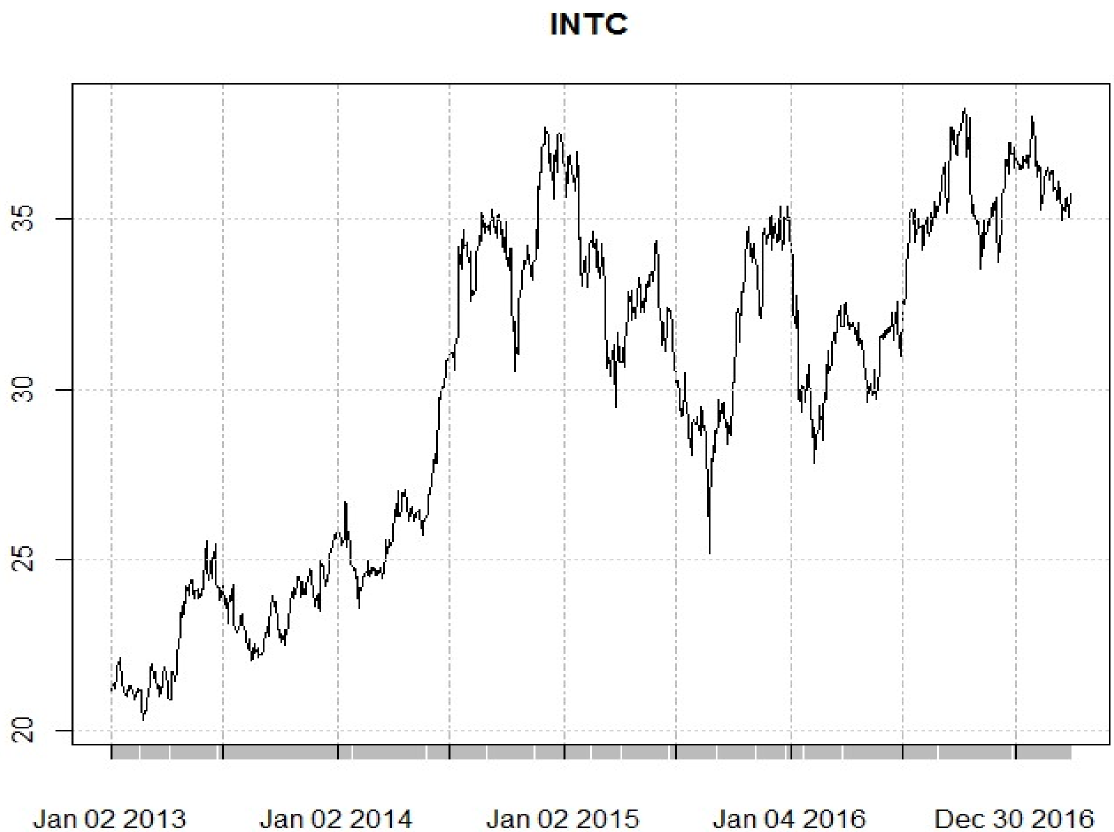
2017

Final

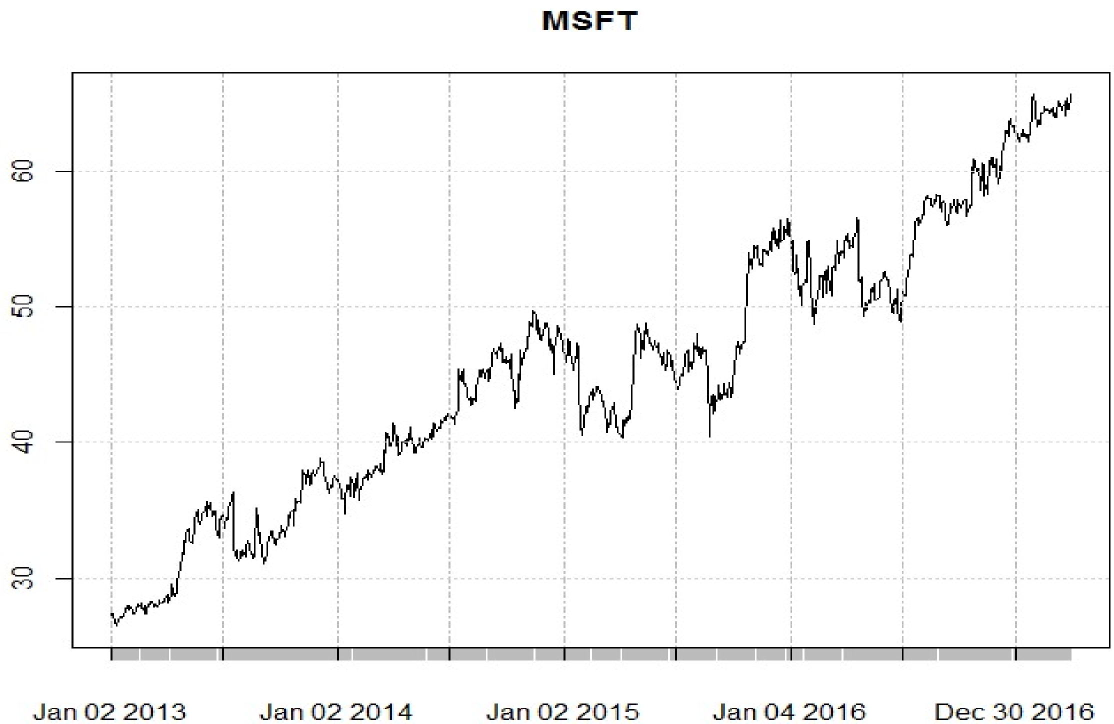
PREDICT 413

DAVID PAZMINO

1. **Download daily stock prices for the following major technology companies: Intel (INTC) and Microsoft (MSFT) and compute the log returns using the date range January 1, 2013 – March 31, 2017.**



The above plot is the time series plot for INTC. We can see a random walk pattern and the variance and mean do not seem constant. There is an upward trend and not real seasonality since there is no discernable pattern by time. We do see a large decrease in 2015 and then a steady uptrend upwards. There does not seem to be any cyclical patterns.



We can see a steady uptrend also with Microsoft. Again, we see a random walk as there seems to be no consistent variance around the mean. There might be some cyclical patterns between 2013 and 2015. We don't see the large drop as we did with INTC so wondering if these assets are correlated somewhat but have some differences in key areas.

```
library(quantmod)
library(PerformanceAnalytics)
getSymbols('MSFT', src = "google",  from ="2013-01-01", to = "2017-03-31")
getSymbols('INTC', src = "google",  from ="2013-01-01", to = "2017-03-31")

MSFT.copy = MSFT[, "MSFT.Close", drop=F]
INTC.copy = INTC[, "INTC.Close", drop=F]

MSFT.copy.ret = CalculateReturns(MSFT.copy, method="log")
INTC.copy.ret = CalculateReturns(INTC.copy, method="log")

MSFT.ret = MSFT.copy.ret[-1,]
INTC.ret = INTC.copy.ret[-1,]
colnames(MSFT.ret) ="MSFT"
colnames(INTC.ret) = "INTC"

plot(MSFT)

plot(INTC)
```

(a) Compute the sample mean, standard deviation, skewness, excess kurtosis, minimum, and maximum of the log returns for each series.

Series	Minimum	Maximum	Mean	Standard Deviation	Skewness	Excess Kurtosis
INTC	-0.095	0.089	0.000	0.014	-0.215	2.520
MSFT	-0.121	0.099	0.001	0.015	-0.249	7.952

The above table shows the summary statistics for the log returns of the series. Both series have a 0 mean which is good. Both have a high excess kurtosis meaning the distrubutions have heavy tails. The best excess kurtosis is 0. A large excess kurtosis means that a random sample taken from this data tends to contain more extreme values. These extreme values will mean that the probability of hitting the mean is lower than it would with a near zero excess kurtosis.

```
library(fBasics)
basicStats(MSFT.ret)
basicStats(INTC.ret)
```

(b) Test the null hypothesis that the mean of each of the series log returns is zero. Also, construct a 95% confidence interval for the daily log returns of each stock.

MSFT

```
One Sample t-test

data:  MSFT.ret
t = 1.8083, df = 1068, p-value = 0.07083
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -6.915516e-05  1.694957e-03
sample estimates:
mean of x
0.000812901
```

The null hypothesis for our t-test is that the mean is equal to zero. Our p-value is not significant since it is over .05. This fact means we cannot reject the null hypothesis and the mean of MSFT is zero.

```
t.test(MSFT.ret)
```

INTC

```
One Sample t-test

data:  INTC.ret
t = 1.1489, df = 1068, p-value = 0.2509
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
```

```
-0.0003463456  0.0013248406
sample estimates:
mean of x
0.0004892475
```

The p-value for the log return of INTC is .2509 which is greater than .05. We cannot reject the null hypothesis for this asset either and the mean is equal to 0 with 95% confidence.

```
t.test(INTC.ret)
```

The 95% confidence intervals for our data are the following:

	Lower Bound	Upper Bound
INTC	-0.0003463456	0.0013248406
MSFT	-6.915516e-05	1.694957e-03

(c) Test Ho: m3 = 0 vs. Ha: m3 != 0, where m3 denotes the skewness of the log return.

INTC

Running a T-Test against skewness gives a p-value of 2 with a t-statistic of -2.869535. The null hypothesi s of this test is that the data is not skewed. Because 2 is not significant, we do not have enough evidence t o reject the null hypothesis. This fact means with 95% confidence we do not see the data as being skewed .

```
tm3=skewness(INTC.ret)/sqrt(6/length(INTC.ret))
tm3
pp= 2 * (1-pnorm(tm3))
pp
```

MSFT

The T-Test for MSFT for skewness is -3.33. The P-value is 2, so we cannot reject the null hypotheses an d the data is not skewed with 95% confidence.

```
tm3=skewness(MSFT.ret)/sqrt(6/length(MSFT.ret))
tm3
pp= 2 * (1-pnorm(tm3))
pp
```

(d) Test Ho: K = 3 vs. Ha: K != 3, where K denotes the kurtosis.

INTC

```
Anscombe-Glynn kurtosis test

data:  INTC.ret
kurt = 8.5363, z = 11.6200, p-value < 2.2e-16
alternative hypothesis: kurtosis is not equal to 3
```

We can perform our Anscombe test to determine if the kurtosis of log returns of INTC is equal to 3. Our p-value in the above test is less than .05. This fact means we have sufficient evidence to reject the null hy pothesis and that the kurtosis is not equal to 3 for the log returns of INTC.

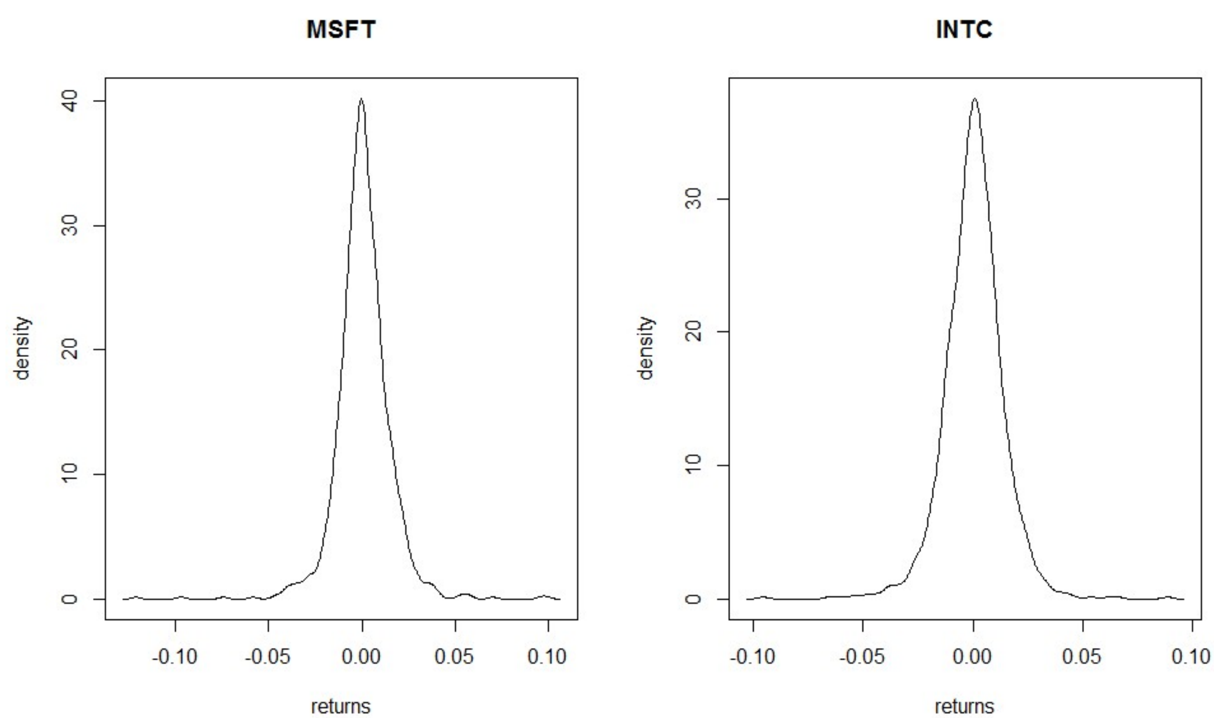
```
library(moments)
anscombe.test(INTC.ret)
```

MSFT

```
Anscombe-Glynn kurtosis test

data:  MSFT.ret
kurt = 13.978, z = 14.449, p-value < 2.2e-16
alternative hypothesis: kurtosis is not equal to 3
```

The Anscombe-Glynn test for the log returns of MSFT also show a p-value less than .05. We can reject the null hypothesis with 95% confidence and we can assume the log return of MSFT is not equal to 3.

(e) Obtain the empirical density plot of the daily log returns of each series, and select an appropriate distribution (Gaussian, t, etc.).

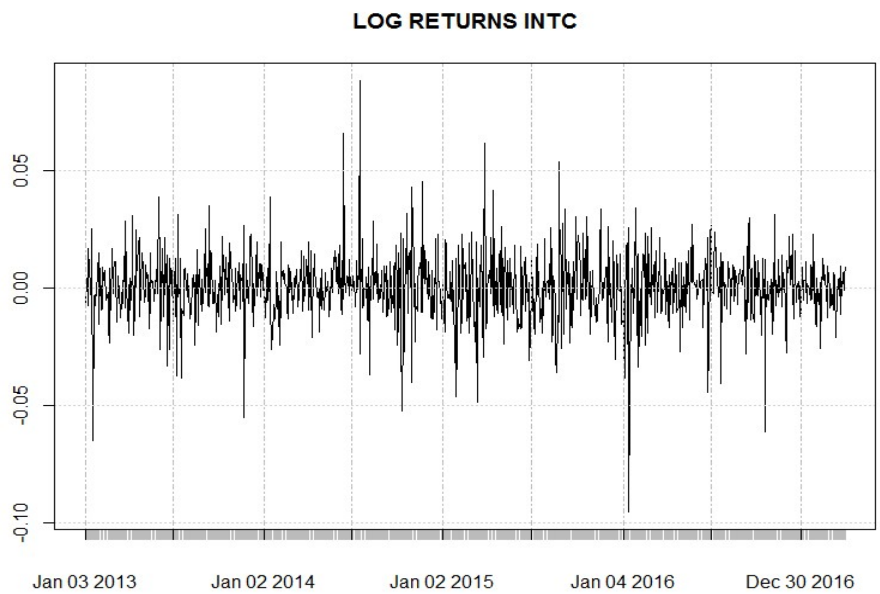


The plots above show our heavy tails as our distributions are thin but they are still normal (gaussian) distributions.

(f) Use the Box-Jenkins methodology to perform univariate time series model fitting to each of the series. Include details of each step of the process, and support your final model selection for each series.

INTC

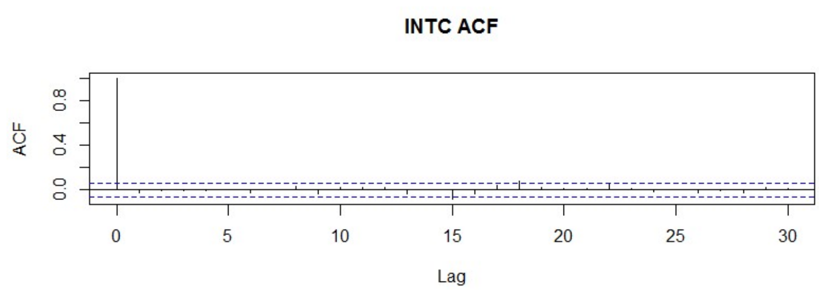
Stationarity

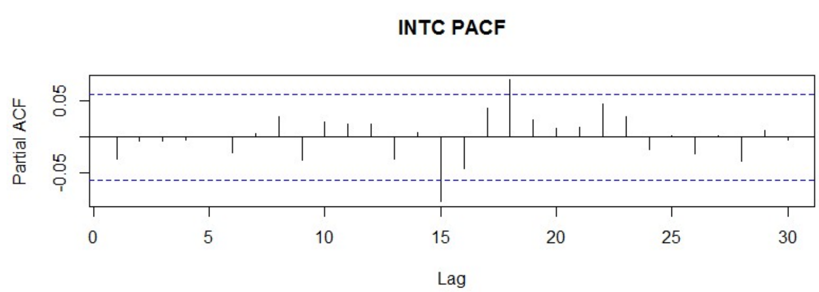


We need to make sure that the log return of INTC are stationary in order to pick a model. The above plot shows our returns which looks stationary but we will need to perform more tests.

```
plot(INTC.ret, main="LOG RETURNS INTC")
```

We can start with the ACF and PACF of the data.





We can see that the data is stationary since ACF does not have many lags. It also seems we have an eith an MA(15) or and ARMA(15,15) model since our lags are at 15 for both ACF and PACF. The quick decay of the ACF might make this just an MA model.

```
par(mfcol=c(2,1))
acf(INTC.ret, main="INTC ACF")
pacf(INTC.ret, main="INTC PACF")
par(mfcol=c(1,1))
```

We can also use a Dicky Fuller test to test for stationarity.

Augmented Dickey-Fuller Test

data: INTC.ret
Dickey-Fuller = -9.668, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary

The null hypothesis of the Dickey-Fuller test is that the time series is non-stationary. Since our p-value is significant (below .05), we can reject the null hypothesis and assume our series is stationary.

```
library(tseries)
adf.test(INTC.ret, alternative = "stationary")
```

The KPSS test is another unit root test to with a null hypothesis that the series is stationary.

KPSS Test for Level Stationarity

data: INTC.ret
KPSS Level = 0.073468, Truncation lag parameter = 7, p-value = 0.1

The p-value of the KPSS test shows that we do not have enough evidence to reject the hypothesis that the INTC series is stationary.

```
kpss.test(INTC.ret)
```

We should also check if there are serial correlations in the data.

```
Box-Ljung test

data:  INTC.ret
X-squared = 15, df = 15, p-value = 0.5
```

Our box test has a non-significant p-value meaning we do not have enough evidence to reject the null hypothesis that there are no serial correlations on the data, which is what we want.

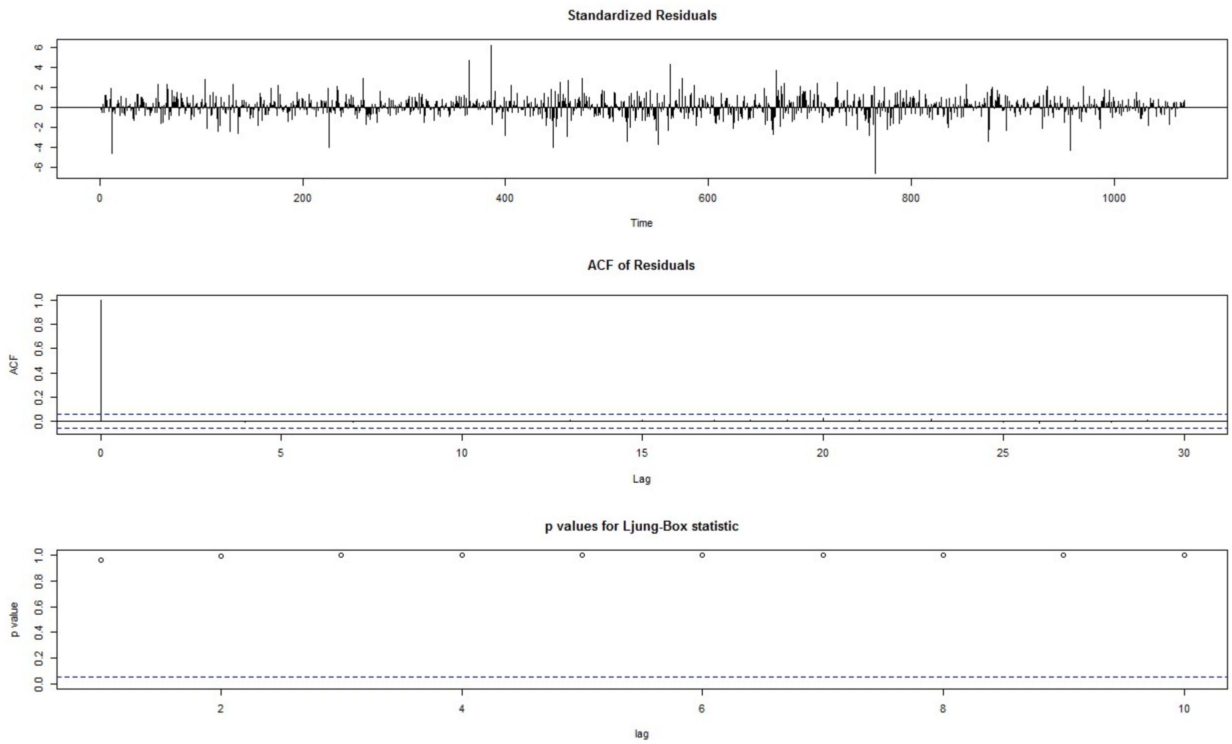
```
Box.test(INTC.ret,lag=15,type='Ljung')
```

Model Identification

ARMA(15,15)

```
Call:
arima(x = INTC.ret, order = c(15, 0, 15), include.mean = F)

Coefficients:
        ar1     ar2     ar3     ar4     ar5     ar6     ar7    ar8     ar9    ar10
      0.309  -0.693  -0.104  -0.118  -0.274  -0.230  -0.346  0.051  -0.524  -0.029
s.e.  0.604   0.302   0.409   0.455   0.466   0.423   0.384  0.311   0.267   0.349
```

```
       ar11    ar12   ar13    ar14   ar15     ma1    ma2    ma3    ma4    ma5
      0.003  -0.265  0.130  -0.098  0.049  -0.345  0.707  0.078  0.112  0.281
s.e.  0.526   0.432  0.271   0.211  0.584   0.598  0.304  0.409  0.457  0.470

        ma6    ma7     ma8    ma9   ma10    ma11   ma12    ma13   ma14    ma15
      0.202  0.366  -0.046  0.502  0.065  -0.021  0.296  -0.176  0.142  -0.187
s.e.  0.432  0.387   0.319  0.274  0.354   0.569  0.447   0.274  0.214   0.604
```



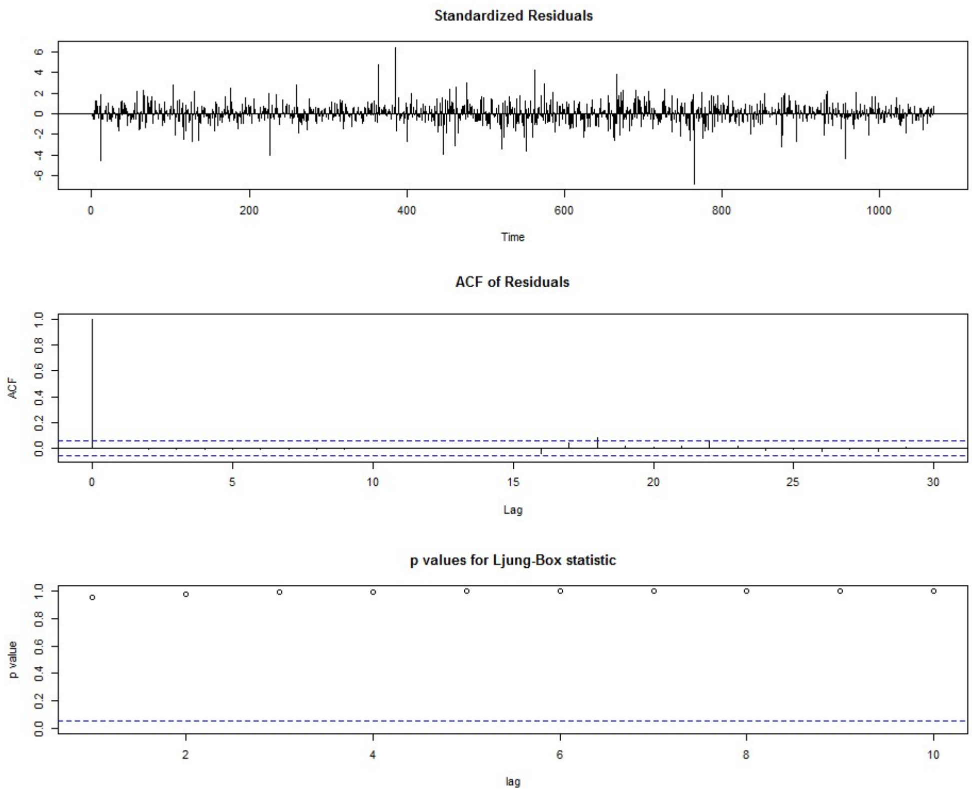
The standardized residuals display a white noise pattern which is what we want. There are no significant lags in the residuals. The p-values for the ljung-box test are not significant meaning there are no serial correlations in the residuals.

```
arma_15_15=arima(INTC.ret,order=c(15,0,15),include.mean=F)
tsdiag(arma_15_15)
summary(arma_15_15)
```

MA(15)

```
Call:
arima(x = INTC.ret, order = c(0, 0, 15), include.mean = F)

Coefficients:
          ma1     ma2     ma3     ma4     ma5      ma6     ma7     ma8      ma9   ma10
```

```
      -0.034  0.002  0.010  0.005  0.012  -0.021  0.017  0.037  -0.032  0.017
s.e.   0.030  0.031  0.031  0.031  0.031   0.031  0.031  0.032   0.029  0.031

       ma11   ma12    ma13   ma14    ma15
      0.010  0.016  -0.038  0.002  -0.099
s.e.  0.029  0.032   0.032  0.033   0.032
```



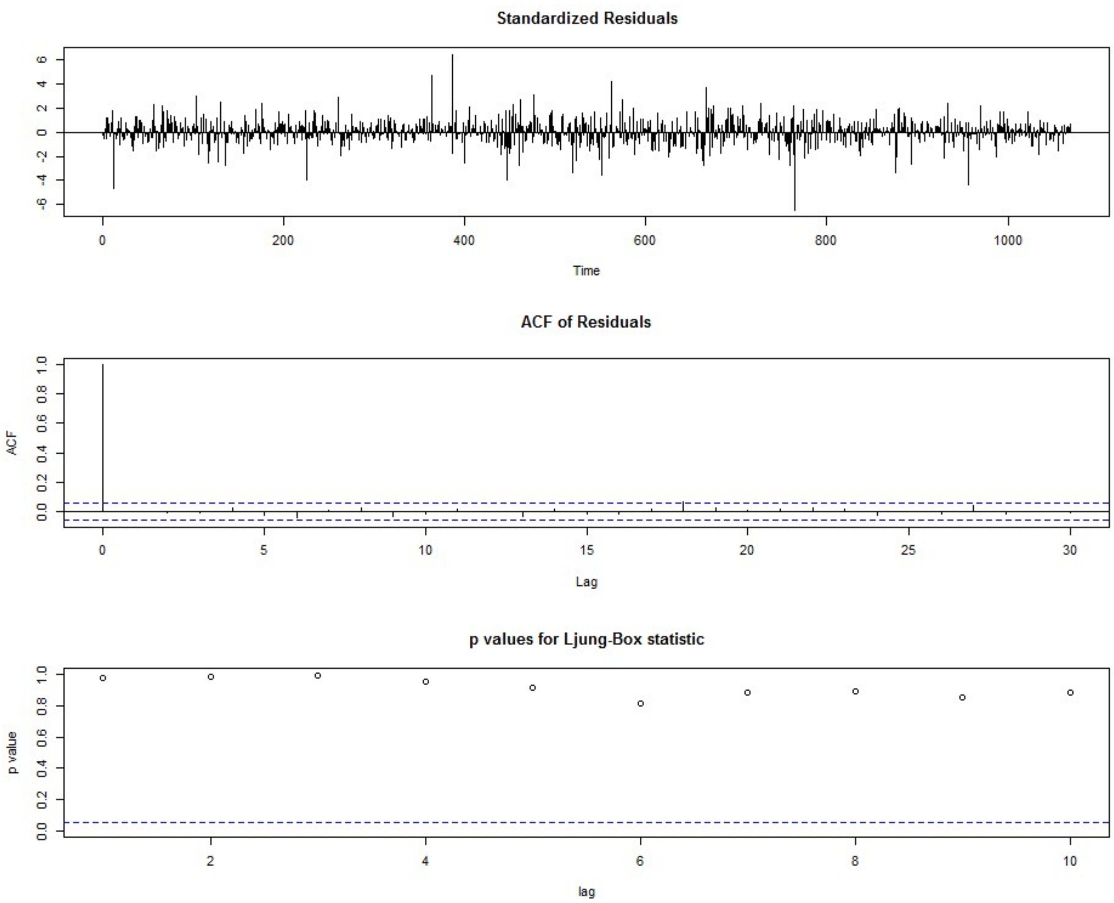
As in the previous model, the MA(15) model has white noise for the residuals and no significant lags, with the exception of lag 18. The Box tests are not significant meaning there are no serial correlations in the residuals.

```
ma_15=arima(INTC.ret,order=c(0,0,15),include.mean=F)
tsdiag(ma_15)
summary(ma_15)
```

Seasonal

```
Call:
arima(x = INTC.ret, order = c(0, 0, 15), seasonal = list(order = c(1, 0, 1),
    period = 12), include.mean = F)
```

```
Coefficients:
          ma1     ma2     ma3      ma4     ma5     ma6     ma7     ma8     ma9    ma10
       -0.039   0.002   0.006   -0.028   0.030   0.021   0.002   0.005   0.000   0.043
s.e.    0.031   0.030   0.032    0.023   0.022   0.021   0.023   0.026   0.024   0.025

         ma11     ma12    ma13     ma14     ma15    sar1    sma1
       -0.006   -0.730   0.024   -0.012   -0.082   0.710   0.025
s.e.    0.024    0.156   0.035    0.030    0.040   0.173   0.048
```



The residuals represent a white noise model. There are no significant lags in the acf of the residuals and the box statistics are no significant.

ARMA(15,1,5)

```
Call:
arima(x = INTC.ret, order = c(15, 1, 15), include.mean = F)
```

```
Coefficients:
         ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8      ar9     ar10
      -0.093   -0.293   -0.513   -0.478   -0.799   -0.556   -0.505   -0.423   -0.861   -0.412
s.e.   0.110    0.110    0.081    0.086    0.104    0.073    0.073    0.102    0.086    0.080

        ar11     ar12     ar13     ar14     ar15      ma1     ma2     ma3      ma4     ma5
      -0.235   -0.241   -0.217   -0.676   -0.048   -0.952   0.250   0.195   -0.027   0.334
s.e.   0.070    0.119    0.066    0.071    0.040    0.107   0.206   0.162    0.030   0.128

         ma6      ma7      ma8     ma9     ma10     ma11    ma12     ma13    ma14     ma15
      -0.263   -0.014   -0.083   0.419   -0.416   -0.182   0.006   -0.042   0.525   -0.750
s.e.   0.086      NaN    0.032   0.059    0.092    0.061   0.035    0.169   0.133    0.063
```

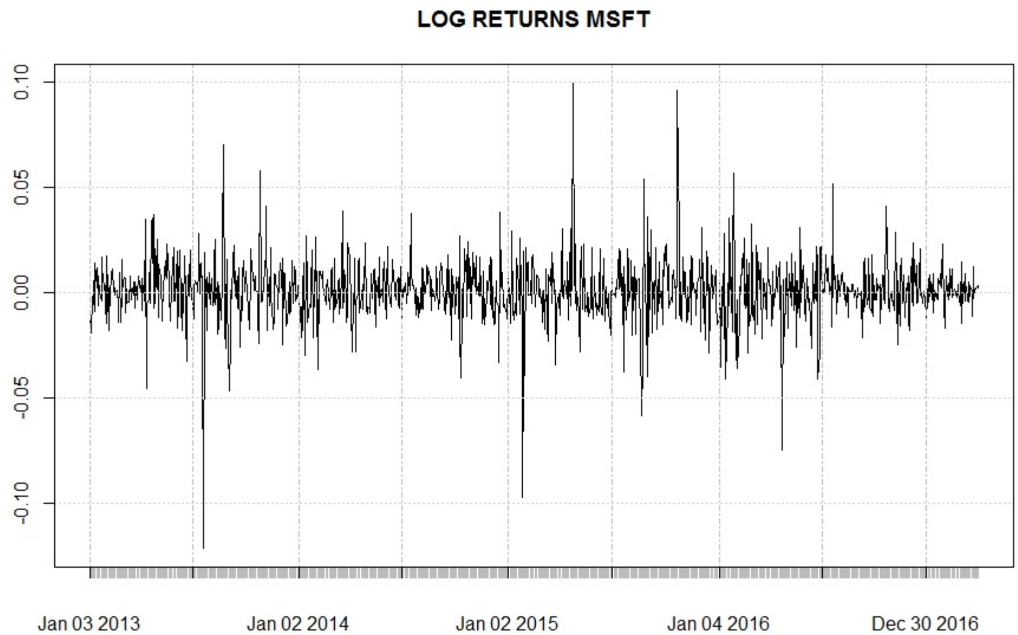
Model is not valid as there is an NAN in the std error, meaning the optimization method could not converge.

```
ma_15=arima(INTC.ret,order=c(15,1,15),include.mean=F)
tsdiag(ma_15)
summary(ma_15)
qqnorm(ma_15$residuals, main="Residuals (MA(15,1,15")
qqline(ma_15$residuals)
```

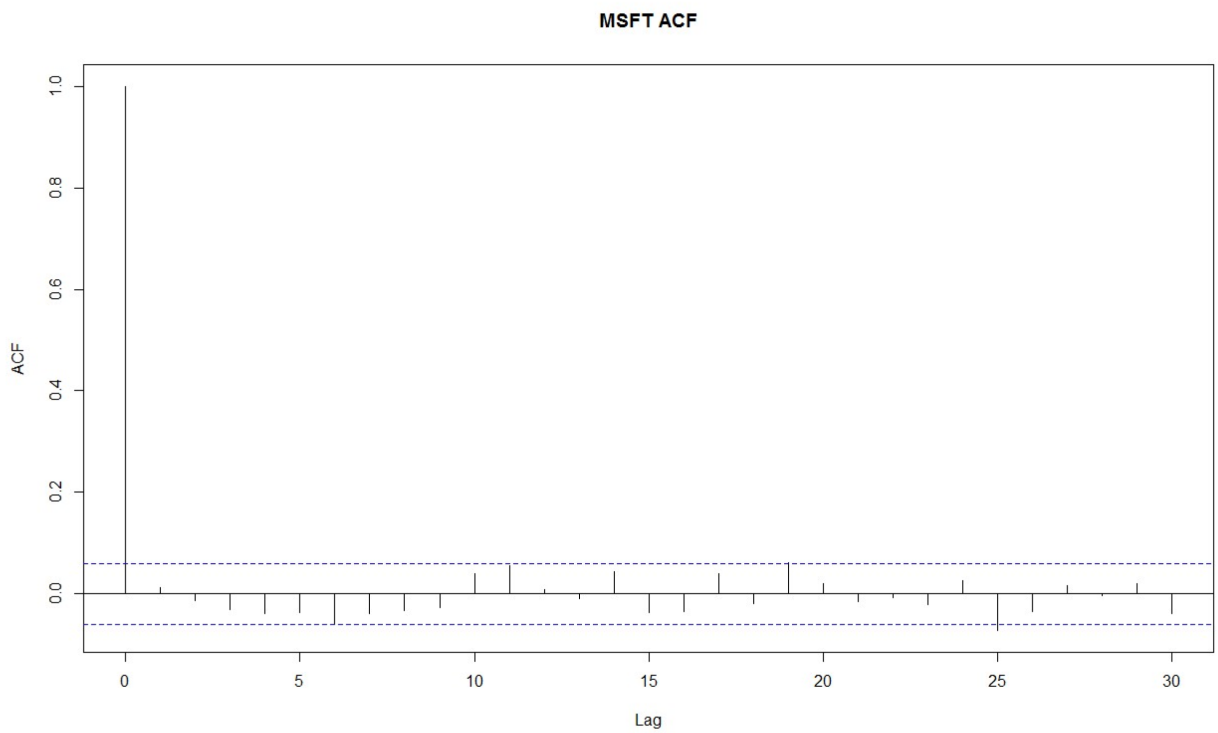
	RMSE	**AIC**	**LOG LIKELIHOOD**	**Variance**
ARMA15_15	0.01369	-6078	3070	0.000187
MA(15)	0.01382	-6088	3060	0.000191
Seasonal	0.01378	-6090	3063	0.00019
ARIMA(15,1,15) (not valid)	0.01357	-6071	3066	0.000184

Looking at our data, the ARMA(15,15) has the best AIC, RMSE, and LOG LIKELIHOOD so this will be our model of choice.

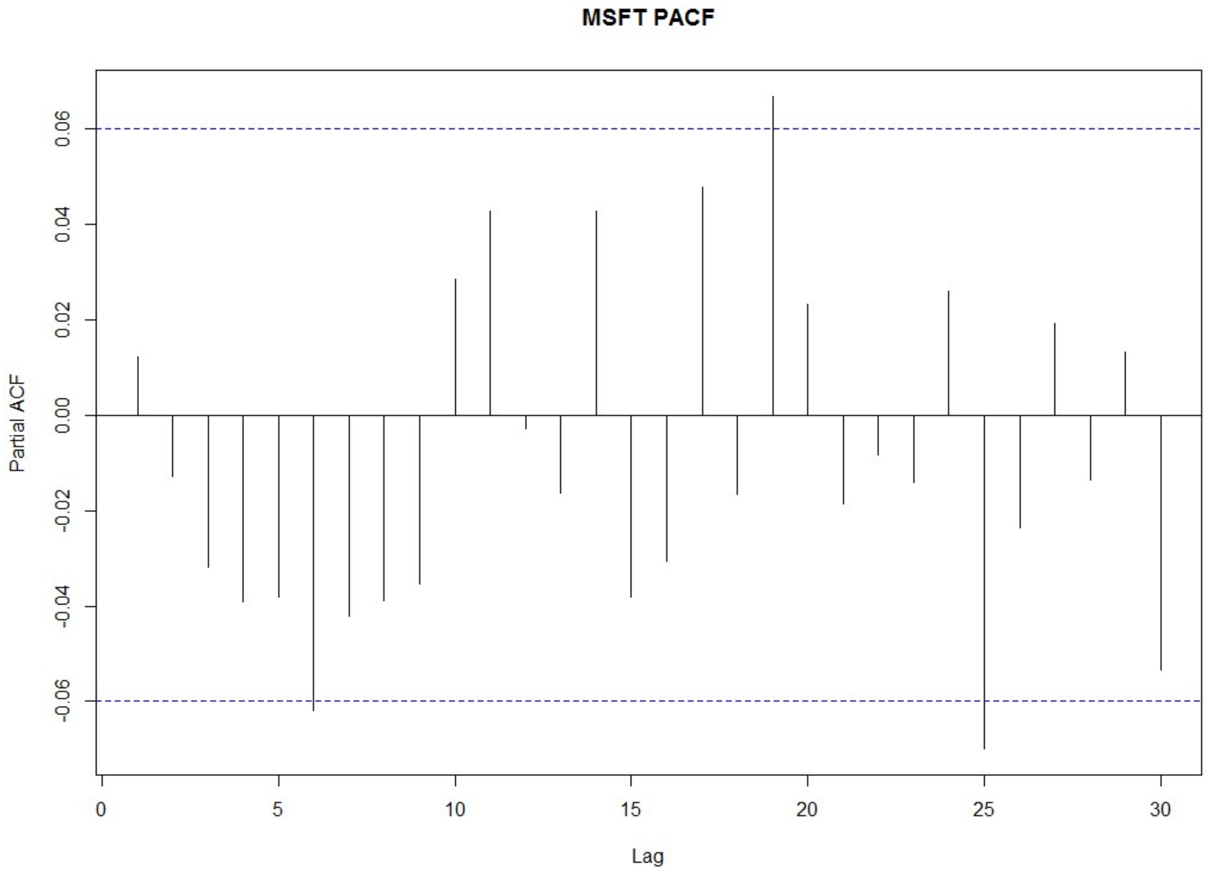
MSFT



The above is the plot of the log returns of MSFT. The data look stationary. Let's perform the tests.



The ACF of MSFT looks stationary. There are no significant lags in the data. 6, 19 and 25 are close to the confidence intervals with 25 the only one crossing the interval.



The PACF is showing some seasonality but still stationary. There are significant lags in 6, 19 and 25.

```
plot(MSFT.ret, main="LOG RETURNS MSFT")
par(mfcol=c(2,1))
acf(MSFT.ret, main="MSFT ACF")
pacf(MSFT.ret, main="MSFT PACF")
par(mfcol=c(1,1))
```

Augmented Dickey-Fuller Test

data: MSFT.ret
Dickey-Fuller = -10.552, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary

The p-value in our ADF test is significant meaning we can reject the null hypothesis that the series is not stationary.

KPSS Test for Level Stationarity

data: MSFT.ret
KPSS Level = 0.034952, Truncation lag parameter = 7, p-value = 0.1

The p-value for the KPSS test is not significant, meaning we cannot reject the null hypothesis that the series is stationary.

Box-Ljung test

data: MSFT.ret
X-squared = 28.473, df = 19, p-value = 0.07474

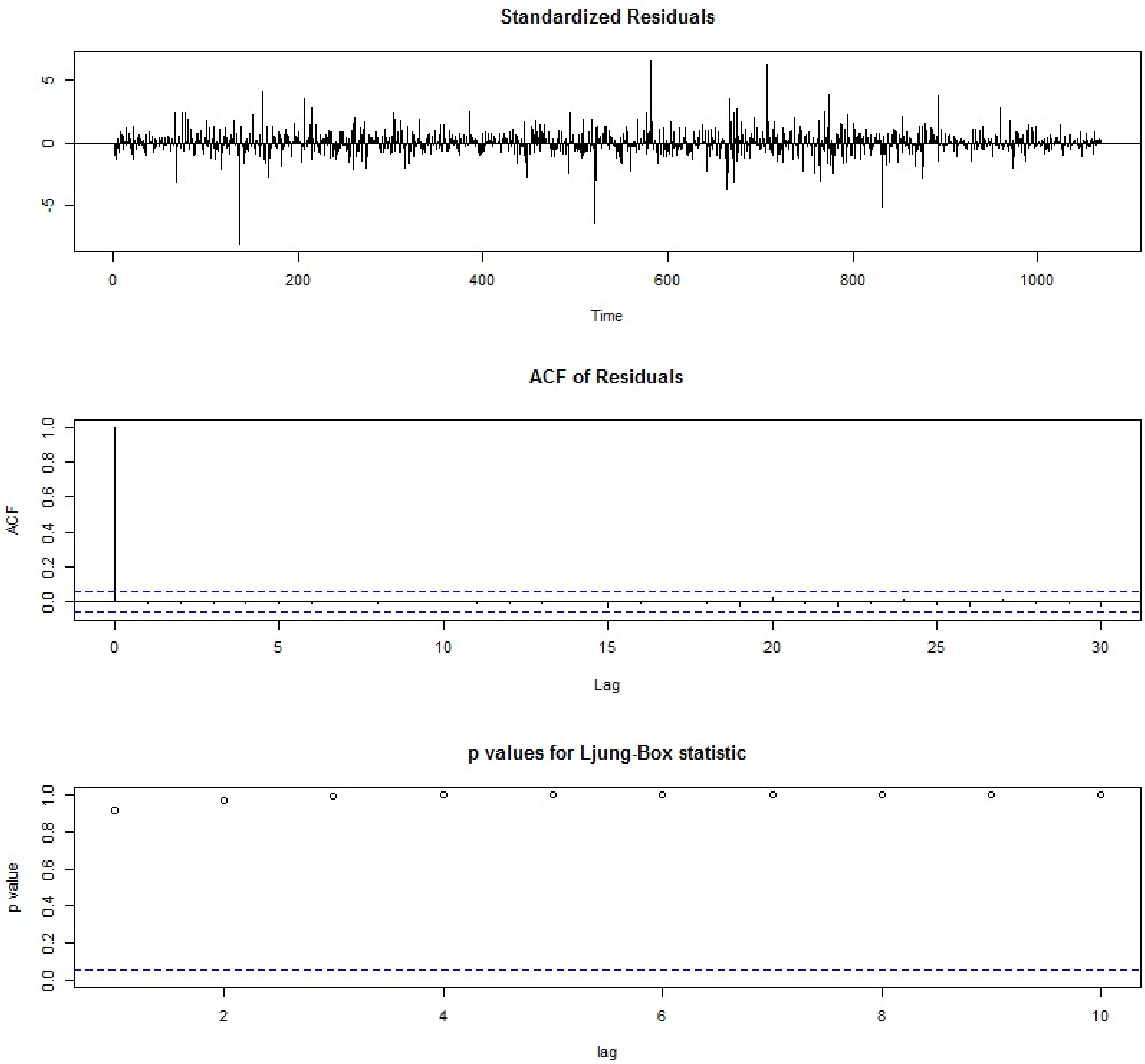
The Box_Ljung test has a p-value that is not significant meaning we cannot reject the null hypothesis that there are no serial correlations in the data.

```
library(tseries)
adf.test(MSFT.ret, alternative = "stationary")

kpss.test(MSFT.ret)

Box.test(MSFT.ret,lag=19,type='Ljung')
```

Models

ARMA(6,19)

```
Call:
arima(x = na.omit(MSFT.ret), order = c(6, 0, 19), include.mean = F)

Coefficients:
         ar1      ar2      ar3      ar4     ar5      ar6      ma1     ma2     ma3     ma4      ma5    ma6
       0.371  -0.088  -0.143  -0.023  0.142  -0.766  -0.372  0.073  0.115  0.009  -0.166  0.737
s.e.   0.098   0.121   0.134   0.100  0.098   0.077   0.101  0.124  0.135  0.101   0.103  0.081

          ma7      ma8      ma9    ma10    ma11     ma12     ma13    ma14     ma15     ma16    ma17     ma18
       -0.018  -0.028  -0.055  0.022  0.033  -0.054  -0.023  0.048  -0.041  -0.004  0.107  -0.020
s.e.    0.040   0.040   0.041  0.039  0.041   0.041   0.040  0.040   0.041   0.045  0.041   0.037
        ma19
       0.078
s.e.   0.039

sigma^2 estimated as 0.000206:  log likelihood = 3017,  aic = -5981

Training set error measures:
                      ME    RMSE       MAE MPE MAPE   MASE       ACF1
Training set 0.0008491 0.01436 0.009831 NaN  Inf 0.6707 -0.003308
```



The residuals show a white noise pattern. We also see no lags in the ACF, meaning we do not see any significant correlations. The p-values are not significant meaning the data is independent of previous lags.

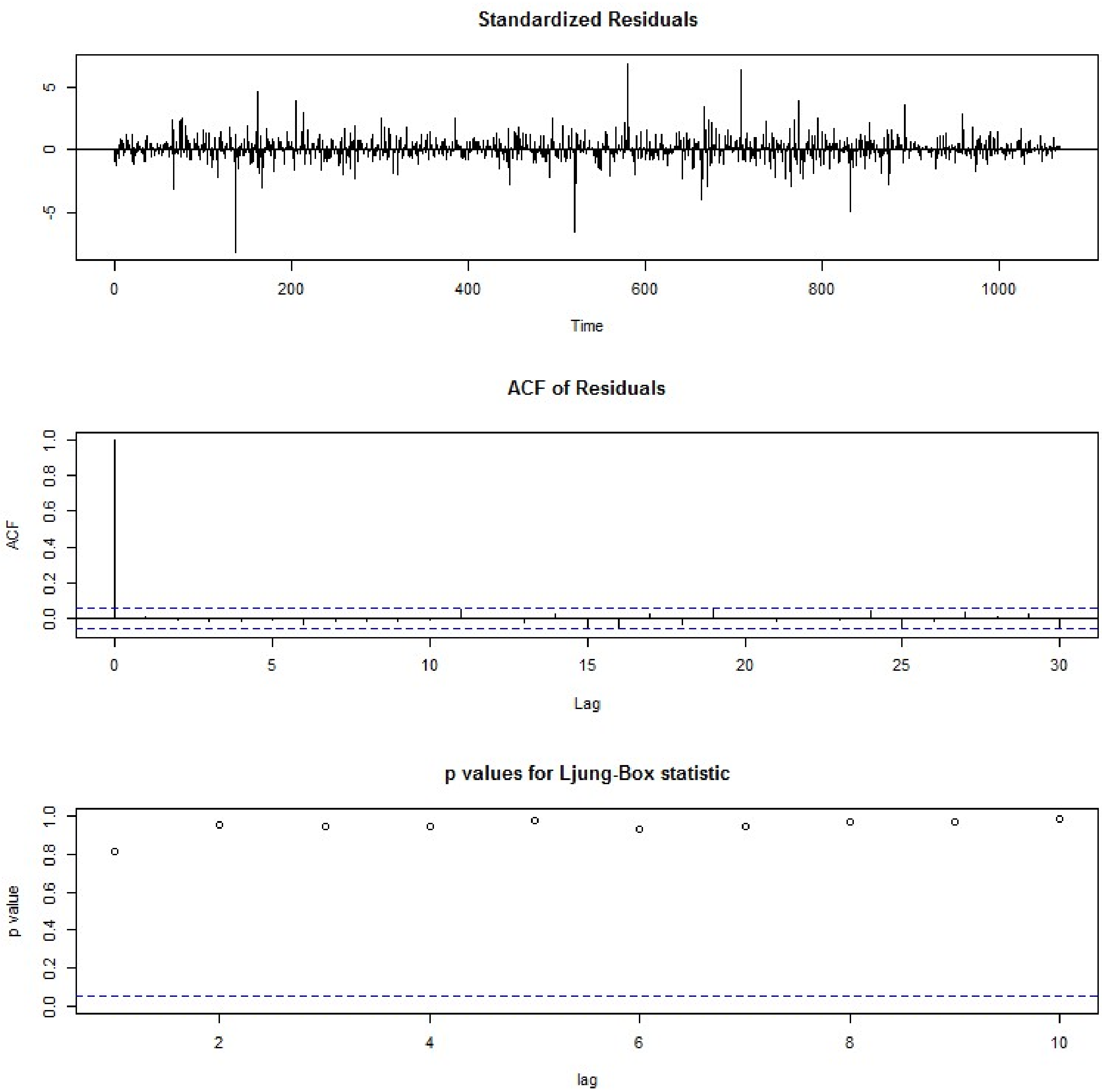
```
arm_1=arima(na.omit(MSFT.ret),order=c(6,0,19),include.mean=F)
round((1-pnorm(abs(arm_1$coef)/sqrt(diag(arm_1$var.coef))))*2, 4)
tsdiag(arm_1)
summary(arm_1)
```

Seasonal

```
Call:
arima(x = MSFT.ret, order = c(2, 0, 2), seasonal = list(order = c(1, 0, 1),
    period = 10), include.mean = F)
```

```
Coefficients:
         ar1      ar2      ma1     ma2    sar1     sma1
      1.8637  -0.9660  -1.8720  0.9654  0.2724  -0.2271
s.e.  0.0237   0.0236   0.0234  0.0246  0.6098   0.6195

sigma^2 estimated as 0.0002135:  log likelihood = 3000.49,  aic = -5986.98
```

I just tried a seasonal ARMA(2,2) model



The plot of the residuals show a white noise pattern. The ACF do not show any significant lags the box statistics do not show any serial correalations

	RMSE	AIC	LOG LIKELIHOOD	Variance
ARMA(6,19)	0.01436	-5981	3017	0.000206
Seasonal		-5986.98	3000.49	0.0002135

It seems the ARMA(6,19) model is superior in AIC, LOG LIKELIHOOD and Variance so we will keep this model.

(g) Using the model you selected in part f), compute forecasts for the daily returns for April 2017 (trading days) as well as 95% confidence intervals for the forecast.

MSFT

```
Point Forecast      Lo 95       Hi 95
1070   0.0009212633 -0.02724887 0.02909140
1071   0.0017497015 -0.02642044 0.02991984
1072  -0.0024850457 -0.03065804 0.02568794
1073   0.0008541687 -0.02733473 0.02904306
1074  -0.0002364878 -0.02843430 0.02796132
1075  -0.0023433984 -0.03055332 0.02586652
1076   0.0003076460 -0.02791797 0.02853326
1077  -0.0025393414 -0.03077469 0.02569601
1078   0.0006511352 -0.02759225 0.02889452
1079   0.0013826588 -0.02687682 0.02964214
1080  -0.0010678280 -0.02934132 0.02720567
1081   0.0026316831 -0.02570921 0.03097258
1082   0.0003758436 -0.02796517 0.02871686
1083   0.0022784476 -0.02606666 0.03062355
1084   0.0008041982 -0.02756929 0.02917769
1085  -0.0008125894 -0.02918672 0.02756154
1086   0.0008840751 -0.02749403 0.02926218
1087  -0.0014670164 -0.02986804 0.02693401
1088  -0.0003991016 -0.02880063 0.02800243
1089  -0.0017580446 -0.03028133 0.02676524
```

The above are the first 20 forecasts for the log returns for MSFT. The first 20 forecasts will include the trading days in April.

```
library(forecast)
forecast(arm_1, h = 20)
```

INTC

```
Point Forecast       Lo 95      Hi 95
1070   4.264599e-03 -0.02256419 0.03109339
1071   6.225925e-04 -0.02622378 0.02746897
1072  -8.921831e-04 -0.02773872 0.02595436
1073  -2.313154e-04 -0.02707786 0.02661522
```

```
1074   6.300368e-04 -0.02621690 0.02747697
1075  -2.143926e-03 -0.02899195 0.02470410
1076   1.212932e-03 -0.02563689 0.02806276
1077  -1.182281e-04 -0.02697224 0.02673578
1078  -2.909633e-04 -0.02715673 0.02657480
1079  -1.339508e-03 -0.02821479 0.02553577
1080  -8.647510e-04 -0.02774810 0.02601859
1081   2.742982e-04 -0.02660937 0.02715796
1082   2.453554e-04 -0.02663926 0.02712997
1083  -7.378993e-04 -0.02764585 0.02617005
1084   1.514287e-05 -0.02689316 0.02692345
1085   6.175755e-04 -0.02645044 0.02768559
1086   7.417854e-04 -0.02634722 0.02783080
1087   7.764387e-04 -0.02632945 0.02788233
1088  -2.969797e-04 -0.02748017 0.02688621
1089   2.349876e-04 -0.02694826 0.02741823
```

The above it INTC first 20 forecasts for the month of April and the 95% confidence interval.

(h) Are there ARCH effect in the log return series? Why or why not?

INTC

```
ARCH heteroscedasticity test for residuals
alternative: heteroscedastic

Portmanteau-Q test:
     order    PQ p.value
       4  4.48   0.345
       8  5.04   0.754
       12  9.32   0.676
       16 13.52   0.635
       20 13.76   0.842
       24 23.63   0.483
```

I ran an arch test on the squared residuals of the ARMA(15,15) model. The null hypothesis is that the squared residuals are a sequence of white noise. No p-values are significant in the different orders, so we can assume that there are no arch effects in the data.

```
arch.test(arma_15_15)
```

MSFT

```
ARCH heteroscedasticity test for residuals
alternative: heteroscedastic

Portmanteau-Q test:
     order   PQ p.value
[1,]     4 5.02   0.285
[2,]     8 5.54   0.698
[3,]    12 5.66   0.932
[4,]    16 6.91   0.975
```

```
[5,]    20 7.91   0.992
[6,]    24 8.57   0.998
```

We do not see any significant p-values in the MSFT arch test so we can assume that there are no arch effects in the data.

```
arch.test(arm_1)
```

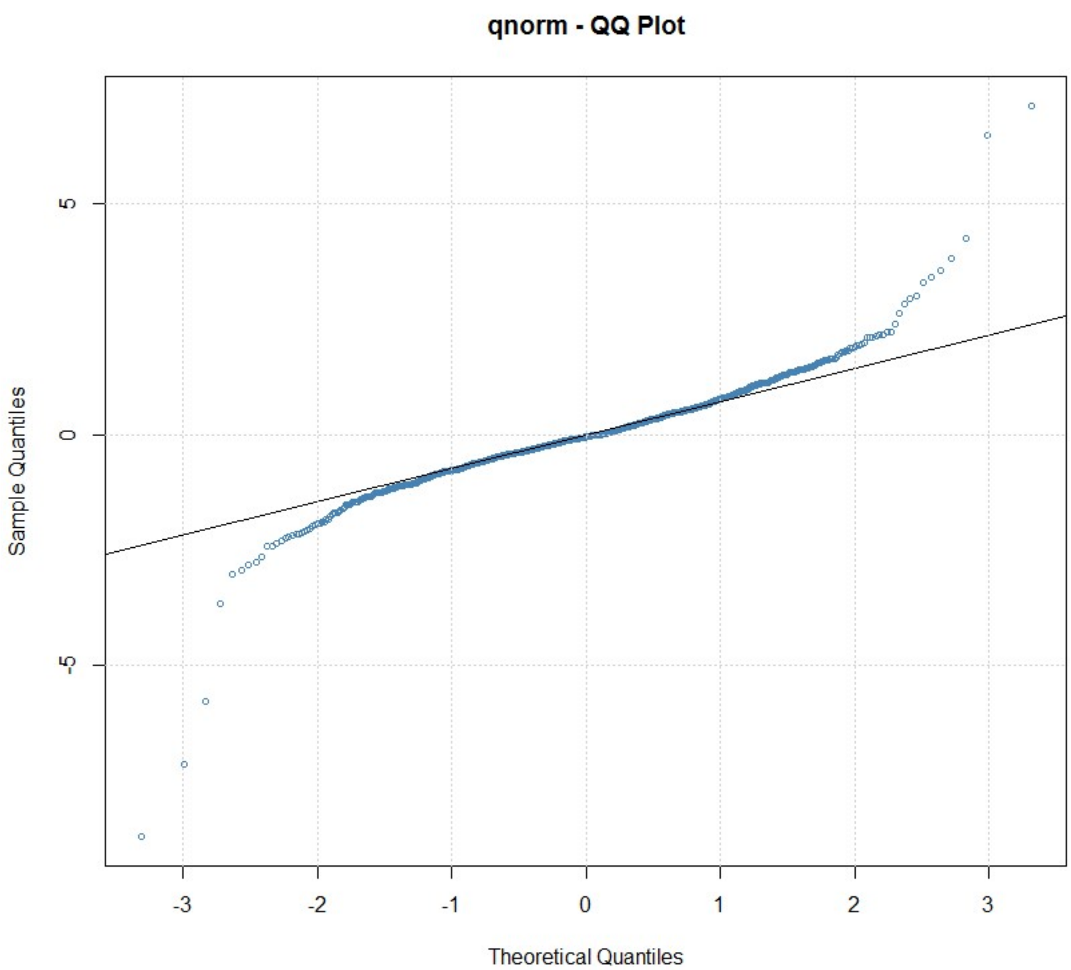
(i) Fit a Gaussian ARMA-GARCH model to each of the log return series. Obtain the normal QQ-plot of the standardized residuals, and write down the fitted model. Is the model adequate? Why or why not?

MSFT

```
Title:
 GARCH Modelling 

Call:
 garchFit(formula = ~arma(0, 1) + garch(1, 1), data = MSFT.ret, 
    trace = F) 

Mean and Variance Equation:
 data ~ arma(0, 1) + garch(1, 1)
<environment: 0x0000000005f46c18>
 [data = MSFT.ret]

Conditional Distribution:
 norm 

Coefficient(s):
        mu         ma1       omega      alpha1       beta1  
1.0767e-03  4.7181e-03  6.0182e-05  2.1158e-01  5.5348e-01  

Std. Errors:
 based on Hessian 

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)    
mu     1.077e-03   4.191e-04    2.569  0.01020 *  
ma1    4.718e-03   3.795e-02    0.124  0.90106    
omega  6.018e-05   1.504e-05    4.000 6.32e-05 ***
alpha1 2.116e-01   7.067e-02    2.994  0.00275 ** 
beta1  5.535e-01   7.810e-02    7.087 1.37e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 3012.474    normalized:  2.81803 

Description:
 Tue Jun 06 20:34:20 2017 by user: david 


Standardised Residuals Tests:
                                Statistic p-Value
```

```
 Ljung-Box Test     R    Q(10)  14.04986  0.1707294
 Ljung-Box Test     R    Q(15)  21.99311  0.1079848
 Ljung-Box Test     R    Q(20)  31.14708  0.05327847
 Ljung-Box Test     R^2  Q(10)  1.995724  0.9963728
 Ljung-Box Test     R^2  Q(15)  3.05468   0.9995502
 Ljung-Box Test     R^2  Q(20)  3.539286  0.9999832
 LM Arch Test       R    TR^2   2.049761  0.9993257

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.626704 -5.603438 -5.626748 -5.617890
```

The box tests of our model are not significant meaning there are no serial correlations in our model, which is what we want.



The QQ plot for the MSFT ARMA-GARCH model is not normal so this model is not adequate.

The model is

$$r_t = 1.077e\text{-}03 + a_t - 4.718e\text{-}03a_{t-1}$$
$$\sigma 2 = 6.018e\text{-}05 + 2.116e\text{-}01a^2_{t-1} + 5.535e\text{-}01\sigma^2_{t-1}$$
$$a_t = \sigma_t \epsilon_t$$
$$\epsilon \sim N(0,1)$$

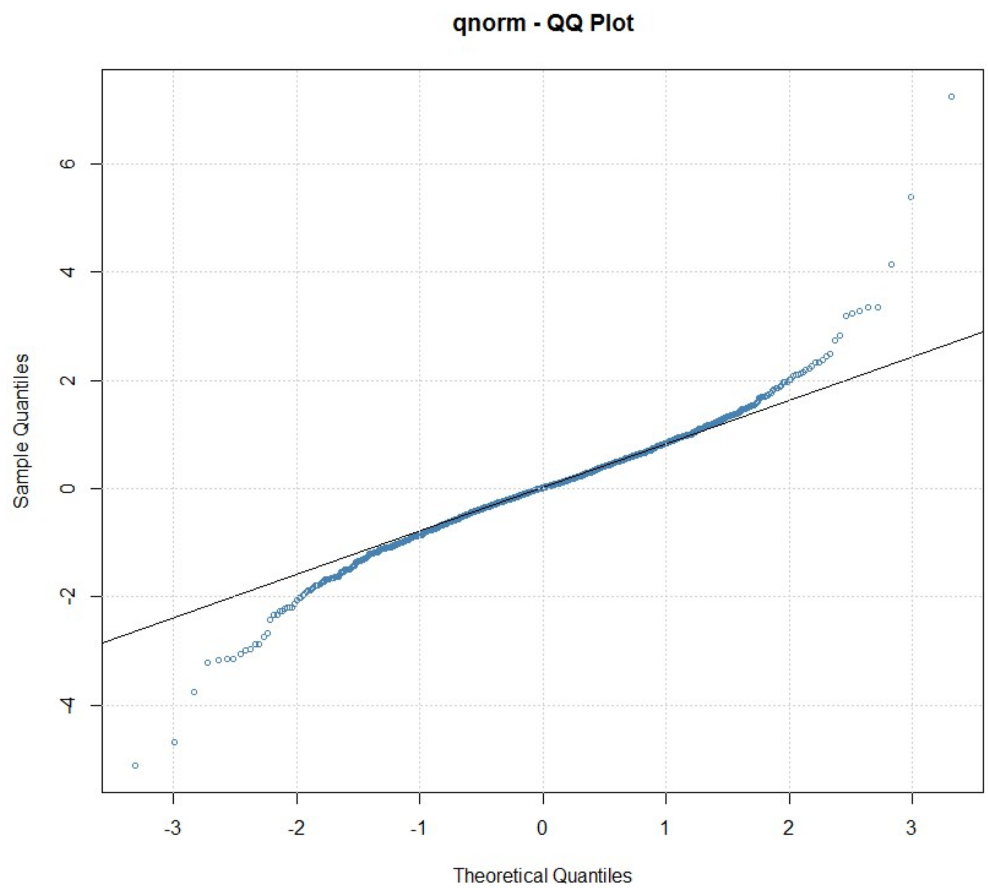
```
library(fGarch)
gar_MSFT <- garchFit(~arma(0,1) + garch(1,1), data=MSFT.ret, trace=F)
plot(gar_MSFT)
```

INTC

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(0, 1) + garch(1, 1), data = INTC.ret,
    trace = F)

Mean and Variance Equation:
 data ~ arma(0, 1) + garch(1, 1)
<environment: 0x000000001f04a8e8>
 [data = INTC.ret]

Conditional Distribution:
 norm

Coefficient(s):
         mu          ma1        omega       alpha1        beta1
 0.00059331  -0.01240022   0.00009054   0.25689751   0.31463728

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      5.933e-04   3.868e-04    1.534  0.12507
ma1    -1.240e-02   3.824e-02   -0.324  0.74575
omega   9.054e-05   1.625e-05    5.571 2.53e-08 ***
alpha1  2.569e-01   5.784e-02    4.441 8.94e-06 ***
beta1   3.146e-01   9.573e-02    3.287  0.00101 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 3073.819    normalized:  2.875415

Description:
 Tue Jun 06 21:40:45 2017 by user: david
```

```
Standardised Residuals Tests:
                                Statistic p-Value

 Ljung-Box Test     R    Q(10)  3.305554  0.973292
 Ljung-Box Test     R    Q(15)  14.04086  0.5224321
 Ljung-Box Test     R    Q(20)  25.50946  0.1826307
 Ljung-Box Test     R^2  Q(10)  2.809723  0.9855539
 Ljung-Box Test     R^2  Q(15)  8.591831  0.8978747
 Ljung-Box Test     R^2  Q(20)  10.13472  0.9656627
 LM Arch Test       R    TR^2   4.213223  0.9792741

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.741476 -5.718209 -5.741520 -5.732662
```

The box tests of the residuals are not significant meaning there are no serial correlations.



The QQ Plot is closer to normal bit still inadequate.

The fitted model is

$$r_t = 5.933e\text{-}04 + a_t - 1.240e\text{-}02a_{t-1}$$
$$\sigma 2 = 9.054e\text{-}05 + 2.569e\text{-}01a^2_{t-1} + 3.146e\text{-}01\sigma^2_{t-1}$$
$$a_t = \sigma_t \, \epsilon_t$$

$$\in \sim N(0,1)$$

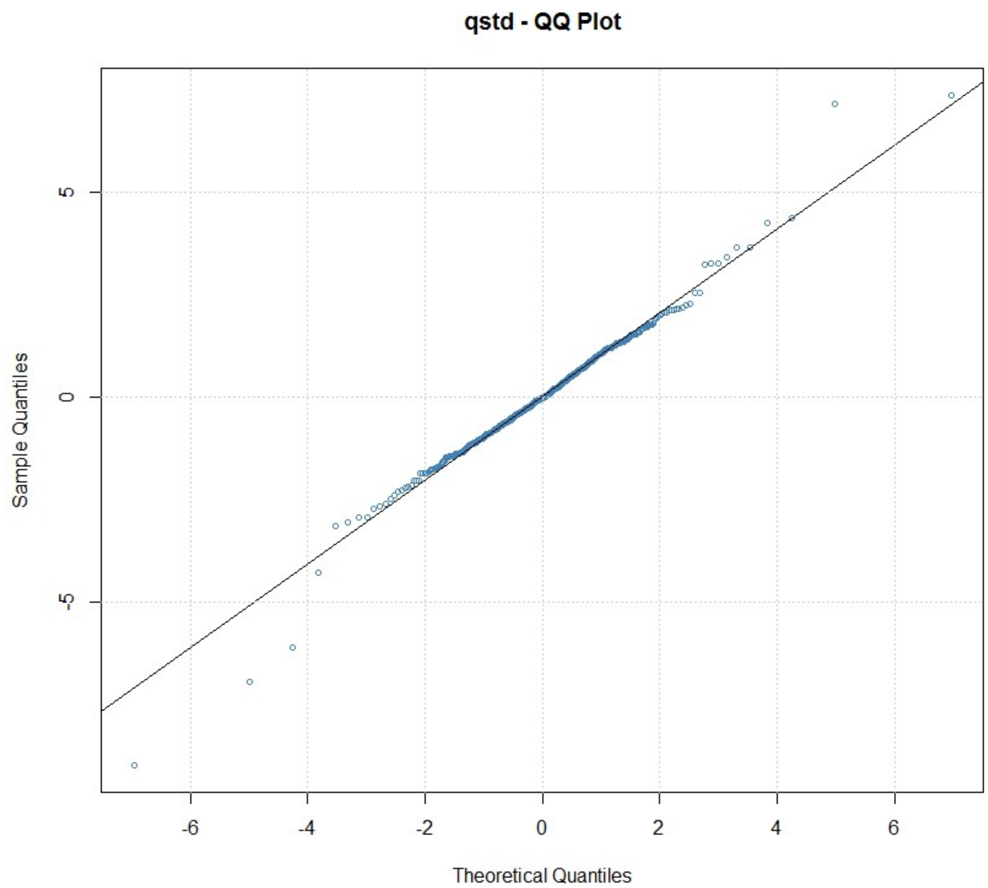
```
gar_INTC <- garchFit(~arma(0,1) + garch(1,1), data=INTC.ret, trace=F)
summary(gar_INTC)
plot(gar_INTC)
```

(j) Build an ARMA-GARCH model with Student-t innovations for the log return series. Perform model checking and write down the fitted model. Is this model better or worse than part (i)?

MSFT

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(0, 1) + garch(1, 1), data = MSFT.ret,
    cond.dist = "std", trace = F)

Conditional Distribution:
 std

Coefficient(s):
         mu           ma1         omega        alpha1         beta1         shape
 7.1159e-04  -4.2796e-02   1.0449e-05   7.1732e-02   8.8767e-01   3.4022e+00


Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      7.116e-04   3.141e-04    2.266   0.0235 *
ma1    -4.280e-02   2.832e-02   -1.511   0.1308
omega   1.045e-05   7.728e-06    1.352   0.1763
alpha1  7.173e-02   3.140e-02    2.285   0.0223 *
beta1   8.877e-01   5.683e-02   15.618   <2e-16 ***
shape   3.402e+00   3.786e-01    8.986   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 3156.382    normalized:  2.952649


Standardised Residuals Tests:
                                Statistic p-Value
Ljung-Box Test     R    Q(10)  14.42344  0.1545383
 Ljung-Box Test     R    Q(15)  22.18724  0.1029876
 Ljung-Box Test     R    Q(20)  31.51835  0.0487088
 Ljung-Box Test     R^2  Q(10)  2.148691  0.9950682
 Ljung-Box Test     R^2  Q(15)  3.131029  0.9994763
 Ljung-Box Test     R^2  Q(20)  3.876868  0.9999641
 LM Arch Test       R    TR^2   2.425862  0.9984173

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.894073 -5.866152 -5.894135 -5.883496
```

One of the Ljung-box is close to being significant, which means the mean equation might not be adequate .



This model is adequate as the QQ plot of the residuals follow a normal distribution.

The model is:

$$r_t = 7.116\text{e-}04 + a_t - 4.280\text{e-}02 a_{t-1}$$
$$\sigma^2 = 1.045\text{e-}05 + 7.173\text{e-}02 a^2_{t-1} + 8.877\text{e-}01 \sigma^2_{t-1}$$
$$at = \sigma t \in t$$

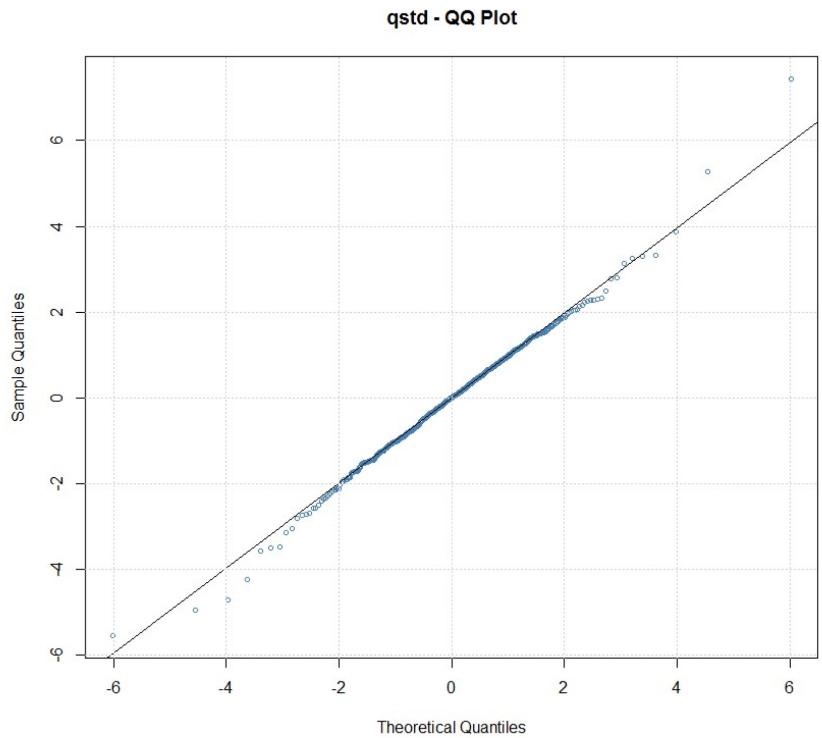
$$\in_t \sim t_{3.402}$$

```
gars_MSFT <-
garchFit(~arma(0,1)+garch(1,1),data=MSFT.ret,trace=F,cond.dist="std")
summary(gars_MSFT)
plot(gars_MSFT)
```

INTC

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(0, 1) + garch(1, 1), data = INTC.ret,
    cond.dist = "std", trace = F)

Mean and Variance Equation:
 data ~ arma(0, 1) + garch(1, 1)
<environment: 0x0000000027cb98f0>
 [data = INTC.ret]

Conditional Distribution:
 std

Coefficient(s):
         mu          ma1        omega       alpha1        beta1        shape
 7.6267e-04  -5.2363e-03   1.7589e-05   6.6831e-02   8.4651e-01   4.1795e+00

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      7.627e-04   3.486e-04    2.188   0.0287 *
ma1    -5.236e-03   3.044e-02   -0.172   0.8634
omega   1.759e-05   1.779e-05    0.989   0.3228
alpha1  6.683e-02   4.934e-02    1.355   0.1756
beta1   8.465e-01   1.303e-01    6.498 8.16e-11 ***
shape   4.180e+00   5.399e-01    7.742 9.77e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 3142.578    normalized:  2.939736

Description:
 Tue Jun 06 21:56:53 2017 by user: david


Standardised Residuals Tests:
                                Statistic p-Value

 Ljung-Box Test     R    Q(10)  3.604437  0.9634327
 Ljung-Box Test     R    Q(15)  13.19808  0.5870015
 Ljung-Box Test     R    Q(20)  23.46903  0.2663549
 Ljung-Box Test     R^2  Q(10)  3.826095  0.9548513
 Ljung-Box Test     R^2  Q(15)  5.830944  0.9824496
 Ljung-Box Test     R^2  Q(20)  7.746061  0.9934189
 LM Arch Test       R    TR^2   3.858051  0.9858637

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.868246 -5.840326 -5.868309 -5.857669
```

Residuals are not significant which is what we want. There are no serial correlations in the residuals.



The residuals show a normal distribution as show by the QQ plot. This model is adequate.

Just the fact that the "J" models' residuals follow a normal distribution means that they are better than the "I" models. We can also compare the statistics.

Series	AIC	LOG LIKELIHOOD	BIC
INTC I	-5.741476	3073.819	-5.718209
INTC J	-5.868246	3142.578	-5.840326
MSFT I	-5.626704	3012.474	-5.603438
MSFT J	-5.894073	3156.382	-5.866152

Looking above, you can see that INTCJ is the better model as the AIC and the BIC is lower than the INTCI. Plus, the log likelihood is larger. The same can be said for MSFTJ respectively.

(k) Obtain 1-step ahead mean and volatility forecasts using the fitted ARMA-GARCH model with Student-t innovations with 95% confidence intervals for April 2017 (trading days). Note that you will need to download April 2017 data to do this forecasting.

I wasn't sure if we were going to forecast the fitted garch model by bringing in April and using the S4 Component to get the fitted values. Instead I did a 10 step forecast based on Aprils data.

INTC

Mean Forecast	Mean Error	Standard Deviation
0.0007198836	0.01149227	0.01149227
0.0007626695	0.01175668	0.01175652
0.0007626695	0.01199295	0.01199279
0.0007626695	0.01220474	0.01220458
0.0007626695	0.01239502	0.01239486
0.0007626695	0.01256630	0.01256613
0.0007626695	0.01272071	0.01272054
0.0007626695	0.01286013	0.01285995
0.0007626695	0.01298615	0.01298598
0.0007626695	0.01310019	0.01310002

MSFT

Mean Forecast	Mean Error	Standard Deviation
0.0006388974	0.01041606	0.01041606
0.0007115915	0.01071155	0.01070227
0.0007115915	0.01097940	0.01096984
0.0007115915	0.01123038	0.01122056
0.0007115915	0.01146600	0.01145594
0.0007115915	0.01168760	0.01167731
0.0007115915	0.01189632	0.01188582
0.0007115915	0.01209318	0.01208248

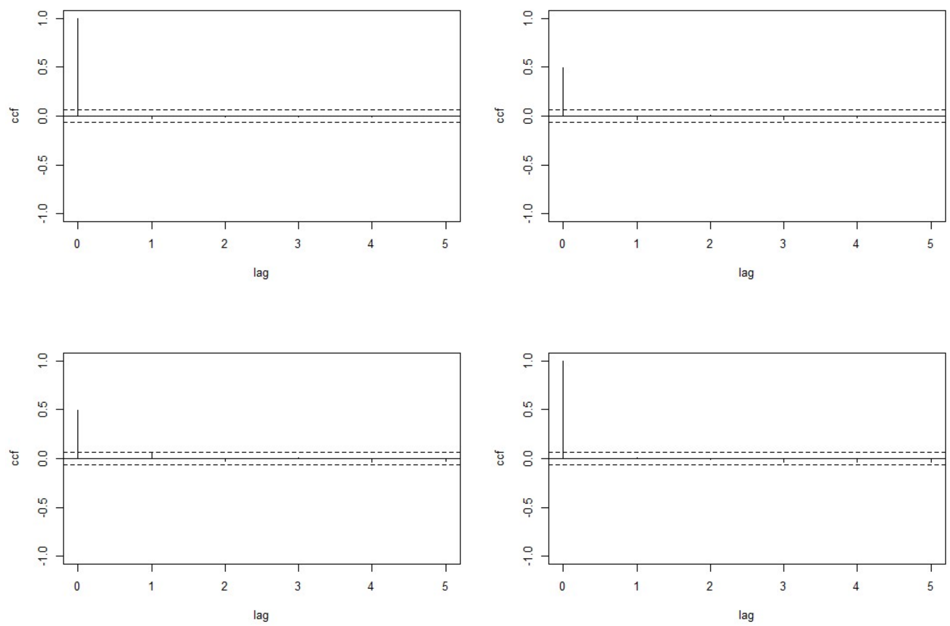
0.0007115915	0.01227909	0.01226819
0.0007115915	0.01245484	0.01244377

```
MSFT_APR <- getSymbols("MSFT", env=NULL, src = "google",  from ="2017-03-
31", to = "2017-04-29")
INTC_APR <- getSymbols("INTC", env=NULL, src = "google",  from ="2017-03-
01", to = "2017-04-29")

MSFT_APR.copy = MSFT_APR[, "MSFT.Close", drop=F]
INTC_APR.copy = INTC_APR[, "INTC.Close", drop=F]

MSFT_APR.copy.ret = CalculateReturns(MSFT_APR.copy, method="log")
INTC_APR.copy.ret = CalculateReturns(INTC_APR.copy, method="log")

MSFT_APR.ret = MSFT_APR.copy.ret[-1,]
INTC_APR.ret = INTC_APR.copy.ret[-1,]

predict(gars_INTC, 10 , data=INTC_APR.ret)
predict(gars_MSFT, 10 , data=IMSFT_APR.ret)
```

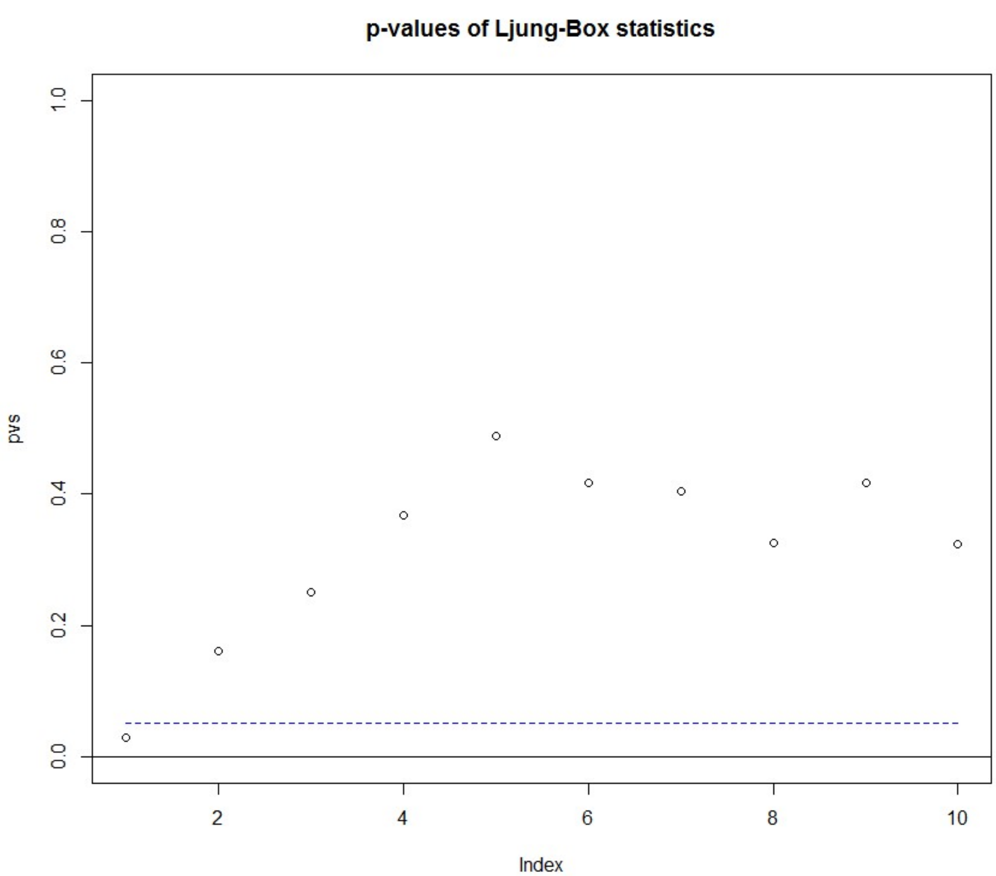
(l) Is there significant cross-correlation in the log returns for the 2 companies?

```
 Q(m)    DF         p-value:
 1.00000 10.733735  0.029725
 2.00000 11.80197   0.16026
 3.00000 14.84100   0.25025
 4.00000 17.28573   0.36734
 5.00000 19.51062   0.48889
 6.00000 24.77359   0.41815
 7.00000 29.15543   0.40468
 8.00000 35.03310   0.32614
 9.00000 37.10132   0.41802
10.00000 43.53528   0.32338
```

We can see from the Ljung-Box statistics, that the p-value is significant at Q(1). This means we can reject the null hypothesis that there are no cross correlations for lag 1.



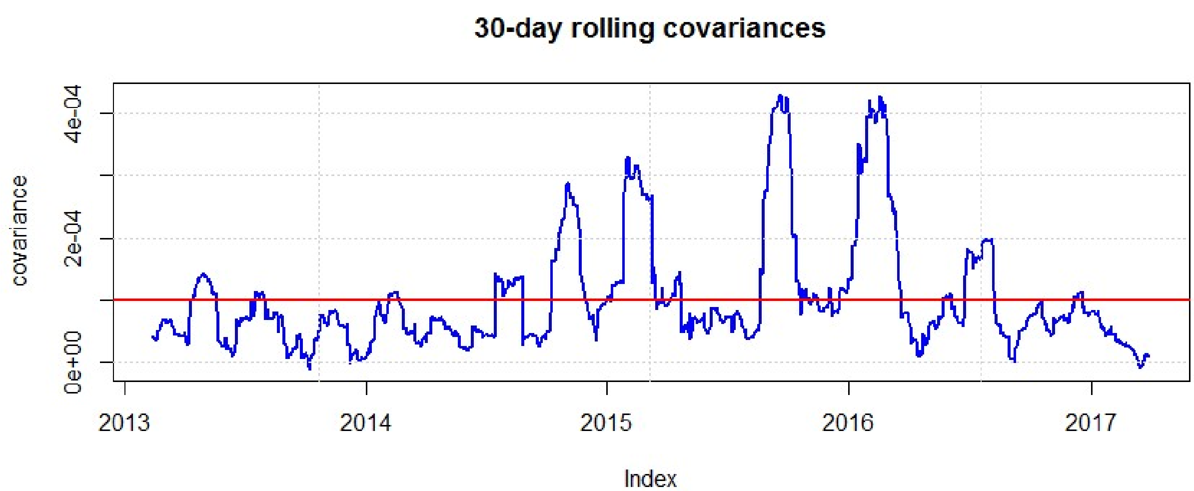
Looking at our ACF plots, we can also see the significant lag 1 in the lower left hand corner.

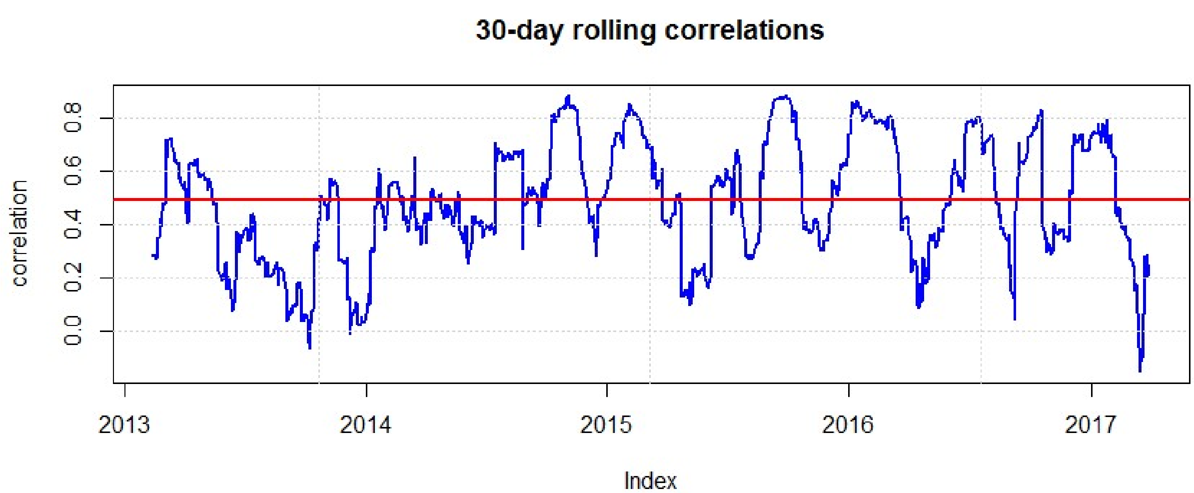


Lastly, in our p-value graph, we can see the significant lag at 1.

```
df<- data.frame(INTC.ret, MSFT.ret)

source("E:/ccm.R")

ccm(df,5)

source("E:/mq.R")

mq(df,10)
```

(m) Using a 30-day moving window, compute and plot rolling covariances and correlations. Briefly comment on what you see.





The 30 day rolling correlations tell us how each asset moves in regard to the other asset. If the correlation coefficient is 1, then both assets move in the same direction. If the correlation coefficient is -1, then as one asset rises, another one falls. Most of our data in the correlation plot is greater than zero, meaning that both assets seem to move in the same direction most of the time. The red line represents the correlation of the whole series. The more values you have in a rolling window, the smoother the plot, which is why we have a straight line. Looking at the plot, it seems the values have the same pattern above and below the line.

Covariance is the measure of the strength of the correlation. These values are not limited to 0 and 1. Covariances measures the expected value of the variations from expected values in the data. We can see that most of the time the covariance is positive, meaning there is some relationship in the data but minimal since the value is very close to zero.. There are large spikes in the data meaning the data is more highly correlated during those spikes.

```
cor.fun = function(x){
  cor(x)[1,2]
}

cov.fun = function(x){
  cov(x)[1,2]
}
roll.cov = rollapply(as.zoo(df), FUN=cov.fun, width=30,
                     by.column=FALSE, align="right")
roll.cor = rollapply(as.zoo(df), FUN=cor.fun, width=30,
                     by.column=FALSE, align="right")
par(mfrow=c(2,1))
plot(roll.cov, main="30-day rolling covariances",
     ylab="covariance", lwd=2, col="blue")
grid()
abline(h=cov(df)[1,2], lwd=2, col="red")

plot(roll.cor, main="30-day rolling correlations",
     ylab="correlation", lwd=2, col="blue")
grid()
abline(h=cor(df)[1,2], lwd=2, col="red")
par(mfrow=c(1,1))
```

(n) Let r_t = (r_INTC,t, r_MSFT,t)^T. Using the dccfit() function from the rmgarch package, estimate the normal-DCC(1,1) model. Briefly comment on the estimated coefficients and the fit of the model.

```
*---------------------------------*
*          DCC GARCH Fit          *
*---------------------------------*

Distribution         :  mvnorm
Model                :  DCC(1,1)
```

```
No. Parameters       :  11
[VAR GARCH DCC UncQ] : [0+8+2+1]
No. Series           :  2
No. Obs.             :  1069
Log-Likelihood       :  6227.013
Av.Log-Likelihood    :  5.83

Optimal Parameters
-----------------------------------
                Estimate  Std. Error  t value Pr(>|t|)
[INTC].mu       0.000588    0.000420   1.3979 0.162138
[INTC].omega    0.000091    0.000026   3.4902 0.000483
[INTC].alpha1   0.259022    0.109041   2.3754 0.017528
[INTC].beta1    0.312571    0.126043   2.4799 0.013143
[MSFT].mu       0.000853    0.000445   1.9167 0.055270
[MSFT].omega    0.000002    0.000002   1.2425 0.214059
[MSFT].alpha1   0.017095    0.004318   3.9589 0.000075
[MSFT].beta1    0.973952    0.003787 257.1625 0.000000
[Joint]dcca1    0.046574    0.022431   2.0764 0.037861
[Joint]dccb1    0.845478    0.068993  12.2546 0.000000
```

With DCC models, we are concerned with alpha1 and beta1 and the dcca1 and the dccb1. The alpha1 and beta1 of INTC is significant meaning that the GARCH(1,1) model is ok for this series. For MSFT, alpha1 and beta1 are also significant, meaning that the GARCH(1,1) model is adequate for this series.

As for if a DCC model makes sense, the dcca1 and dccb2 are jointly significant meaning the DCC model is a good fit.

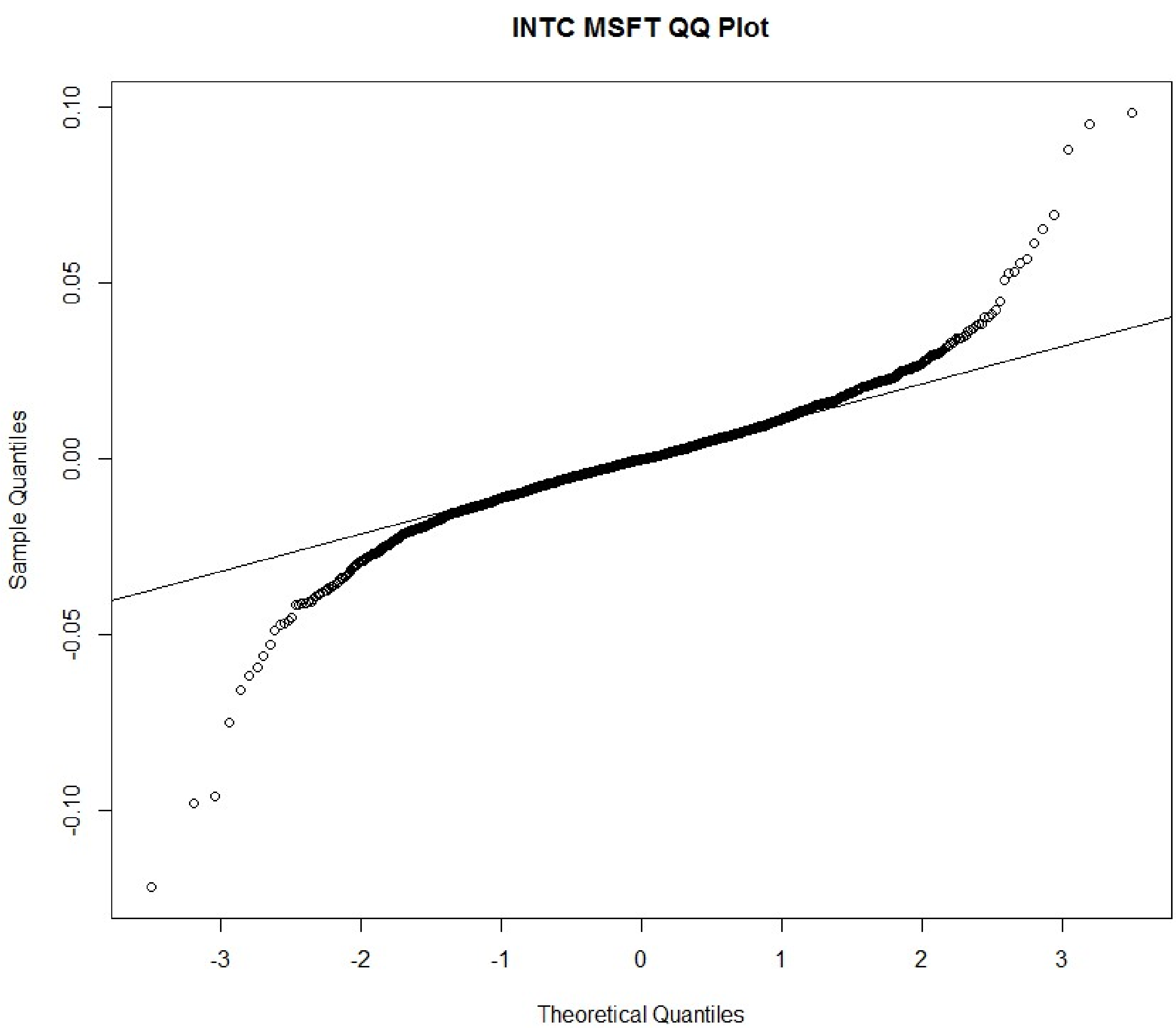
```
garch11.spec = ugarchspec(mean.model = list(armaOrder = c(0,0)),
                          variance.model = list(garchOrder = c(1,1),
                                                model = "sGARCH"),
                          distribution.model = "norm")


dcc.garch11.spec = dccspec(uspec = multispec( replicate(2, garch11.spec) ),
                           dccOrder = c(1,1),
                           distribution = "mvnorm")
dcc.garch11.spec
dcc.fit = dccfit(dcc.garch11.spec, data = df)



dcc.fit
```

```
                    Box-Ljung test

            data:  residuals(dcc.fit)$MSFT
       X-squared = 13.855, df = 10, p-value = 0.1797


                    Box-Ljung test

            data:  residuals(dcc.fit)$INTC
       X-squared = 4.3247, df = 10, p-value = 0.9315
```

The Box-Ljung tests can check for serial correlations in the residuals. The p values for both tests are not significant which means we cannot reject the null hypothesis that there are no serial correlations in the residuals. This is what we want for our model.

```
                    Box-Ljung test

       data:  residuals(dcc.fit)$MSFT^2
X-squared = 4.752, df = 10, p-value = 0.9071


                    Box-Ljung test

       data:  residuals(dcc.fit)$INTC^2
X-squared = 5.7411, df = 10, p-value = 0.8365
```

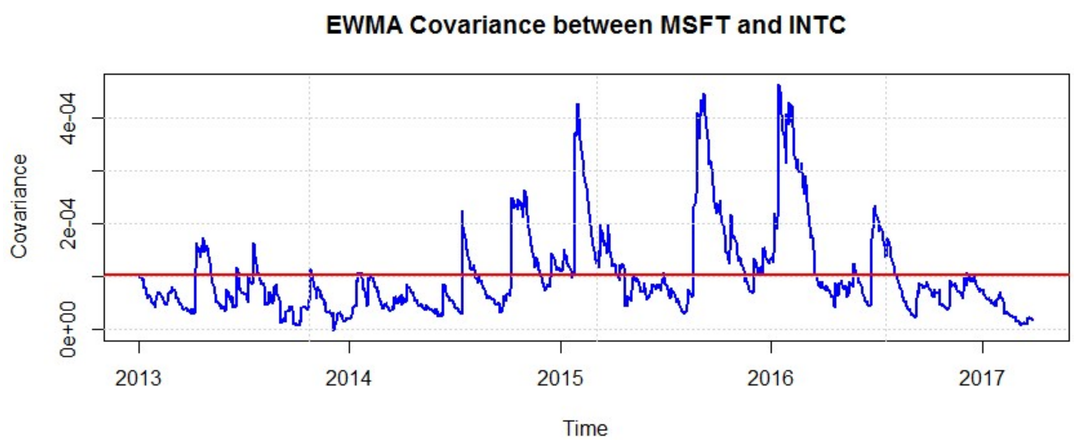
Above is the test for squared residuals. Both p-values are not signficant meaning there are no arch effects in our residuals which is what we want.

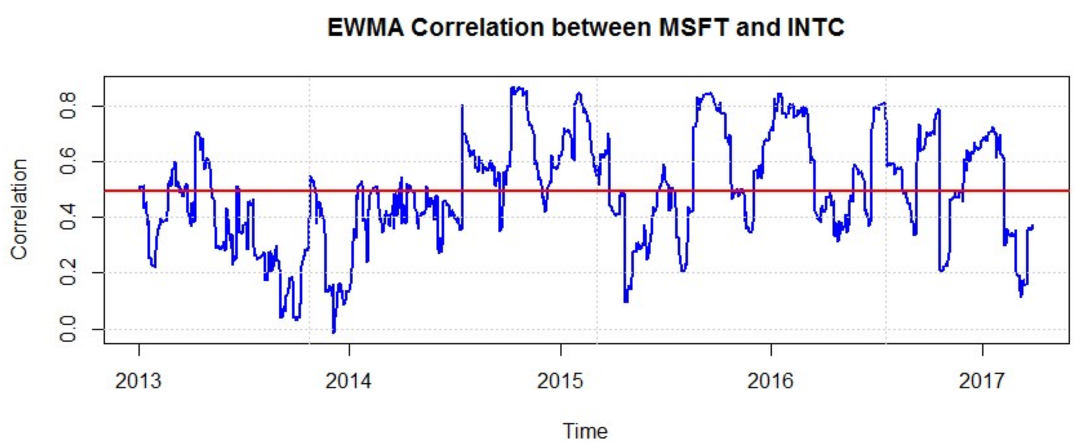


Our qq plot of the residuals are not normal which is the only problem with the fit of the model.

```
Box.test(residuals(dcc.fit)$MSFT^2,lag=10,type='Ljung')
Box.test(residuals(dcc.fit)$INTC^2,lag=10,type='Ljung')
qqnorm(residuals(dcc.fit), main="INTC MSFT QQ Plot")
qqline(residuals(dcc.fit))
```

(o) Plot the estimated in-sample conditional covariances and correlations. Compare the EWMA and rolling estimates.



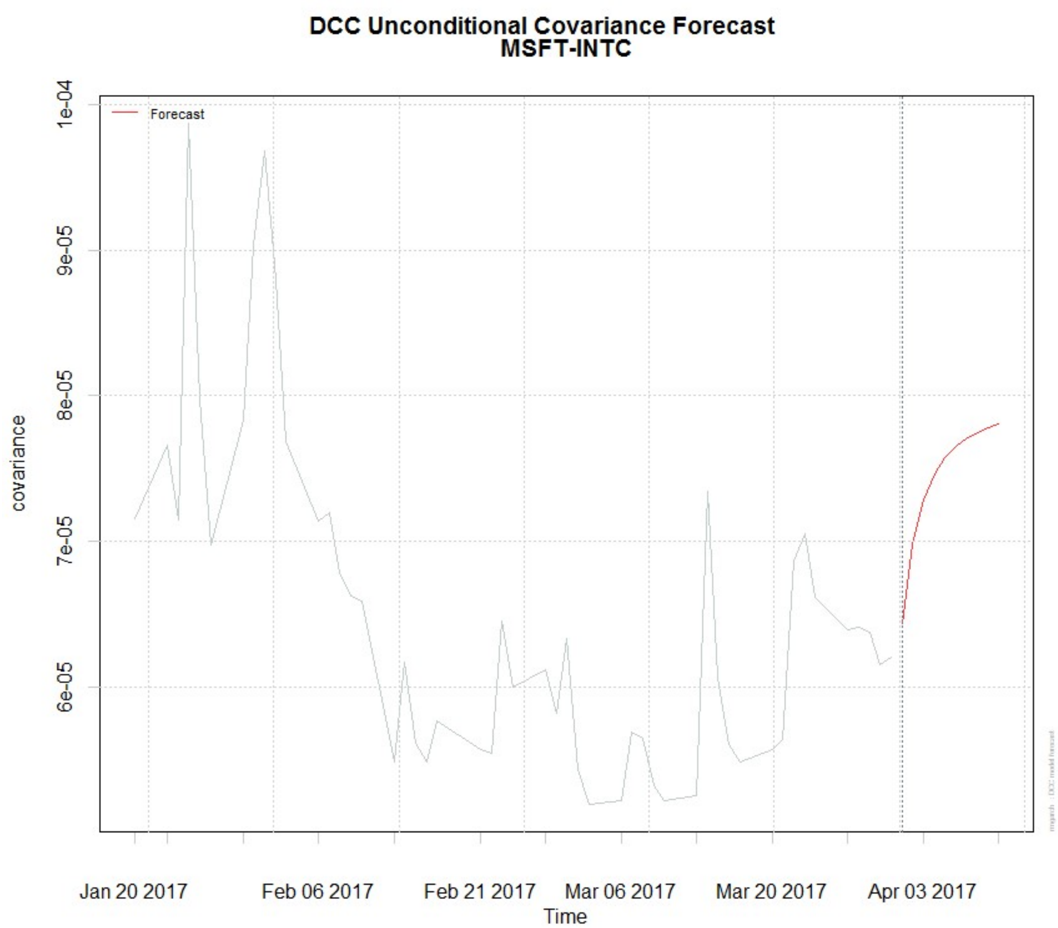


The conditional correlations and the conditional covariances match the 30 day rolling correlations and covariances of our data.

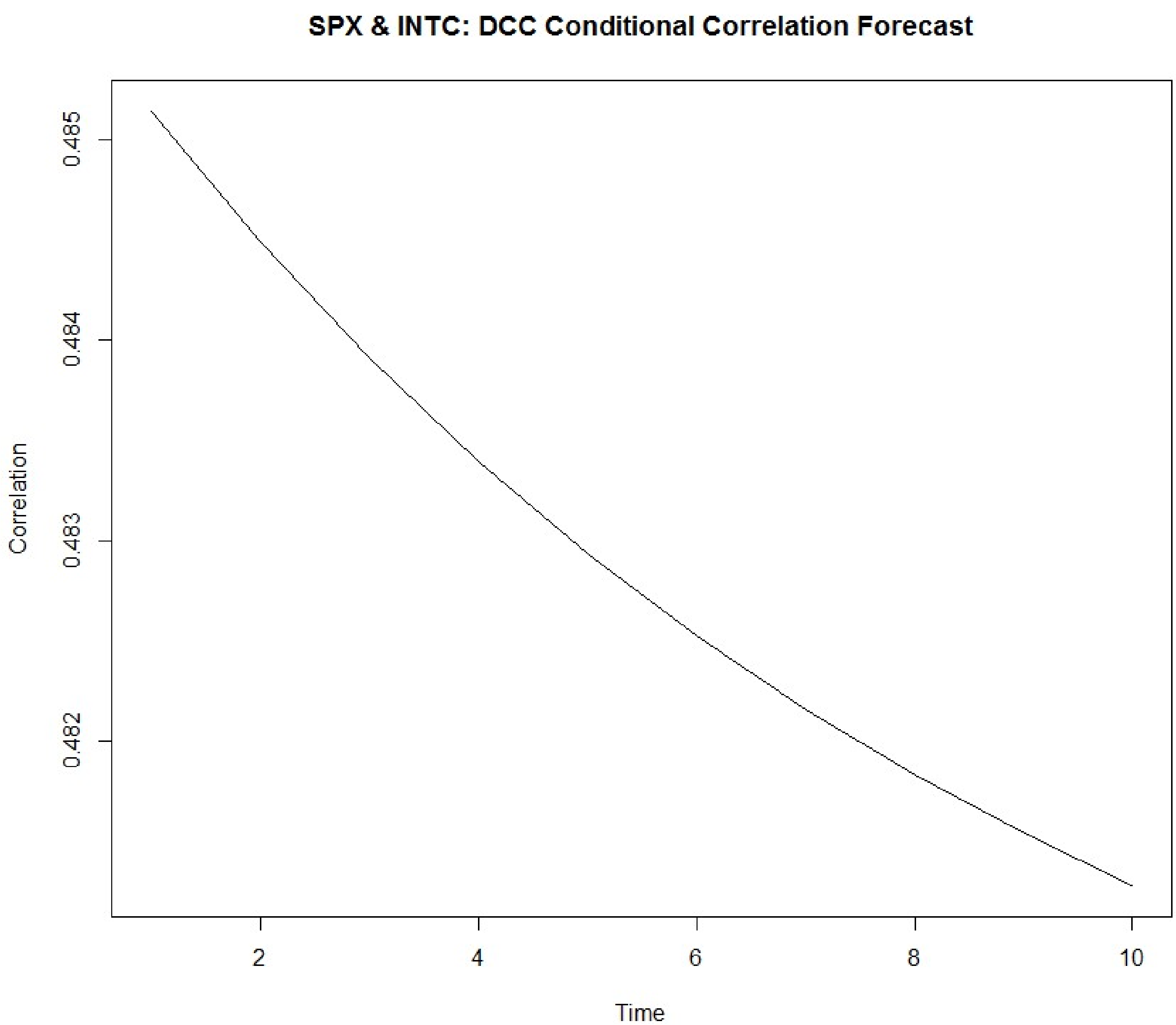
```
source("E:/covEWMA.r")
lambda <- 0.94
cov.ewma <- covEWMA(as.data.frame(df), lambda=lambda)

df.cond.cov <- cov.ewma[,2,1];
```

```
### conditional correlation
t <- length(cov.ewma[,1,1]);
df.cond.cor<- rep(0,t);
for (i in 1:t) {
  df.cond.cor[i]<- cov2cor(cov.ewma[i,,])[1,2];
}

par(mfrow=c(2,1))
plot(x=time(as.zoo(df)), y=df.cond.cov,
     type="l", xlab="Time", ylab="Covariance", lwd=2, col="blue",
     main="EWMA Covariance between MSFT and INTC");
grid()
abline(h=cov(df)[1,2], lwd=2, col="red")
plot(x=time(as.zoo(df)), y=df.cond.cor,
     type="l", xlab="Time", ylab="Correlation", lwd=2, col="blue",
     main="EWMA Correlation between MSFT and INTC");
grid()
abline(h=cor(df)[1,2], lwd=2, col="red")
par(mfrow=c(1,1))
```

(p) Using the Estimated DCC(1,1) model, compute (using dccforecast() function) and plot April 2017's (trading days) 1-step ahead forecasts of conditional covariance and correlation.

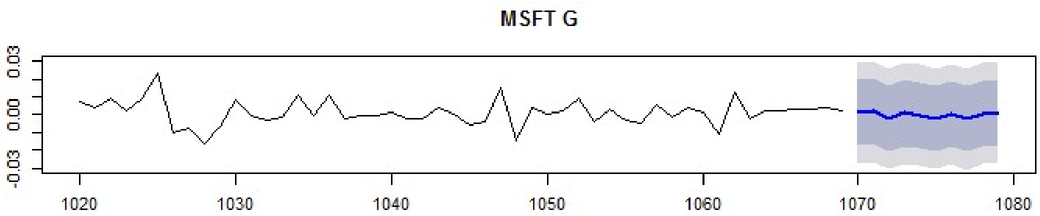


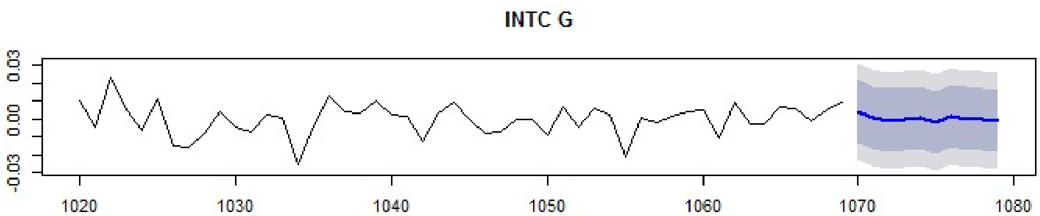
```
dcc.fcst = dccforecast(dcc.fit, n.ahead=10)
class(dcc.fcst)
slotNames(dcc.fcst)
class(dcc.fcst@mforecast)
names(dcc.fcst@mforecast)

dcc.fcst

plot(dcc.fcst)
```

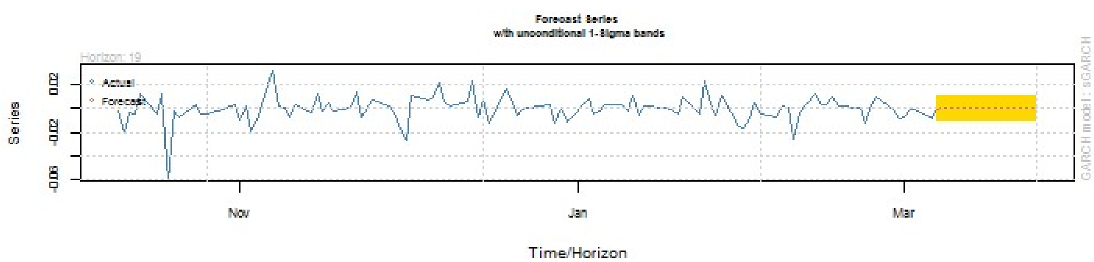


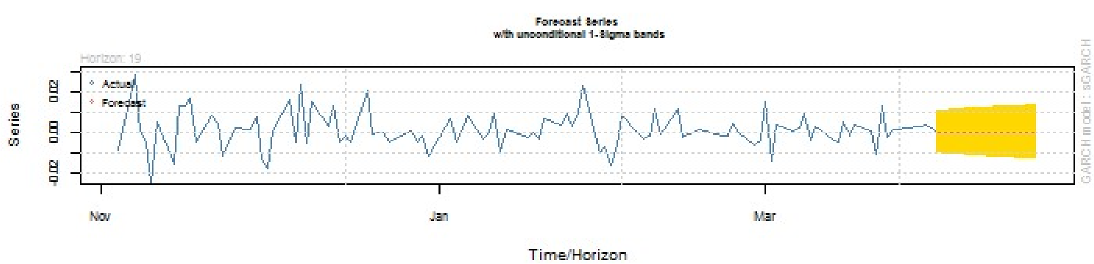

```
ts.plot(rcor(dcc.fcst)[[1]][1,2,], main = "SPX & INTC: DCC Conditional
Correlation Forecast",
        ylab = "Correlation" )
```

(q) Compare your mean and volatility forecast from part (p) with parts (g) and (k).

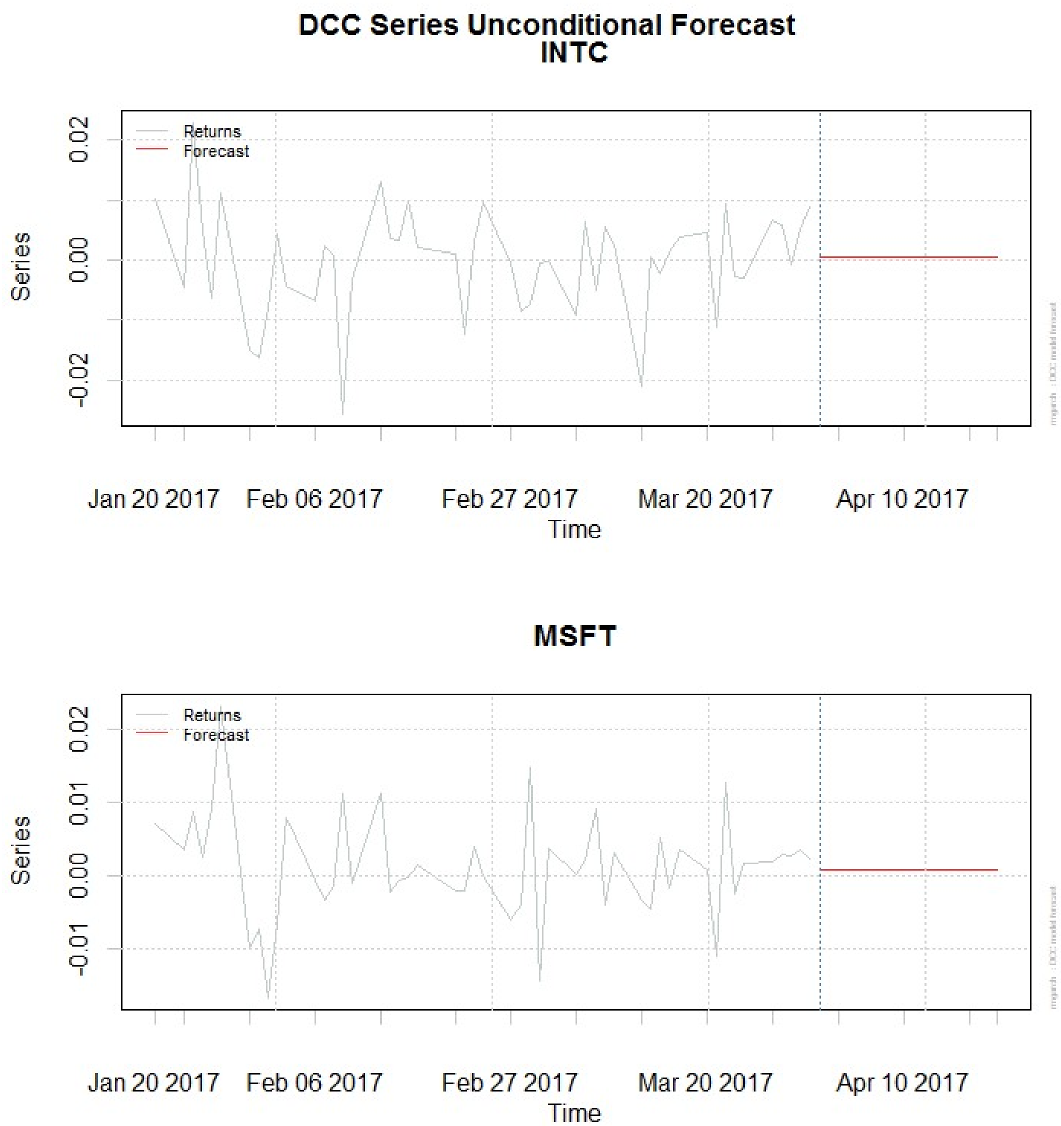








G forecasts are on the top and K forecasts on the bottom. It seems that K forecasts are flatter than the G models. We can also see that K forecasts for returns go to zero which the G forecasts decrease and cycle around zero.

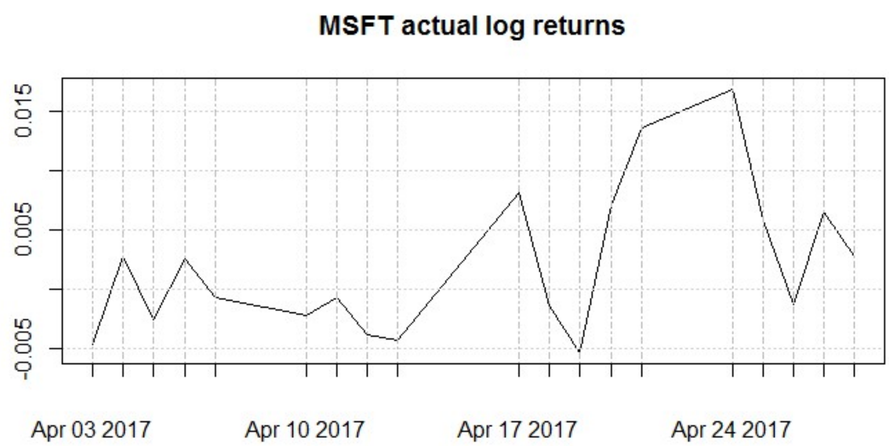


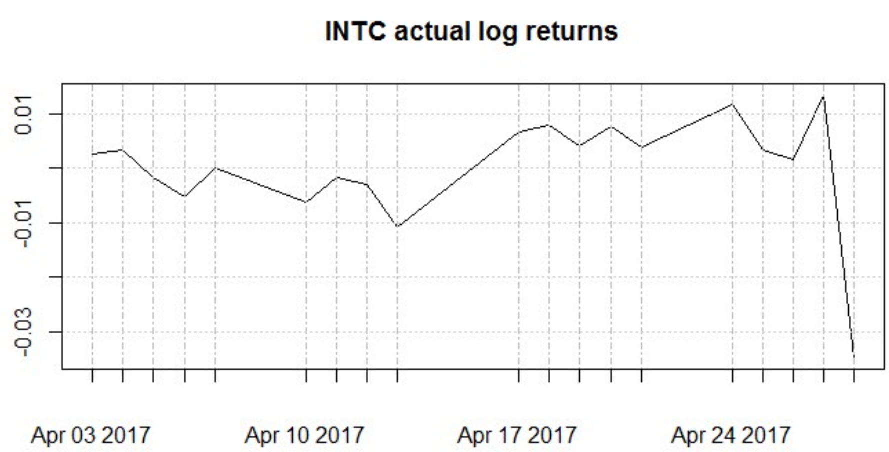
The P forecasts also go to zero, matching the K forecasts.

```
spec <- ugarchspec(variance.model = list(garchOrder = c(1, 1),
                                    model = "sGARCH"),
           mean.model = list(armaOrder = c(0, 1),include.mean = F),
           distribution.model = "std")


INTC_GARCH <- ugarchfit(spec, data = INTC.ret, out.sample =19 )
MSFT_GARCH <- ugarchfit(spec, data = MSFT.ret)

par(mfcol=c(4,1))

plot(forecast::forecast(arm_1), main = "MSFT G", include=50)
abline(h=MSFT_APR.ret, lwd=2, col="red")
```

```
plot(forecast::forecast(arma_15_15), main = "INTC G", include=50)

plot(ugarchforecast(INTC_GARCH, data = null, n.ahead = 19), which=1)
plot(ugarchforecast(MSFT_GARCH, data = null, n.ahead = 19), which = 1)
par(mfcol=c(1,1))

plot(dcc.fcst, which = 1)
```

(r) Compare each of the forecasts in part (g), (p), and (k) to the ACTUAL LOG RETURNS. Do the confidence intervals include the actual values? Which is the best model?





The above graphs depict the actual log returns of the data. We can see that both cycle around zero but there is an upward trend. We can see that the MSFT G model and the INTC G model match the up and down movements of the actual returns so the arima model is the better model.

	INTC.Close	Forecast	LO	HIGH	In Confidence Interval
4/3/2017	0.002492041	0.000622593	-0.02622378	0.02746897	Yes
4/4/2017	0.00331309	-0.000892183	-0.02773872	0.02595436	Yes
4/5/2017	-0.001655173	-0.000231315	-0.02707786	0.02661522	Yes
4/6/2017	-0.005259528	0.000630037	-0.0262169	0.02747697	Yes
4/7/2017	0	-0.002143926	-0.02899195	0.0247041	Yes
4/10/2017	-0.006404031	0.001212932	-0.02563689	0.02806276	Yes
4/11/2017	-0.001677384	-0.000118228	-0.02697224	0.02673578	Yes
4/12/2017	-0.00308253	-0.000290963	-0.02715673	0.0265748	Yes
4/13/2017	-0.01072245	-0.001339508	-0.02821479	0.02553577	Yes
4/17/2017	0.006503628	-0.000864751	-0.0277481	0.02601859	Yes
4/18/2017	0.007860793	0.000274298	-0.02660937	0.02715796	Yes
4/19/2017	0.004185858	0.000245355	-0.02663926	0.02712997	Yes
4/20/2017	0.007490672	-0.000737899	-0.02764585	0.02617005	Yes
4/21/2017	0.003862074	0.000015143	-0.02689316	0.02692345	Yes
4/24/2017	0.011769672	0.000617576	-0.02645044	0.02768559	Yes
4/25/2017	0.003259987	0.000741785	-0.02634722	0.0278308	Yes
4/26/2017	0.001626017	0.000776439	-0.02632945	0.02788233	Yes
4/27/2017	0.013448293	-0.000296980	-0.02748017	0.02688621	Yes
4/28/2017	-0.034795573	0.000234988	-0.02694826	0.02741823	No

We can see in the above table that The G forecasts are in the confidence intervals predicts.

	MSFT.Close	**Forecast**	**LO**	**HIGH**	**In Confidence Interval**
4/3/2017	-0.005	0.002	-0.026	0.030	Yes
4/4/2017	0.003	-0.002	-0.031	0.026	Yes
4/5/2017	-0.003	0.001	-0.027	0.029	Yes
4/6/2017	0.003	0.000	-0.028	0.028	Yes
4/7/2017	-0.001	-0.002	-0.031	0.026	Yes
4/10/2017	-0.002	0.000	-0.028	0.029	Yes
4/11/2017	-0.001	-0.003	-0.031	0.026	Yes
4/12/2017	-0.004	0.001	-0.028	0.029	Yes
4/13/2017	-0.004	0.001	-0.027	0.030	Yes
4/17/2017	0.008	-0.001	-0.029	0.027	Yes
4/18/2017	-0.001	0.003	-0.026	0.031	Yes
4/19/2017	-0.005	0.000	-0.028	0.029	Yes

4/20/2017	0.007	0.002	-0.026	0.031	Yes
4/21/2017	0.014	0.001	-0.028	0.029	Yes
4/24/2017	0.017	-0.001	-0.029	0.028	Yes
4/25/2017	0.006	0.001	-0.027	0.029	Yes
4/26/2017	-0.001	-0.001	-0.030	0.027	Yes
4/27/2017	0.006	0.000	-0.029	0.028	Yes
4/28/2017	0.003	-0.002	-0.030	0.027	Yes

The same is true with MSFT.