Time Series Analysis and Forecasting on COST using GARCH Model

I. Introduction

Costco Wholesale Corporation (COST), the world's second largest retailer right after Walmart, is a multinational membership-based American company founded in 1983 and went public in 1985 with an initial share price of $10. For the fiscal year of 2018, Costco reported an increase of 9.7% of earnings compared to the previous year, and it is ranked the 15th largest United States corporation by revenue according to the Fortune 500. The company was valued at over $95.7 billion for its market capitalization, while within the past 52 week the stock was traded at the highest $245 per share and the lowest $176 per share. Noticing the 39% raise of COST share price with its business reputation in mind, an exploratory time series analysis of COST stock return including the near future forecasting would be helpful for both current investors and potential investors to take action.

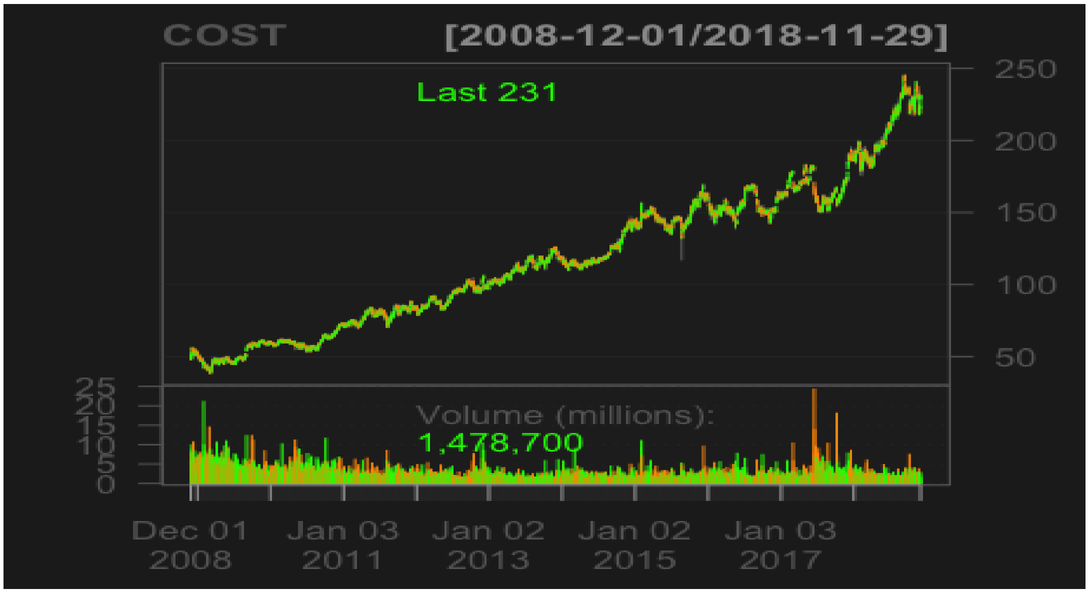
In time series analysis, there are several models that can be used to model the data variations. ARMA(Autoregressive Moving Average) and ARFIMA(Autoregressive Integrated Moving Average) can analyze the time series data linearly. While ARCH(Autoregressive Conditionally Heteroscedastic) and GARCH(Generalized Autoregressive Conditionally Heteroscedastic) models can do it nonlinearly, they are used to analyze the changes of variance overtime, and are largely utilized in economic and financial field using stock data for instance. This project is to conduct a time series analysis and forecasting of Costco's daily stock return through GARCH model for the past 10 years, assuming the error variance in the GARCH model fits an ARMA model. Packages "quantmod" and "rugarch" are used through the analysis.

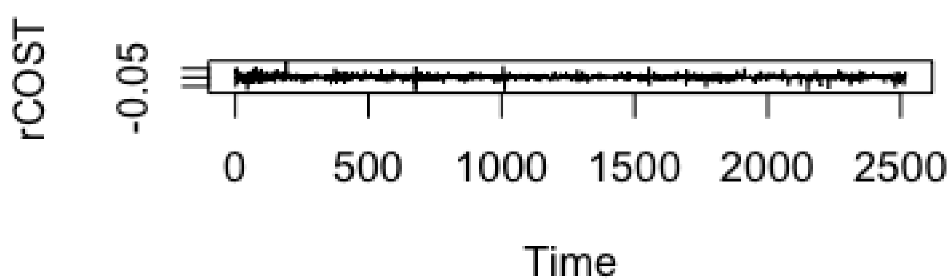
II. Model Estimation

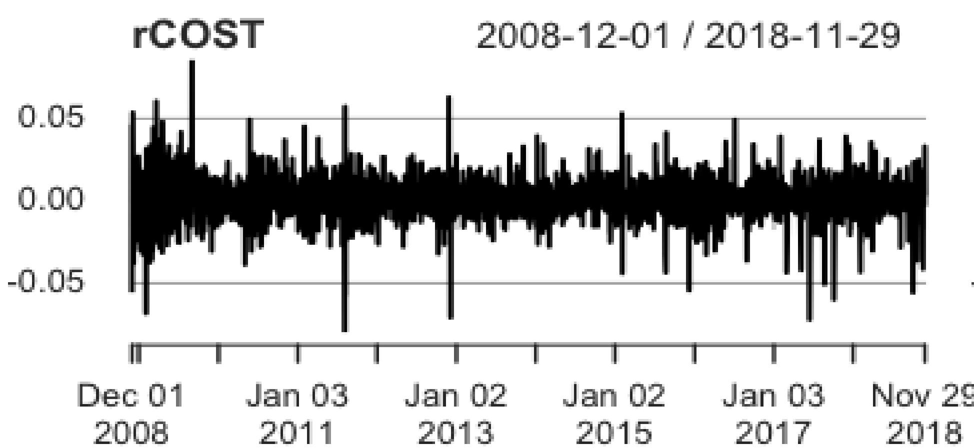
To start the time series analysis, we retrieve the stock data of COST for the past ten years (12/1/2008 - 11/30/2018) from finance.yahoo.com, and we plot the data as shown below. For some unknown reason, the data for 11/30/2018 is missing, so we need to proceed with an omission of the missing value. We can see clusters of volatility in the plot, and that the variance of change is not constant through the time series. Then we plot the daily return of COST data as well as the ACF and PACF. We can see a more apparent cluster of volatility in the time series plot. In ACF plot all returns are insignificant, and in PACF plot, there is only one significant return at lag 26.

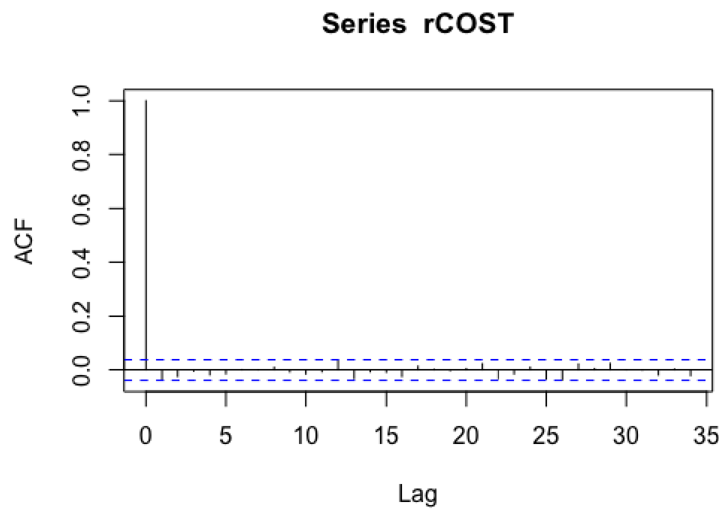
Data Source: https://finance.yahoo.com/quote/COST/history?p=COST

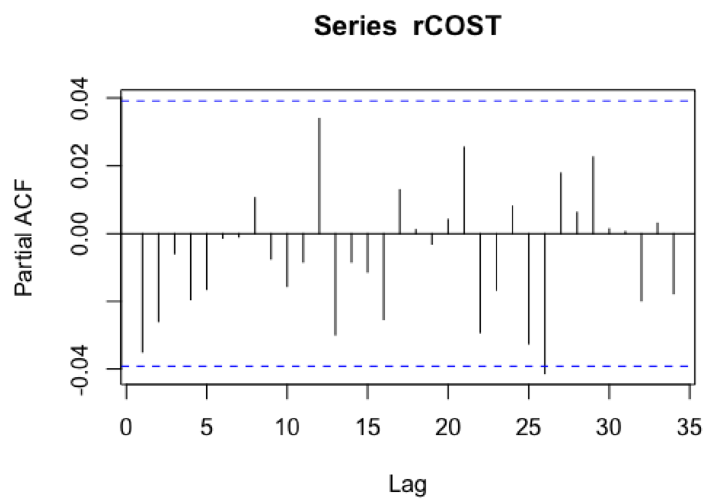
Coding Language: R


COST
[2008-12-01/2018-11-29]
Last 231
Volume (millions):
1,478,700
Dec 01
2008
Jan 03
2011
Jan 02
2013
Jan 02
2015
Jan 03
2017


rCOST
-0.05
0
500
1000
1500
2000
2500
Time


rCOST
2008-12-01 / 2018-11-29
0.05
0.00
-0.05
Dec 01
2008
Jan 03
2011
Jan 02
2013
Jan 02
2015
Jan 03
2017
Nov 29
2018





Next, we specify that GARCH is the model we will be using. The standard GARCH model for the estimation in R is (1, 1) by default as shown under the GARCH Model Spec table. Then we can start estimating the GARCH model using ARMA (1,0), (0,1), (1,1), (1,2), (2,1), and (2,2). Here we only print out the result for ARMA(1,0) below as an example. In the Information Criterion, Akaike value or AIC value is -6.0271. Each estimated model has an unique AIC value, we need to compare them and choose the model with the smallest AIC value to be the best fitted model for further forecasting. And here, we have model (1,0) and (0,1) to be the best models which both possess the smallest AIC value of -6.0271 as seen in the AIC comparison table below. In this case, if we can round the values to over five decimals, there might be a winner between these two models, however we can use the ugarchfit function to extract the best fitted model and compare the return predictions for the next 20 days. The Optimal Parameters section in the table shows the optimal variable coefficients for the model estimated.

```
*---------------------------------*
*       GARCH Model Spec          *
*---------------------------------*

Conditional Variance Dynamics
------------------------------------
GARCH Model       : sGARCH(1,1)
Variance Targeting      : FALSE

Conditional Mean Dynamics
------------------------------------
Mean Model        : ARFIMA(1,0,1)
Include Mean            : TRUE
GARCH-in-Mean           : FALSE

Conditional Distribution
------------------------------------
Distribution      :  norm
Includes Skew     :  FALSE
Includes Shape    :  FALSE
Includes Lambda:  FALSE



*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,1)
Mean Model  : ARFIMA(1,0,0)
Distribution      : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.000751    0.000225   3.33540 0.000852
ar1    -0.009374    0.021330  -0.43945 0.660332
omega   0.000004    0.000000  10.96449 0.000000
alpha1  0.033132    0.002359  14.04627 0.000000
beta1   0.941781    0.004001 235.39460 0.000000

Robust Standard Errors:
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.000751    0.000219   3.42146 0.000623
ar1    -0.009374    0.024319  -0.38546 0.699900
omega   0.000004    0.000001   4.55495 0.000005
alpha1  0.033132    0.005389   6.14838 0.000000
beta1   0.941781    0.004051 232.49819 0.000000

LogLikelihood : 7593.151

Information Criteria
------------------------------------

Akaike       -6.0271
Bayes        -6.0155
Shibata      -6.0271
Hannan-Quinn -6.0229

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
```

```
Lag[1]                          0.01592  0.8996
Lag[2*(p+q)+(p+q)-1][2]         0.68235  0.9051
Lag[4*(p+q)+(p+q)-1][5]         1.79085  0.7705
d.o.f=1
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                      4.107 0.04271
Lag[2*(p+q)+(p+q)-1][5]     5.178 0.13951
Lag[4*(p+q)+(p+q)-1][9]     6.030 0.29477
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]  0.005906 0.500 2.000  0.9387
ARCH Lag[5]  0.026441 1.440 1.667  0.9980
ARCH Lag[7]  0.612549 2.315 1.543  0.9671

Nyblom stability test
------------------------------------
Joint Statistic:  7.2116
Individual Statistics:
mu     0.04141
ar1    0.59845
omega  0.10214
alpha1 0.26503
beta1  0.15513

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:        1.28 1.47 1.88
Individual Statistic:   0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value    prob sig
Sign Bias           0.2981 0.76568
Negative Sign Bias  1.9074 0.05658   *
Positive Sign Bias  0.7530 0.45150
Joint Effect        5.5255 0.13712


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     116.7    4.589e-16
2    30     129.8    1.029e-14
3    40     134.4    2.075e-12
4    50     151.0    2.495e-12
```

Model	Garch 1	Garch 2	Garch 3	Garch 4	Garch 5	Garch 6
ARMA	(1,0)	(0,1)	(1,1)	(1,2)	(2,1)	(2,2)
AIC	-6.0271	-6.0271	-6.0263	-6.0235	-6.0253	-6.0247

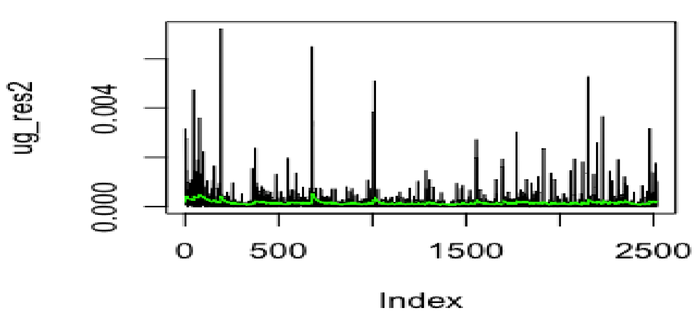
Surprisingly, if we use ugarchfit function for the estimation of the model and to find the best parameters, the best fitted model will be (1,1) as shown below, although its AIC value is not the lowest according to the comparison of all of the six models. According to the GARCH (m,s) model, where

$$\begin{aligned} y_t &= \sigma_t \epsilon_t \\ \sigma_t^2 &= \beta_1 \sigma_{t-1}^2 + \cdots + \beta_s \sigma_{t-s}^2 + \alpha_0 + \alpha_1 y_{t-1}^2 + \cdots + \alpha_m y_{t-m}^2 \end{aligned}$$

The optimal parameters for the GARCH model are as shown, where ar1=-0.687981 is the coefficient of the mean model, alpha 1=0.033768 is the coefficient of the squared residuals, and beta1=0.940305 is the coefficient to the lagged variance. Then we can plot the squared residuals for the (1, 1) model and the estimated conditional variance together, where the variance is in green for ug_res2. Form the graph, we can see a few big volatility, and that the large volatility happens with large residuals. Next, we will do the forecasting. We will see the differences of the forecasted stock return for the next 20 days among these three models in the next section.

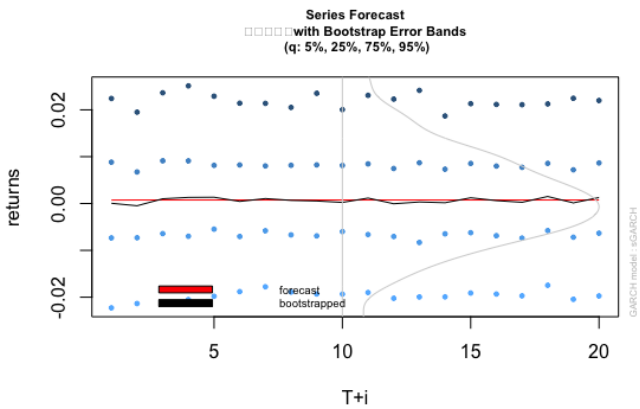
```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,1)
Mean Model  : ARFIMA(1,0,1)
Distribution      : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000749    0.000228   3.2850  0.00102
ar1    -0.687981    0.471914  -1.4579  0.14488
ma1     0.694867    0.468541   1.4830  0.13806
omega   0.000004    0.000000  11.9502  0.00000
alpha1  0.033768    0.002389  14.1328  0.00000
beta1   0.940305    0.004040 232.7623  0.00000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000749    0.000220   3.4038 0.000665
ar1    -0.687981    0.197431  -3.4847 0.000493
ma1     0.694867    0.200265   3.4697 0.000521
omega   0.000004    0.000001   5.1392 0.000000
alpha1  0.033768    0.004800   7.0343 0.000000
beta1   0.940305    0.004161 225.9923 0.000000

LogLikelihood : 7593.171

Information Criteria
------------------------------------

Akaike       -6.0263
Bayes        -6.0125
Shibata      -6.0264
Hannan-Quinn -6.0213

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.7286  0.3933
Lag[2*(p+q)+(p+q)-1][5]    2.1548  0.9215
Lag[4*(p+q)+(p+q)-1][9]    2.8437  0.9129
```

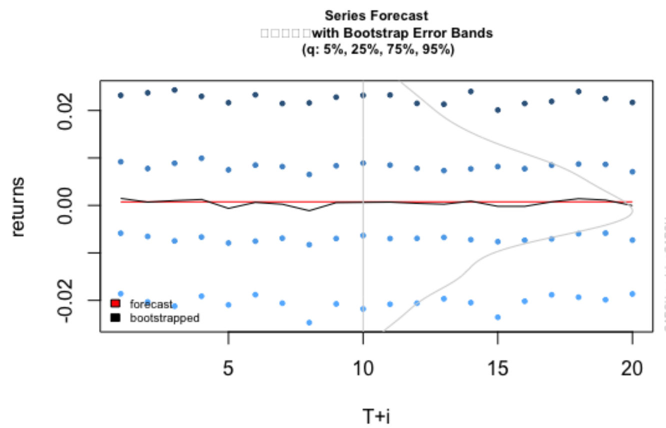
```
d.o.f=2
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                      4.173 0.04107
Lag[2*(p+q)+(p+q)-1][5]     5.206 0.13747
Lag[4*(p+q)+(p+q)-1][9]     6.018 0.29609
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]  0.005754 0.500 2.000  0.9395
ARCH Lag[5]  0.025966 1.440 1.667  0.9981
ARCH Lag[7]  0.570989 2.315 1.543  0.9715

Nyblom stability test
------------------------------------
Joint Statistic:  9.512
Individual Statistics:
mu     0.04101
ar1    0.48800
ma1    0.48422
omega  0.15809
alpha1 0.27749
beta1  0.15997

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           0.2163 0.82879
Negative Sign Bias  1.9698 0.04897  **
Positive Sign Bias  0.7270 0.46731
Joint Effect        5.6393 0.13054


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     116.9    4.115e-16
2    30     124.1    9.892e-14
3    40     136.5    9.743e-13
4    50     141.1    7.452e-11
```



III. Model Forecasting

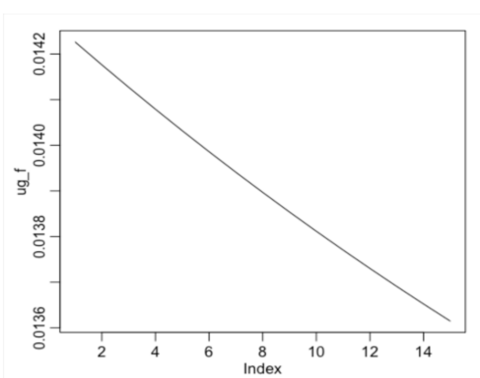
For the forecasting, ugarchboot function is used for both (1,0) and (0,1) models, and their forecasts are named rcostPredict1 and rcostPreict2. As shown in two of these forecasting graph, the red line represents the forecasting line, and they are very similar to each other visually speaking. If we look at the forecasted value of the return for the first day T+1, ARMA(1,0) produces a return of 0.000047, and ARM(0,1) produces a return of 0.001481. We want to compare the numbers with what ARMA(1,1) produce.

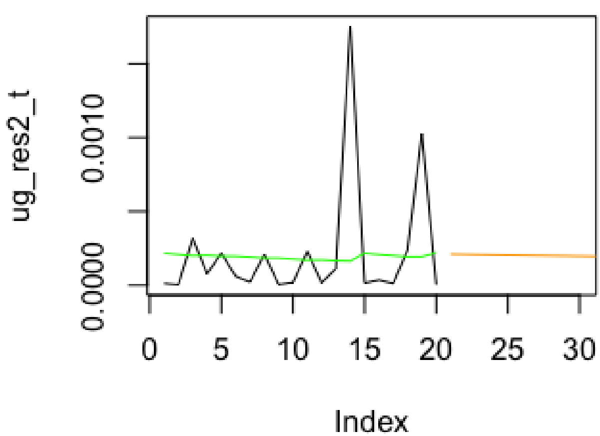




The forecasting for model (1, 1) is processed slightly different as ugarchfore function and the two drawers inside, @model and @forecast, are used this time to produce the forecasting line of ug_f of the sigma or the forecast of the variance. As we can see in the graph, there is a decreasing volatility. Then we put the forecast, the variance and the squared residual altogether, and it gives us the three-colored graph below. The orange line represents the forecast, the green line represents the variance, and the squared residual is depicted in black.

The forecasting for model (1, 1) is processed slightly different, ugarchfore function and the two drawers inside, @model and @forecast, are used this time to produce the forecasting line of ug_f of the sigma or the forecast of the variance. As we can see in the graph, there is a decreasing volatility. Then we put the forecast, the variance and the squared residual altogether, and it gives us three-colored graph below. The orange line represents the forecast, the green line represents the variance, and the squared residual is depicted in black.





It is convenient to get the time series forecast using GARCH Bootstrap, and from the result shown below, the mean forecasted value for the first day, that is t+1, 11/30/2018 is 0.0006601 with a sigma of 0.01452, which is right between the first day forecast for the other two models ARMA(1,0) and ARMA (0,1), that is 0.000047 and 0.001481. Thus, although (1,1) does not have the smallest AIC, since the purpose of this project is to help investors take action in terms of buying or selling the stock, among the three predicted stock return, the highest being the best scenario, the lowest being the worst scenario, we want to select the predicted return in the middle from model (1,1). And the model would be:

```
*------------------------------------*
*       GARCH Model Forecast         *
*------------------------------------*
Model: sGARCH
Horizon: 20
Roll Steps: 0
Out of Sample: 0

0-roll forecast [T0=2018-11-29]:
        Series   Sigma
T+1  0.0006601 0.01452
T+2  0.0008101 0.01446
T+3  0.0007069 0.01440
T+4  0.0007779 0.01435
T+5  0.0007291 0.01429
T+6  0.0007627 0.01424
T+7  0.0007395 0.01419
T+8  0.0007554 0.01414
T+9  0.0007445 0.01409
T+10 0.0007520 0.01404
T+11 0.0007469 0.01399
T+12 0.0007504 0.01395
T+13 0.0007480 0.01390
T+14 0.0007496 0.01386
T+15 0.0007485 0.01381
T+16 0.0007493 0.01377
T+17 0.0007487 0.01373
T+18 0.0007491 0.01369
T+19 0.0007489 0.01365
T+20 0.0007490 0.01361
```

- **Code**

```
> startDate = as.Date("2008-12-01")
> endDate = as.Date("2018-11-30")
> getSymbols("COST", from = startDate, to = endDate)
> rCOST <- dailyReturn(COST)
> plot.ts(rCOST)
> plot(rCOST)
> library(quantmod)
> library(rugarch)
> ug_spec = ugarchspec()
> ug_spec

> rcost1 <- ugarchspec(variance.model =
list(model="sGARCH",garchOrder=c(1,1)),mean.model=list(armaOrder=c(1,0)),distributi
on.model="norm")
> rcostGarch1 <- ugarchfit(spec = rcost0, data = rCOST)
> rcostGarch1

> rcost2 <- ugarchspec(variance.model =
list(model="sGARCH",garchOrder=c(1,1)),mean.model=list(armaOrder=c(0,1)),distributi
on.model="norm")
> rcostGarch2 <- ugarchfit(spec = rcost2, data = rCOST)
> rcostGarch2

> rcost3 <- ugarchspec(variance.model =
list(model="sGARCH",garchOrder=c(1,1)),mean.model=list(armaOrder=c(1,1)),distributi
on.model="norm")
> rcostGarch3 <- ugarchfit(spec = rcost3, data = rCOST)
> rcostGarch3

> rcost4 <- ugarchspec(variance.model =
list(model="sGARCH",garchOrder=c(1,1)),mean.model=list(armaOrder=c(1,2)),distributi
on.model="norm")
> rcostGarch4 <- ugarchfit(spec = rcost4, data = rCOST)
> rcostGarch4

> rcost5 <- ugarchspec(variance.model =
list(model="sGARCH",garchOrder=c(1,1)),mean.model=list(armaOrder=c(2,1)),distributi
on.model="norm")
> rcostGarch5 <- ugarchfit(spec = rcost5, data = rCOST)
> rcostGarch5
```

```
> rcost6 <- ugarchspec(variance.model =
list(model="sGARCH",garchOrder=c(1,1)),mean.model=list(armaOrder=c(2,2)),distributi
on.model="norm")
> rcostGarch6 <- ugarchfit(spec = rcost6, data = rCOST)
> rcostGarch6

> rcostPredict1<-ugarchboot(rcostGarch1, n.ahead=20, method=c("Partial",
"Full")[1])
> rcostPredict1
> plot(rcostPredict1)
Make a plot selection (or 0 to exit):
1:   Parameter Density Plots
2:   Series Standard Error Plots
3:   Sigma  Standard Error Plots
Selection: 2

> rcostPredict2<-ugarchboot(rcostGarch2, n.ahead=20, method=c("Partial",
"Full")[1])
> rcostPredict2
> plot(rcostPredict2)
Make a plot selection (or 0 to exit):
1:   Parameter Density Plots
2:   Series Standard Error Plots
3:   Sigma  Standard Error Plots
Selection: 2


> ugfit=ugarchfit(spec = ug_spec,data=rCOST)
> ugfit

> paste("Elements in the @model slot")
> names(ugfit@model)
> paste("Elements in the @fit slot")
> names(ugfit@fit)

> ugfit@fit$coef
> ug_var <- ugfit@fit$var   # save the estimated conditional variances
> ug_res2 <- (ugfit@fit$residuals)^2   # save the estimated squared residuals
> plot(ug_res2, type = "l")

> lines(ug_var, col = "green")
> ugfore <- ugarchforecast(ugfit, n.ahead = 20)
> ug_f <- ugfore@forecast$sigmaFor
> plot(ug_f, type = "l")

> ug_var_t <- c(tail(ug_var,20),rep(NA,10))  # gets the last 20 observations
```

```
> ug_res2_t <- c(tail(ug_res2,20),rep(NA,10))  # gets the last 20 observations
> ug_f <- c(rep(NA,20),(ug_f)^2)

> plot(ug_res2_t, type = "l")
> lines(ug_f, col = "orange")
> lines(ug_var_t, col = "green")
> ugfore
```