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)
```

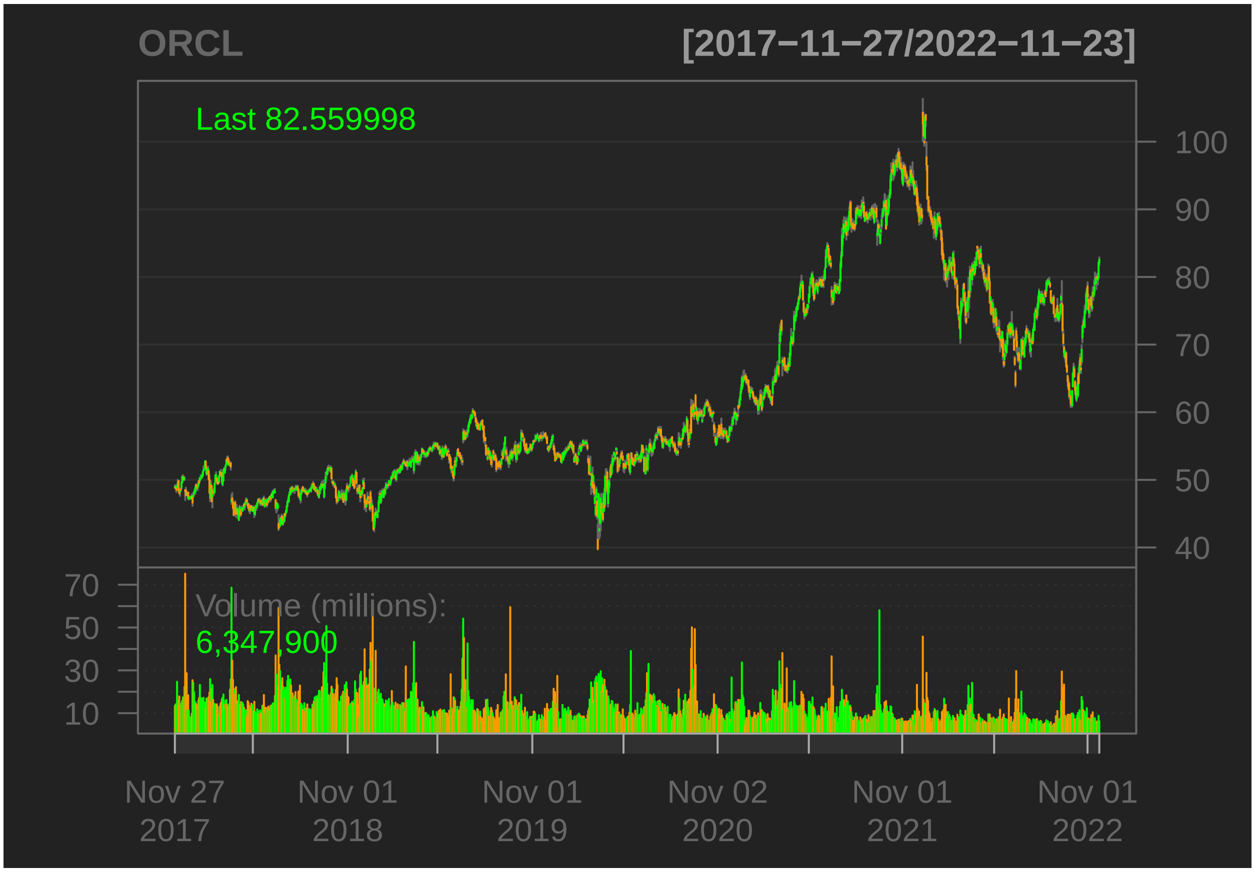
```
##
## Attaching package: 'rmgarch'
```

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

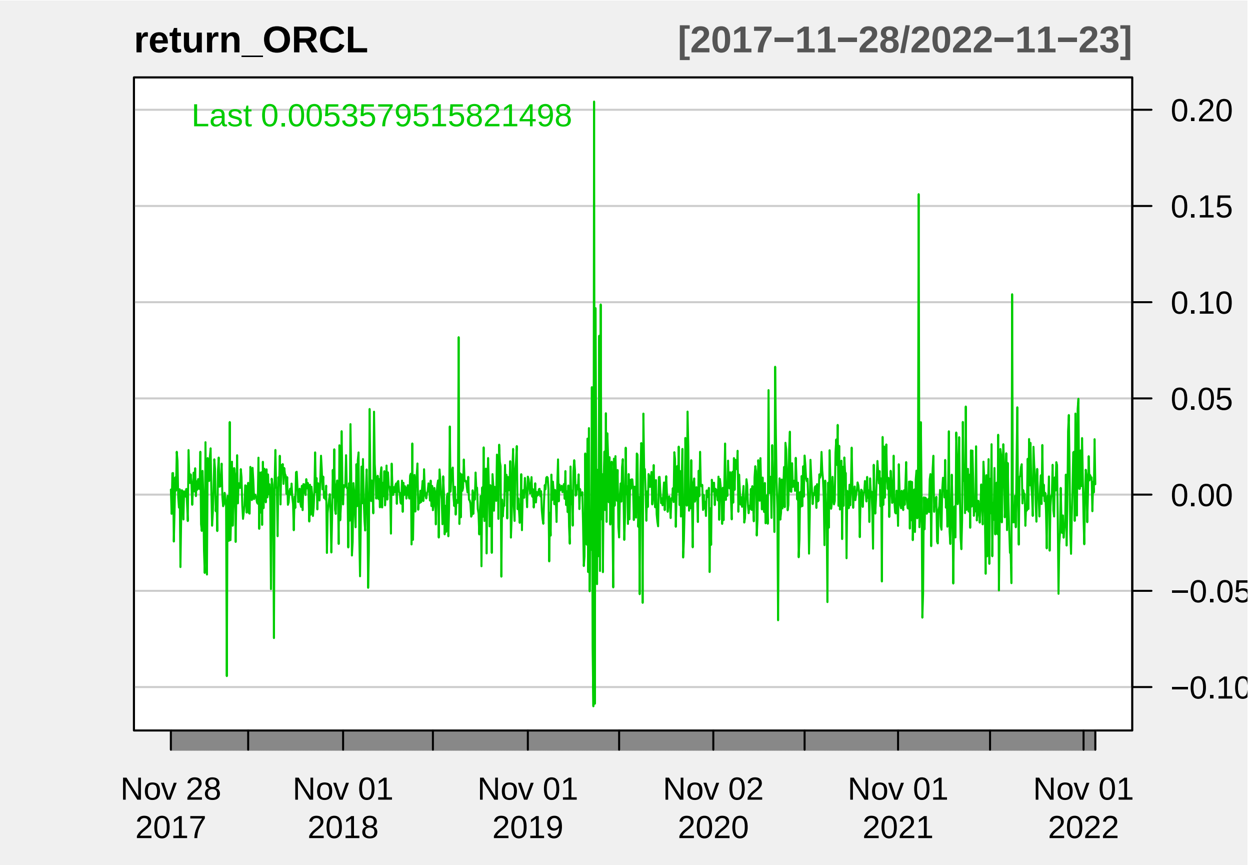
```
startDate = as.Date("2017-11-27")
endDate = as.Date("2022-11-25")
getSymbols("ORCL", from = startDate, to = endDate)
```

```
## [1] "ORCL"
```

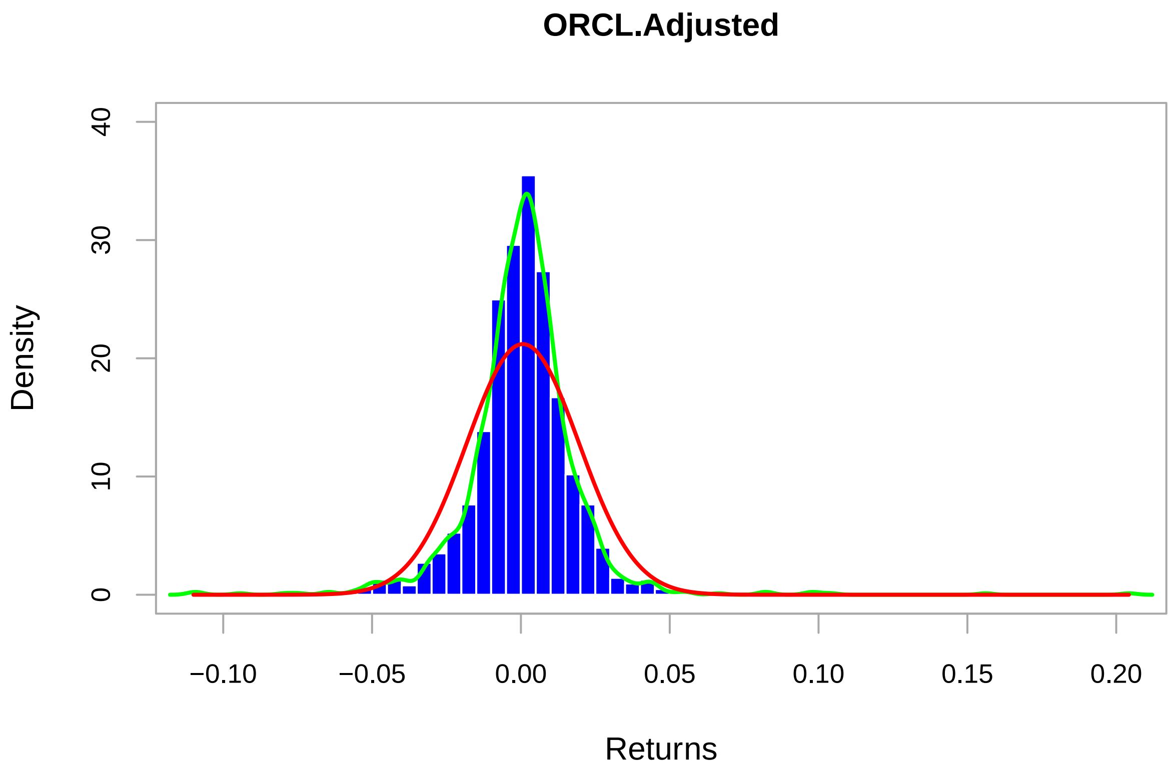
```
chartSeries(ORCL)
```



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



```
chart.Histogram(return_ORCL,
                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
```

```
ORCL_garch_1 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                          variance.model = list(model = 'sGARCH',
                                                garchOrder = c(1,1)),
                          distribution = 'std') # standard GARCH model
ORCL_fit_garch_1 <- ugarchfit(spec = ORCL_garch_1,
                              data = na.omit(return_ORCL))
infocriteria(ORCL_fit_garch_1)
```

```
##
## Akaike       -5.601523
## Bayes        -5.577005
## Shibata      -5.601568
## Hannan-Quinn -5.592308
```

```
ORCL_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      0.001063    0.000330  3.22367 0.001266
## ma1    -0.005205    0.029331 -0.17746 0.859146
## omega   0.000022    0.000007  3.27996 0.001038
## alpha1  0.224346    0.048901  4.58775 0.000004
## beta1   0.744724    0.043000 17.31923 0.000000
## shape   3.713564    0.407861  9.10497 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001063    0.000330  3.22369 0.001265
## ma1    -0.005205    0.030974 -0.16804 0.866549
## omega   0.000022    0.000007  2.93687 0.003315
## alpha1  0.224346    0.055703  4.02755 0.000056
## beta1   0.744724    0.047987 15.51924 0.000000
## shape   3.713564    0.447155  8.30487 0.000000
##
## LogLikelihood : 3526.557
##
## Information Criteria
## ------------------------------------
##
## Akaike       -5.6015
## Bayes        -5.5770
## Shibata      -5.6016
## Hannan-Quinn -5.5923
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.01776  0.8940
## Lag[2*(p+q)+(p+q)-1][2]   0.21981  0.9983
## Lag[4*(p+q)+(p+q)-1][5]   1.19260  0.9140
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.3091  0.5782
## Lag[2*(p+q)+(p+q)-1][5]    0.7633  0.9105
## Lag[4*(p+q)+(p+q)-1][9]    1.0648  0.9834
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.3107 0.500 2.000  0.5772
## ARCH Lag[5]    0.3418 1.440 1.667  0.9290
## ARCH Lag[7]    0.5748 2.315 1.543  0.9711
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.6098
## Individual Statistics:
## mu     0.1691
## ma1    0.2272
## omega  0.7338
## alpha1 0.5119
## beta1  0.7760
## shape  0.7696
##
## 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           2.2446 0.02497  **
## Negative Sign Bias  0.6306 0.52843
## Positive Sign Bias  0.2691 0.78788
## Joint Effect        7.5846 0.05543   *
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     14.96       0.7248
## 2    30     27.22       0.5596
## 3    40     41.68       0.3550
## 4    50     39.54       0.8306
##
##
## Elapsed time : 0.2011509
```

```r
#coef(ORCL_fit_garch_1)
```

```r
ORCL_garch_2 <- ugarchspec(mean.model = list(armaOrder = c(0,1)),
                           variance.model = list(model = "sGARCH",
                                                 garchOrder = c(1,2)),
                           distribution = 'std')
ORCL_fit_garch_2 <- ugarchfit(spec = ORCL_garch_2,
                              data = na.omit(return_ORCL))
# ORCL_fit_garch_2
infocriteria(ORCL_fit_garch_2)
```

```
##
## Akaike       -5.600580
## Bayes        -5.571976
## Shibata      -5.600642
## Hannan-Quinn -5.589830
```

```
#coef(ORCL_fit_garch_2)
```

```
ORCL_garch_3 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(1,3)),
                           distribution = 'std')

ORCL_fit_garch_3 <- ugarchfit(spec = ORCL_garch_3, data = na.omit(return_ORCL))
#ORCL_fit_garch_3
infocriteria(ORCL_fit_garch_3)
```

```
##
## Akaike       -5.599288
## Bayes        -5.566597
## Shibata      -5.599368
## Hannan-Quinn -5.587002
```

```
#coef(ORCL_fit_garch_3)
```

```
ORCL_garch_4 <- ugarchspec(mean.model = list(armaOrder=c(0,01)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(2,1)),
                           distribution = 'std')

ORCL_fit_garch_4 <- ugarchfit(spec = ORCL_garch_4, data = na.omit(return_ORCL))
#ORCL_fit_garch_4
infocriteria(ORCL_fit_garch_4)
```

```
##
## Akaike       -5.600031
## Bayes        -5.571427
## Shibata      -5.600093
## Hannan-Quinn -5.589281
```

```
#coef(ORCL_fit_garch_4)
```

```
ORCL_garch_5 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(2,2)),
                           distribution = 'std')

ORCL_fit_garch_5 <- ugarchfit(spec = ORCL_garch_5, data = na.omit(return_ORCL))
#ORCL_fit_garch_5
infocriteria(ORCL_fit_garch_5)
```

```
##
## Akaike       -5.598989
## Bayes        -5.566299
## Shibata      -5.599070
## Hannan-Quinn -5.586703
```

```r
#coef(ORCL_fit_garch_5)
```

```r
ORCL_garch_11 <- ugarchspec(mean.model = list(armaOrder=c(0,0)),
                          variance.model = list(model = 'sGARCH',
                                                garchOrder = c(1,1)),
                          distribution = 'std')
ORCL_fit_garch_11 <- ugarchfit(spec = ORCL_garch_11,
                              data = na.omit(return_ORCL))
infocriteria(ORCL_fit_garch_11)
```

```
##
## Akaike       -5.603089
## Bayes        -5.582657
## Shibata      -5.603120
## Hannan-Quinn -5.595410
```

```r
#ORCL_fit_garch_11
#coef(ORCL_fit_garch_11)
```

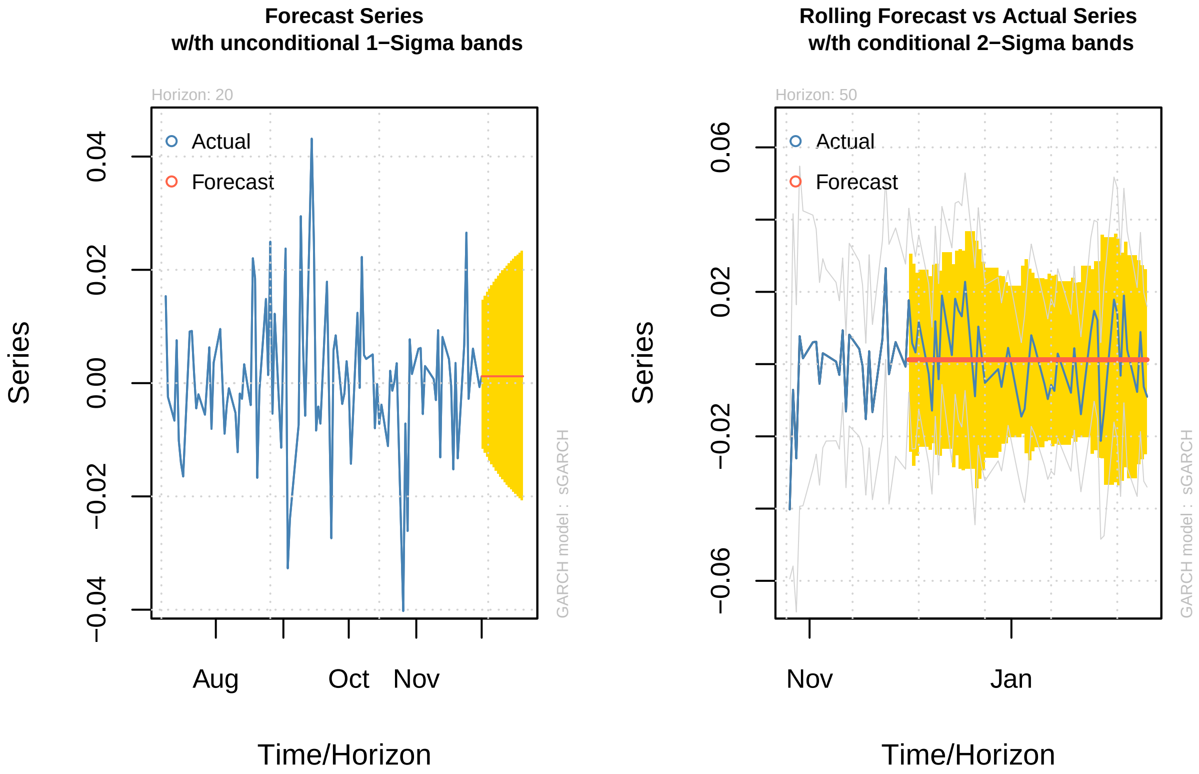
```r
ORCL_forecast <- ugarchforecast(ORCL_fit_garch_11,
                                 data = na.omit(return_ORCL),
                                 n.ahead = 20)
ORCL_forecast
```

```
##
## *------------------------------------*
## *       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  0.001062 0.01578
## T+2  0.001062 0.01622
## T+3  0.001062 0.01663
## T+4  0.001062 0.01702
## T+5  0.001062 0.01738
## T+6  0.001062 0.01773
## T+7  0.001062 0.01807
## T+8  0.001062 0.01838
## T+9  0.001062 0.01868
## T+10 0.001062 0.01897
## T+11 0.001062 0.01924
## T+12 0.001062 0.01951
## T+13 0.001062 0.01976
## T+14 0.001062 0.02000
## T+15 0.001062 0.02023
## T+16 0.001062 0.02045
## T+17 0.001062 0.02066
## T+18 0.001062 0.02086
## T+19 0.001062 0.02105
## T+20 0.001062 0.02124
```

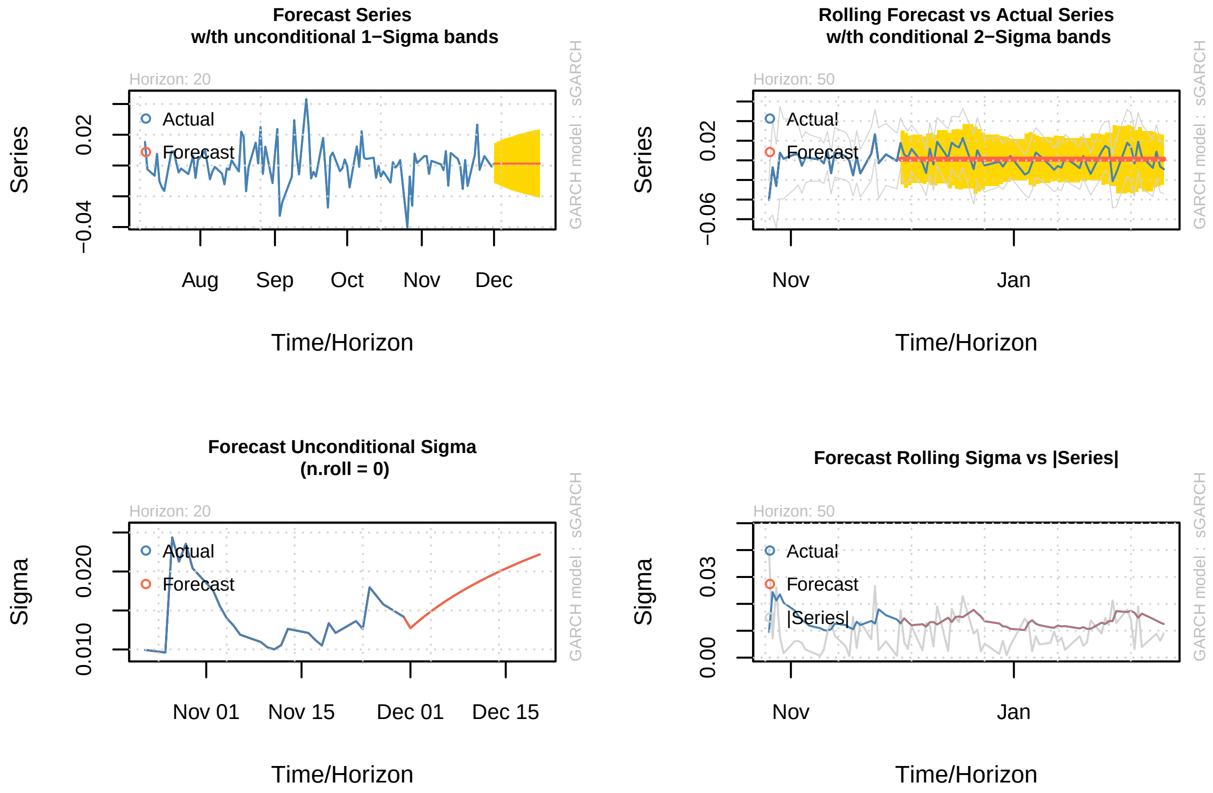
```
ORCL_fit_roll <- ugarchfit(ORCL_garch_11,
                           data = na.omit(return_ORCL),
                           out.sample = 500)
ORCL_fore_roll <- ugarchforecast(ORCL_fit_roll,
                                 n.ahead = 20,
                                 n.roll=50)
ORCL_fore_roll
```

```
##
## *------------------------------------*
## *       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  0.001207 0.01274
## T+2  0.001207 0.01350
## T+3  0.001207 0.01421
## T+4  0.001207 0.01488
## T+5  0.001207 0.01550
## T+6  0.001207 0.01610
## T+7  0.001207 0.01666
## T+8  0.001207 0.01720
## T+9  0.001207 0.01771
## T+10 0.001207 0.01820
## T+11 0.001207 0.01867
## T+12 0.001207 0.01912
## T+13 0.001207 0.01956
## T+14 0.001207 0.01997
## T+15 0.001207 0.02038
## T+16 0.001207 0.02076
## T+17 0.001207 0.02114
## T+18 0.001207 0.02150
## T+19 0.001207 0.02185
## T+20 0.001207 0.02219
```

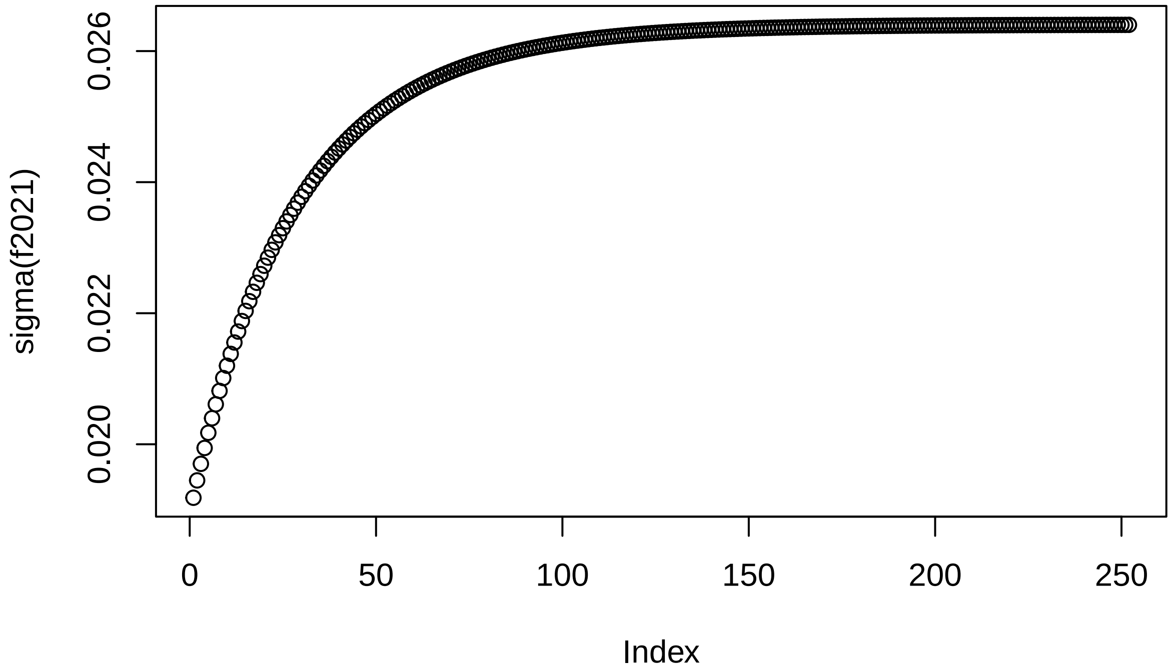
```
par(mfrow = c(1,2))
plot(ORCL_fore_roll, which=1)
plot(ORCL_fore_roll, which=2)
```



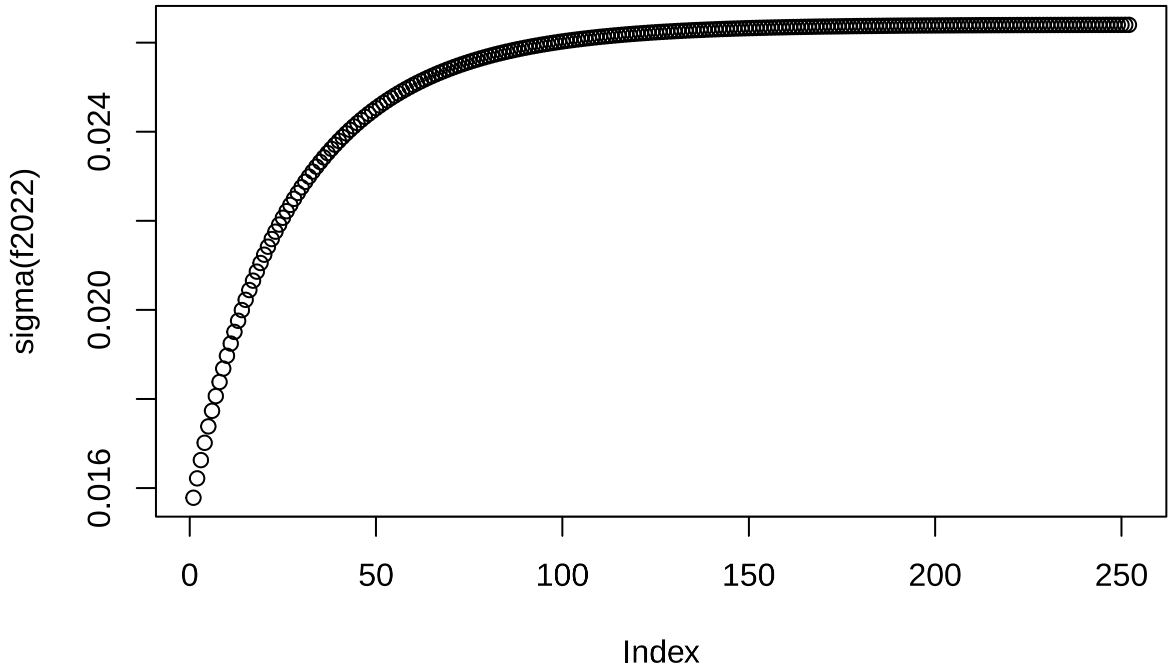
```r
plot(ORCL_fore_roll, which='all')
```



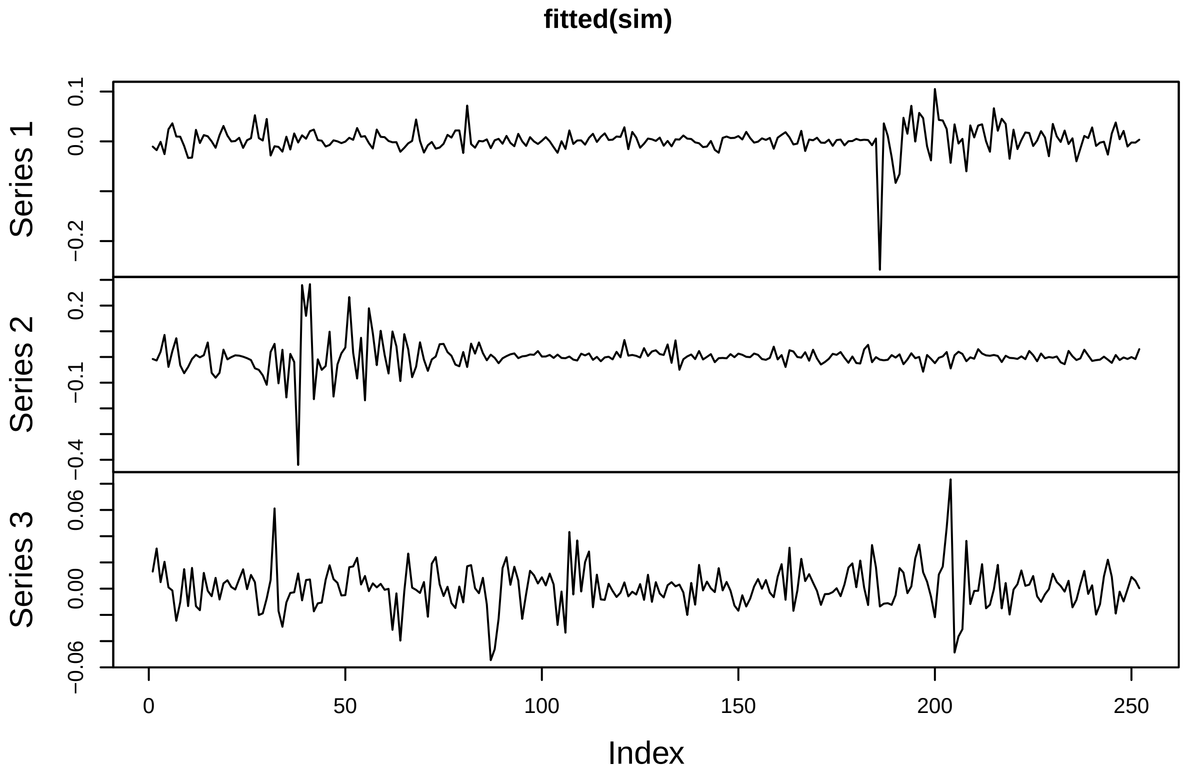
```r
# We chose model with lowest AIC and Information Criteria
s <- ORCL_garch_11
m <- ORCL_fit_garch_11
#Merge parameter
sfinal <- s
setfixed(sfinal) <- as.list(coef(m))

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



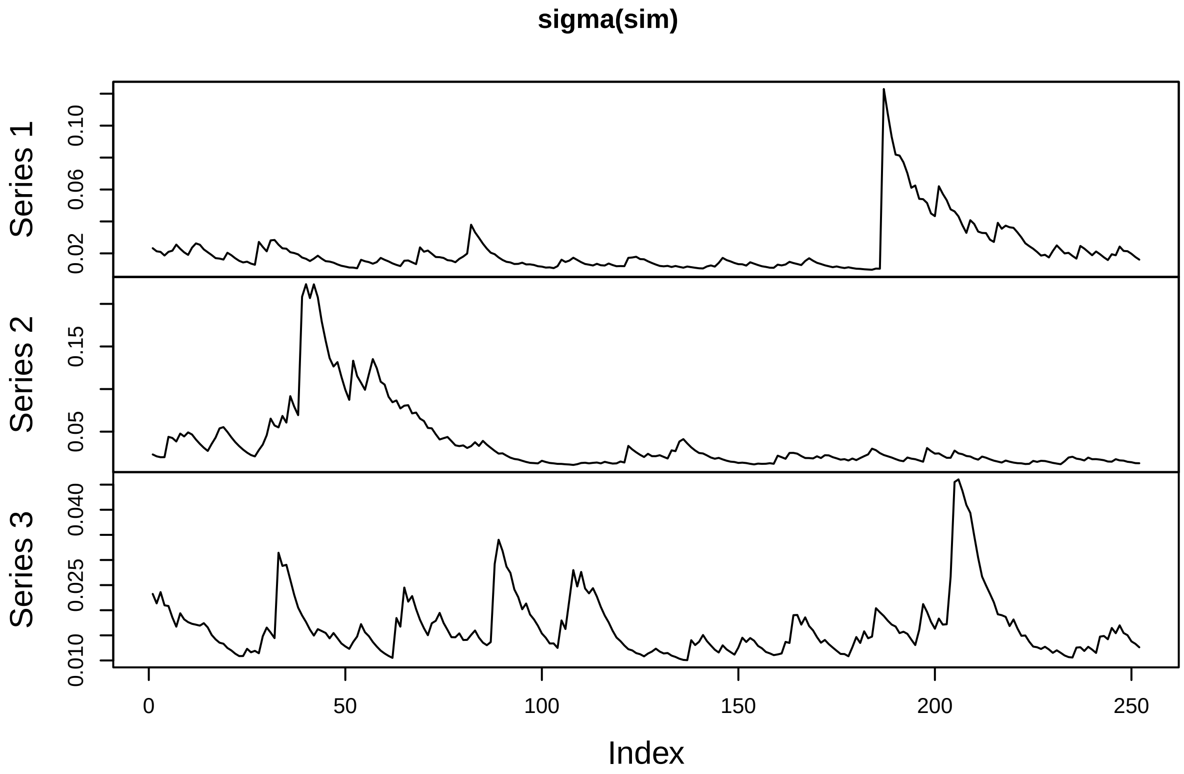
```
plot(sigma(f2022))
```



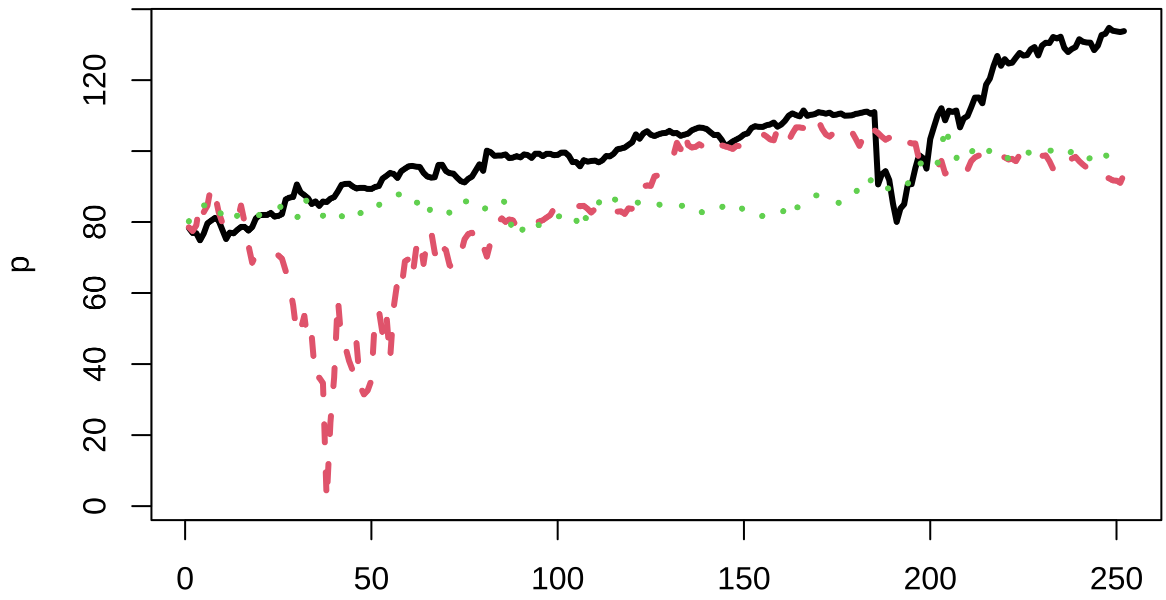
```
sim <- ugarchpath(spec = sfinal,
                  m.sim = 3,
                  n.sim = 1*252,
                  rseed = 123)
plot.zoo(fitted(sim))
```



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



```
p <- 79.18  *apply(fitted(sim),2,'cumsum')+79.18
matplot(p, type = "l", lwd=3)
```



p
0
20
40
60
80
120
0
50
100
150
200
250