R Notebook

```
library(readr)
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: ’zoo’
```

```
## The following objects are masked from ’package:base’:
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by ’quantmod’:
##   method            from
##   as.zoo.data.frame zoo
```

```
library(xts)
library(PerformanceAnalytics)
```

```
##
## Attaching package: ’PerformanceAnalytics’
```

```
## The following object is masked from ’package:graphics’:
##
##     legend
```

```
library(rugarch)
```

```
## Loading required package: parallel
```

```
##
## Attaching package: ’rugarch’
```

```
## The following object is masked from ’package:stats’:
##
##     sigma
```

```
library(FinTS)
library(e1071)
```

```
##
## Attaching package: ’e1071’

## The following objects are masked from ’package:PerformanceAnalytics’:
##
##     kurtosis, skewness
```

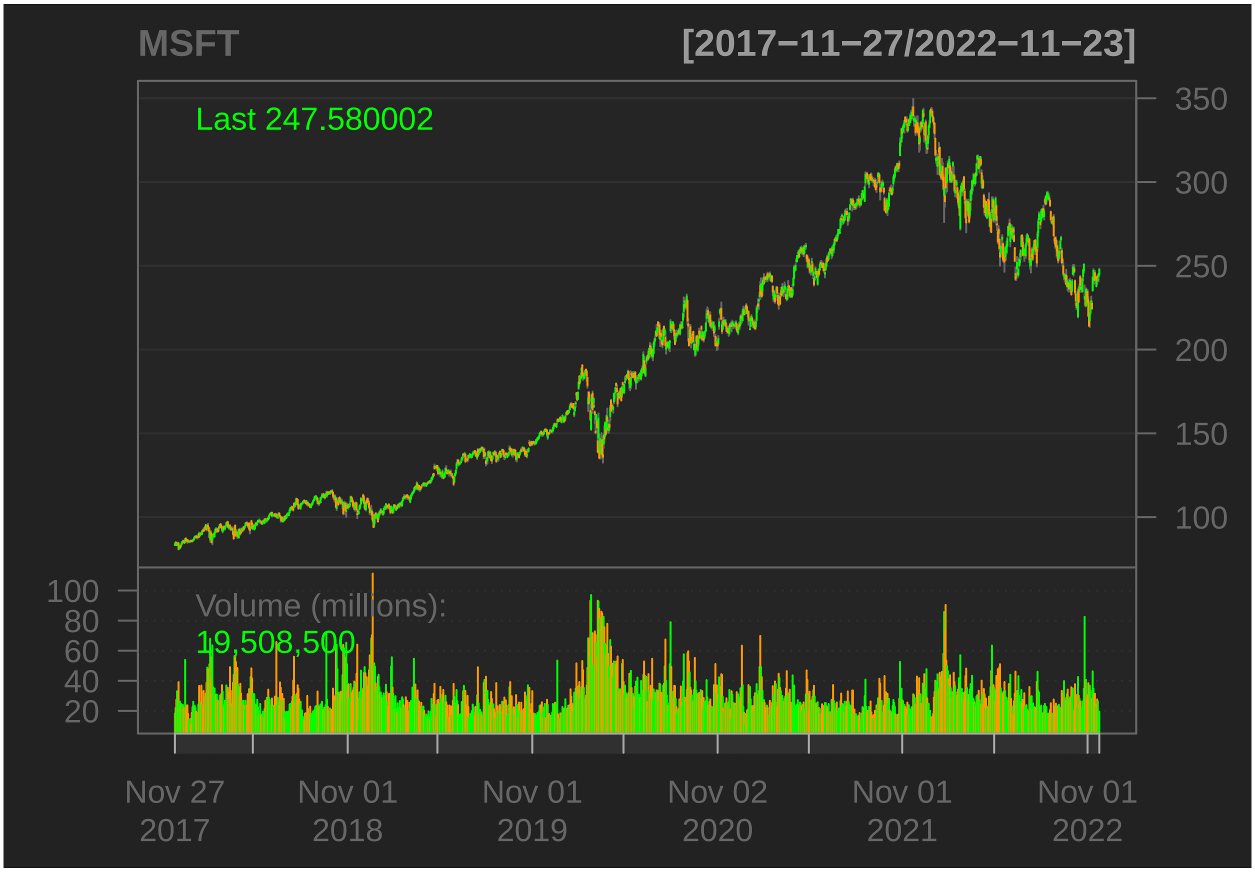
```
library(tseries)
library(rmgarch)
```

```
##
## Attaching package: ’rmgarch’

## The following objects are masked from ’package:xts’:
##
##     first, last
```

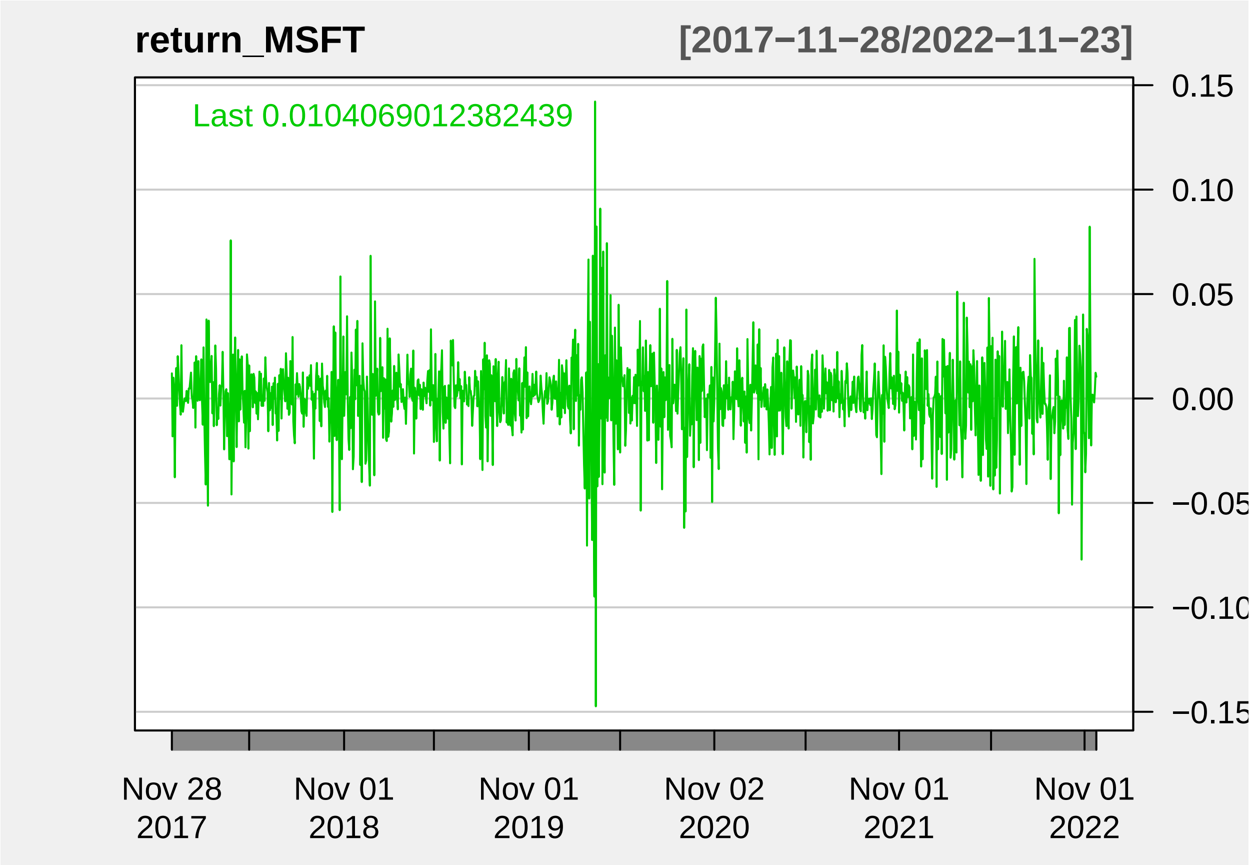
```
startDate = as.Date("2017-11-27")
endDate = as.Date("2022-11-25")
getSymbols("MSFT", from = startDate, to = endDate)
```

```
## [1] "MSFT"
```

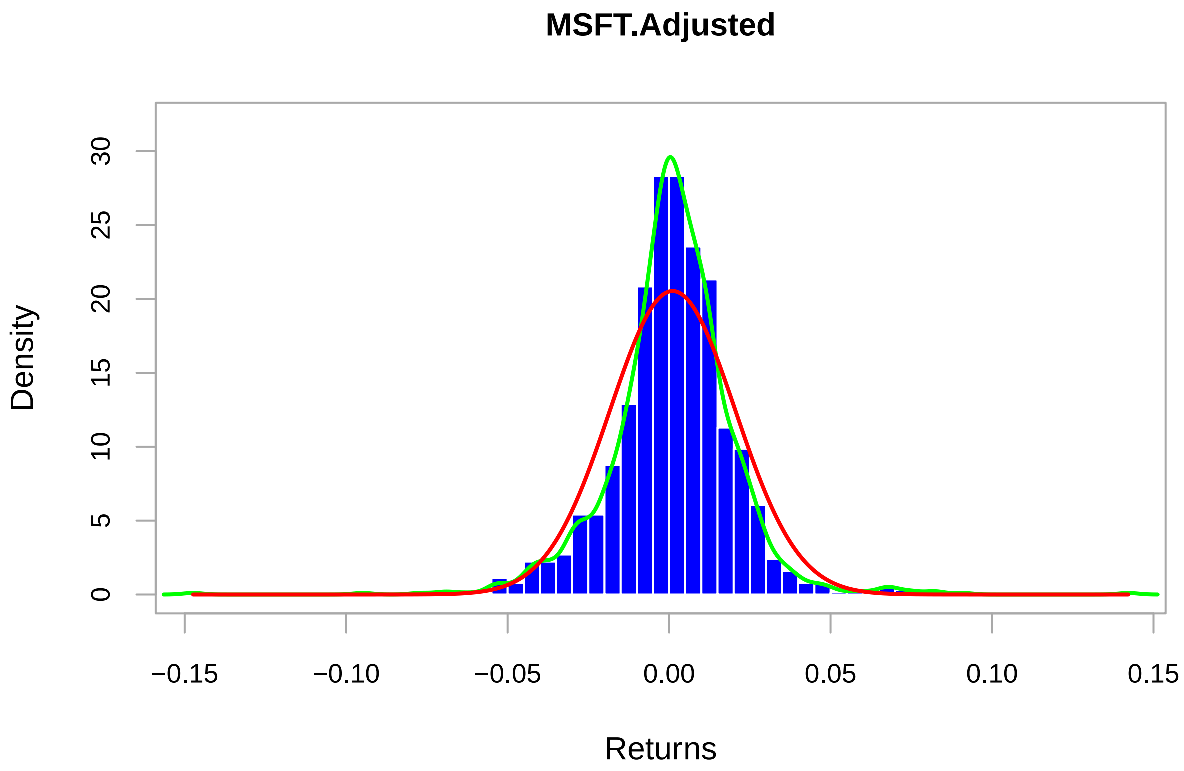
```
chartSeries(MSFT)
```



```
# Daily returns
return_MSFT <- CalculateReturns(MSFT$MSFT.Adjusted)
return_MSFT <- return_MSFT[-1]
chartSeries(return_MSFT, theme = 'white')
```



```
chart.Histogram(return_MSFT,
                methods = c('add.density','add.normal'),
                colorset = c('blue','green','red'))
```



```
# Green line is higher than normal distribution (red line). Hence, student t distribution (heavier tail
```

```
MSFT_garch_1 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                          variance.model = list(model = 'sGARCH',
                                                garchOrder = c(1,1)),
                          distribution = 'std') # standard GARCH model
MSFT_fit_garch_1 <- ugarchfit(spec = MSFT_garch_1,
                              data = na.omit(return_MSFT))
#infocriteria(MSFT_fit_garch_1)
MSFT_fit_garch_1
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,1)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001872    0.000341   5.4883 0.000000
## ma1    -0.103690    0.030868  -3.3591 0.000782
## omega   0.000011    0.000010   1.1465 0.251578
## alpha1  0.169752    0.056318   3.0142 0.002577
## beta1   0.810233    0.033180  24.4191 0.000000
## shape   6.596837    2.213295   2.9806 0.002877
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001872    0.000346  5.41554 0.000000
## ma1    -0.103690    0.028767 -3.60450 0.000313
## omega   0.000011    0.000037  0.29862 0.765228
## alpha1  0.169752    0.206066  0.82378 0.410068
## beta1   0.810233    0.088651  9.13954 0.000000
## shape   6.596837    7.369114  0.89520 0.370680
##
## LogLikelihood : 3400.134
##
## Information Criteria
## ------------------------------------
##
## Akaike       -5.4004
## Bayes        -5.3759
## Shibata      -5.4004
## Hannan-Quinn -5.3912
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.3445  0.5572
## Lag[2*(p+q)+(p+q)-1][2]    0.7964  0.8532
## Lag[4*(p+q)+(p+q)-1][5]    1.4557  0.8573
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.02206  0.8819
## Lag[2*(p+q)+(p+q)-1][5]   0.29463  0.9843
## Lag[4*(p+q)+(p+q)-1][9]   1.60246  0.9468
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1642 0.500 2.000  0.6853
## ARCH Lag[5]    0.5885 1.440 1.667  0.8572
## ARCH Lag[7]    0.9577 2.315 1.543  0.9203
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  10.3331
## Individual Statistics:
```

```
## mu     0.1526
## ma1    0.5581
## omega  1.0918
## alpha1 0.2071
## beta1  0.4326
## shape  0.1418
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.49 1.68 2.12
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias           0.3522 0.7247
## Negative Sign Bias  1.5976 0.1104
## Positive Sign Bias  1.2211 0.2223
## Joint Effect        6.1803 0.1032
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     22.82      0.24514
## 2    30     45.36      0.02712
## 3    40     58.48      0.02322
## 4    50     48.37      0.49858
##
##
## Elapsed time : 0.222996
```

```
infocriteria(MSFT_fit_garch_1)
```

```
##
## Akaike       -5.400372
## Bayes        -5.375854
## Shibata      -5.400417
## Hannan-Quinn -5.391158
```

```
#coef(MSFT_fit_garch_1)
```

```
MSFT_garch_2 <- ugarchspec(mean.model = list(armaOrder = c(0,1)),
                           variance.model = list(model = "sGARCH",
                                                 garchOrder = c(1,2)),
                           distribution = 'std')
MSFT_fit_garch_2 <- ugarchfit(spec = MSFT_garch_2,
                              data = na.omit(return_MSFT))
# MSFT_fit_garch_2
infocriteria(MSFT_fit_garch_2)
```

```
##
## Akaike       -5.398798
## Bayes        -5.370194
## Shibata      -5.398859
## Hannan-Quinn -5.388048
```

```
#coef(MSFT_fit_garch_2)
```

```
MSFT_garch_3 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(1,3)),
                           distribution = 'std')

MSFT_fit_garch_3 <- ugarchfit(spec = MSFT_garch_3, data = na.omit(return_MSFT))
#MSFT_fit_garch_3
infocriteria(MSFT_fit_garch_3)
```

```
##
## Akaike       -5.397252
## Bayes        -5.364562
## Shibata      -5.397332
## Hannan-Quinn -5.384966
```

```
#coef(MSFT_fit_garch_3)
```

```
MSFT_garch_4 <- ugarchspec(mean.model = list(armaOrder=c(0,01)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(2,1)),
                           distribution = 'std')

MSFT_fit_garch_4 <- ugarchfit(spec = MSFT_garch_4, data = na.omit(return_MSFT))
#MSFT_fit_garch_4
infocriteria(MSFT_fit_garch_4)
```

```
##
## Akaike       -5.398796
## Bayes        -5.370192
## Shibata      -5.398858
## Hannan-Quinn -5.388046
```

```
#coef(MSFT_fit_garch_4)
```

```
MSFT_garch_5 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(2,2)),
                           distribution = 'std')

MSFT_fit_garch_5 <- ugarchfit(spec = MSFT_garch_5, data = na.omit(return_MSFT))
#MSFT_fit_garch_5
infocriteria(MSFT_fit_garch_5)
```

```
##
```

```
## Akaike       -5.397207
## Bayes        -5.364516
## Shibata      -5.397287
## Hannan-Quinn -5.384921
```

```
#coef(MSFT_fit_garch_5)
```

```
MSFT_garch_6 <- ugarchspec(mean.model = list(armaOrder=c(0,0)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(1,1)),
                           distribution = 'std')

MSFT_fit_garch_6 <- ugarchfit(spec = MSFT_garch_6, data = na.omit(return_MSFT))
#MSFT_fit_garch_6
infocriteria(MSFT_fit_garch_6)
```

```
##
## Akaike       -5.393106
## Bayes        -5.372674
## Shibata      -5.393137
## Hannan-Quinn -5.385427
```

```
#coef(MSFT_fit_garch_6)
```

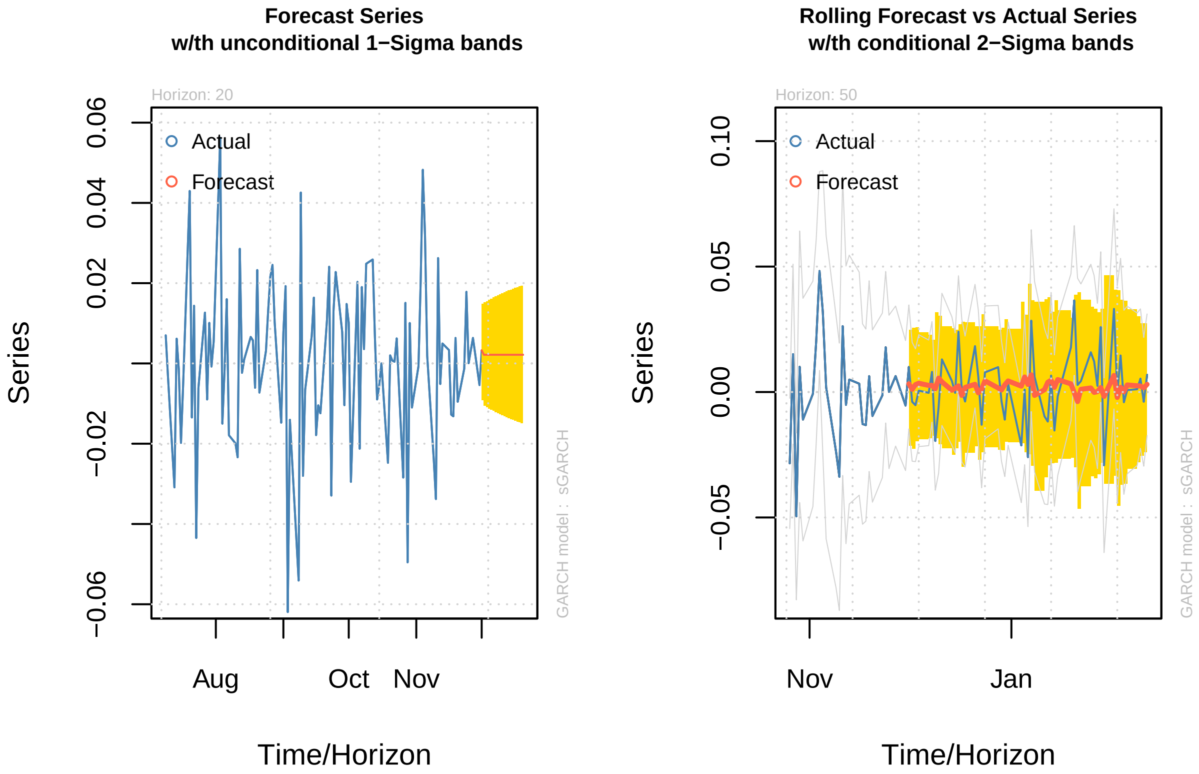
```
MSFT_forecast <- ugarchforecast(MSFT_fit_garch_1,
                                data = na.omit(return_MSFT),
                                n.ahead = 20)
MSFT_forecast
```

```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 20
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=2022-11-23]:
##         Series   Sigma
## T+1  0.0008733 0.01912
## T+2  0.0018718 0.01922
## T+3  0.0018718 0.01932
## T+4  0.0018718 0.01941
## T+5  0.0018718 0.01951
## T+6  0.0018718 0.01960
## T+7  0.0018718 0.01968
## T+8  0.0018718 0.01977
## T+9  0.0018718 0.01985
## T+10 0.0018718 0.01993
## T+11 0.0018718 0.02001
## T+12 0.0018718 0.02009
## T+13 0.0018718 0.02017
## T+14 0.0018718 0.02024
## T+15 0.0018718 0.02031
## T+16 0.0018718 0.02038
## T+17 0.0018718 0.02045
## T+18 0.0018718 0.02052
## T+19 0.0018718 0.02059
## T+20 0.0018718 0.02065
```

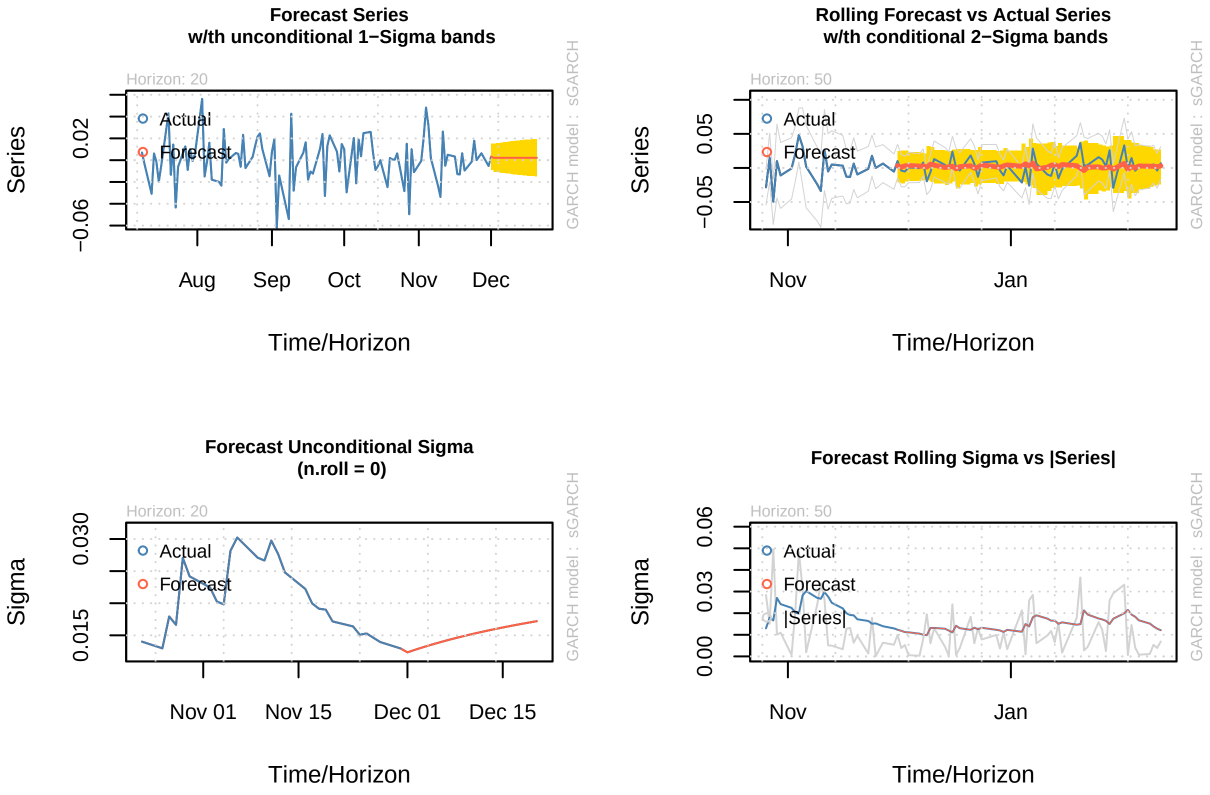
```
MSFT_fit_roll <- ugarchfit(MSFT_garch_1,
                           data = na.omit(return_MSFT),
                           out.sample = 500)
MSFT_fore_roll <- ugarchforecast(MSFT_fit_roll,
                                 n.ahead = 20,
                                 n.roll=50)
MSFT_fore_roll
```

```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 20
## Roll Steps: 50
## Out of Sample: 20
##
## 0-roll forecast [T0=2020-11-30]:
##        Series   Sigma
## T+1  0.003280 0.01235
## T+2  0.002163 0.01271
## T+3  0.002163 0.01305
## T+4  0.002163 0.01337
## T+5  0.002163 0.01369
## T+6  0.002163 0.01399
## T+7  0.002163 0.01427
## T+8  0.002163 0.01455
## T+9  0.002163 0.01481
## T+10 0.002163 0.01507
## T+11 0.002163 0.01532
## T+12 0.002163 0.01555
## T+13 0.002163 0.01579
## T+14 0.002163 0.01601
## T+15 0.002163 0.01622
## T+16 0.002163 0.01643
## T+17 0.002163 0.01663
## T+18 0.002163 0.01683
## T+19 0.002163 0.01702
## T+20 0.002163 0.01720
```

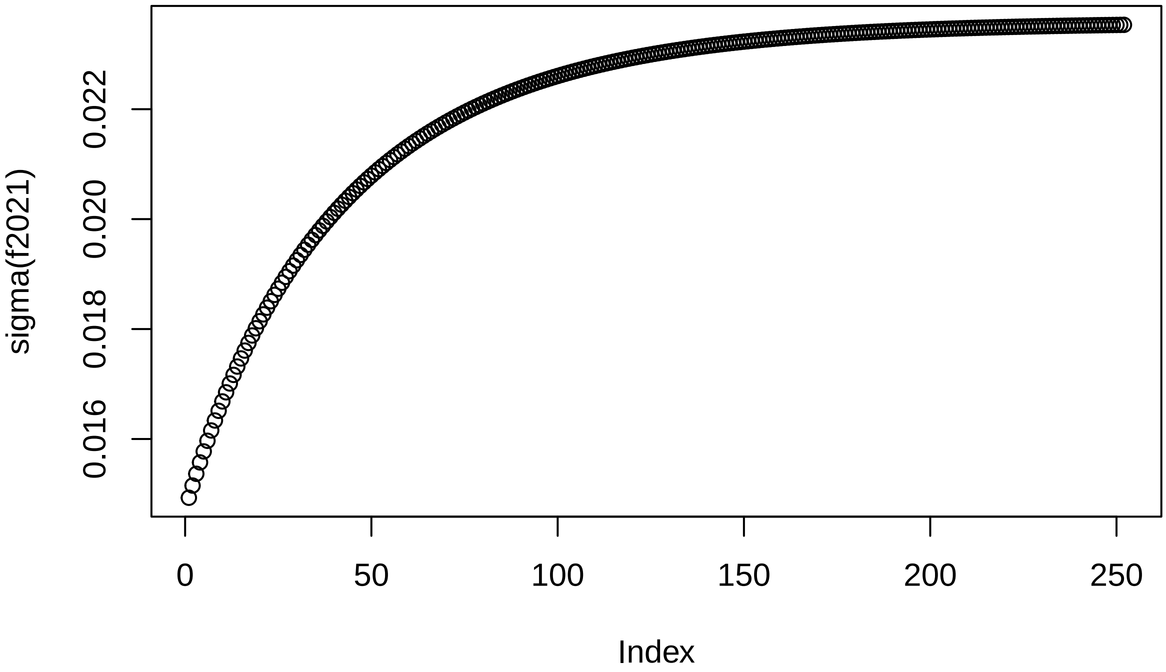
```
par(mfrow = c(1,2))
plot(MSFT_fore_roll, which=1)
plot(MSFT_fore_roll, which=2)
```



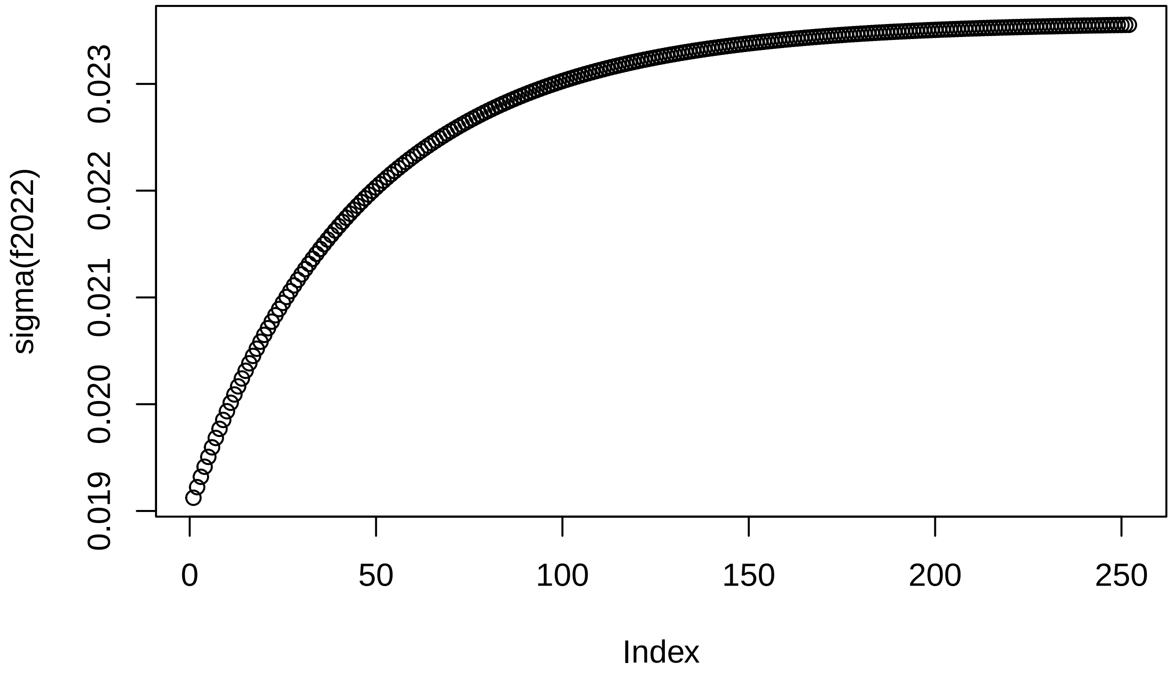
```
plot(MSFT_fore_roll, which='all')
```



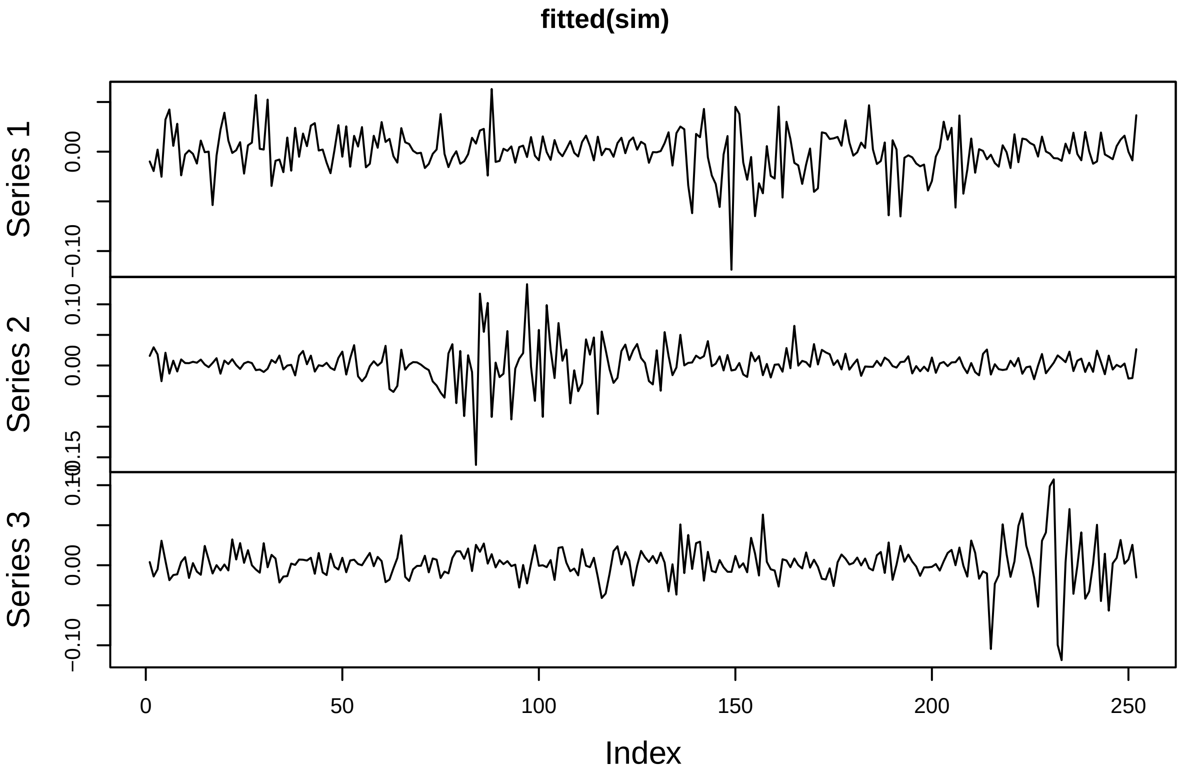
```
# We chose model with lowest AIC and Information Criteria
s <- MSFT_garch_1
m <- MSFT_fit_garch_1
#Merge parameter
sfinal <- s
setfixed(sfinal) <- as.list(coef(m))

f2021 <- ugarchforecast(data = na.omit(return_MSFT["/2021-12"]),
                        fitORspec = sfinal,
                        n.ahead = 252)
f2022 <- ugarchforecast(data = na.omit(return_MSFT["/2022-12"]),
                        fitORspec = sfinal,
                        n.ahead = 252)
#Forecasting future variance
par(mfrow=c(1,1))
plot(sigma(f2021))
```



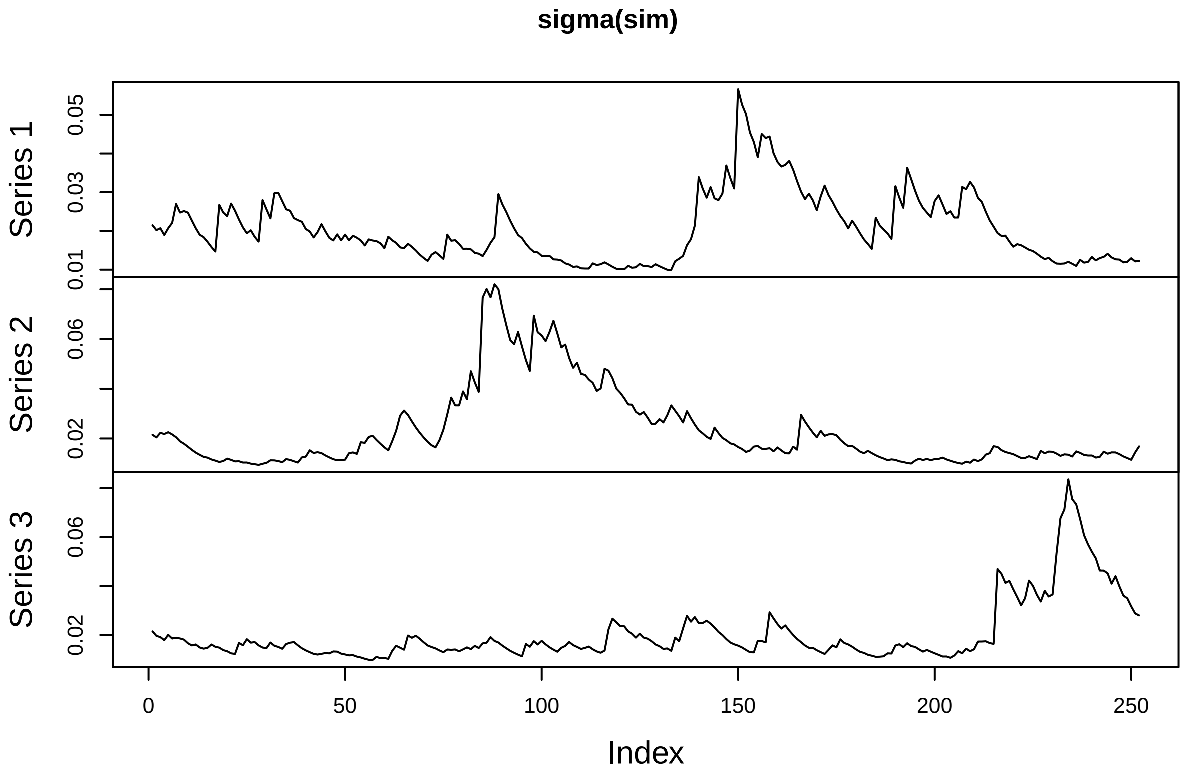
```
plot(sigma(f2022))
```



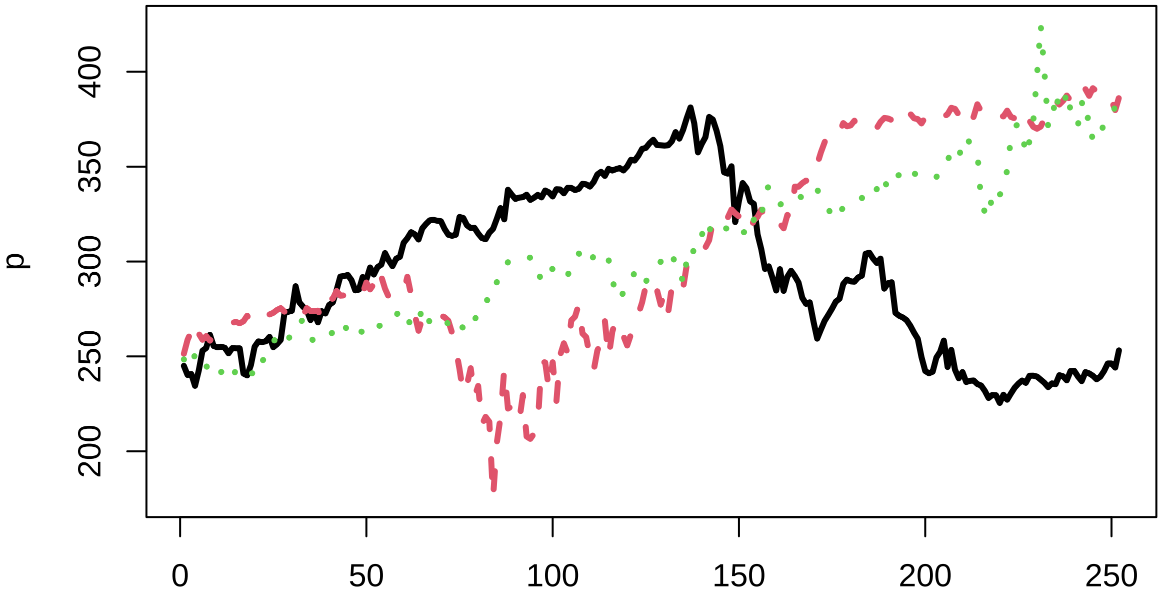
```
sim <- ugarchpath(spec = sfinal,
                  m.sim = 3,
                  n.sim = 1*252,
                  rseed = 123)
plot.zoo(fitted(sim))
```



```
plot.zoo(sigma(sim))
```



```
p <- 247.49 *apply(fitted(sim),2,'cumsum')+247.49
matplot(p, type = "l", lwd=3)
```


p
200
250
300
350
400
0
50
100
150
200
250