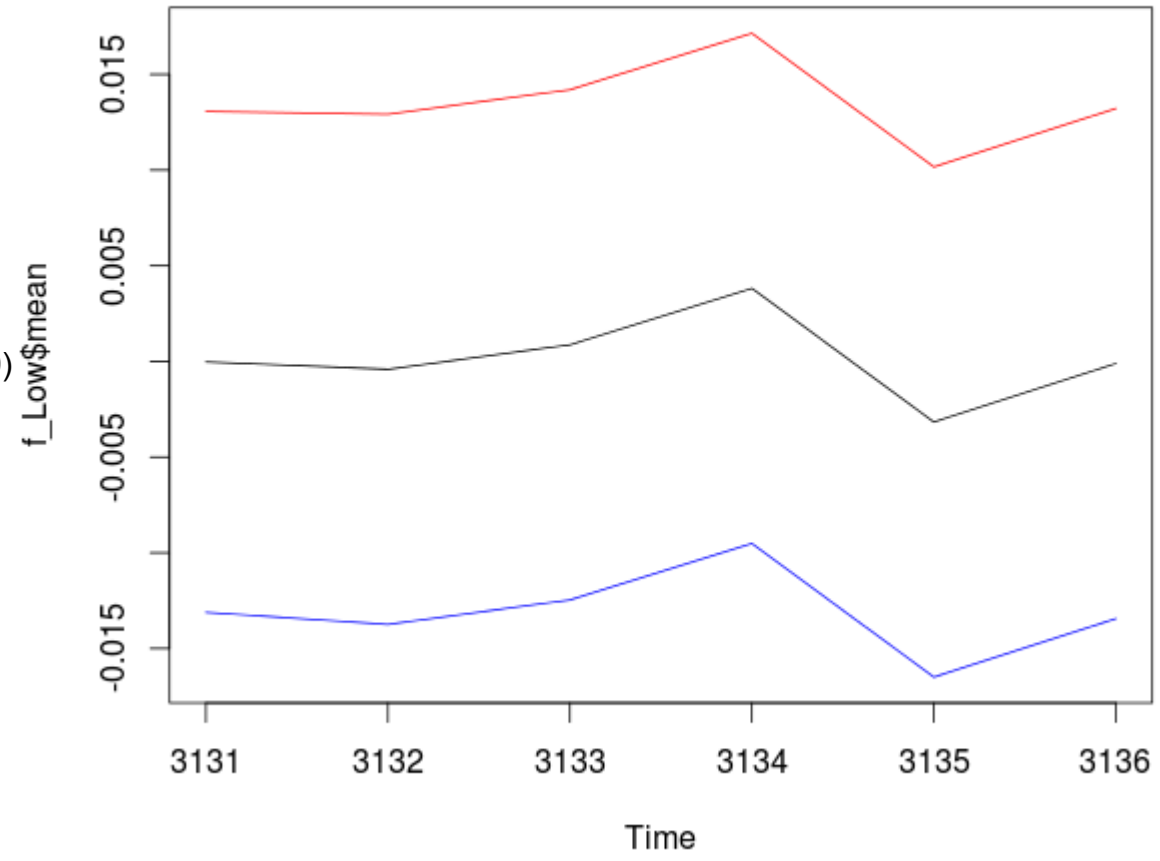
18)


Normal Q-Q Plot
Sample Quantiles
0.04
0.02
0.00
-0.02
-0.04
-3
-2
-1
0
1
2
3
Theoretical Quantiles

19)


f_High$mean
0.015
0.005
-0.005
-0.015
3131
3132
3133
3134
3135
3136
Time

20)


f_Low$mean
0.015
0.005
-0.005
-0.015
3131
3132
3133
3134
3135
3136
Time

APPENDIX C

```
#load required libraries
library(fBasics)
library(lmtest)
library(forecast)
library(fpp)

#set variable dataSet to the dataset
dataSet<-EURUSD_Candlestick_1_D_BID_01.01.2004.31.12.2013

##transform the data

##remove the data from dataSet when no trading happen so when volume is 0
dataSet=dataSet[dataSet$Volume!=0,]


###############
# Close Model #
###############
adf.test(dataSet$Close)
kpss.test(dataSet$Close)
#just look at the close data
data <- diff(dataSet$Close, lag=1)
adf.test(data)
kpss.test(data)

##data is no longer stationary

#Create Histogram
hist(data,xlab="Close",prob=TRUE, main="Histogram")
#add approximating normal density curve
xfit<-seq(min(data),max(data),length=length(data))
yfit<-dnorm(xfit,mean=mean(data),sd=sd(data))
lines(xfit,yfit, col="blue", lwd=2)

#Create Normal Probability Plot
qqnorm(data)
qqline(data,col=2)

#Normality Tests
normalTest(data,method=c("jb"))

#plot acf and pacf values
acf(data,plot=T)
pacf(data,plot=T)

#Box test
Box.test(data,lag=35,type="Ljung")


###############
# High Model  #
###############
```

```
adf.test(dataSet$High)
kpss.test(dataSet$High)
#just look at the High data
data <- diff(dataSet$High, lag=1)
adf.test(data)
kpss.test(data)

#Create Histogram
hist(data,xlab="High",prob=TRUE, main="Histogram")
#add approximating normal density curve
xfit<-seq(min(data),max(data),length=length(data))
yfit<-dnorm(xfit,mean=mean(data),sd=sd(data))
lines(xfit,yfit, col="blue", lwd=2)

#Create Normal Probability Plot
qqnorm(data)
qqline(data,col=2)

#Normality Tests
normalTest(data,method=c("jb"))

#plot acf and pacf values
acf(data,plot=T)
pacf(data,plot=T)

#Box test
Box.test(data,lag=35,type="Ljung")

#fit the model
m1=arima(data, order = c(0,0,1), seasonal = list(order = c(1, 0, 1), period = 6))
m1

#diagnostics
coeftest(m1)
plot(m1$residuals,main="High Model Residuals")
acf(m1$residuals)
qqnorm(m1$residuals)
qqline(m1$residuals)
normalTest(m1$residuals,method=c("jb"))
Box.test(m1$residuals,lag=35, type="Ljung",fitdf=1)

#compute predictions for up to 6 days ahead(1 week in market) also after 6 values repeat
f_High=forecast.Arima(m1,h=6)
f_High
plot(f_High$mean, type="l",ylim=c(min(f_Low$lower[,2]),max(f_Low$upper[,2])))
##had to add then subtract the mean to get indexing right. It looks funny but it works
lines(f_High$mean-f_High$mean+f_High$upper[,2],type="l",col="red")
lines(f_High$mean-f_High$mean+f_High$lower[,2],type="l",col="blue")

###############
# Low Model   #
###############
adf.test(dataSet$Low)
kpss.test(dataSet$Low)
```

```
#just look at the Low data
data <- diff(dataSet$Low, lag=1)
adf.test(data)
kpss.test(data)


#Create Histogram
hist(data,xlab="Low",prob=TRUE, main="Histogram")
#add approximating normal density curve
xfit<-seq(min(data),max(data),length=length(data))
yfit<-dnorm(xfit,mean=mean(data),sd=sd(data))
lines(xfit,yfit, col="blue", lwd=2)

#Create Normal Probability Plot
qqnorm(data)
qqline(data,col=2)

#Normality Tests
normalTest(data,method=c("jb"))

#plot acf and pacf values
acf(data,plot=T)
pacf(data,plot=T)

#Box test
Box.test(data,lag=35,type="Ljung")

#fit the model
data=ts(data)
m1=arima(data, order = c(0,0,1), seasonal = list(order = c(1, 0, 1), period = 6))
m1

#diagnostics
coeftest(m1)
plot(m1$residuals,main="Low Models Residuals")
acf(m1$residuals)
qqnorm(m1$residuals)
qqline(m1$residuals)
Box.test(m1$residuals,lag=35, type="Ljung",fitdf=1)
normalTest(m1$residuals,method=c("jb"))

#compute predictions for up to 6 days ahead(1 week in market) also after 6 values repeat
f_Low=forecast.Arima(m1,h=6)
f_Low
plot(f_Low$mean, type="l",ylim=c(min(f_Low$lower[,2]),max(f_Low$upper[,2])))
lines(f_Low$mean-f_Low$mean+f_Low$upper[,2],type="l",col="red")
lines(f_Low$mean-f_Low$mean+f_Low$lower[,2],type="l",col="blue")


##############################
# Forcast Models Together   #
##############################

##add the high and the low prediction models together on same plot
```

```
plot(f_Low$mean, type="l",ylim=c(min(f_Low$lower[,2]),max(f_High$upper[,2])),col='blue',main="Both Low and
High Predictions",ylab="Change in Price")
lines(f_Low$mean-f_Low$mean+f_Low$upper[,2],type="l",col="blue")
lines(f_Low$mean-f_Low$mean+f_Low$lower[,2],type="l",col="blue")
lines(f_High$mean, type="l",col='red')
lines(f_High$mean-f_High$mean+f_High$upper[,2],type="l",col="red")
lines(f_High$mean-f_High$mean+f_High$lower[,2],type="l",col="red")
```