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'
```

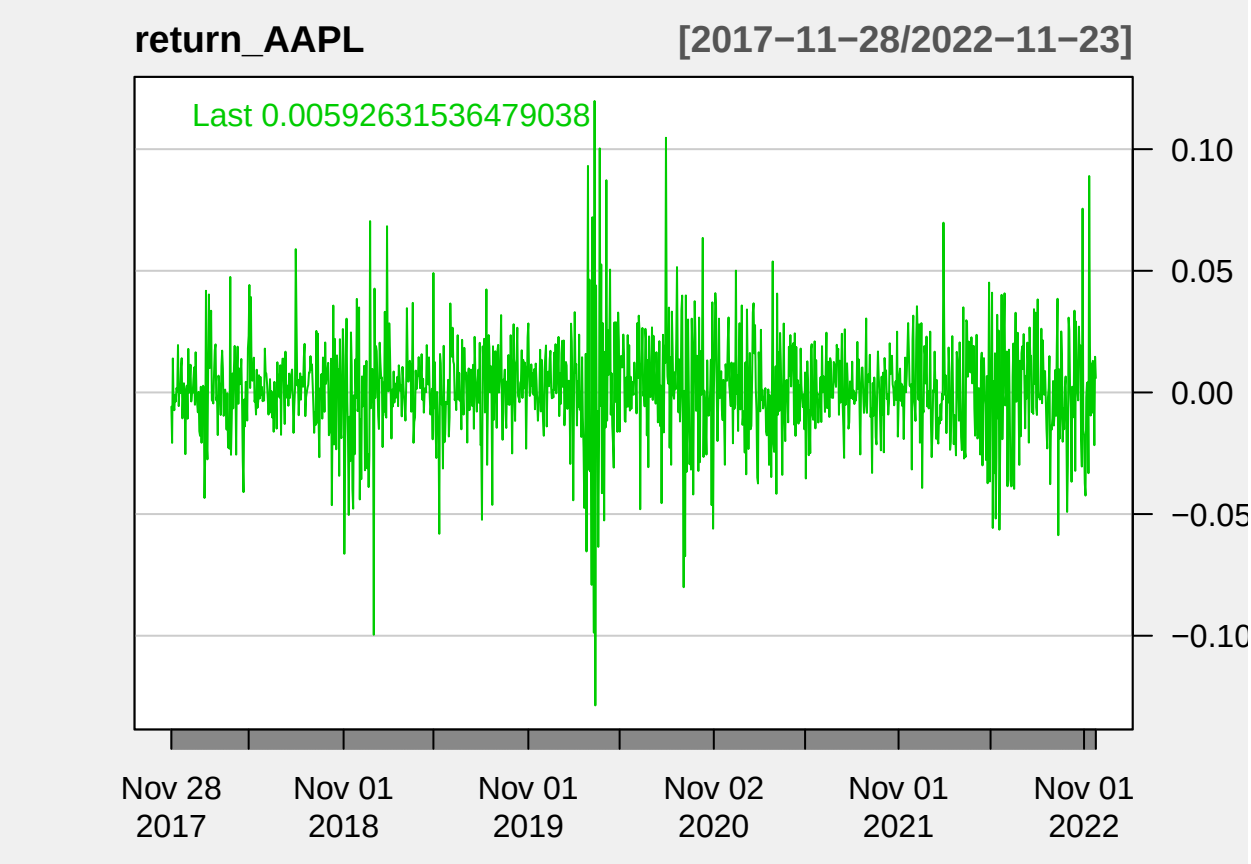
```
## The following objects are masked from 'package:xts':
##
##     first, last
```

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

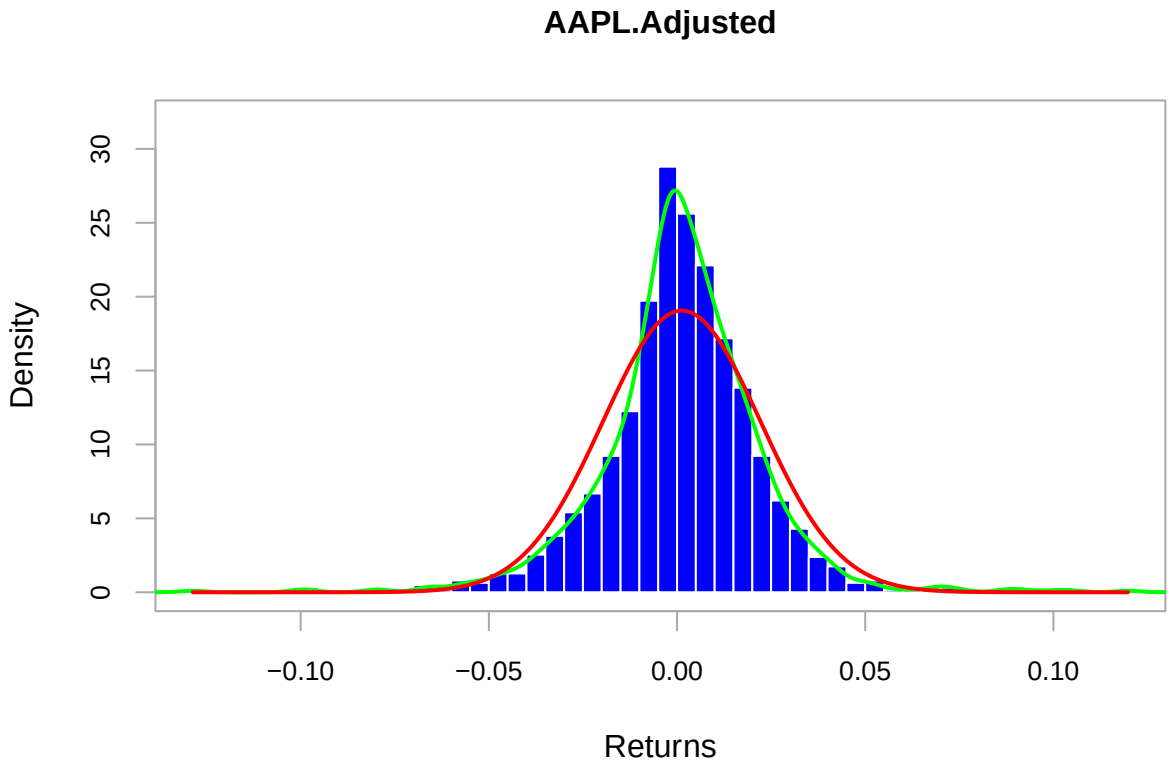
```
## [1] "AAPL"
```

```
chartSeries(AAPL)
```


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




```
chart.Histogram(return_AAPL,
                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
```

```
AAPL_garch_1 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(1,1)),
                           distribution = 'std') # standard GARCH model
AAPL_fit_garch_1 <- ugarchfit(spec = AAPL_garch_1,
                              data = na.omit(return_AAPL))
#infocriteria(AAPL_fit_garch_1)
AAPL_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      0.001955    0.000426   4.5922 0.000004
## ma1    -0.039501    0.030129  -1.3111 0.189835
## omega   0.000016    0.000005   3.4397 0.000582
## alpha1  0.136386    0.026341   5.1777 0.000000
## beta1   0.833926    0.024469  34.0812 0.000000
## shape   5.639228    0.939621   6.0016 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001955    0.000439   4.4498 0.000009
## ma1    -0.039501    0.029600  -1.3345 0.182050
## omega   0.000016    0.000005   3.3683 0.000756
## alpha1  0.136386    0.030848   4.4212 0.000010
## beta1   0.833926    0.022323  37.3577 0.000000
## shape   5.639228    0.935934   6.0252 0.000000
##
## LogLikelihood : 3251.026
##
## Information Criteria
## ------------------------------------
##
## Akaike       -5.1631
## Bayes        -5.1386
## Shibata      -5.1632
## Hannan-Quinn -5.1539
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.0278  0.8676
## Lag[2*(p+q)+(p+q)-1][2]    0.8388  0.8314
## Lag[4*(p+q)+(p+q)-1][5]    2.4117  0.5917
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.4755  0.4905
## Lag[2*(p+q)+(p+q)-1][5]    0.9945  0.8609
## Lag[4*(p+q)+(p+q)-1][9]    2.2634  0.8713
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.7541 0.500 2.000  0.3852
## ARCH Lag[5]    0.8885 1.440 1.667  0.7663
## ARCH Lag[7]    2.1011 2.315 1.543  0.6959
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  7.4624
## Individual Statistics:
```

```
## mu     0.2172
## ma1    0.1785
## omega  0.2205
## alpha1 0.3415
## beta1  0.7166
## shape  0.9298
##
## 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           1.2181 0.2234
## Negative Sign Bias  0.4274 0.6692
## Positive Sign Bias  0.4327 0.6653
## Joint Effect        3.4375 0.3290
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     23.05       0.2352
## 2    30     37.87       0.1253
## 3    40     39.13       0.4639
## 4    50     49.17       0.4665
##
##
## Elapsed time : 0.1949339
```

```
#coef(AAPL_fit_garch_1)
```

```
AAPL_garch_2 <- ugarchspec(mean.model = list(armaOrder = c(0,1)),
                           variance.model = list(model = "sGARCH",
                                                 garchOrder = c(1,2)),
                           distribution = 'std')
AAPL_fit_garch_2 <- ugarchfit(spec = AAPL_garch_2,
                              data = na.omit(return_AAPL))
# AAPL_fit_garch_2
infocriteria(AAPL_fit_garch_2)
```

```
##
## Akaike       -5.161228
## Bayes        -5.132624
## Shibata      -5.161289
## Hannan-Quinn -5.150478
```

```
#coef(AAPL_fit_garch_2)
```

```
AAPL_garch_11 <- ugarchspec(mean.model = list(armaOrder=c(0,0)),
                            variance.model = list(model = 'sGARCH',
```

```
                                                garchOrder = c(1,1)),
                           distribution = 'std')
AAPL_fit_garch_11 <- ugarchfit(spec = AAPL_garch_11,
                               data = na.omit(return_AAPL))
infocriteria(AAPL_fit_garch_11)
```

```
##
## Akaike       -5.163353
## Bayes        -5.142921
## Shibata      -5.163384
## Hannan-Quinn -5.155674
```

```
#AAPL_fit_garch_11
#coef(AAPL_fit_garch_11)
```

```
AAPL_garch_3 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(1,3)),
                           distribution = 'std')

AAPL_fit_garch_3 <- ugarchfit(spec = AAPL_garch_3, data = na.omit(return_AAPL))
#AAPL_fit_garch_3
infocriteria(AAPL_fit_garch_3)
```

```
##
## Akaike       -5.159793
## Bayes        -5.127102
## Shibata      -5.159873
## Hannan-Quinn -5.147507
```

```
#coef(AAPL_fit_garch_3)
```

```
AAPL_garch_4 <- ugarchspec(mean.model = list(armaOrder=c(0,01)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(2,1)),
                           distribution = 'std')

AAPL_fit_garch_4 <- ugarchfit(spec = AAPL_garch_4, data = na.omit(return_AAPL))
#AAPL_fit_garch_4
infocriteria(AAPL_fit_garch_4)
```

```
##
## Akaike       -5.161423
## Bayes        -5.132819
## Shibata      -5.161484
## Hannan-Quinn -5.150672
```

```
#coef(AAPL_fit_garch_4)
```

```
AAPL_garch_5 <- ugarchspec(mean.model = list(armaOrder=c(0,1)),
                           variance.model = list(model = 'sGARCH',
                                                 garchOrder = c(2,2)),
                           distribution = 'std')

AAPL_fit_garch_5 <- ugarchfit(spec = AAPL_garch_5, data = na.omit(return_AAPL))
#AAPL_fit_garch_5
infocriteria(AAPL_fit_garch_5)
```

```
##
## Akaike       -5.160949
## Bayes        -5.128258
## Shibata      -5.161029
## Hannan-Quinn -5.148663
```

```
#coef(AAPL_fit_garch_5)
```

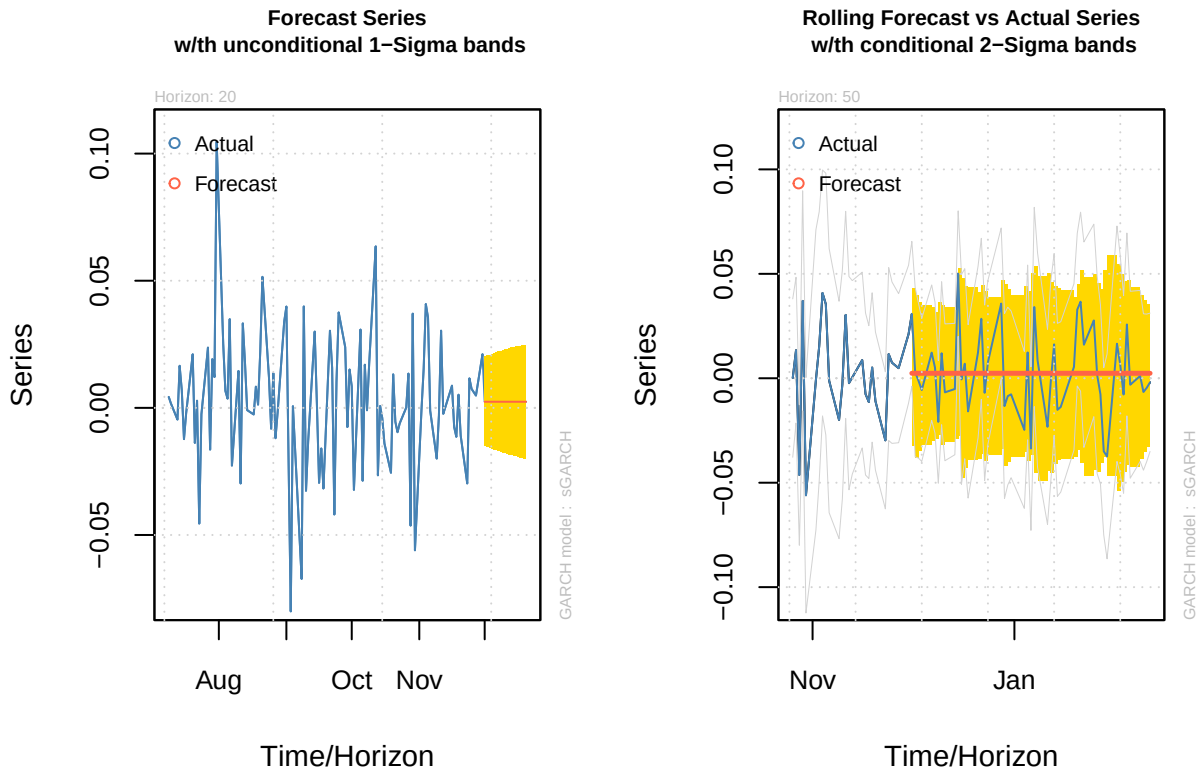
```
AAPL_forecast <- ugarchforecast(AAPL_fit_garch_11,
                                data = na.omit(return_AAPL),
                                n.ahead = 20)
AAPL_forecast
```

```
##
## *------------------------------------*
## *       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  0.001942 0.02269
## T+2  0.001942 0.02271
## T+3  0.001942 0.02273
## T+4  0.001942 0.02275
## T+5  0.001942 0.02277
## T+6  0.001942 0.02279
## T+7  0.001942 0.02281
## T+8  0.001942 0.02283
## T+9  0.001942 0.02285
## T+10 0.001942 0.02287
## T+11 0.001942 0.02289
## T+12 0.001942 0.02290
## T+13 0.001942 0.02292
## T+14 0.001942 0.02294
## T+15 0.001942 0.02295
## T+16 0.001942 0.02297
## T+17 0.001942 0.02298
## T+18 0.001942 0.02299
## T+19 0.001942 0.02301
## T+20 0.001942 0.02302
```

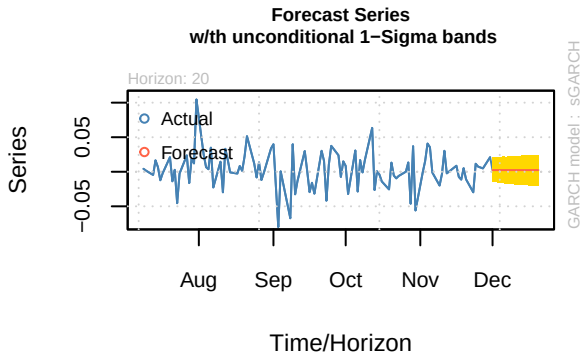
```
AAPL_fit_roll <- ugarchfit(AAPL_garch_11,
                           data = na.omit(return_AAPL),
                           out.sample = 500)
AAPL_fore_roll <- ugarchforecast(AAPL_fit_roll,
                                 n.ahead = 20,
                                 n.roll=50)
AAPL_fore_roll
```

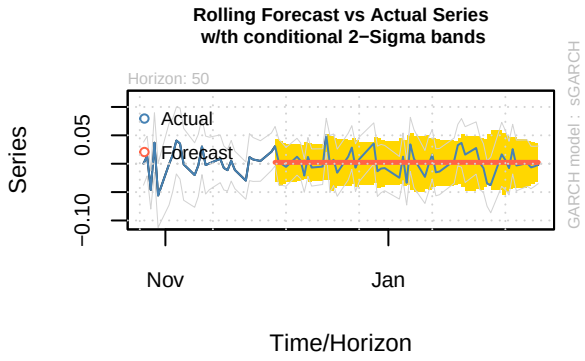
```
##
## *----------------------------------*
## *       GARCH Model Forecast       *
## *----------------------------------*
## Model: sGARCH
## Horizon: 20
## Roll Steps: 50
## Out of Sample: 20
##
## 0-roll forecast [T0=2020-11-30]:
##        Series   Sigma
## T+1  0.002392 0.01735
## T+2  0.002392 0.01773
## T+3  0.002392 0.01809
## T+4  0.002392 0.01844
## T+5  0.002392 0.01877
## T+6  0.002392 0.01909
## T+7  0.002392 0.01939
## T+8  0.002392 0.01969
## T+9  0.002392 0.01997
## T+10 0.002392 0.02024
## T+11 0.002392 0.02050
## T+12 0.002392 0.02075
## T+13 0.002392 0.02099
## T+14 0.002392 0.02122
## T+15 0.002392 0.02145
## T+16 0.002392 0.02166
## T+17 0.002392 0.02187
## T+18 0.002392 0.02207
## T+19 0.002392 0.02227
## T+20 0.002392 0.02245
```

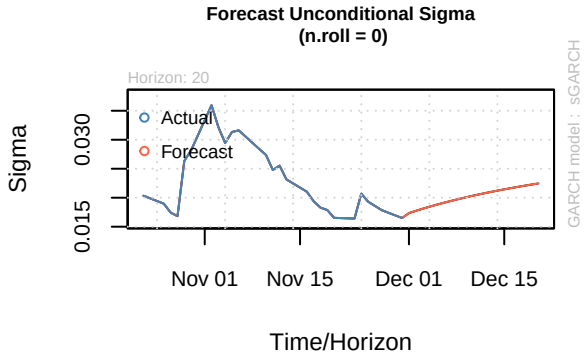
```
par(mfrow = c(1,2))
plot(AAPL_fore_roll, which=1)
plot(AAPL_fore_roll, which=2)
```

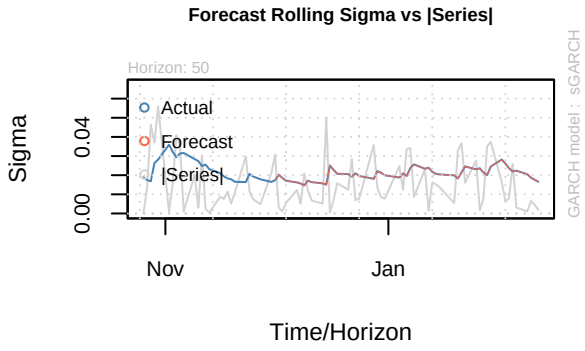



```
plot(AAPL_fore_roll, which='all')
```

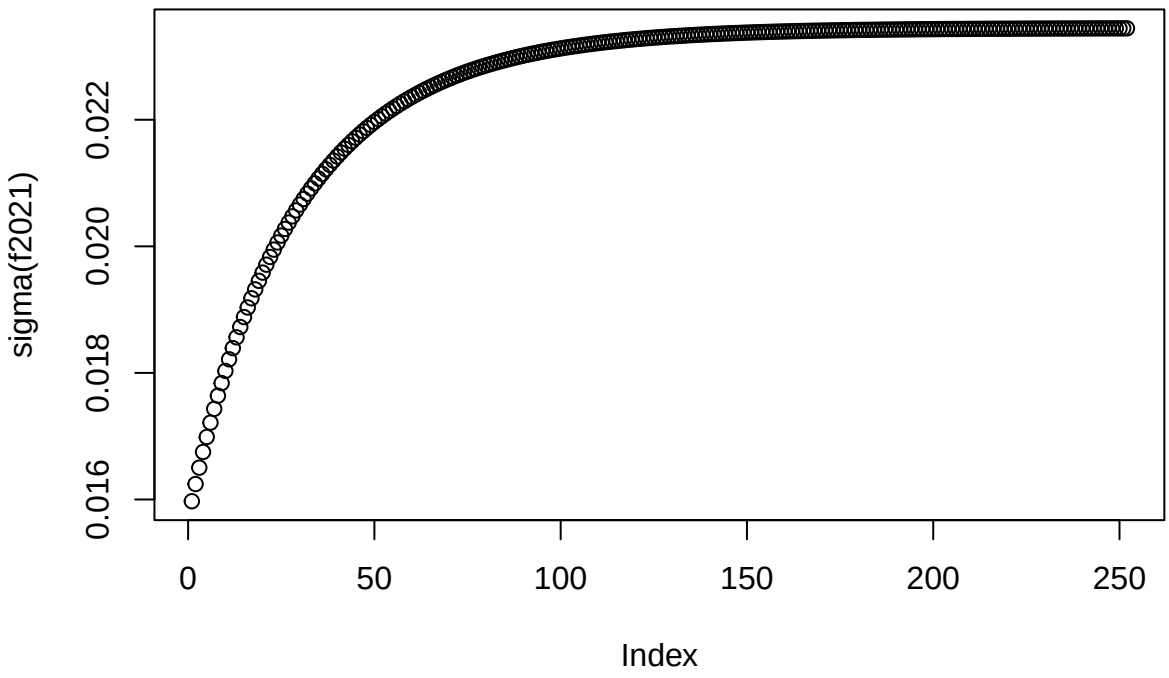




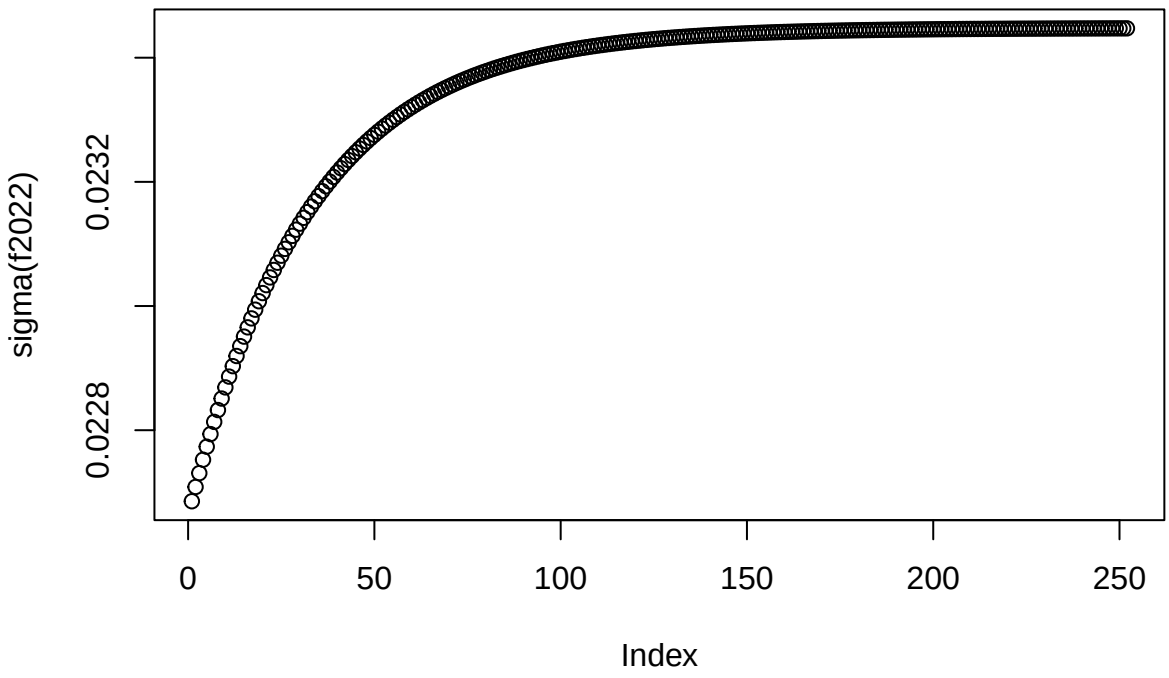




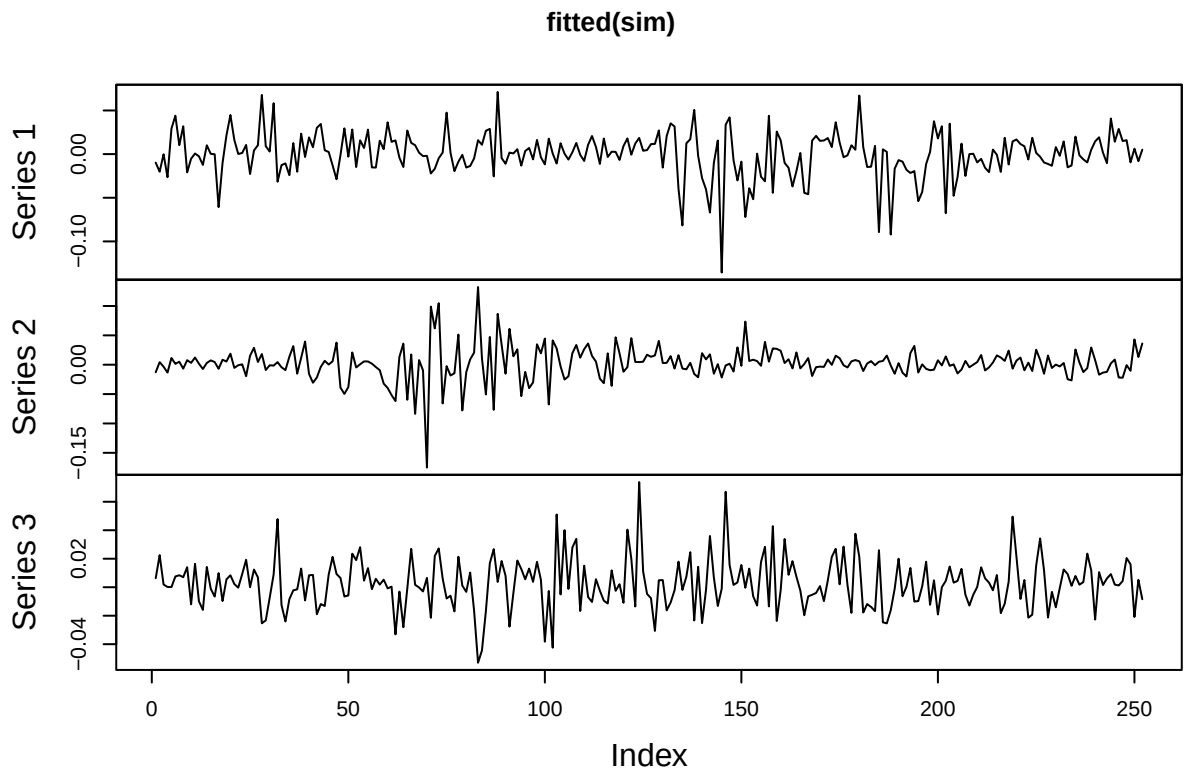

```
# We chose model with lowest AIC and Information Criteria
s <- AAPL_garch_11
m <- AAPL_fit_garch_11
#Merge parameter
sfinal <- s
setfixed(sfinal) <- as.list(coef(m))

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



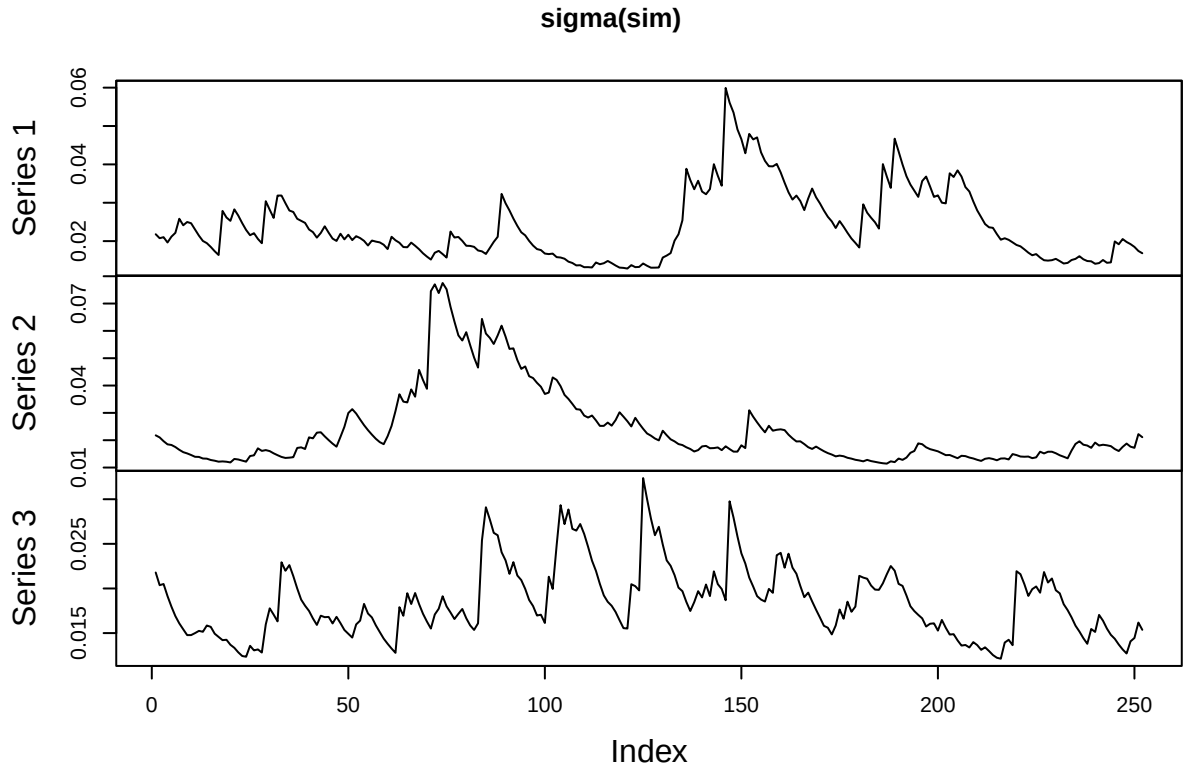

```
plot(sigma(f2022))
```



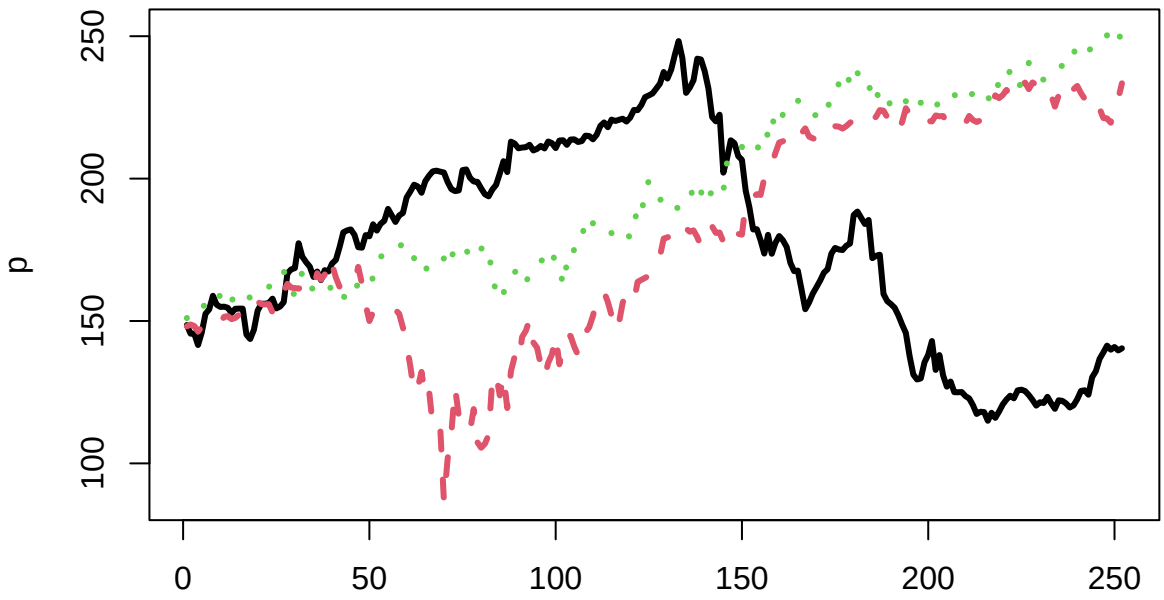

```
sim <- ugarchpath(spec = sfinal,
                  m.sim = 3,
                  n.sim = 1*252,
                  rseed = 123)
plot.zoo(fitted(sim))
```




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




```
p <- 150.04 *apply(fitted(sim),2,'cumsum')+150.04
matplot(p, type = "l", lwd=3)
```


p
100
150
200
250
0
50
100
150
200
250