Circuit Breakers and LULD efficacy

Davide Zicca

Motivation

This work is meant to provide to the literature more evidence in support of the idea that Circuit Breaker regulations were not effective in avoiding huge spikes (as Flash Crashes continue to be a burden for financial markets) and thus, reducing volatility. The first analysis, which has been conducted by fitting a ARFIMA(1,0,0) - eGARCH(1,1) with Student's t-distribution, has given a further confirmation of the results already achieved by the literature (Santoni and Liu, 1993, Subrahmanyam, 1994, Liang, 2017). In particular, the announcement of new 'Circuit Breaker' regulations tends to increase volatility, as traders tend to increase the trading volume, to exit from markets where a Circuit Breaker has been activated and thus, increasing price variability. The novelty, proposed in this work, is the second analysis: the LULD Regulation and the effects that its implementation have brought to the U.S. stock volatility. It has been tested whether the two LULD Phases affected volatility. The estimated ARFIMA(1,0,0) - eGARCH(1,1) model, based on the Normal Distribution, provided a clear indication that only the first Phase affected the volatility, by increasing it.

The following ones are the lines of code, necessary to achieve the results that have been already showed:

```
#DATA
library(tidyquant)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
## Loading required package: PerformanceAnalytics
```

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

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

```
##
## Attaching package: 'zoo'
```

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

```
##
## Attaching package: 'PerformanceAnalytics'
```

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

```
## Loading required package: quantmod
```

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

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

```
## == Need to Learn tidyquant? ============================================
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio Optimizat
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

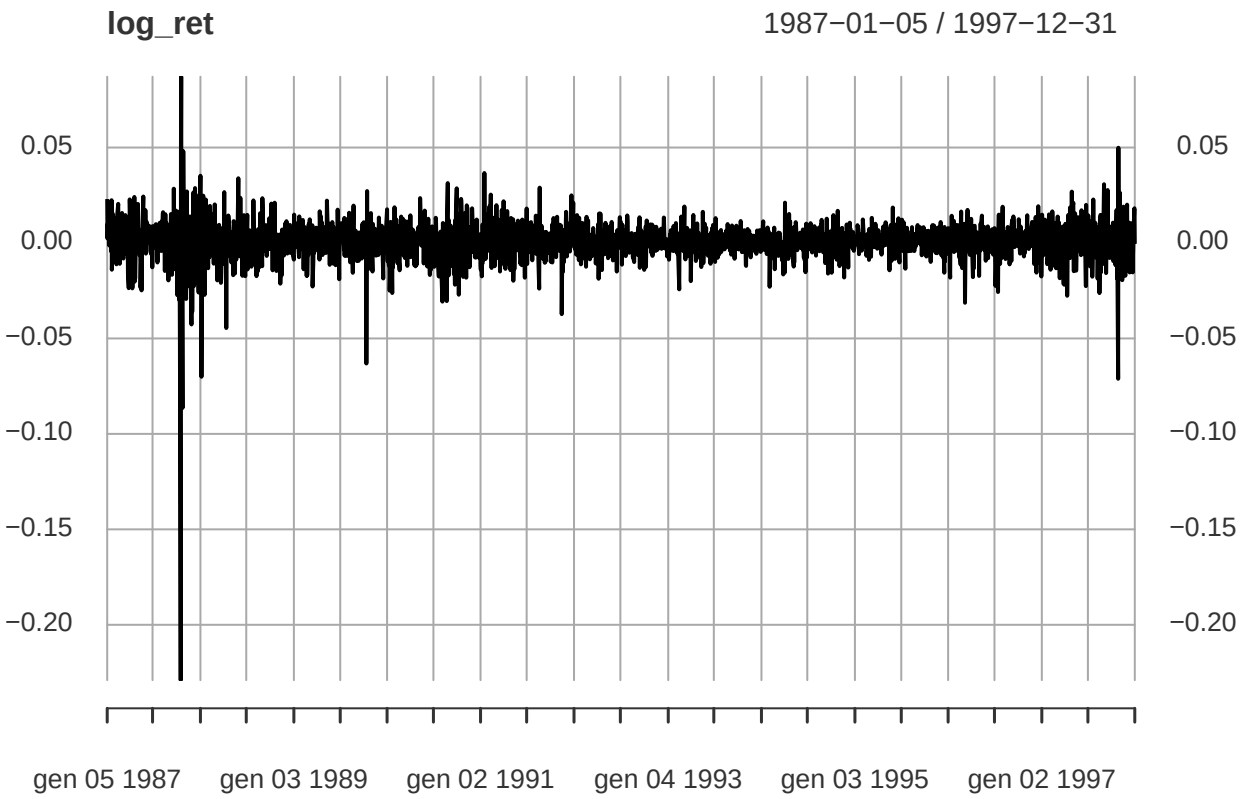
```
# Downloading SP500 using the library tidyquant

SP500= getSymbols("^GSPC", from = '1987-01-01',
                  to = "1998-01-01", auto.assign = FALSE)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
head(SP500)
```

```
##            GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted
## 1987-01-02    242.17    246.45   242.17     246.45    91880000        246.45
## 1987-01-05    246.45    252.57   246.45     252.19   181900000        252.19
## 1987-01-06    252.20    253.99   252.14     252.78   189300000        252.78
## 1987-01-07    252.78    255.72   252.65     255.33   190900000        255.33
## 1987-01-08    255.36    257.28   254.97     257.28   194500000        257.28
## 1987-01-09    257.26    259.20   256.11     258.73   193000000        258.73
```

```
logret = diff(log(SP500$GSPC.Adjusted))
log_ret <- logret[-1]

plot(log_ret, length(log_ret))
```




```
#Stationarity test
library(tseries)
adf.test(log_ret, alternative = "stationary")
```

```
## Warning in adf.test(log_ret, alternative = "stationary"): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  log_ret
## Dickey-Fuller = -14.056, Lag order = 14, p-value = 0.01
## alternative hypothesis: stationary
```

```
#ARCH EFFECT

library(fDMA)
archtest(as.vector(log_ret))
```

```
##
##  Engle's LM ARCH Test
##
## data:  as.vector(log_ret)
## statistic = 34.188, lag = 1, p-value = 5.002e-09
## alternative hypothesis: ARCH effects of order 1 are present
```

First analysis on Circuit Breakers effects on stock volatility

```
#DUMMY VARIABLES

n=length(log_ret)
#10/22/87

e1=which(row.names(as.matrix(log_ret))=="1987-10-20")
f1=which(row.names(as.matrix(log_ret))=="1987-10-21")
g1=which(row.names(as.matrix(log_ret))=="1987-10-22")
h1=which(row.names(as.matrix(log_ret))=="1987-10-23")
j1=which(row.names(as.matrix(log_ret))=="1987-10-26")

d1=rep(0,n)

d1[e1]=1
d1[f1]=1
d1[g1]=1
d1[h1]=1
d1[j1]=1

#1/14/88

e2=which(row.names(as.matrix(log_ret))=="1988-01-12")
f2=which(row.names(as.matrix(log_ret))=="1988-01-13")
g2=which(row.names(as.matrix(log_ret))=="1988-01-14")
h2=which(row.names(as.matrix(log_ret))=="1988-01-15")
j2=which(row.names(as.matrix(log_ret))=="1988-01-18")

d2=rep(0,n)

d2[e2]=1
d2[f2]=1
d2[g2]=1
d2[h2]=1
d2[j2]=1

#2/4/88
e3=which(row.names(as.matrix(log_ret))=="1988-02-02")
f3=which(row.names(as.matrix(log_ret))=="1988-02-03")
g3=which(row.names(as.matrix(log_ret))=="1988-02-04")
h3=which(row.names(as.matrix(log_ret))=="1988-02-05")
j3=which(row.names(as.matrix(log_ret))=="1988-02-08")

d3=rep(0,n)

d3[e3]=1
d3[f3]=1
d3[g3]=1
d3[h3]=1
d3[j3]=1

#10/20/88
```

```
e4=which(row.names(as.matrix(log_ret))=="1988-10-18")
f4=which(row.names(as.matrix(log_ret))=="1988-10-19")
g4=which(row.names(as.matrix(log_ret))=="1988-10-20")
h4=which(row.names(as.matrix(log_ret))=="1988-10-21")
j4=which(row.names(as.matrix(log_ret))=="1988-10-24")

d4=rep(0,n)

d4[e4]=1
d4[f4]=1
d4[g4]=1
d4[h4]=1
d4[j4]=1
#8/1/90

e5=which(row.names(as.matrix(log_ret))=="1990-07-30")
f5=which(row.names(as.matrix(log_ret))=="1990-07-31")
g5=which(row.names(as.matrix(log_ret))=="1990-08-01")
h5=which(row.names(as.matrix(log_ret))=="1990-08-02")
j5=which(row.names(as.matrix(log_ret))=="1990-08-03")

d5=rep(0,n)

d5[e5]=1
d5[f5]=1
d5[g5]=1
d5[h5]=1
d5[j5]=1
#7/22/96

e6=which(row.names(as.matrix(log_ret))=="1996-07-18")
f6=which(row.names(as.matrix(log_ret))=="1996-07-19")
g6=which(row.names(as.matrix(log_ret))=="1996-07-22")
h6=which(row.names(as.matrix(log_ret))=="1996-07-23")
j6=which(row.names(as.matrix(log_ret))=="1996-07-24")

d6=rep(0,n)

d6[e6]=1
d6[f6]=1
d6[g6]=1
d6[h6]=1
d6[j6]=1
#3/3/97
e7=which(row.names(as.matrix(log_ret))=="1997-02-27")
f7=which(row.names(as.matrix(log_ret))=="1997-02-28")
g7=which(row.names(as.matrix(log_ret))=="1997-03-03")
h7=which(row.names(as.matrix(log_ret))=="1997-03-04")
j7=which(row.names(as.matrix(log_ret))=="1997-03-05")

d7=rep(0,n)

d7[e7]=1
```

```
d7[f7]=1
d7[g7]=1
d7[h7]=1
d7[j7]=1

df_dummies=matrix(c(d1,d2,d3,d4,d5,d6,d7), ncol = 7)

# GARCH MODELS assuming a Normal distribution
#sGARCH
library(rugarch)
```

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

```
##
## Attaching package: 'rugarch'
```

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

```
s_garchMod <- ugarchspec(mean.model = list(armaOrder = c(1, 0), include.mean = TRUE
),
variance.model = list(model = 'sGARCH',
                      garchOrder = c(1, 1),
                      external.regressors = df_dummies
),
distribution.model = "norm")

s_garchFit <- ugarchfit(spec=s_garchMod, data=log_ret)
s_garchFit
```

```
##
## *---------------------------------*
## *          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      0.000681    0.000139   4.885991 0.000001
## ar1     0.034238    0.019737   1.734658 0.082801
## omega   0.000002    0.000001   3.763065 0.000168
## alpha1  0.105806    0.000893 118.426971 0.000000
## beta1   0.875760    0.003518 248.906968 0.000000
## vxreg1  0.000000    0.001222   0.000007 0.999994
## vxreg2  0.000000    0.000042   0.000124 0.999901
## vxreg3  0.000000    0.000009   0.000375 0.999701
## vxreg4  0.000000    0.000003   0.010005 0.992017
## vxreg5  0.000039    0.000030   1.303237 0.192494
```

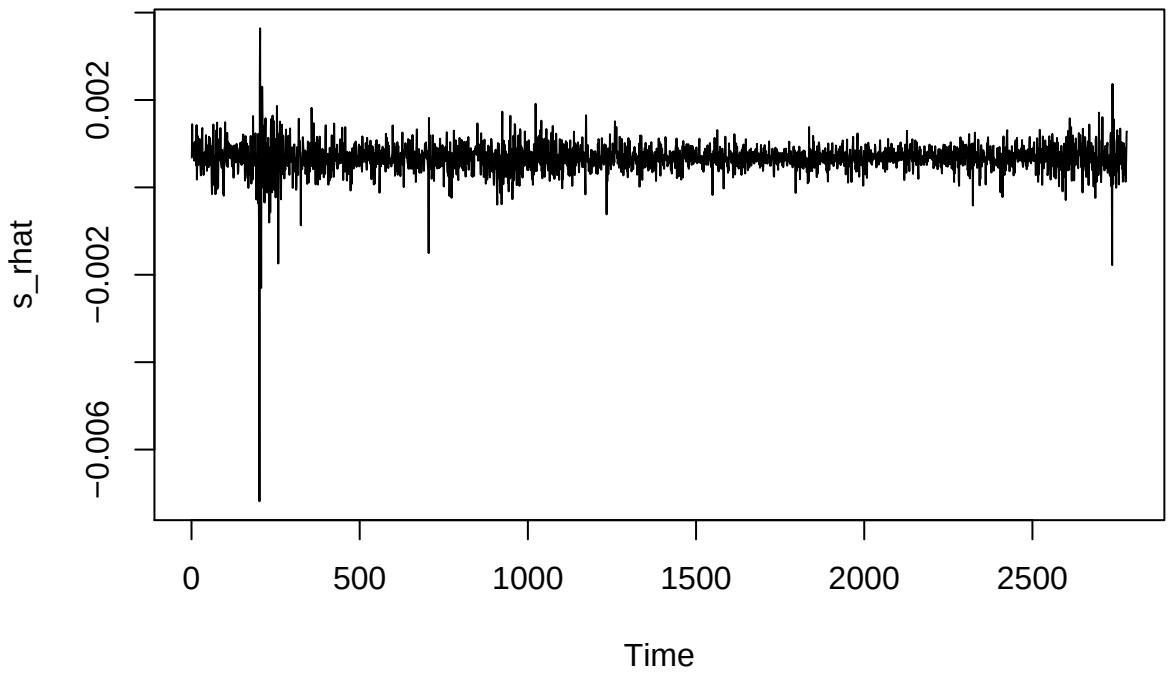
```
## vxreg6  0.000000    0.000029   0.000000 1.000000
## vxreg7  0.000000    0.000003   0.006297 0.994976
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000681    0.000253   2.690277 0.007139
## ar1     0.034238    0.028212   1.213570 0.224912
## omega   0.000002    0.000003   0.630088 0.528637
## alpha1  0.105806    0.001017 104.010095 0.000000
## beta1   0.875760    0.011113  78.806466 0.000000
## vxreg1  0.000000    0.005394   0.000002 0.999999
## vxreg2  0.000000    0.000103   0.000051 0.999960
## vxreg3  0.000000    0.000024   0.000136 0.999892
## vxreg4  0.000000    0.000014   0.001962 0.998435
## vxreg5  0.000039    0.000013   2.976127 0.002919
## vxreg6  0.000000    0.000041   0.000000 1.000000
## vxreg7  0.000000    0.000020   0.001093 0.999128
##
## LogLikelihood : 9384.27
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.7402
## Bayes        -6.7146
## Shibata      -6.7403
## Hannan-Quinn -6.7310
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      2.645 0.10385
## Lag[2*(p+q)+(p+q)-1][2]     2.671 0.06628
## Lag[4*(p+q)+(p+q)-1][5]     5.715 0.05930
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.466  0.2260
## Lag[2*(p+q)+(p+q)-1][5]     1.541  0.7298
## Lag[4*(p+q)+(p+q)-1][9]     1.963  0.9093
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]  0.000941 0.500 2.000  0.9755
## ARCH Lag[5]  0.148262 1.440 1.667  0.9772
## ARCH Lag[7]  0.530647 2.315 1.543  0.9755
##
## Nyblom stability test
## ------------------------------------
```

```
## Joint Statistic:  60.104
## Individual Statistics:
## mu     0.193060
## ar1    0.325178
## omega  1.841970
## alpha1 0.424701
## beta1  0.478137
## vxreg1 9.890775
## vxreg2 6.275548
## vxreg3 0.026626
## vxreg4 2.178759
## vxreg5 0.001121
## vxreg6 0.332870
## vxreg7 0.051772
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.69 2.96 3.51
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## -----------------------------------
##                    t-value      prob sig
## Sign Bias           0.03479 0.9722517
## Negative Sign Bias  3.39207 0.0007034 ***
## Positive Sign Bias  1.21699 0.2237118
## Joint Effect       19.57646 0.0002077 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     129.4    1.871e-18
## 2    30     143.0    5.042e-17
## 3    40     151.6    3.357e-15
## 4    50     169.9    3.019e-15
##
##
## Elapsed time : 1.314238
```

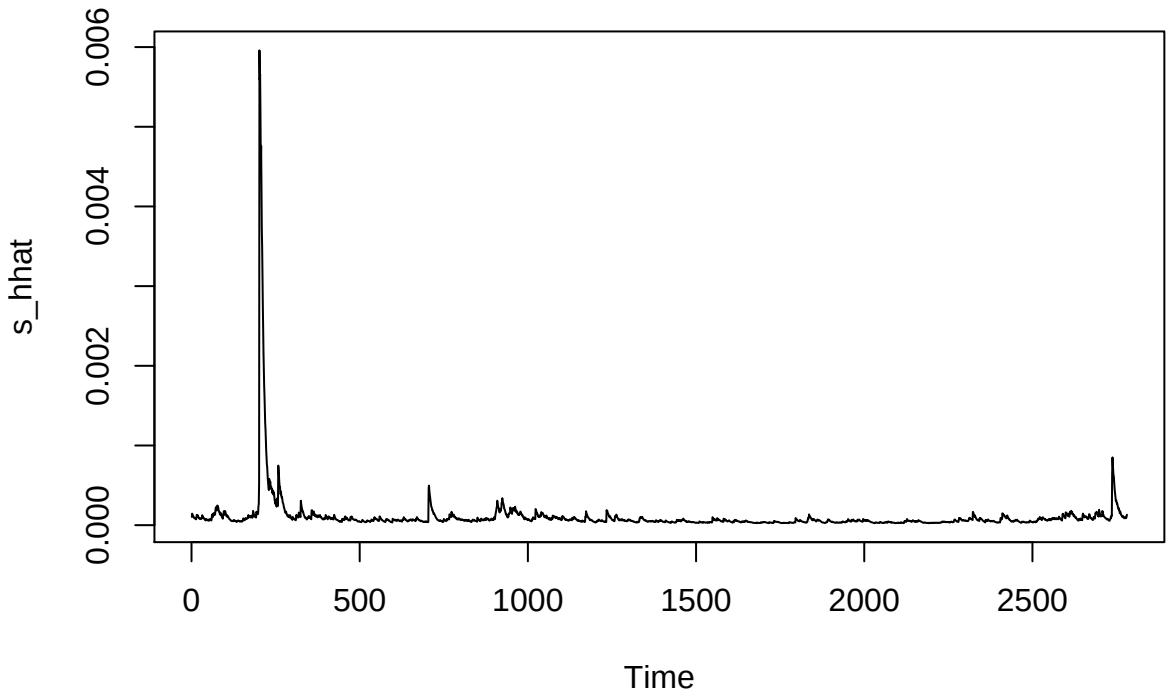
```
## Results review 1
coef(s_garchFit)
```

```
##           mu          ar1        omega       alpha1        beta1       vxreg1
## 6.811716e-04 3.423771e-02 2.129495e-06 1.058060e-01 8.757601e-01 9.032225e-09
##       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6       vxreg7
## 5.196942e-09 3.310499e-09 2.822589e-08 3.896673e-05 6.353554e-13 2.135644e-08
```

```
s_rhat <- s_garchFit@fit$fitted.values
plot.ts(s_rhat)
```




```
s_hhat <- ts(s_garchFit@fit$sigma^2)
plot.ts(s_hhat)
```




```
## Results review 2
fit.val     <- coef(s_garchFit)
fit.sd      <- diag(vcov(s_garchFit))
true.val = s_garchFit@fit$tval

fit.conf.lb <- fit.val + qnorm(0.025) * fit.sd
fit.conf.ub <- fit.val + qnorm(0.975) * fit.sd
print(fit.val)
```

```
##           mu          ar1        omega       alpha1        beta1       vxreg1
## 6.811716e-04 3.423771e-02 2.129495e-06 1.058060e-01 8.757601e-01 9.032225e-09
##       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6       vxreg7
## 5.196942e-09 3.310499e-09 2.822589e-08 3.896673e-05 6.353554e-13 2.135644e-08
```

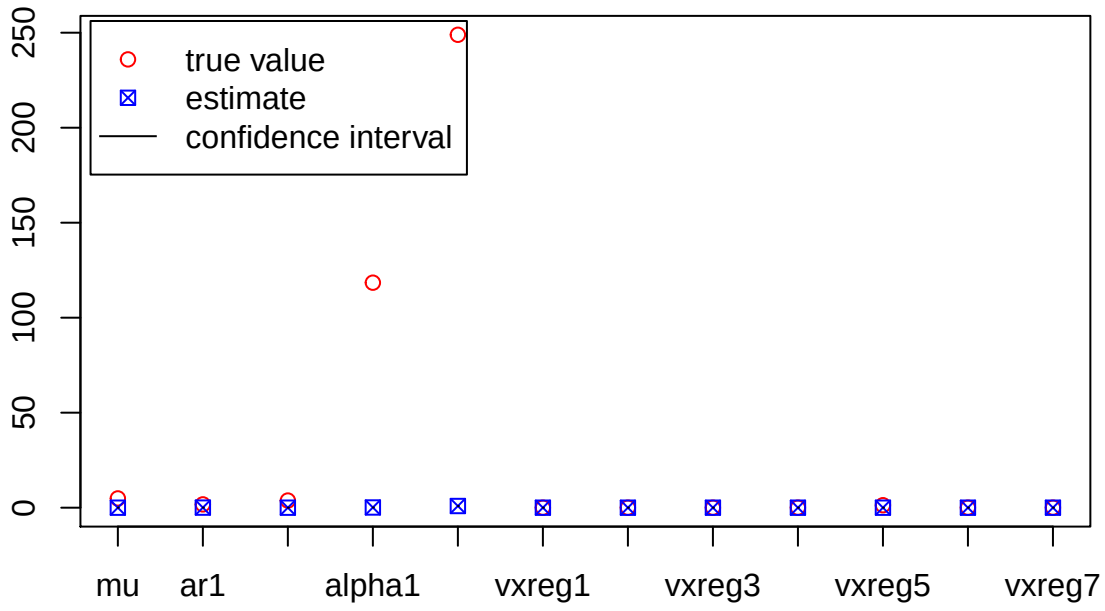
```
print(fit.sd)
```

```
##  [1]  1.943604e-08  3.895666e-04  3.202357e-13  7.982141e-07  1.237930e-05
##  [6] -1.493874e-06 -1.745845e-09 -7.810591e-11 -7.959431e-12  8.940069e-10
## [11]  8.448609e-10 -1.150296e-11
```

```
print(true.val)
```

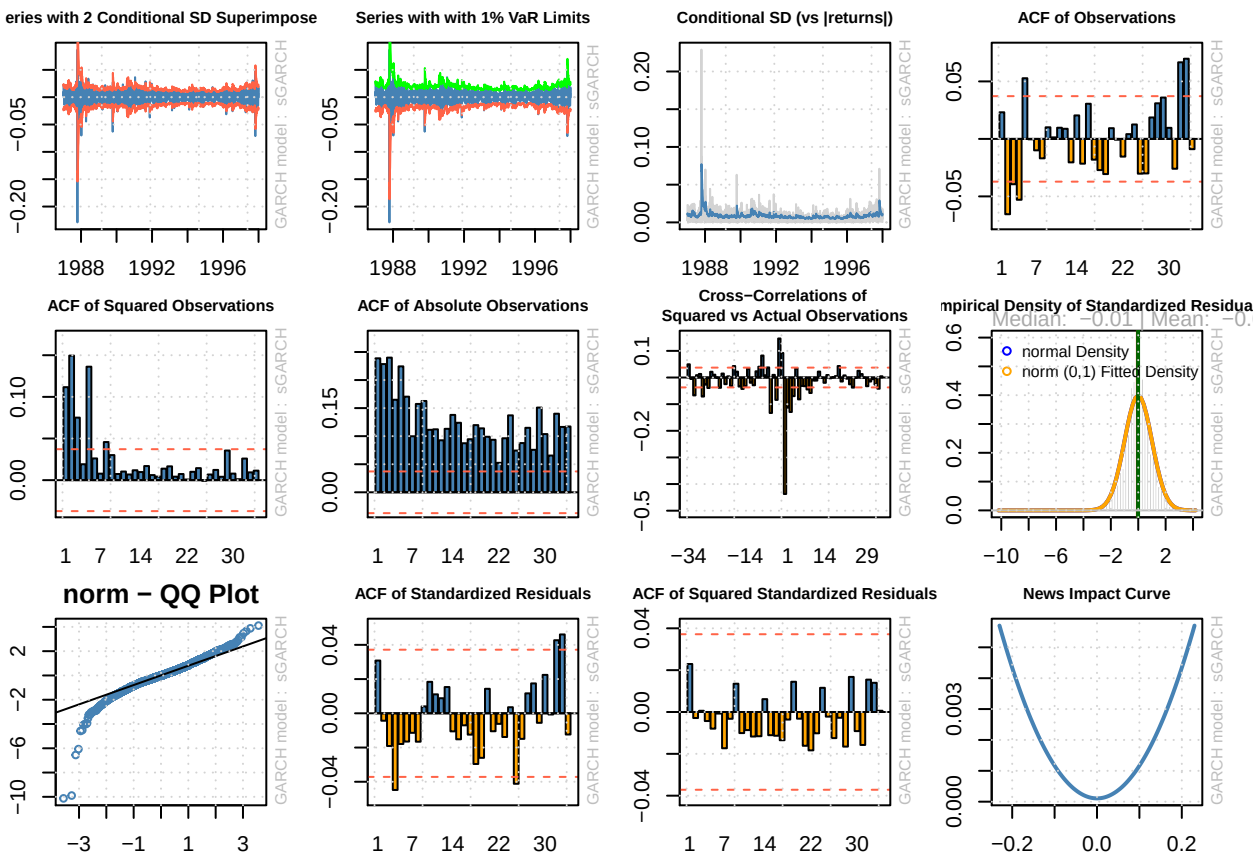
```
##           mu          ar1        omega       alpha1        beta1       vxreg1
## 4.885991e+00 1.734658e+00 3.763065e+00 1.184270e+02 2.489070e+02 7.389886e-06
##       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6       vxreg7
## 1.243784e-04 3.745860e-04 1.000476e-02 1.303237e+00 2.185869e-08 6.296858e-03
```

```
plot(true.val, pch = 1, col = "red",
     ylim = range(c(fit.conf.lb, fit.conf.ub, true.val)),
     xlab = "", ylab = "", axes = FALSE)
box(); axis(1, at = 1:length(fit.val), labels = names(fit.val)); axis(2)
points(coef(s_garchFit), col = "blue", pch = 7)
for (i in 1:length(fit.val)) {
  lines(c(i,i), c(fit.conf.lb[i], fit.conf.ub[i]))
}
legend( "topleft", legend = c("true value", "estimate", "confidence interval"),
        col = c("red", "blue", 1), pch = c(1, 7, NA), lty = c(NA, NA, 1), inset = 0.01)
```




```
par(mfrow=c(2, 3))
par(mar = c(2, 2, 2, 2))
plot(s_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#gjrGARCH
gjr_garchMod <- ugarchspec(
  variance.model=list(model="gjrGARCH",
                      garchOrder=c(1,1), external.regressors = df_dummies ),
  mean.model=list(armaOrder=c(1,0),
                  include.mean=TRUE
  ),
  distribution.model="norm"
)
gjr_garchFit <- ugarchfit(spec=gjr_garchMod, data=log_ret)
coef(gjr_garchFit)
```

```
##           mu          ar1        omega       alpha1        beta1       gamma1
## 4.706382e-04 4.779133e-02 2.713057e-06 3.542937e-02 8.740424e-01 1.232974e-01
##       vxreg1       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6
## 9.200975e-09 5.765240e-09 5.258136e-09 3.479895e-08 9.636875e-08 3.098914e-10
##       vxreg7
## 1.908787e-08
```

```
gjr_garchFit
```

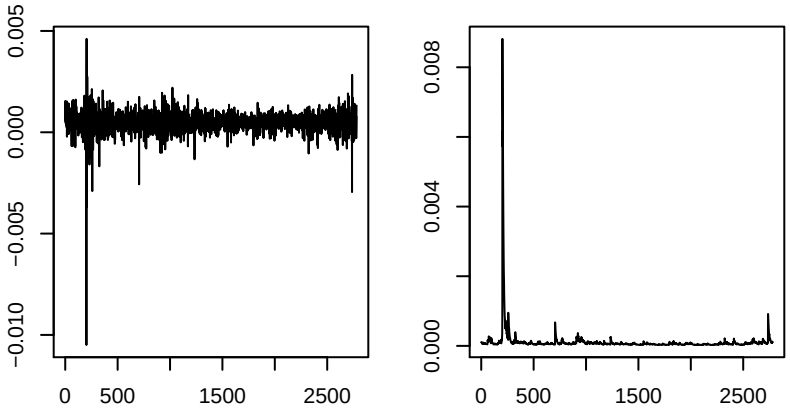
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
```

```
## ------------------------------------
## GARCH Model  : gjrGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000471    0.000106   4.428157 0.000010
## ar1     0.047791    0.019788   2.415197 0.015727
## omega   0.000003    0.000000   7.215401 0.000000
## alpha1  0.035429    0.003209  11.040880 0.000000
## beta1   0.874042    0.004868 179.559601 0.000000
## gamma1  0.123297    0.006114  20.165937 0.000000
## vxreg1  0.000000    0.001054   0.000009 0.999993
## vxreg2  0.000000    0.000148   0.000039 0.999969
## vxreg3  0.000000    0.000002   0.003019 0.997591
## vxreg4  0.000000    0.000001   0.062847 0.949888
## vxreg5  0.000000    0.000008   0.012061 0.990377
## vxreg6  0.000000    0.000026   0.000012 0.999991
## vxreg7  0.000000    0.000011   0.001713 0.998634
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000471    0.000344  1.367242 0.171550
## ar1     0.047791    0.025801  1.852309 0.063981
## omega   0.000003    0.000002  1.397693 0.162205
## alpha1  0.035429    0.053475  0.662544 0.507623
## beta1   0.874042    0.014364 60.848264 0.000000
## gamma1  0.123297    0.091052  1.354136 0.175693
## vxreg1  0.000000    0.003746  0.000002 0.999998
## vxreg2  0.000000    0.000451  0.000013 0.999990
## vxreg3  0.000000    0.000004  0.001180 0.999059
## vxreg4  0.000000    0.000001  0.026586 0.978790
## vxreg5  0.000000    0.000015  0.006497 0.994816
## vxreg6  0.000000    0.000156  0.000002 0.999998
## vxreg7  0.000000    0.000020  0.000971 0.999225
##
## LogLikelihood : 9404.956
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.7544
## Bayes        -6.7267
## Shibata      -6.7544
## Hannan-Quinn -6.7444
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.094  0.2957
## Lag[2*(p+q)+(p+q)-1][2]     1.100  0.6789
## Lag[4*(p+q)+(p+q)-1][5]     3.681  0.2808
```

```
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.08143  0.7754
## Lag[2*(p+q)+(p+q)-1][5]   0.34966  0.9782
## Lag[4*(p+q)+(p+q)-1][9]   0.89798  0.9901
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.05316 0.500 2.000  0.8177
## ARCH Lag[5]   0.20537 1.440 1.667  0.9642
## ARCH Lag[7]   0.65759 2.315 1.543  0.9620
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  30.6529
## Individual Statistics:
## mu      0.50349
## ar1     0.33048
## omega   0.35684
## alpha1  0.68378
## beta1   0.73125
## gamma1  0.43983
## vxreg1 11.72363
## vxreg2  6.75182
## vxreg3  0.32478
## vxreg4  1.02959
## vxreg5  0.97464
## vxreg6  0.34190
## vxreg7  0.09879
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.89 3.15 3.69
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           0.2058 0.83695
## Negative Sign Bias  2.0601 0.03948  **
## Positive Sign Bias  1.1551 0.24815
## Joint Effect        9.9600 0.01891  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     116.2    5.613e-16
## 2    30     121.8    2.463e-13
```

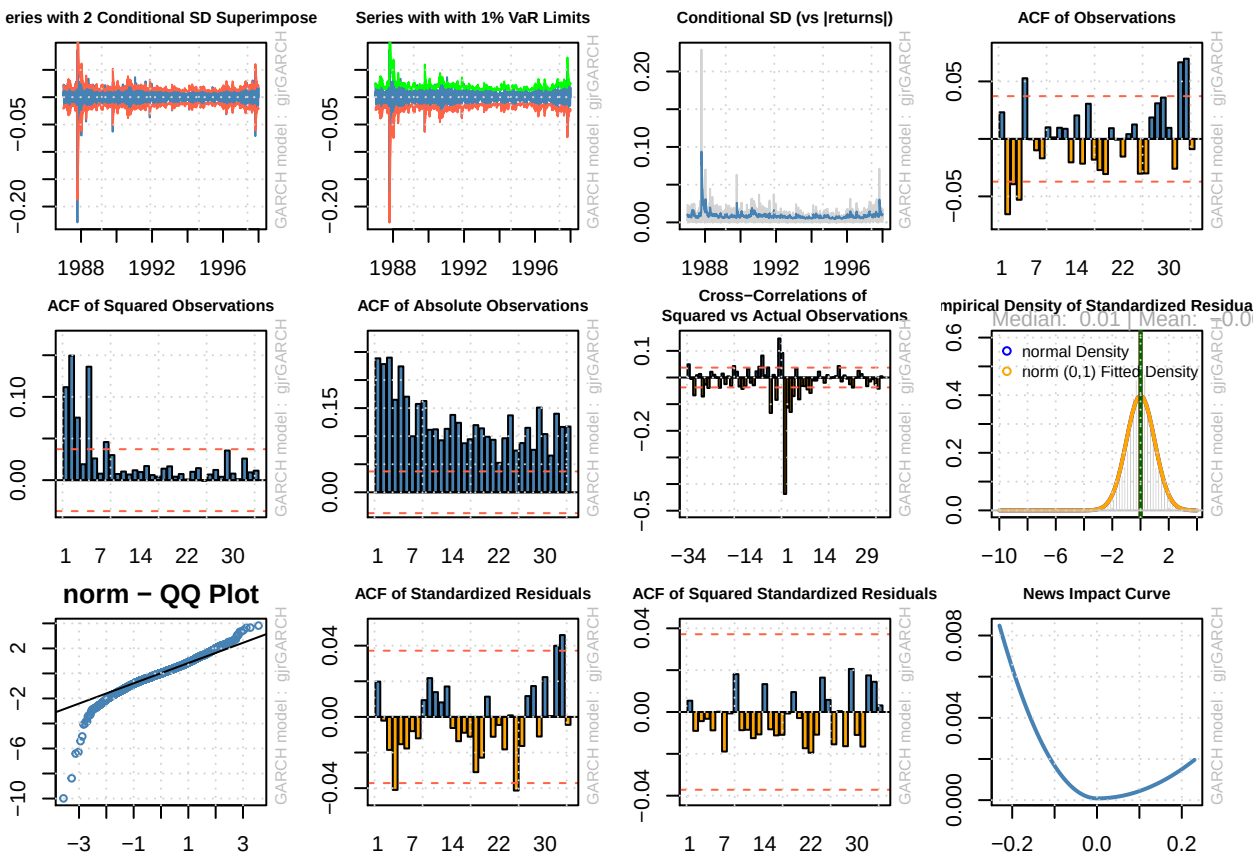
```
## 3    40     144.0    5.880e-14
## 4    50     146.1    1.381e-11
##
##
## Elapsed time : 1.509532
```

```
gjr_rhat <- gjr_garchFit@fit$fitted.values
plot.ts(gjr_rhat)
gjr_hhat <- ts(gjr_garchFit@fit$sigma^2)
plot.ts(gjr_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(gjr_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#eGARCH
e_garchMod <- ugarchspec(
  variance.model=list(model="eGARCH",
                      garchOrder=c(1,1), external.regressors = df_dummies ),
  mean.model=list(armaOrder=c(1,0),
                  include.mean=TRUE
  ),
  distribution.model="norm"
)
e_garchFit <- ugarchfit(spec=e_garchMod, data=log_ret)
coef(e_garchFit)
```

```
##           mu          ar1        omega       alpha1        beta1
##  0.0003827276  0.0384915691 -0.2046065717 -0.0897420253  0.9773735780
##        gamma1       vxreg1       vxreg2       vxreg3       vxreg4
##  0.1612022863 -0.1784463983 -0.1148208973 -0.0945015117  0.0574193815
##       vxreg5       vxreg6       vxreg7
##  0.1548938323 -0.2241254159 -0.0352817475
```

```
e_garchFit
```

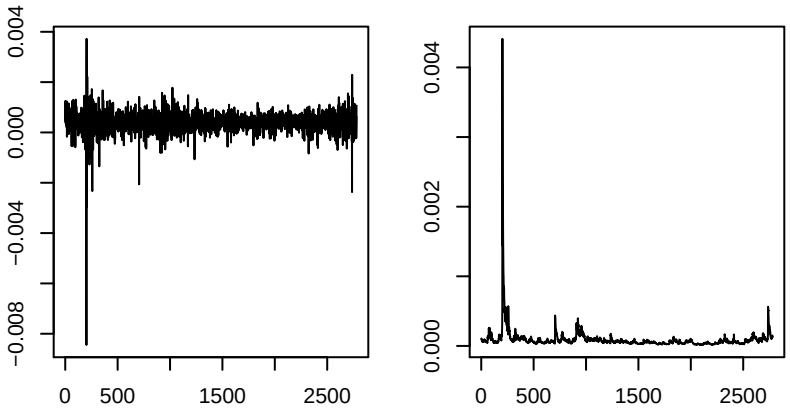
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
```

```
## ------------------------------------
## GARCH Model  : eGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000383    0.000103   3.73113 0.000191
## ar1     0.038492    0.018484   2.08240 0.037306
## omega  -0.204607    0.011862 -17.24859 0.000000
## alpha1 -0.089742    0.012712  -7.05985 0.000000
## beta1   0.977374    0.001316 742.70921 0.000000
## gamma1  0.161202    0.018927   8.51700 0.000000
## vxreg1 -0.178446    0.111477  -1.60074 0.109434
## vxreg2 -0.114821    0.118882  -0.96584 0.334123
## vxreg3 -0.094502    0.130444  -0.72446 0.468782
## vxreg4  0.057419    0.122627   0.46824 0.639610
## vxreg5  0.154894    0.128575   1.20470 0.228319
## vxreg6 -0.224125    0.122193  -1.83420 0.066625
## vxreg7 -0.035282    0.129604  -0.27223 0.785448
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000383    0.000079   4.85477 0.000001
## ar1     0.038492    0.017878   2.15306 0.031314
## omega  -0.204607    0.013442 -15.22126 0.000000
## alpha1 -0.089742    0.033457  -2.68229 0.007312
## beta1   0.977374    0.001351 723.53874 0.000000
## gamma1  0.161202    0.049277   3.27138 0.001070
## vxreg1 -0.178446    0.098822  -1.80573 0.070960
## vxreg2 -0.114821    0.066012  -1.73939 0.081967
## vxreg3 -0.094502    0.062659  -1.50819 0.131506
## vxreg4  0.057419    0.067522   0.85038 0.395112
## vxreg5  0.154894    0.067504   2.29459 0.021756
## vxreg6 -0.224125    0.084755  -2.64438 0.008184
## vxreg7 -0.035282    0.064352  -0.54827 0.583509
##
## LogLikelihood : 9419.462
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.7648
## Bayes        -6.7371
## Shibata      -6.7648
## Hannan-Quinn -6.7548
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.771  0.1832
## Lag[2*(p+q)+(p+q)-1][2]     1.822  0.2805
## Lag[4*(p+q)+(p+q)-1][5]     4.581  0.1472
```

```
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.8656  0.3522
## Lag[2*(p+q)+(p+q)-1][5]    0.9060  0.8806
## Lag[4*(p+q)+(p+q)-1][9]    1.3730  0.9653
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.01591 0.500 2.000  0.8996
## ARCH Lag[5]   0.08464 1.440 1.667  0.9896
## ARCH Lag[7]   0.42214 2.315 1.543  0.9849
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.3795
## Individual Statistics:
## mu     0.4839837
## ar1    0.2798858
## omega  0.6104227
## alpha1 0.0933716
## beta1  0.5983516
## gamma1 0.0978513
## vxreg1 0.0012195
## vxreg2 0.0003748
## vxreg3 0.0013128
## vxreg4 0.0009336
## vxreg5 0.0005639
## vxreg6 0.0016619
## vxreg7 0.0005274
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.89 3.15 3.69
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value     prob sig
## Sign Bias           0.3204 0.748676
## Negative Sign Bias  3.1572 0.001610 ***
## Positive Sign Bias  1.0562 0.290980
## Joint Effect       14.5938 0.002199 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     117.1    3.806e-16
## 2    30     124.1    9.871e-14
```

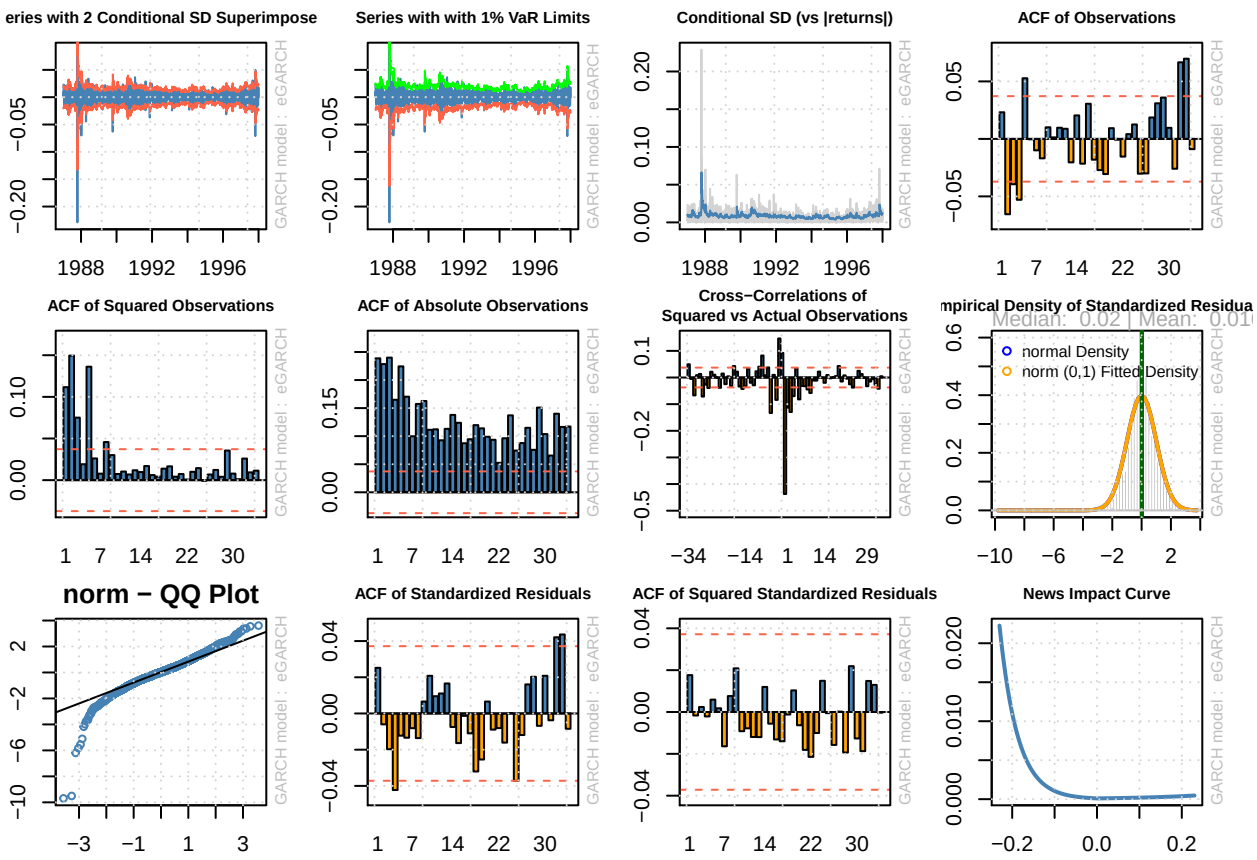
```
## 3    40     136.2    1.089e-12
## 4    50     159.4    1.300e-13
##
##
## Elapsed time : 3.489493
```

```
e_rhat <- e_garchFit@fit$fitted.values
plot.ts(e_rhat)
e_hhat <- ts(e_garchFit@fit$sigma^2)
plot.ts(e_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(e_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
# apARCH
apARCHMod <- ugarchspec(variance.model=list(model="apARCH",
                                            garchOrder=c(1,1),
                                           external.regressors = df_dummies ),
                        mean.model=list(armaOrder=c(1,0)),
                        distribution.model="norm")
apARCHFit <- ugarchfit(spec=apARCHMod, data= log_ret)
coef(apARCHFit)
```

```
##           mu          ar1        omega       alpha1        beta1       gamma1
## 4.659356e-04 4.892219e-02 2.409258e-07 7.505760e-02 8.567407e-01 2.905291e-01
##        delta       vxreg1       vxreg2       vxreg3       vxreg4       vxreg5
## 2.522832e+00 9.994818e-09 9.930641e-09 9.872127e-09 1.036614e-08 1.067708e-08
##       vxreg6       vxreg7
## 9.699512e-09 1.032048e-08
```

```
apARCHFit
```

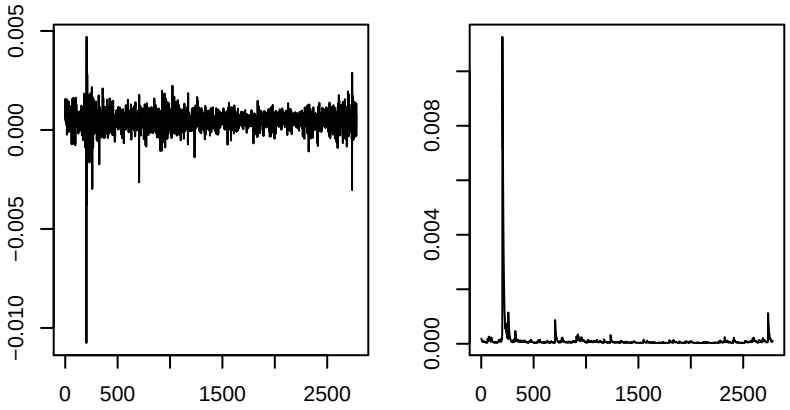
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : apARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
```

```
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000466    0.000150  3.112241 0.001857
## ar1     0.048922    0.021431  2.282731 0.022446
## omega   0.000000    0.000000  1.192373 0.233115
## alpha1  0.075058    0.002070 36.253493 0.000000
## beta1   0.856741    0.024203 35.398467 0.000000
## gamma1  0.290529    0.053116  5.469753 0.000000
## delta   2.522832    0.065925 38.268433 0.000000
## vxreg1  0.000000    0.000310  0.000032 0.999974
## vxreg2  0.000000    0.000000  0.552629 0.580517
## vxreg3  0.000000    0.000000  0.038988 0.968900
## vxreg4  0.000000    0.000000  0.040922 0.967358
## vxreg5  0.000000    0.000000  0.042247 0.966302
## vxreg6  0.000000    0.000000  0.038247 0.969490
## vxreg7  0.000000    0.000000  0.040818 0.967441
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000466    0.000264 1.766075 0.077383
## ar1     0.048922    0.024110 2.029088 0.042449
## omega   0.000000    0.000005 0.044990 0.964116
## alpha1  0.075058    0.032539 2.306677 0.021073
## beta1   0.856741    0.391588 2.187864 0.028680
## gamma1  0.290529    0.587117 0.494840 0.620713
## delta   2.522832    0.932690 2.704900 0.006833
## vxreg1  0.000000    0.001000 0.000010 0.999992
## vxreg2  0.000000    0.000000 0.048696 0.961162
## vxreg3  0.000000    0.000004 0.002383 0.998098
## vxreg4  0.000000    0.000004 0.002499 0.998006
## vxreg5  0.000000    0.000004 0.002591 0.997933
## vxreg6  0.000000    0.000004 0.002332 0.998139
## vxreg7  0.000000    0.000004 0.002503 0.998003
##
## LogLikelihood : 9393.595
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.7455
## Bayes        -6.7156
## Shibata      -6.7455
## Hannan-Quinn -6.7347
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.120  0.2900
## Lag[2*(p+q)+(p+q)-1][2]     1.128  0.6613
## Lag[4*(p+q)+(p+q)-1][5]     3.829  0.2540
## d.o.f=1
```

```
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                   0.009334  0.9230
## Lag[2*(p+q)+(p+q)-1][5]  0.365290  0.9763
## Lag[4*(p+q)+(p+q)-1][9]  0.927661  0.9891
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1090 0.500 2.000  0.7413
## ARCH Lag[5]    0.2857 1.440 1.667  0.9440
## ARCH Lag[7]    0.7464 2.315 1.543  0.9510
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  NaN
## Individual Statistics:
## mu      0.4113
## ar1     0.3405
## omega  29.8360
## alpha1  0.5634
## beta1   0.8381
## gamma1  0.1392
## delta   0.9494
## vxreg1 12.7770
## vxreg2     NaN
## vxreg3     NaN
## vxreg4     NaN
## vxreg5     NaN
## vxreg6     NaN
## vxreg7     NaN
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          3.08 3.34 3.9
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           0.3194 0.74948
## Negative Sign Bias  1.8156 0.06954   *
## Positive Sign Bias  1.1725 0.24108
## Joint Effect        9.2255 0.02644  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     114.2    1.328e-15
## 2    30     121.0    3.281e-13
```

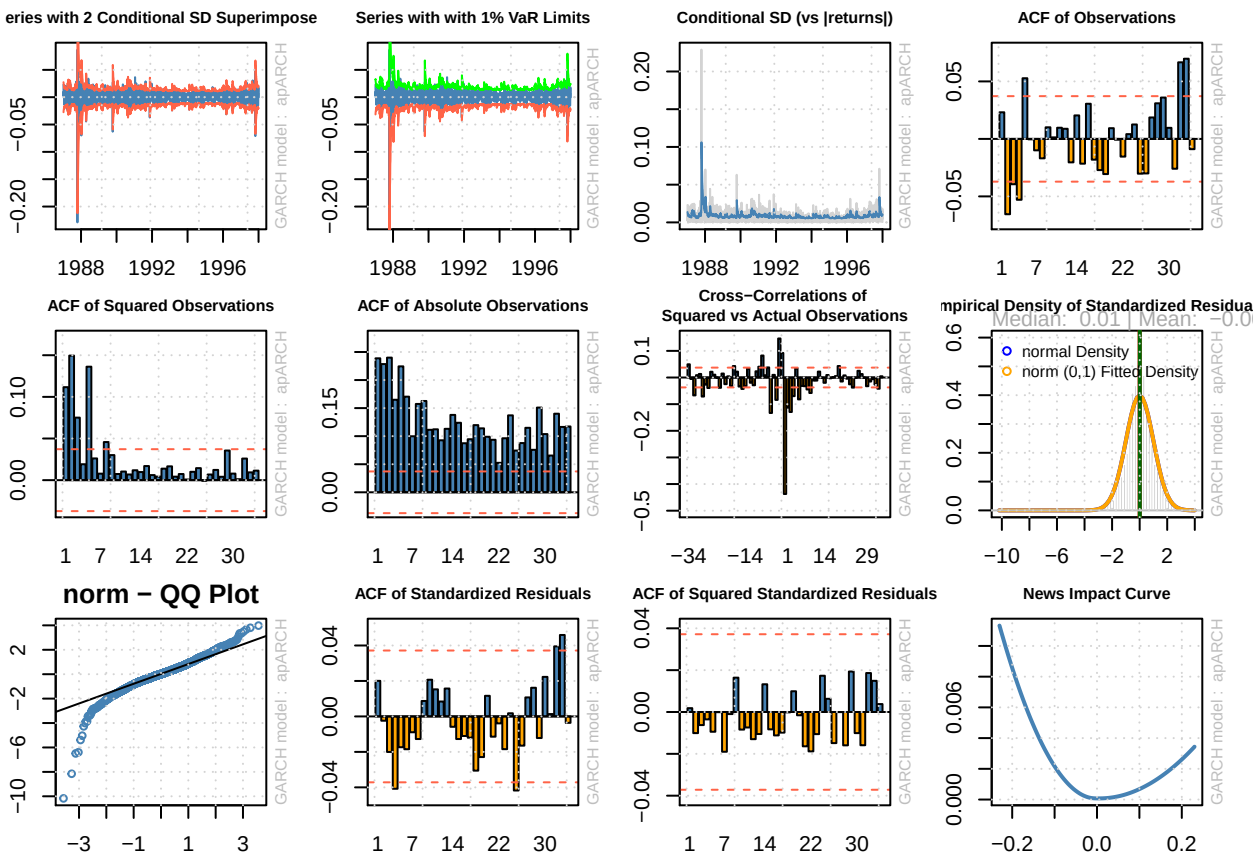
```
## 3    40     145.5    3.431e-14
## 4    50     153.9    9.159e-13
##
##
## Elapsed time : 1.979514
```

```
apARCH_rhat <- apARCHFit@fit$fitted.values
plot.ts(apARCH_rhat)
apARCH_hhat <- ts(apARCHFit@fit$sigma^2)
plot.ts(apARCH_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(apARCHFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
# IGARCH
i_garchMod <- ugarchspec(variance.model=list(model="iGARCH",
                                             garchOrder=c(1,1),
                                             external.regressors = df_dummies ),
                         mean.model=list(armaOrder=c(1,0)),
                         distribution.model="norm")
i_garchFit <- ugarchfit(spec=i_garchMod, data= log_ret)
coef(i_garchFit)
```

```
##           mu          ar1        omega       alpha1        beta1       vxreg1
## 6.718728e-04 3.460114e-02 1.304707e-06 1.107062e-01 8.892938e-01 9.129943e-09
##       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6       vxreg7
## 5.410431e-09 3.307950e-09 1.726805e-08 1.948471e-07 4.802796e-12 1.505417e-08
```

```
i_garchFit
```

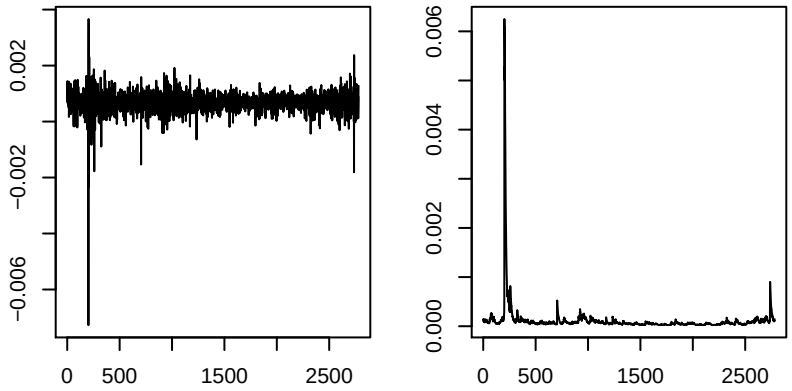
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : iGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
```

```
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000672    0.000166  4.045965 0.000052
## ar1     0.034601    0.022170  1.560742 0.118585
## omega   0.000001    0.000000  2.960788 0.003069
## alpha1  0.110706    0.001635 67.707776 0.000000
## beta1   0.889294          NA        NA       NA
## vxreg1  0.000000    0.000932  0.000010 0.999992
## vxreg2  0.000000    0.000013  0.000429 0.999657
## vxreg3  0.000000    0.000002  0.001761 0.998595
## vxreg4  0.000000    0.000013  0.001370 0.998907
## vxreg5  0.000000    0.000005  0.036663 0.970754
## vxreg6  0.000000    0.000031  0.000000 1.000000
## vxreg7  0.000000    0.000010  0.001449 0.998844
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000672    0.000396  1.698429 0.089427
## ar1     0.034601    0.059119  0.585283 0.558358
## omega   0.000001    0.000003  0.447497 0.654516
## alpha1  0.110706    0.010167 10.889161 0.000000
## beta1   0.889294          NA        NA       NA
## vxreg1  0.000000    0.001276  0.000007 0.999994
## vxreg2  0.000000    0.000017  0.000324 0.999741
## vxreg3  0.000000    0.000007  0.000497 0.999603
## vxreg4  0.000000    0.000032  0.000547 0.999564
## vxreg5  0.000000    0.000013  0.015228 0.987850
## vxreg6  0.000000    0.000036  0.000000 1.000000
## vxreg7  0.000000    0.000050  0.000303 0.999758
##
## LogLikelihood : 9378.587
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.7368
## Bayes        -6.7134
## Shibata      -6.7369
## Hannan-Quinn -6.7284
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      3.294 0.06953
## Lag[2*(p+q)+(p+q)-1][2]     3.306 0.01895
## Lag[4*(p+q)+(p+q)-1][5]     6.372 0.03374
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.261  0.2615
```

```
## Lag[2*(p+q)+(p+q)-1][5]     1.399  0.7647
## Lag[4*(p+q)+(p+q)-1][9]     1.908  0.9156
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]  0.008871 0.500 2.000  0.9250
## ARCH Lag[5]  0.253244 1.440 1.667  0.9524
## ARCH Lag[7]  0.722511 2.315 1.543  0.9540
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  44.3559
## Individual Statistics:
## mu      0.1944
## ar1     0.3350
## omega   7.7849
## alpha1  0.1040
## vxreg1 14.4927
## vxreg2 10.6094
## vxreg3  3.3986
## vxreg4  2.9878
## vxreg5  0.5660
## vxreg6  0.5595
## vxreg7  0.0348
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.49 2.75 3.27
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                     t-value      prob sig
## Sign Bias           0.09377 0.9253025
## Negative Sign Bias  3.19389 0.0014194 ***
## Positive Sign Bias  1.35132 0.1767030
## Joint Effect       19.01009 0.0002721 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     138.8    3.107e-20
## 2    30     150.7    2.190e-18
## 3    40     168.4    5.233e-18
## 4    50     182.1    3.385e-17
##
##
## Elapsed time : 0.7031522
```

```
i_rhat <- i_garchFit@fit$fitted.values
plot.ts(i_rhat)
i_hhat <- ts(i_garchFit@fit$sigma^2)
```

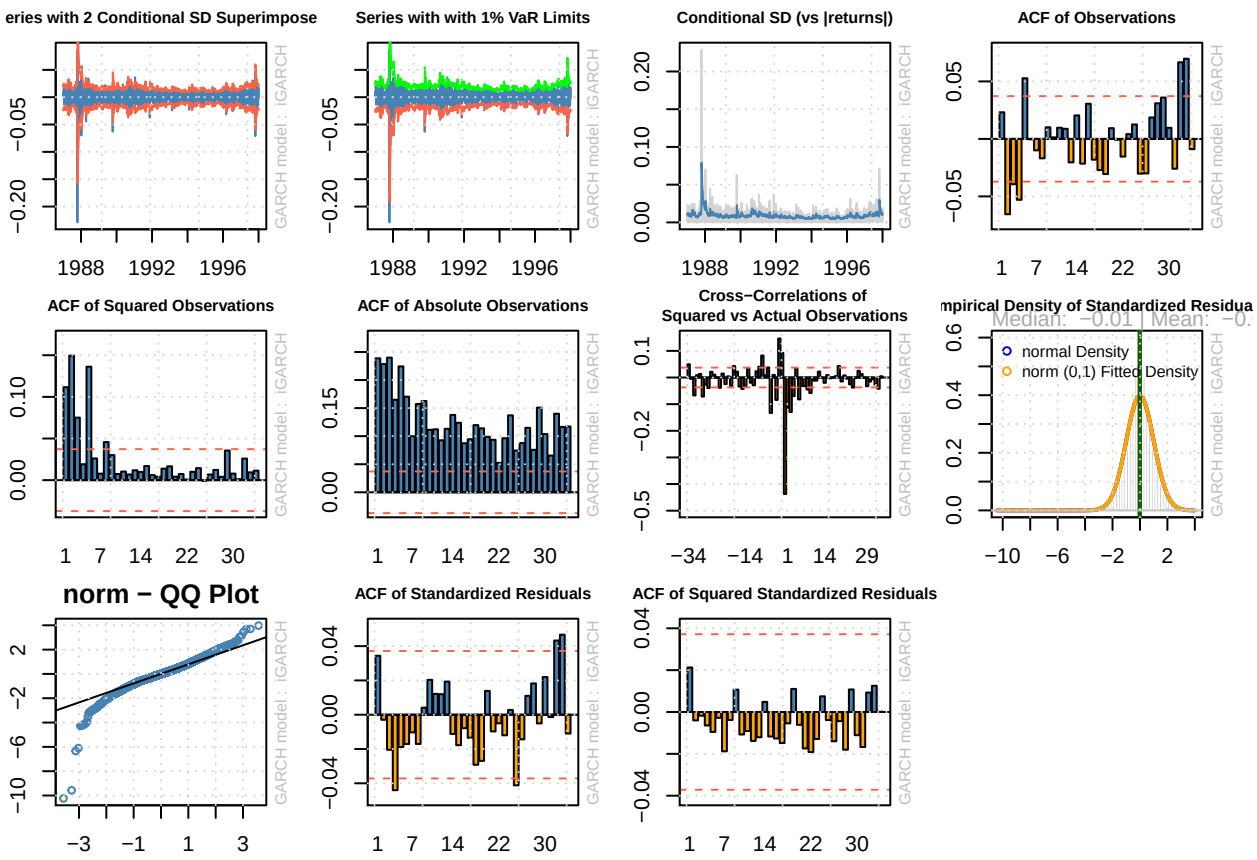
```r
plot.ts(i_hhat)

par(mfrow=c(2, 3))
```




```r
par(mar = c(2, 2, 2, 2))
plot(i_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```

```
## Warning in FUN(x):
## plot-->: iGARCH and fiGARCH newsimpact not available
```




```
# GARCH MODELS assuming a Student's t-distribution
# Student's t-sGARCH --->  tsGARCH
library(rugarch)

ts_garchMod <- ugarchspec(mean.model = list(armaOrder = c(1, 0), include.mean = TRUE
),
variance.model = list(model = 'sGARCH',
                      garchOrder = c(1, 1),
                      external.regressors = df_dummies
),
distribution.model = "std")

ts_garchFit <- ugarchfit(spec=ts_garchMod, data=log_ret)
ts_garchFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : std
##
```

```
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000681    0.000125  5.440424  0.00000
## ar1     0.016918    0.018124  0.933480  0.35057
## omega   0.000001    0.000001  0.716074  0.47395
## alpha1  0.046134    0.002604 17.719872  0.00000
## beta1   0.944010    0.010702 88.210378  0.00000
## vxreg1  0.000000    0.000745  0.000013  0.99999
## vxreg2  0.000000    0.000092  0.000108  0.99991
## vxreg3  0.000000    0.000092  0.000107  0.99991
## vxreg4  0.000000    0.000005  0.001912  0.99848
## vxreg5  0.000000    0.000005  0.002245  0.99821
## vxreg6  0.000000    0.000016  0.000610  0.99951
## vxreg7  0.000000    0.000002  0.004791  0.99618
## shape   4.874184    0.552438  8.823038  0.00000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000681    0.000281 2.422255 0.015425
## ar1     0.016918    0.025356 0.667226 0.504628
## omega   0.000001    0.000011 0.072052 0.942560
## alpha1  0.046134    0.028186 1.636776 0.101677
## beta1   0.944010    0.114764 8.225629 0.000000
## vxreg1  0.000000    0.008467 0.000001 0.999999
## vxreg2  0.000000    0.001194 0.000008 0.999993
## vxreg3  0.000000    0.000349 0.000028 0.999977
## vxreg4  0.000000    0.000111 0.000090 0.999928
## vxreg5  0.000000    0.000072 0.000143 0.999886
## vxreg6  0.000000    0.000160 0.000062 0.999950
## vxreg7  0.000000    0.000054 0.000188 0.999850
## shape   4.874184    1.764929 2.761688 0.005750
##
## LogLikelihood : 9561.356
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.8669
## Bayes        -6.8391
## Shibata      -6.8669
## Hannan-Quinn -6.8568
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic   p-value
## Lag[1]                      5.889 1.524e-02
## Lag[2*(p+q)+(p+q)-1][2]     5.895 5.124e-05
## Lag[4*(p+q)+(p+q)-1][5]     8.789 3.606e-03
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
```

```
##                           statistic  p-value
## Lag[1]                        9.663 0.001880
## Lag[2*(p+q)+(p+q)-1][5]      10.695 0.006096
## Lag[4*(p+q)+(p+q)-1][9]      11.110 0.028745
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.7348 0.500 2.000  0.3913
## ARCH Lag[5]    0.7478 1.440 1.667  0.8089
## ARCH Lag[7]    0.8544 2.315 1.543  0.9360
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  461.1365
## Individual Statistics:
## mu      0.220534
## ar1     0.096322
## omega  71.164742
## alpha1  0.195380
## beta1   0.199804
## vxreg1  7.965580
## vxreg2  6.415530
## vxreg3  3.725879
## vxreg4  0.005702
## vxreg5  2.052022
## vxreg6  0.740233
## vxreg7  0.178243
## shape   0.150245
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.89 3.15 3.69
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value      prob sig
## Sign Bias           0.8530 3.937e-01
## Negative Sign Bias  5.9148 3.729e-09 ***
## Positive Sign Bias  0.8026 4.223e-01
## Joint Effect       41.0696 6.321e-09 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     27.39      0.09583
## 2    30     42.03      0.05580
## 3    40     59.23      0.01989
## 4    50     69.83      0.02690
##
##
## Elapsed time : 1.329262
```

```
## Results review 1
coef(ts_garchFit)
```

```
##           mu          ar1        omega       alpha1        beta1       vxreg1
## 6.808605e-04 1.691814e-02 7.986854e-07 4.613415e-02 9.440098e-01 9.986932e-09
##       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6       vxreg7
## 9.947039e-09 9.929000e-09 1.002196e-08 1.031877e-08 9.961744e-09 1.018089e-08
##        shape
## 4.874184e+00
```

```
ts_rhat <- ts_garchFit@fit$fitted.values
plot.ts(ts_rhat)
ts_hhat <- ts(ts_garchFit@fit$sigma^2)
plot.ts(ts_hhat)

## Results review 2
tfit.val     <- coef(ts_garchFit)
tfit.sd      <- diag(vcov(ts_garchFit))
ttrue.val = ts_garchFit@fit$tval

tfit.conf.lb <- tfit.val + qnorm(0.025) * tfit.sd
tfit.conf.ub <- tfit.val + qnorm(0.975) * tfit.sd
print(tfit.val)
```

```
##           mu          ar1        omega       alpha1        beta1       vxreg1
## 6.808605e-04 1.691814e-02 7.986854e-07 4.613415e-02 9.440098e-01 9.986932e-09
##       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6       vxreg7
## 9.947039e-09 9.929000e-09 1.002196e-08 1.031877e-08 9.961744e-09 1.018089e-08
##        shape
## 4.874184e+00
```

```
print(tfit.sd)
```

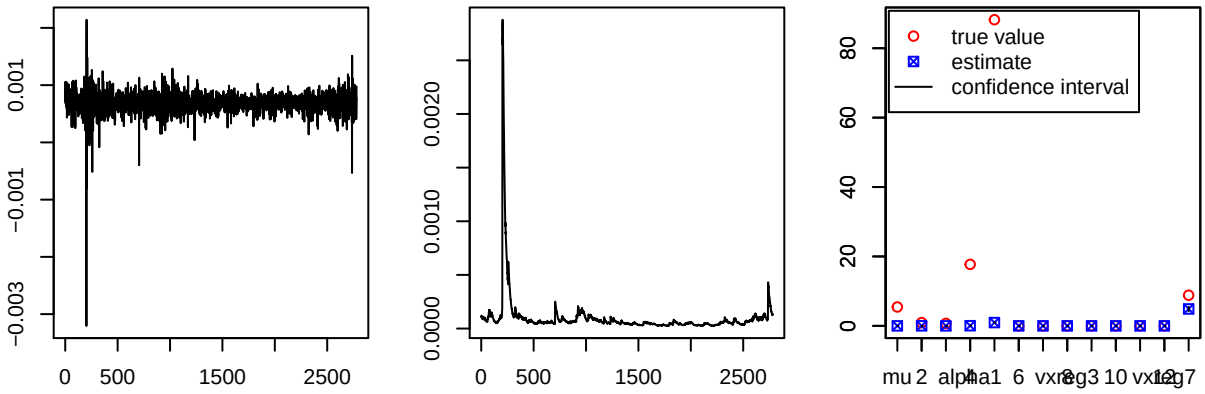
```
##  [1]  1.566213e-08  3.284693e-04 -1.244043e-12 -6.778347e-06 -1.145285e-04
##  [6] -5.547588e-07  8.488128e-09  8.550484e-09 -2.747755e-11 -2.112168e-11
## [11] -2.664448e-10  4.514746e-12  3.051880e-01
```

```
print(ttrue.val)
```

```
##           mu          ar1        omega       alpha1        beta1       vxreg1
## 5.440424e+00 9.334802e-01 7.160742e-01 1.771987e+01 8.821038e+01 1.340849e-05
##       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6       vxreg7
## 1.079662e-04 1.073767e-04 1.911892e-03 2.245245e-03 6.102837e-04 4.791476e-03
##        shape
## 8.823038e+00
```

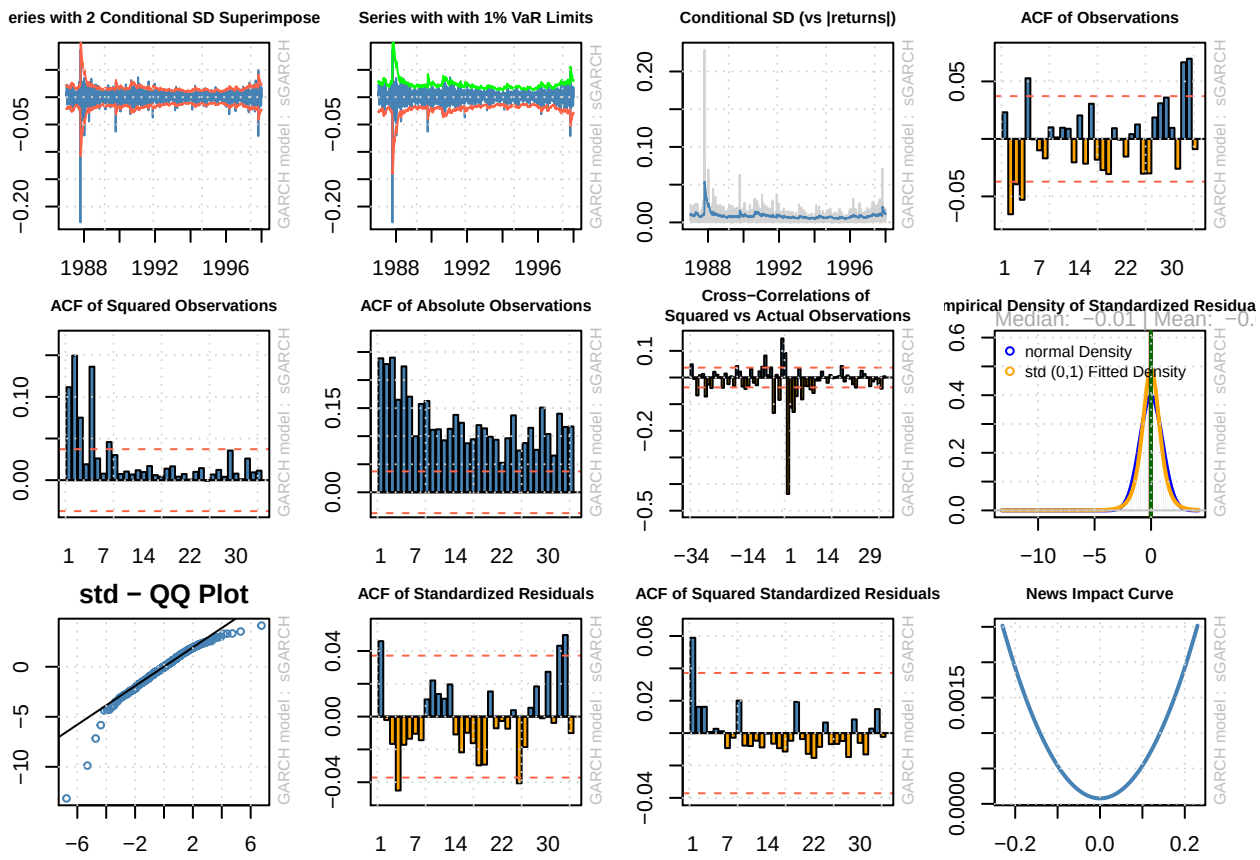
```
plot(ttrue.val, pch = 1, col = "red",
     ylim = range(c(tfit.conf.lb, tfit.conf.ub, ttrue.val)),
     xlab = "", ylab = "", axes = TRUE)
box(); axis(1, at = 1:length(tfit.val), labels = names(tfit.val)); axis(2)
points(coef(ts_garchFit), col = "blue", pch = 7)
for (i in 1:length(tfit.val)) {
  lines(c(i,i), c(tfit.conf.lb[i], tfit.conf.ub[i]))
}
legend( "topleft", legend = c("true value", "estimate", "confidence interval"),
        col = c("red", "blue", 1), pch = c(1, 7, NA), lty = c(NA, NA, 1), inset = 0.01)
```

```
par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(ts_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#Student's t-gjrGARCH --->tgjrGARCH
tgjr_garchMod <- ugarchspec(
  variance.model=list(model="gjrGARCH",
                      garchOrder=c(1,1), external.regressors = df_dummies ),
  mean.model=list(armaOrder=c(1,0),
                  include.mean=TRUE
  ),
  distribution.model="std"
)
tgjr_garchFit <- ugarchfit(spec=tgjr_garchMod, data=log_ret)
coef(tgjr_garchFit)
```

```
##           mu          ar1        omega       alpha1        beta1       gamma1
## 6.495544e-04 2.037559e-02 8.937411e-07 3.027894e-02 9.444314e-01 2.250641e-02
##       vxreg1       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6
## 7.747261e-09 2.140205e-09 6.018121e-17 3.547505e-08 5.518537e-05 8.569491e-11
##       vxreg7        shape
## 1.107205e-05 5.025285e+00
```

```
tgjr_garchFit
```

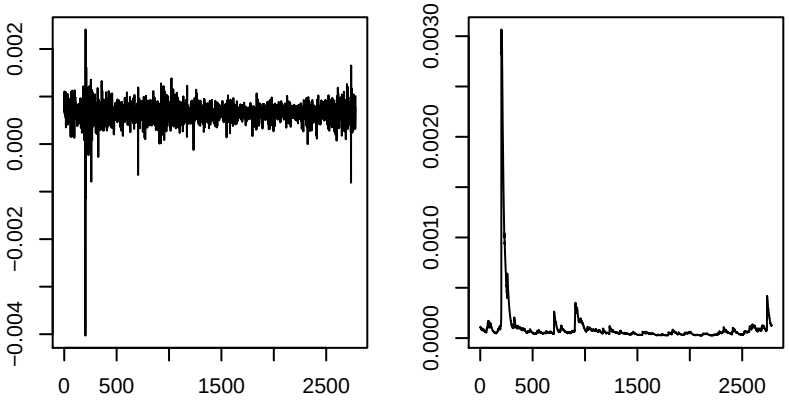
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
```

```
## ------------------------------------
## GARCH Model  : gjrGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000650    0.000129 5.0455e+00 0.000000
## ar1     0.020376    0.018301 1.1134e+00 0.265547
## omega   0.000001    0.000000 2.4964e+00 0.012547
## alpha1  0.030279    0.002633 1.1501e+01 0.000000
## beta1   0.944431    0.000276 3.4200e+03 0.000000
## gamma1  0.022506    0.003266 6.8918e+00 0.000000
## vxreg1  0.000000    0.000453 1.7000e-05 0.999986
## vxreg2  0.000000    0.000183 1.2000e-05 0.999991
## vxreg3  0.000000    0.000114 0.0000e+00 1.000000
## vxreg4  0.000000    0.000008 4.2800e-03 0.996585
## vxreg5  0.000055    0.000034 1.6304e+00 0.103017
## vxreg6  0.000000    0.000002 3.7000e-05 0.999970
## vxreg7  0.000011    0.000026 4.3092e-01 0.666529
## shape   5.025285    0.519785 9.6680e+00 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000650    0.000116  5.586597 0.000000
## ar1     0.020376    0.017365  1.173353 0.240654
## omega   0.000001    0.000001  0.685138 0.493257
## alpha1  0.030279    0.017647  1.715801 0.086199
## beta1   0.944431    0.010215 92.455574 0.000000
## gamma1  0.022506    0.003971  5.668008 0.000000
## vxreg1  0.000000    0.001862  0.000004 0.999997
## vxreg2  0.000000    0.000632  0.000003 0.999997
## vxreg3  0.000000    0.000401  0.000000 1.000000
## vxreg4  0.000000    0.000016  0.002236 0.998216
## vxreg5  0.000055    0.000023  2.419107 0.015559
## vxreg6  0.000000    0.000003  0.000026 0.999979
## vxreg7  0.000011    0.000086  0.128287 0.897922
## shape   5.025285    1.297528  3.872967 0.000108
##
## LogLikelihood : 9568.568
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.8713
## Bayes        -6.8415
## Shibata      -6.8714
## Hannan-Quinn -6.8605
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      4.149 0.04165
```

```
## Lag[2*(p+q)+(p+q)-1][2]     4.155 0.00305
## Lag[4*(p+q)+(p+q)-1][5]     7.126 0.01721
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic  p-value
## Lag[1]                      8.138 0.004334
## Lag[2*(p+q)+(p+q)-1][5]     9.052 0.016057
## Lag[4*(p+q)+(p+q)-1][9]     9.488 0.064405
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.7338 0.500 2.000  0.3917
## ARCH Lag[5]    0.7430 1.440 1.667  0.8104
## ARCH Lag[7]    0.8567 2.315 1.543  0.9357
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  434.3182
## Individual Statistics:
## mu      0.265596
## ar1     0.110481
## omega  62.932791
## alpha1  0.226081
## beta1   0.237378
## gamma1  0.188567
## vxreg1  9.473552
## vxreg2  6.398403
## vxreg3  3.435132
## vxreg4  0.051169
## vxreg5  0.002212
## vxreg6  0.782884
## vxreg7  0.019797
## shape   0.150751
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          3.08 3.34 3.9
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value      prob sig
## Sign Bias           0.6861 4.927e-01
## Negative Sign Bias  5.3773 8.188e-08 ***
## Positive Sign Bias  0.7909 4.291e-01
## Joint Effect       34.7582 1.370e-07 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
```

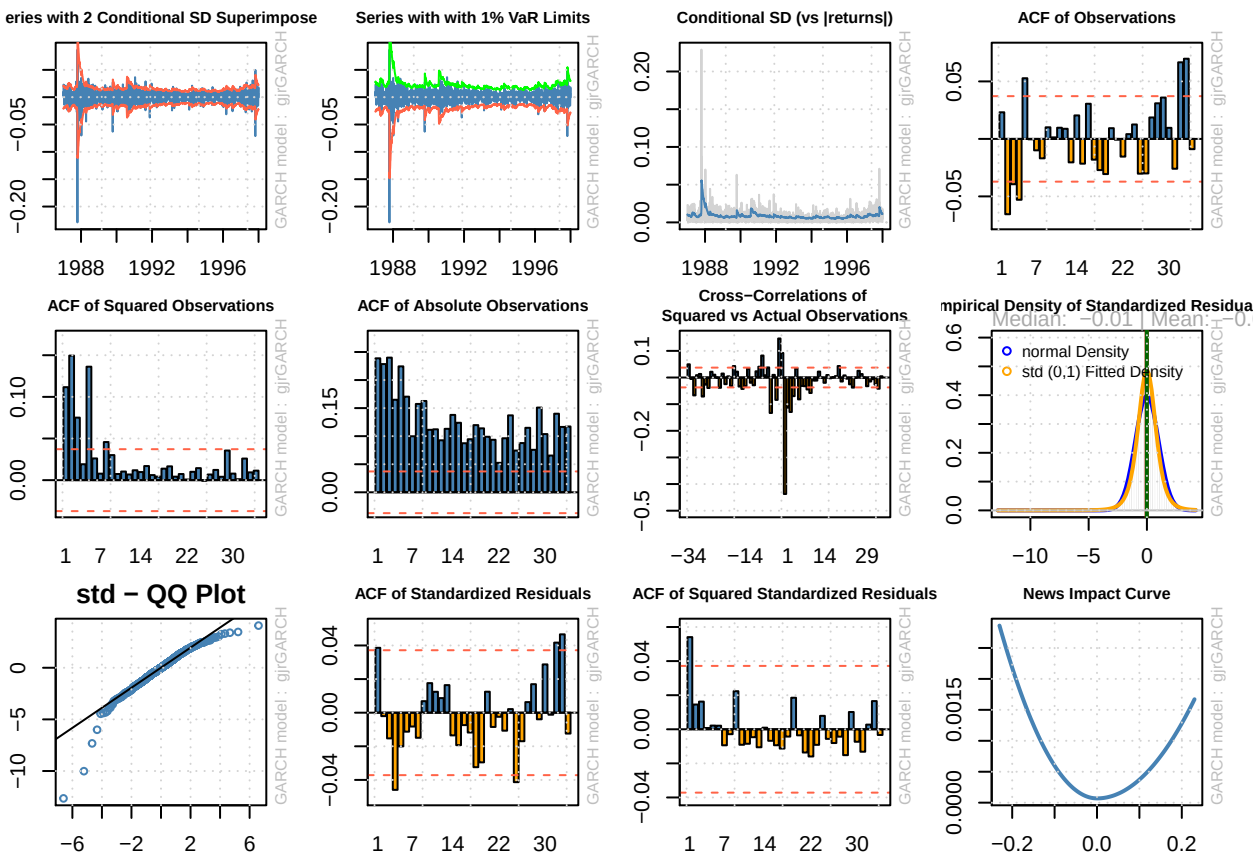
```
##   group statistic p-value(g-1)
## 1    20     28.27      0.07837
## 2    30     26.37      0.60577
## 3    40     57.16      0.03034
## 4    50     61.48      0.10866
##
##
## Elapsed time : 2.963501
```

```
tgjr_rhat <- tgjr_garchFit@fit$fitted.values
plot.ts(tgjr_rhat)
tgjr_hhat <- ts(tgjr_garchFit@fit$sigma^2)
plot.ts(tgjr_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(tgjr_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#Student's t-eGARCH --->teGARCH
te_garchMod <- ugarchspec(
  variance.model=list(model="eGARCH",
                      garchOrder=c(1,1), external.regressors = df_dummies ),
  mean.model=list(armaOrder=c(1,0),
                  include.mean=TRUE
  ),
  distribution.model="std"
)
te_garchFit <- ugarchfit(spec=te_garchMod, data=log_ret)
coef(te_garchFit)
```

```
##            mu           ar1         omega        alpha1         beta1
##  0.0005920684  0.0158498266 -0.0999985207 -0.0383220276  0.9897031782
##         gamma1        vxreg1        vxreg2        vxreg3        vxreg4
##  0.0947133145 -0.0260061486 -0.0502494495 -0.1012987398  0.0264633650
##         vxreg5        vxreg6        vxreg7         shape
##  0.2744948314 -0.0710716406  0.1056797395  5.1484659175
```

```
te_garchFit
```

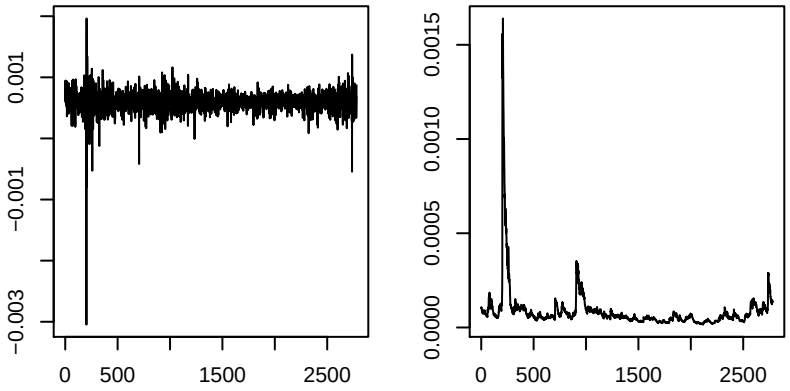
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
```

```
## ------------------------------------
## GARCH Model  : eGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000592    0.000122    4.87149 0.000001
## ar1     0.015850    0.017442    0.90873 0.363491
## omega  -0.099999    0.005057  -19.77544 0.000000
## alpha1 -0.038322    0.012193   -3.14295 0.001673
## beta1   0.989703    0.000528 1874.53024 0.000000
## gamma1  0.094713    0.022845    4.14585 0.000034
## vxreg1 -0.026006    0.130370   -0.19948 0.841888
## vxreg2 -0.050249    0.120515   -0.41696 0.676711
## vxreg3 -0.101299    0.123271   -0.82176 0.411215
## vxreg4  0.026463    0.118940    0.22249 0.823930
## vxreg5  0.274495    0.113468    2.41913 0.015557
## vxreg6 -0.071072    0.102020   -0.69665 0.486024
## vxreg7  0.105680    0.114059    0.92653 0.354169
## shape   5.148466    0.693757    7.42113 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000592    0.000103   5.76118 0.000000
## ar1     0.015850    0.015530   1.02059 0.307450
## omega  -0.099999    0.011216  -8.91598 0.000000
## alpha1 -0.038322    0.021817  -1.75652 0.078999
## beta1   0.989703    0.001180 838.60657 0.000000
## gamma1  0.094713    0.057201   1.65581 0.097760
## vxreg1 -0.026006    0.160699  -0.16183 0.871438
## vxreg2 -0.050249    0.040259  -1.24815 0.211975
## vxreg3 -0.101299    0.049238  -2.05734 0.039653
## vxreg4  0.026463    0.062881   0.42085 0.673864
## vxreg5  0.274495    0.065825   4.17004 0.000030
## vxreg6 -0.071072    0.080319  -0.88486 0.376230
## vxreg7  0.105680    0.076963   1.37312 0.169714
## shape   5.148466    1.572399   3.27427 0.001059
##
## LogLikelihood : 9579.461
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.8792
## Bayes        -6.8493
## Shibata      -6.8792
## Hannan-Quinn -6.8684
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic  p-value
## Lag[1]                      3.912 0.047951
```

```
## Lag[2*(p+q)+(p+q)-1][2]     4.033 0.004001
## Lag[4*(p+q)+(p+q)-1][5]     7.190 0.016227
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic  p-value
## Lag[1]                      9.369 0.002207
## Lag[2*(p+q)+(p+q)-1][5]    11.091 0.004812
## Lag[4*(p+q)+(p+q)-1][9]    11.828 0.019845
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.9059 0.500 2.000  0.3412
## ARCH Lag[5]    1.1939 1.440 1.667  0.6764
## ARCH Lag[7]    1.3687 2.315 1.543  0.8473
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.2799
## Individual Statistics:
## mu     0.3747720
## ar1    0.1383511
## omega  0.3793137
## alpha1 0.0728387
## beta1  0.3880465
## gamma1 0.1458148
## vxreg1 0.0028431
## vxreg2 0.0004189
## vxreg3 0.0007320
## vxreg4 0.0009185
## vxreg5 0.0008567
## vxreg6 0.0035727
## vxreg7 0.0015351
## shape  0.1283689
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          3.08 3.34 3.9
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value      prob sig
## Sign Bias           0.8890 3.741e-01
## Negative Sign Bias  5.8954 4.189e-09 ***
## Positive Sign Bias  0.6116 5.408e-01
## Joint Effect       39.7646 1.195e-08 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
```

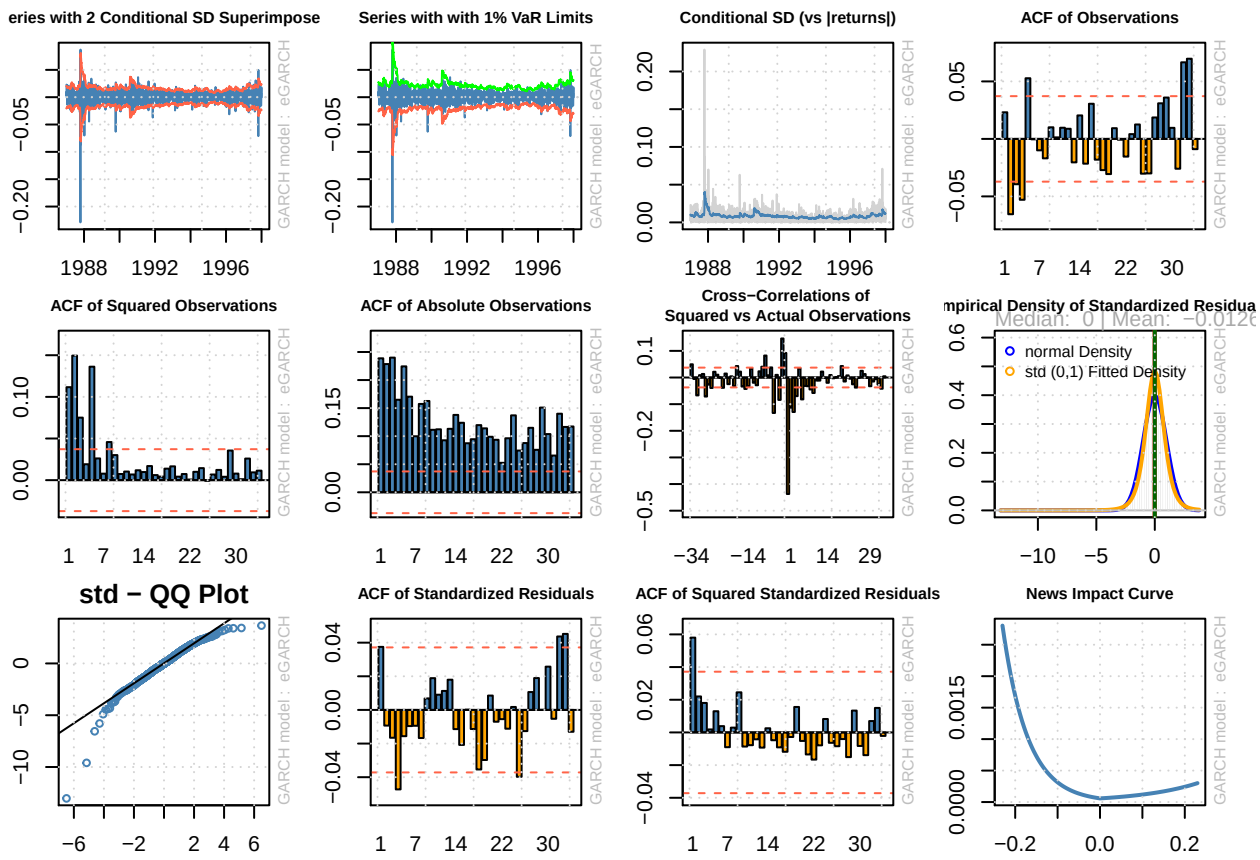
```
##   group statistic p-value(g-1)
## 1    20     22.82       0.2454
## 2    30     35.11       0.2011
## 3    40     42.32       0.3299
## 4    50     53.11       0.3189
##
##
## Elapsed time : 2.651492
```

```
te_rhat <- te_garchFit@fit$fitted.values
plot.ts(te_rhat)
te_hhat <- ts(te_garchFit@fit$sigma^2)
plot.ts(te_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(te_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#Student's t-apARCH ---> tapARCH
tapARCHMod <- ugarchspec(variance.model=list(model="apARCH",
                                             garchOrder=c(1,1),
                                             external.regressors = df_dummies ),
                         mean.model=list(armaOrder=c(1,0)),
                         distribution.model="std")
tapARCHFit <- ugarchfit(spec=tapARCHMod, data= log_ret)
coef(tapARCHFit)
```

```
##           mu          ar1        omega       alpha1        beta1       gamma1
## 6.159577e-04 2.094163e-02 6.378751e-06 5.353966e-02 9.410051e-01 2.772754e-01
##        delta       vxreg1       vxreg2       vxreg3       vxreg4       vxreg5
## 1.586309e+00 7.583722e-09 2.104194e-09 2.357757e-10 1.998229e-08 1.931712e-06
##       vxreg6       vxreg7        shape
## 1.514864e-19 1.801589e-07 5.101146e+00
```

```
tapARCHFit
```

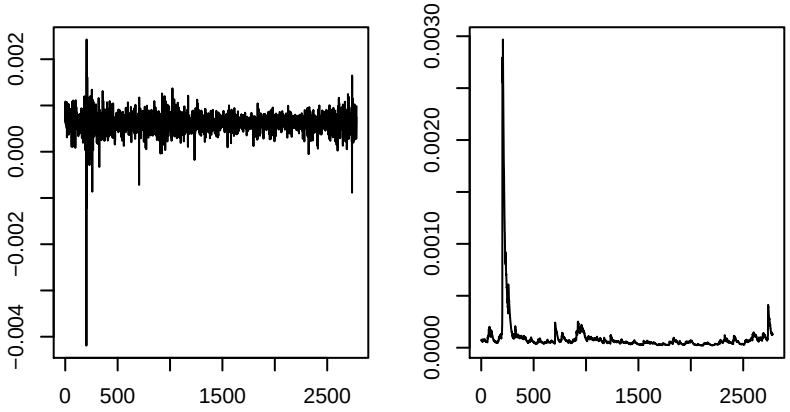
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : apARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
```

```
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000616    0.000121   5.108084  0.00000
## ar1     0.020942    0.018183   1.151696  0.24945
## omega   0.000006    0.000004   1.498394  0.13403
## alpha1  0.053540    0.017326   3.090209  0.00200
## beta1   0.941005    0.002744 342.955312  0.00000
## gamma1  0.277275    0.084261   3.290658  0.00100
## delta   1.586309    0.021228  74.727252  0.00000
## vxreg1  0.000000    0.001177   0.000006  0.99999
## vxreg2  0.000000    0.000254   0.000008  0.99999
## vxreg3  0.000000    0.000072   0.000003  1.00000
## vxreg4  0.000000    0.000047   0.000427  0.99966
## vxreg5  0.000002    0.000041   0.046960  0.96254
## vxreg6  0.000000    0.000064   0.000000  1.00000
## vxreg7  0.000000    0.000037   0.004925  0.99607
## shape   5.101146    0.474191  10.757578  0.00000
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000616    0.000214   2.879988 0.003977
## ar1     0.020942    0.017034   1.229386 0.218927
## omega   0.000006    0.000020   0.323628 0.746220
## alpha1  0.053540    0.075567   0.708508 0.478630
## beta1   0.941005    0.005076 185.399597 0.000000
## gamma1  0.277275    0.563184   0.492336 0.622482
## delta   1.586309    0.159084   9.971507 0.000000
## vxreg1  0.000000    0.003102   0.000002 0.999998
## vxreg2  0.000000    0.000448   0.000005 0.999996
## vxreg3  0.000000    0.000306   0.000001 0.999999
## vxreg4  0.000000    0.000032   0.000629 0.999498
## vxreg5  0.000002    0.000118   0.016440 0.986883
## vxreg6  0.000000    0.000056   0.000000 1.000000
## vxreg7  0.000000    0.000109   0.001653 0.998681
## shape   5.101146    0.787032   6.481501 0.000000
##
## LogLikelihood : 9569.8
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.8715
## Bayes        -6.8395
## Shibata      -6.8715
## Hannan-Quinn -6.8599
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic  p-value
## Lag[1]                      4.302 0.038077
## Lag[2*(p+q)+(p+q)-1][2]     4.312 0.002144
```

```
## Lag[4*(p+q)+(p+q)-1][5]     7.175 0.016451
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                       6.017 0.01417
## Lag[2*(p+q)+(p+q)-1][5]      6.550 0.06659
## Lag[4*(p+q)+(p+q)-1][9]      6.953 0.20311
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.3841 0.500 2.000  0.5354
## ARCH Lag[5]    0.3894 1.440 1.667  0.9157
## ARCH Lag[7]    0.5706 2.315 1.543  0.9716
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  15.6833
## Individual Statistics:
## mu     0.32579
## ar1    0.11537
## omega  0.41155
## alpha1 0.27087
## beta1  0.27901
## gamma1 0.16959
## delta  0.41549
## vxreg1 5.44140
## vxreg2 4.54844
## vxreg3 2.55608
## vxreg4 0.07645
## vxreg5 1.63121
## vxreg6 0.09930
## vxreg7 0.10969
## shape  0.14823
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          3.26 3.54 4.07
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value      prob sig
## Sign Bias           0.5766 5.643e-01
## Negative Sign Bias  4.9687 7.149e-07 ***
## Positive Sign Bias  0.9091 3.634e-01
## Joint Effect       30.8613 9.092e-07 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
```

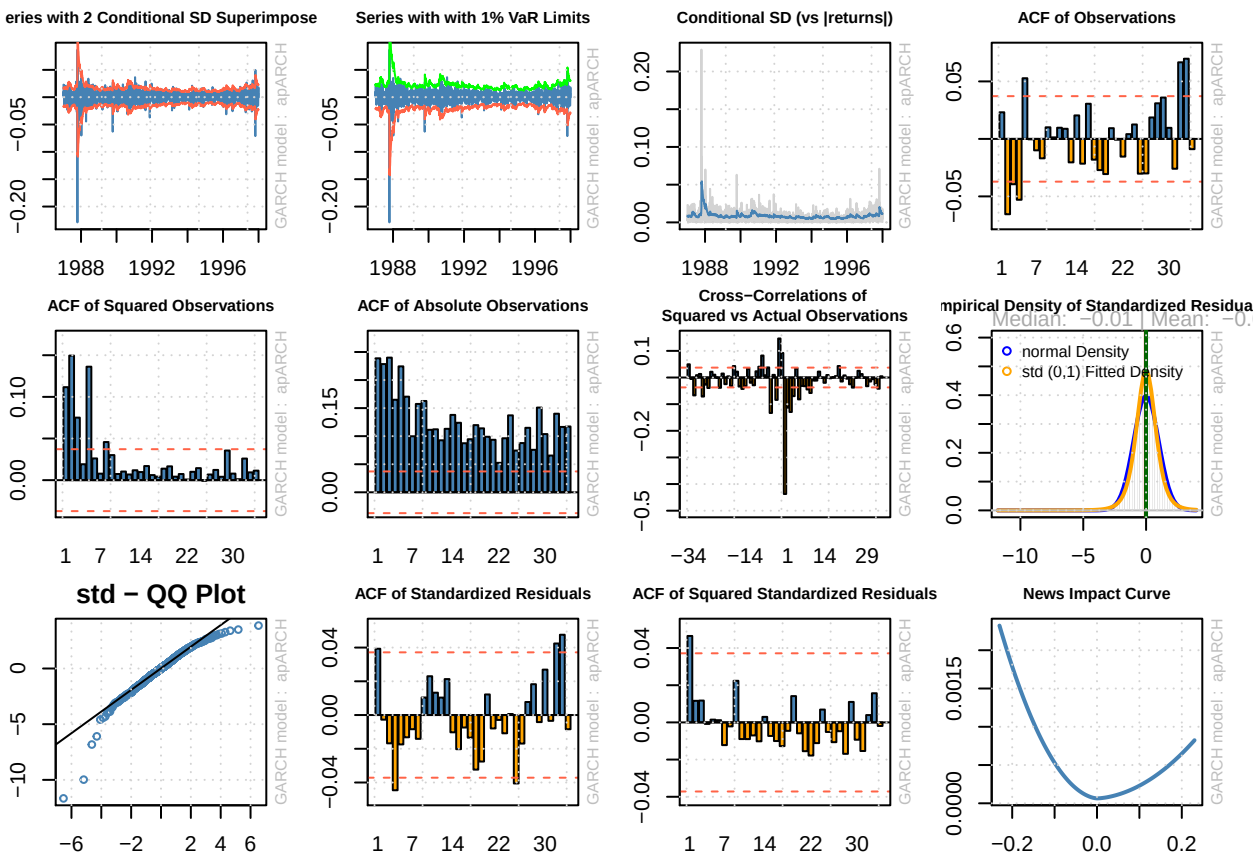
```
##   group statistic p-value(g-1)
## 1    20     22.98       0.2384
## 2    30     26.45       0.6011
## 3    40     47.61       0.1622
## 4    50     55.91       0.2313
##
##
## Elapsed time : 11.18811
```

```r
tapARCH_rhat <- tapARCHFit@fit$fitted.values
plot.ts(tapARCH_rhat)
tapARCH_hhat <- ts(tapARCHFit@fit$sigma^2)
plot.ts(tapARCH_hhat)

par(mfrow=c(2, 3))
```




```r
par(mar = c(2, 2, 2, 2))
plot(tapARCHFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#Student's t-tiGARCH ---> tiGARCH
ti_garchMod <- ugarchspec(variance.model=list(model="iGARCH",
                                              garchOrder=c(1,1),
                                              external.regressors = df_dummies ),
                          mean.model=list(armaOrder=c(1,0), include.mean=TRUE),
                          distribution.model="std")
ti_garchFit <- ugarchfit(spec=ti_garchMod, data= log_ret)
coef(ti_garchFit)
```

```
##           mu          ar1        omega       alpha1        beta1       vxreg1
## 6.701593e-04 1.621996e-02 4.381322e-07 4.996179e-02 9.500382e-01 7.558834e-09
##       vxreg2       vxreg3       vxreg4       vxreg5       vxreg6       vxreg7
## 1.952047e-09 1.498418e-14 1.140755e-08 5.348527e-05 2.366768e-09 2.402249e-07
##        shape
## 4.392296e+00
```

```
ti_garchFit
```

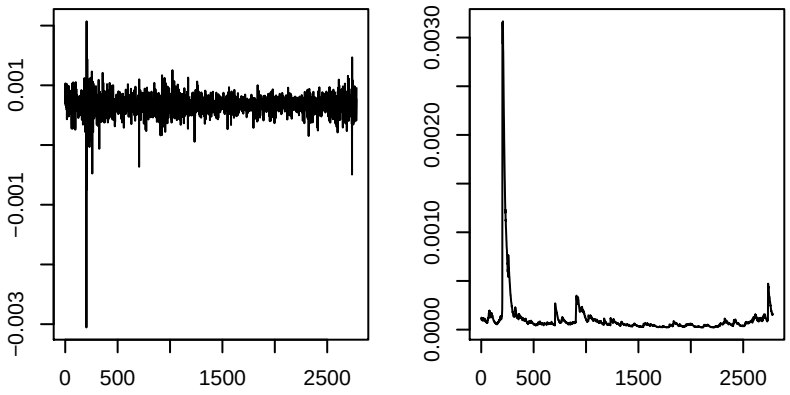
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : iGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
```

```
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000670    0.000124  5.416430 0.000000
## ar1     0.016220    0.017972  0.902495 0.366794
## omega   0.000000    0.000000  1.037008 0.299732
## alpha1  0.049962    0.000786 63.580084 0.000000
## beta1   0.950038          NA        NA       NA
## vxreg1  0.000000    0.000737  0.000010 0.999992
## vxreg2  0.000000    0.000210  0.000009 0.999993
## vxreg3  0.000000    0.000159  0.000000 1.000000
## vxreg4  0.000000    0.000012  0.000946 0.999245
## vxreg5  0.000053    0.000032  1.656886 0.097543
## vxreg6  0.000000    0.000014  0.000168 0.999866
## vxreg7  0.000000    0.000006  0.038706 0.969125
## shape   4.392296    0.338716 12.967475 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000670    0.000172 3.886356 0.000102
## ar1     0.016220    0.023214 0.698716 0.484730
## omega   0.000000    0.000002 0.232836 0.815888
## alpha1  0.049962    0.005530 9.034103 0.000000
## beta1   0.950038          NA       NA       NA
## vxreg1  0.000000    0.001953 0.000004 0.999997
## vxreg2  0.000000    0.000396 0.000005 0.999996
## vxreg3  0.000000    0.000481 0.000000 1.000000
## vxreg4  0.000000    0.000045 0.000256 0.999796
## vxreg5  0.000053    0.000030 1.759295 0.078527
## vxreg6  0.000000    0.000022 0.000106 0.999916
## vxreg7  0.000000    0.000004 0.055097 0.956061
## shape   4.392296    0.897180 4.895668 0.000001
##
## LogLikelihood : 9562.43
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.8683
## Bayes        -6.8428
## Shibata      -6.8684
## Hannan-Quinn -6.8591
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic   p-value
## Lag[1]                      5.270 0.0216917
## Lag[2*(p+q)+(p+q)-1][2]     5.280 0.0002264
## Lag[4*(p+q)+(p+q)-1][5]     8.441 0.0050379
## d.o.f=1
## H0 : No serial correlation
##
```

```
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic  p-value
## Lag[1]                      8.664 0.003245
## Lag[2*(p+q)+(p+q)-1][5]     9.540 0.012069
## Lag[4*(p+q)+(p+q)-1][9]     9.914 0.052334
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.6587 0.500 2.000  0.4170
## ARCH Lag[5]    0.6626 1.440 1.667  0.8348
## ARCH Lag[7]    0.7794 2.315 1.543  0.9466
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  268.9635
## Individual Statistics:
## mu     2.193e-01
## ar1    9.071e-02
## omega  1.111e+02
## alpha1 1.822e-01
## vxreg1 1.364e+01
## vxreg2 1.129e+01
## vxreg3 6.832e+00
## vxreg4 6.429e-01
## vxreg5 2.267e-03
## vxreg6 6.935e-01
## vxreg7 1.271e-01
## shape  2.355e-01
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.69 2.96 3.51
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value      prob sig
## Sign Bias           0.7488 4.540e-01
## Negative Sign Bias  5.5712 2.772e-08 ***
## Positive Sign Bias  0.9001 3.682e-01
## Joint Effect       37.4600 3.677e-08 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     22.55      0.25795
## 2    30     33.29      0.26606
## 3    40     49.19      0.12706
## 4    50     62.64      0.09126
##
##
```

```
## Elapsed time : 1.380489
```

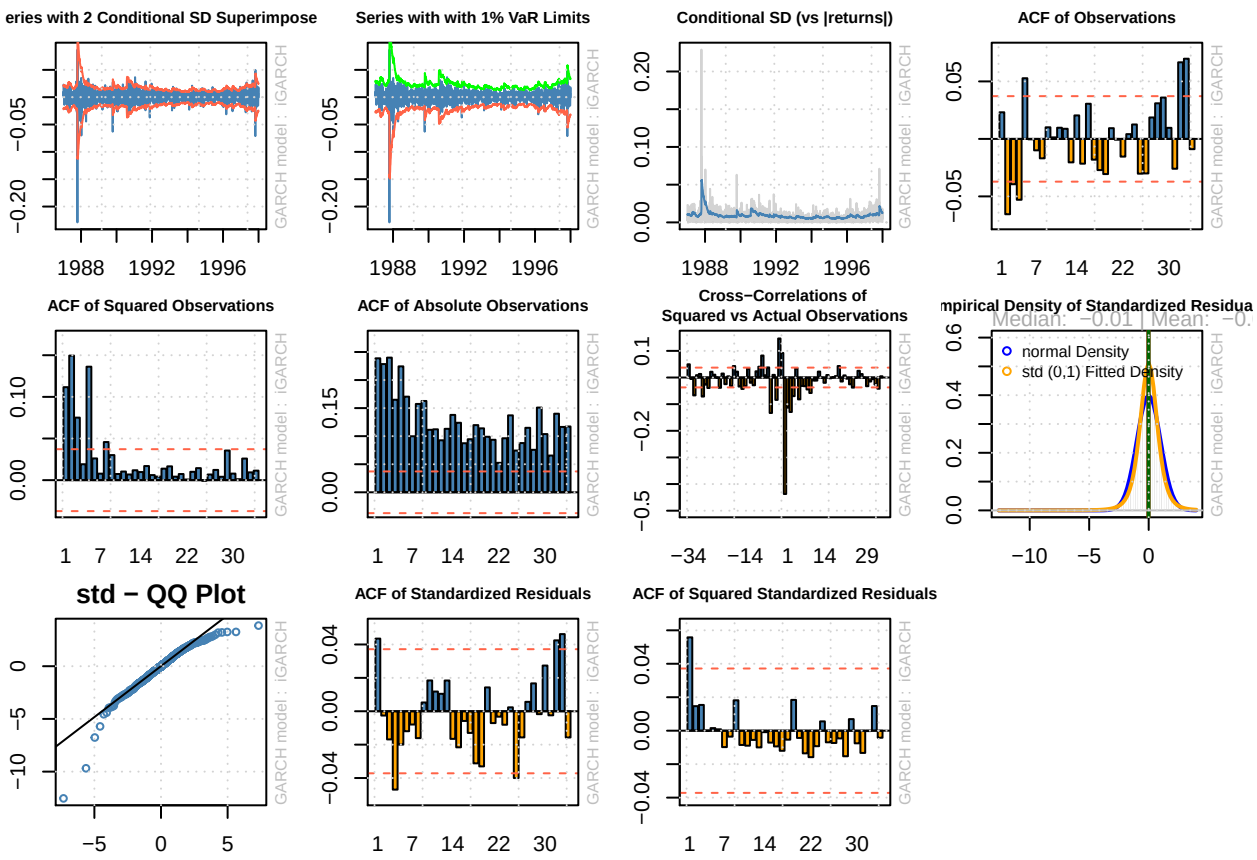
```
ti_rhat <- ti_garchFit@fit$fitted.values
plot.ts(ti_rhat)
ti_hhat <- ts(ti_garchFit@fit$sigma^2)
plot.ts(ti_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(ti_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```

```
## Warning in FUN(x):
## plot-->: iGARCH and fiGARCH newsimpact not available
```



In order to chose the best model:

```
#Choose best GARCH model
#Normal
AIC= c(infocriteria(s_garchFit)[1], infocriteria(gjr_garchFit)[1],
       infocriteria(e_garchFit)[1], infocriteria(apARCHFit)[1],  infocriteria(i_garchFit)[1] )
rowSAIC= c("s_garch", "gjr_garchFit", "e_garchFit", "apARCHFit", "i_garchFit")


AIC_Results= data.frame(AIC, row.names = rowSAIC)
AIC_Results
```

```
##                    AIC
## s_garch      -6.740216
## gjr_garchFit -6.754373
## e_garchFit   -6.764806
## apARCHFit    -6.745484
## i_garchFit   -6.736848
```

```
BestAICNorm=AIC_Results[which.min(AIC_Results$AIC),]
BestAICNorm
```

```
## [1] -6.764806
```

```
BIC= c(infocriteria(s_garchFit)[2], infocriteria(gjr_garchFit)[2],
       infocriteria(e_garchFit)[2], infocriteria(apARCHFit)[2],  infocriteria(i_garchFit)[2] )
rowSBIC= c("s_garch", "gjr_garchFit", "e_garchFit", "apARCHFit", "i_garchFit")
BIC_Results= data.frame(BIC, row.names = rowSBIC)
```

```
BIC_Results
```

```
##                    BIC
## s_garch      -6.714626
## gjr_garchFit -6.726650
## e_garchFit   -6.737083
## apARCHFit    -6.715628
## i_garchFit   -6.713390
```

```
BestBICNorm=BIC_Results[which.min(BIC_Results$BIC),]
BestBICNorm
```

```
## [1] -6.737083
```

```
Best_Norm=c(BestAICNorm,BestBICNorm)
rowSBest_Norm= c("AICbest","BICbest")
Norm_Results= data.frame(Best_Norm, row.names =rowSBest_Norm)
Norm_Results
```

```
##         Best_Norm
## AICbest -6.764806
## BICbest -6.737083
```

```
#Student-t
tAIC= c(infocriteria(ts_garchFit)[1], infocriteria(tgjr_garchFit)[1],
       infocriteria(te_garchFit)[1], infocriteria(tapARCHFit)[1],  infocriteria(ti_garchFit)[1] )
trowSAIC= c("ts_garch", "tgjr_garchFit", "te_garchFit", "tapARCHFit", "ti_garchFit")


tAIC_Results= data.frame(tAIC, row.names = trowSAIC)
tAIC_Results
```

```
##                    tAIC
## ts_garch      -6.866851
## tgjr_garchFit -6.871318
## te_garchFit   -6.879152
## tapARCHFit    -6.871485
## ti_garchFit   -6.868343
```

```
tAIC_Results[which.min(tAIC_Results$tAIC),]
```

```
## [1] -6.879152
```

```
tBIC= c(infocriteria(ts_garchFit)[2], infocriteria(tgjr_garchFit)[2],
       infocriteria(te_garchFit)[2], infocriteria(tapARCHFit)[2],  infocriteria(ti_garchFit)[2]  )
trowSBIC= c("ts_garch", "tgjr_garchFit", "te_garchFit", "tapARCHFit", "ti_garchFit")
tBIC_Results= data.frame(tBIC, row.names = trowSBIC)
tBIC_Results
```

```
##                    tBIC
## ts_garch      -6.839128
## tgjr_garchFit -6.841463
## te_garchFit   -6.849296
## tapARCHFit    -6.839497
## ti_garchFit   -6.842752
```

```
tBIC_Results[which.min(tBIC_Results$tBIC),]
```

```
## [1] -6.849296
```

```
#Goodness of Fit
Distribution=c("Normal", "Normal", "Normal", "Normal", "Normal",
               "Student's-t", "Student's-t", "Student's-t", "Student's-t", "Student's-t" )
rowsGARCH=c("s_garchFit", "gjr_garchFit","e_garchFit","apARCHFit", "i_garchFit",
            "ts_garchFit", "tgjr_garchFit", "te_garchFit", "tapARCHFit", "ti_garchFit")
Best=c("o", "o", "*", "o", "o",
       "o", "o", "*", "o","o")
final_AIC=c(AIC,tAIC)
final_BIC=c(BIC,tBIC)
Results <- data.frame("AIC" = final_AIC, "BIC" = final_BIC , "Distribution" = Distribution,
                      "Best"= Best ,row.names = rowsGARCH)
Results
```

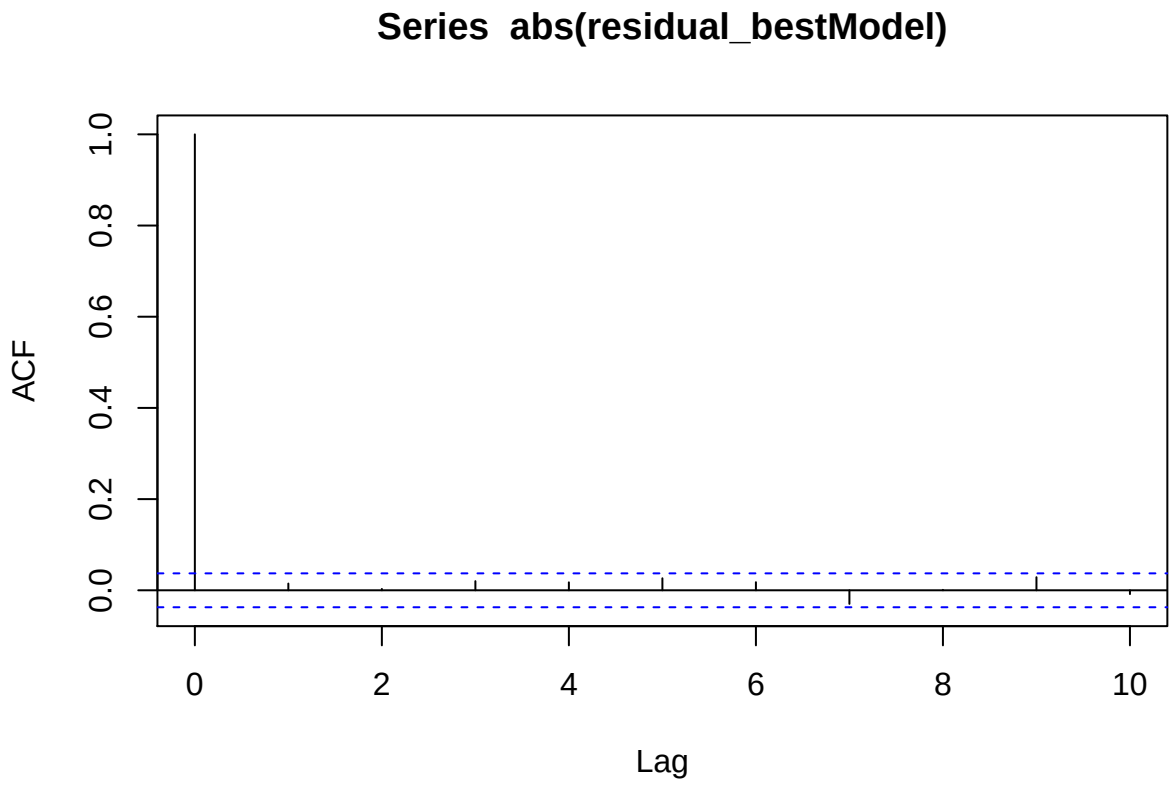
```
##                      AIC       BIC Distribution Best
## s_garchFit    -6.740216 -6.714626       Normal    o
## gjr_garchFit  -6.754373 -6.726650       Normal    o
## e_garchFit    -6.764806 -6.737083       Normal    *
## apARCHFit     -6.745484 -6.715628       Normal    o
## i_garchFit    -6.736848 -6.713390       Normal    o
## ts_garchFit   -6.866851 -6.839128  Student's-t    o
## tgjr_garchFit -6.871318 -6.841463  Student's-t    o
## te_garchFit   -6.879152 -6.849296  Student's-t    *
## tapARCHFit    -6.871485 -6.839497  Student's-t    o
## ti_garchFit   -6.868343 -6.842752  Student's-t    o
```

Validation of the model:

```
#Validation of the model
residual_bestModel= residuals(te_garchFit, standardize = TRUE)
Box.test(abs(residual_bestModel), 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  abs(residual_bestModel)
## X-squared = 10.423, df = 10, p-value = 0.4042
```

```
acf(abs(residual_bestModel), 10)
```

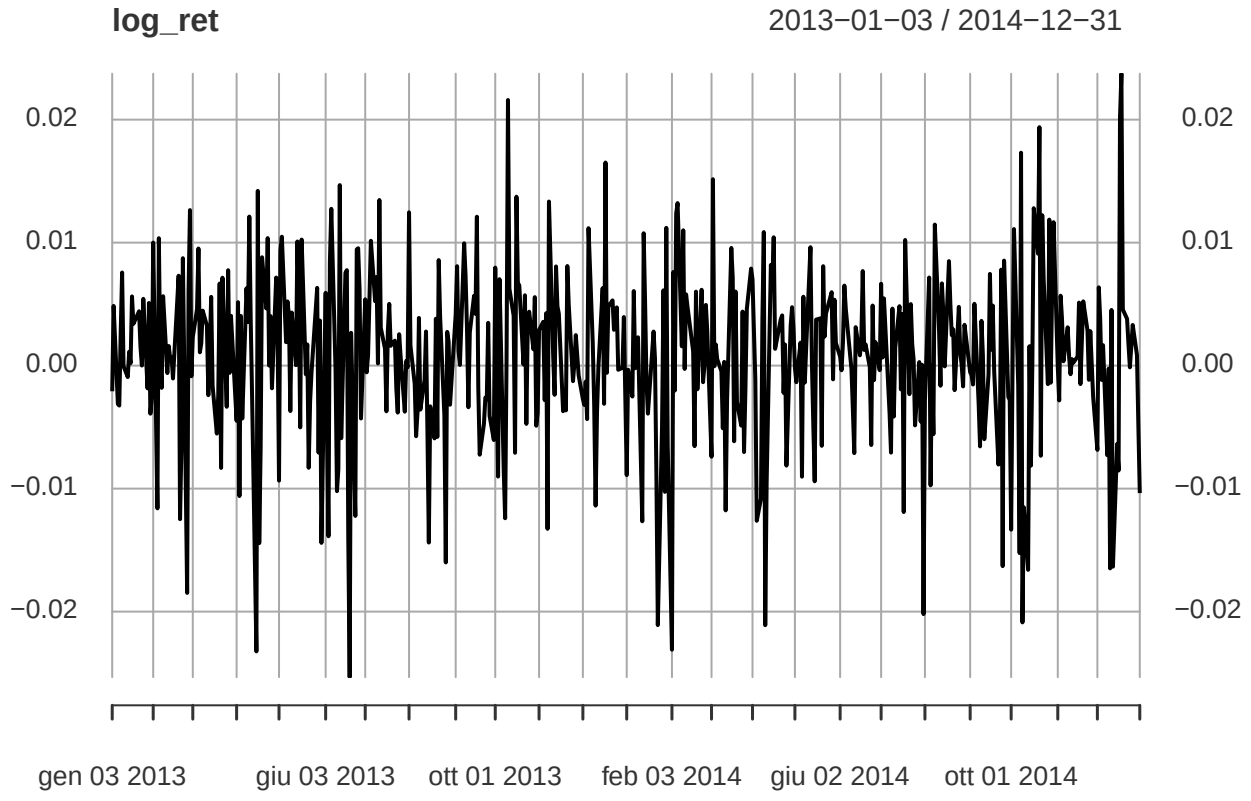


Second analysis on the Limit-Up/Limit-Down Regulation's effect on stock volatility

```
#DATA
library(tidyquant)

# Downloading SP500 using the library tidyquant

SP500= getSymbols("^GSPC", from = '2013-01-01',
                  to = "2015-01-01", auto.assign = FALSE)
head(SP500)
```

```
##            GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted
## 2013-01-02   1426.19   1462.43  1426.19    1462.42  4202600000       1462.42
## 2013-01-03   1462.42   1465.47  1455.53    1459.37  3829730000       1459.37
## 2013-01-04   1459.37   1467.94  1458.99    1466.47  3424290000       1466.47
## 2013-01-07   1466.47   1466.47  1456.62    1461.89  3304970000       1461.89
## 2013-01-08   1461.89   1461.89  1451.64    1457.15  3601600000       1457.15
## 2013-01-09   1457.15   1464.73  1457.15    1461.02  3674390000       1461.02
```

```
logret = diff(log(SP500$GSPC.Adjusted))
log_ret <- logret[-1]

plot(log_ret, length(log_ret))
```




```
log_ret
```

```
##            GSPC.Adjusted
## 2013-01-03 -2.087796e-03
## 2013-01-04  4.853300e-03
## 2013-01-07 -3.128003e-03
## 2013-01-08 -3.247640e-03
## 2013-01-09  2.652346e-03
## 2013-01-10  7.568700e-03
## 2013-01-11 -4.751492e-05
## 2013-01-14 -9.311048e-04
## 2013-01-15  1.128033e-03
## 2013-01-16  1.969725e-04
## 2013-01-17  5.627060e-03
## 2013-01-18  3.397492e-03
## 2013-01-22  4.418332e-03
## 2013-01-23  1.506342e-03
## 2013-01-24  6.614196e-06
## 2013-01-25  5.430709e-03
## 2013-01-28 -1.851334e-03
## 2013-01-29  5.093004e-03
## 2013-01-30 -3.907245e-03
## 2013-01-31 -2.566592e-03
## 2013-02-01  1.000251e-02
## 2013-02-04 -1.160583e-02
## 2013-02-05  1.036263e-02
```

```
## 2013-02-06  5.490198e-04
## 2013-02-07 -1.807031e-03
## 2013-02-08  5.641995e-03
## 2013-02-11 -6.063013e-04
## 2013-02-12  1.594001e-03
## 2013-02-13  5.920875e-04
## 2013-02-14  6.904334e-04
## 2013-02-15 -1.045628e-03
## 2013-02-19  7.309694e-03
## 2013-02-20 -1.248171e-02
## 2013-02-21 -6.323005e-03
## 2013-02-22  8.734214e-03
## 2013-02-25 -1.847928e-02
## 2013-02-26  6.090876e-03
## 2013-02-27  1.264570e-02
## 2013-02-28 -8.644531e-04
## 2013-03-01  2.321159e-03
## 2013-03-04  4.600127e-03
## 2013-03-05  9.520552e-03
## 2013-03-06  1.083925e-03
## 2013-03-07  1.814844e-03
## 2013-03-08  4.471129e-03
## 2013-03-11  3.243819e-03
## 2013-03-12 -2.406146e-03
## 2013-03-13  1.313190e-03
## 2013-03-14  5.587352e-03
## 2013-03-15 -1.619774e-03
## 2013-03-18 -5.525570e-03
## 2013-03-19 -2.425470e-03
## 2013-03-20  6.675164e-03
## 2013-03-21 -8.316924e-03
## 2013-03-22  7.148644e-03
## 2013-03-25 -3.345630e-03
## 2013-03-26  7.754964e-03
## 2013-03-27 -5.885231e-04
## 2013-03-28  4.048463e-03
## 2013-04-01 -4.483617e-03
## 2013-04-02  5.158934e-03
## 2013-04-03 -1.060213e-02
## 2013-04-04  4.040279e-03
## 2013-04-05 -4.304145e-03
## 2013-04-08  6.282959e-03
## 2013-04-09  3.538066e-03
## 2013-04-10  1.211544e-02
## 2013-04-11  3.545956e-03
## 2013-04-12 -2.840798e-03
## 2013-04-15 -2.323413e-02
## 2013-04-16  1.420584e-02
## 2013-04-17 -1.443131e-02
## 2013-04-18 -6.723556e-03
## 2013-04-19  8.808989e-03
## 2013-04-22  4.650798e-03
## 2013-04-23  1.036531e-02
## 2013-04-24  6.340319e-06
```

```
## 2013-04-25  4.026614e-03
## 2013-04-26 -1.843812e-03
## 2013-04-29  7.160315e-03
## 2013-04-30  2.481817e-03
## 2013-05-01 -9.351473e-03
## 2013-05-02  9.364004e-03
## 2013-05-03  1.047956e-02
## 2013-05-06  1.905961e-03
## 2013-05-07  5.216639e-03
## 2013-05-08  4.130539e-03
## 2013-05-09 -3.693918e-03
## 2013-05-10  4.312344e-03
## 2013-05-13  4.288884e-05
## 2013-05-14  1.009107e-02
## 2013-05-15  5.101103e-03
## 2013-05-16 -5.022332e-03
## 2013-05-17  1.024741e-02
## 2013-05-20 -7.078686e-04
## 2013-05-21  1.720905e-03
## 2013-05-22 -8.308074e-03
## 2013-05-23 -2.928115e-03
## 2013-05-24 -5.515174e-04
## 2013-05-28  6.320962e-03
## 2013-05-29 -7.072937e-03
## 2013-05-30  3.663625e-03
## 2013-05-31 -1.441058e-02
## 2013-06-03  5.918440e-03
## 2013-06-04 -5.526048e-03
## 2013-06-05 -1.387555e-02
## 2013-06-06  8.454455e-03
## 2013-06-07  1.274991e-02
## 2013-06-10 -3.468734e-04
## 2013-06-11 -1.020526e-02
## 2013-06-12 -8.404777e-03
## 2013-06-13  1.467607e-02
## 2013-06-14 -5.902401e-03
## 2013-06-17  7.538876e-03
## 2013-06-18  7.760964e-03
## 2013-06-19 -1.394830e-02
## 2013-06-20 -2.532842e-02
## 2013-06-21  2.666219e-03
## 2013-06-24 -1.221937e-02
## 2013-06-25  9.452456e-03
## 2013-06-26  9.544790e-03
## 2013-06-27  6.180691e-03
## 2013-06-28 -4.298789e-03
## 2013-07-01  5.389200e-03
## 2013-07-02 -5.450568e-04
## 2013-07-03  8.237078e-04
## 2013-07-05  1.015005e-02
## 2013-07-08  5.237805e-03
## 2013-07-09  7.203662e-03
## 2013-07-10  1.815761e-04
## 2013-07-11  1.346321e-02
```

```
## 2013-07-12  3.081730e-03
## 2013-07-15  1.373935e-03
## 2013-07-16 -3.715655e-03
## 2013-07-17  2.770206e-03
## 2013-07-18  5.020342e-03
## 2013-07-19  1.608756e-03
## 2013-07-22  2.030962e-03
## 2013-07-23 -1.853653e-03
## 2013-07-24 -3.818503e-03
## 2013-07-25  2.553210e-03
## 2013-07-26  8.279512e-04
## 2013-07-29 -3.743034e-03
## 2013-07-30  3.737472e-04
## 2013-07-31 -1.364188e-04
## 2013-08-01  1.246259e-02
## 2013-08-02  1.639114e-03
## 2013-08-05 -1.480931e-03
## 2013-08-06 -5.739473e-03
## 2013-08-07 -3.813126e-03
## 2013-08-08  3.877921e-03
## 2013-08-09 -3.576348e-03
## 2013-08-12 -1.153586e-03
## 2013-08-13  2.772210e-03
## 2013-08-14 -5.190063e-03
## 2013-08-15 -1.438457e-02
## 2013-08-16 -3.310067e-03
## 2013-08-19 -5.917778e-03
## 2013-08-20  3.813913e-03
## 2013-08-21 -5.796370e-03
## 2013-08-22  8.582442e-03
## 2013-08-23  3.939242e-03
## 2013-08-26 -4.047839e-03
## 2013-08-27 -1.600154e-02
## 2013-08-28  2.743878e-03
## 2013-08-29  1.961477e-03
## 2013-08-30 -3.179367e-03
## 2013-09-03  4.155575e-03
## 2013-09-04  8.084188e-03
## 2013-09-05  1.209132e-03
## 2013-09-06  5.442973e-05
## 2013-09-09  9.943282e-03
## 2013-09-10  7.318940e-03
## 2013-09-11  3.047635e-03
## 2013-09-12 -3.386142e-03
## 2013-09-13  2.711001e-03
## 2013-09-16  5.677009e-03
## 2013-09-17  4.208869e-03
## 2013-09-18  1.210412e-02
## 2013-09-19 -1.844654e-03
## 2013-09-20 -7.243054e-03
## 2013-09-23 -4.730759e-03
## 2013-09-24 -2.600522e-03
## 2013-09-25 -2.743225e-03
## 2013-09-26  3.479366e-03
```

```
## 2013-09-27 -4.082122e-03
## 2013-09-30 -6.047480e-03
## 2013-10-01  7.966725e-03
## 2013-10-02 -6.668919e-04
## 2013-10-03 -9.019973e-03
## 2013-10-04  7.028467e-03
## 2013-10-07 -8.542748e-03
## 2013-10-08 -1.240875e-02
## 2013-10-09  5.737416e-04
## 2013-10-10  2.159562e-02
## 2013-10-11  6.266595e-03
## 2013-10-14  4.066441e-03
## 2013-10-15 -7.088790e-03
## 2013-10-16  1.373280e-02
## 2013-10-17  6.721314e-03
## 2013-10-18  6.527406e-03
## 2013-10-21  9.173211e-05
## 2013-10-22  5.721116e-03
## 2013-10-23 -4.735753e-03
## 2013-10-24  3.252838e-03
## 2013-10-25  4.385215e-03
## 2013-10-28  1.328816e-03
## 2013-10-29  5.568661e-03
## 2013-10-30 -4.887849e-03
## 2013-10-31 -3.846771e-03
## 2013-11-01  2.899215e-03
## 2013-11-04  3.564199e-03
## 2013-11-05 -2.809531e-03
## 2013-11-06  4.256468e-03
## 2013-11-07 -1.327044e-02
## 2013-11-08  1.333821e-02
## 2013-11-11  7.226705e-04
## 2013-11-12 -2.373206e-03
## 2013-11-13  8.062753e-03
## 2013-11-14  4.825597e-03
## 2013-11-15  4.213147e-03
## 2013-11-18 -3.705053e-03
## 2013-11-19 -2.045055e-03
## 2013-11-20 -3.642236e-03
## 2013-11-21  8.095706e-03
## 2013-11-22  4.949190e-03
## 2013-11-25 -1.264141e-03
## 2013-11-26  1.497935e-04
## 2013-11-27  2.481998e-03
## 2013-11-29 -7.859980e-04
## 2013-12-02 -2.722724e-03
## 2013-12-03 -3.197956e-03
## 2013-12-04 -1.304343e-03
## 2013-12-05 -4.349016e-03
## 2013-12-06  1.117520e-02
## 2013-12-09  1.815451e-03
## 2013-12-10 -3.184725e-03
## 2013-12-11 -1.138140e-02
## 2013-12-12 -3.777689e-03
```

```
## 2013-12-13 -1.014154e-04
## 2013-12-16  6.300152e-03
## 2013-12-17 -3.105806e-03
## 2013-12-18  1.651091e-02
## 2013-12-19 -5.800970e-04
## 2013-12-20  4.807155e-03
## 2013-12-23  5.304029e-03
## 2013-12-24  2.911504e-03
## 2013-12-26  4.734305e-03
## 2013-12-27 -3.366415e-04
## 2013-12-30 -1.792699e-04
## 2013-12-31  3.951856e-03
## 2014-01-02 -8.901413e-03
## 2014-01-03 -3.330203e-04
## 2014-01-06 -2.514927e-03
## 2014-01-07  6.063345e-03
## 2014-01-08 -2.122317e-04
## 2014-01-09  3.482487e-04
## 2014-01-10  2.304030e-03
## 2014-01-13 -1.265597e-02
## 2014-01-14  1.075988e-02
## 2014-01-15  5.152889e-03
## 2014-01-16 -1.348028e-03
## 2014-01-17 -3.902781e-03
## 2014-01-21  2.769912e-03
## 2014-01-22  5.746998e-04
## 2014-01-23 -8.929325e-03
## 2014-01-24 -2.109642e-02
## 2014-01-27 -4.888222e-03
## 2014-01-28  6.121875e-03
## 2014-01-29 -1.026170e-02
## 2014-01-30  1.120404e-02
## 2014-01-31 -6.486290e-03
## 2014-02-03 -2.309660e-02
## 2014-02-04  7.612043e-03
## 2014-02-05 -2.030282e-03
## 2014-02-06  1.236305e-02
## 2014-02-07  1.321419e-02
## 2014-02-10  1.568005e-03
## 2014-02-11  1.100138e-02
## 2014-02-12 -2.692985e-04
## 2014-02-13  5.793212e-03
## 2014-02-14  4.797689e-03
## 2014-02-18  1.157803e-03
## 2014-02-19 -6.545862e-03
## 2014-02-20  6.013342e-03
## 2014-02-21 -1.920567e-03
## 2014-02-24  6.167455e-03
## 2014-02-25 -1.348591e-03
## 2014-02-26  2.169971e-05
## 2014-02-27  4.935881e-03
## 2014-02-28  2.778824e-03
## 2014-03-03 -7.405866e-03
## 2014-03-04  1.515232e-02
```

```
## 2014-03-05 -5.335244e-05
## 2014-03-06  1.716933e-03
## 2014-03-07  5.379447e-04
## 2014-03-10 -4.633536e-04
## 2014-03-11 -5.095097e-03
## 2014-03-12  3.051242e-04
## 2014-03-13 -1.177009e-02
## 2014-03-14 -2.825766e-03
## 2014-03-17  9.567718e-03
## 2014-03-18  7.193682e-03
## 2014-03-19 -6.150525e-03
## 2014-03-20  6.022334e-03
## 2014-03-21 -2.936980e-03
## 2014-03-24 -4.876581e-03
## 2014-03-25  4.394271e-03
## 2014-03-26 -7.024937e-03
## 2014-03-27 -1.901892e-03
## 2014-03-28  4.629489e-03
## 2014-03-31  7.892872e-03
## 2014-04-01  7.014688e-03
## 2014-04-02  2.849263e-03
## 2014-04-03 -1.127085e-03
## 2014-04-04 -1.261654e-02
## 2014-04-07 -1.080831e-02
## 2014-04-08  3.743538e-03
## 2014-04-09  1.085904e-02
## 2014-04-10 -2.110597e-02
## 2014-04-11 -9.532060e-03
## 2014-04-14  8.183708e-03
## 2014-04-15  6.734579e-03
## 2014-04-16  1.043387e-02
## 2014-04-17  1.362924e-03
## 2014-04-21  3.768016e-03
## 2014-04-22  4.083789e-03
## 2014-04-23 -2.215767e-03
## 2014-04-24  1.715488e-03
## 2014-04-25 -8.129345e-03
## 2014-04-28  3.230812e-03
## 2014-04-29  4.749460e-03
## 2014-04-30  2.987550e-03
## 2014-05-01 -1.432715e-04
## 2014-05-02 -1.349355e-03
## 2014-05-05  1.869467e-03
## 2014-05-06 -9.029031e-03
## 2014-05-07  5.600755e-03
## 2014-05-08 -1.374569e-03
## 2014-05-09  1.518323e-03
## 2014-05-12  9.626256e-03
## 2014-05-13  4.216689e-04
## 2014-05-14 -4.712090e-03
## 2014-05-15 -9.405904e-03
## 2014-05-16  3.739962e-03
## 2014-05-19  3.837414e-03
## 2014-05-20 -6.519605e-03
```

```
## 2014-05-21  8.083340e-03
## 2014-05-22  2.359444e-03
## 2014-05-23  4.239393e-03
## 2014-05-27  5.969950e-03
## 2014-05-28 -1.114693e-03
## 2014-05-29  5.352759e-03
## 2014-05-30  1.841980e-03
## 2014-06-02  7.275617e-04
## 2014-06-03 -3.792887e-04
## 2014-06-04  1.889877e-03
## 2014-06-05  6.504082e-03
## 2014-06-06  4.617083e-03
## 2014-06-09  9.383313e-04
## 2014-06-10 -2.460142e-04
## 2014-06-11 -3.543311e-03
## 2014-06-12 -7.114139e-03
## 2014-06-13  3.129659e-03
## 2014-06-16  8.363553e-04
## 2014-06-17  2.170212e-03
## 2014-06-18  7.689243e-03
## 2014-06-19  1.276663e-03
## 2014-06-20  1.728564e-03
## 2014-06-23 -1.324730e-04
## 2014-06-24 -6.456107e-03
## 2014-06-25  4.885558e-03
## 2014-06-26 -1.179579e-03
## 2014-06-27  1.909045e-03
## 2014-06-30 -3.723263e-04
## 2014-07-01  6.655573e-03
## 2014-07-02  6.585962e-04
## 2014-07-03  5.464550e-03
## 2014-07-07 -3.931239e-03
## 2014-07-08 -7.073762e-03
## 2014-07-09  4.633516e-03
## 2014-07-10 -4.139628e-03
## 2014-07-11  1.469842e-03
## 2014-07-14  4.831861e-03
## 2014-07-15 -1.933965e-03
## 2014-07-16  4.192285e-03
## 2014-07-17 -1.190461e-02
## 2014-07-18  1.021261e-02
## 2014-07-21 -2.322947e-03
## 2014-07-22  5.003611e-03
## 2014-07-23  1.752901e-03
## 2014-07-24  4.880365e-04
## 2014-07-25 -4.860946e-03
## 2014-07-28  2.881132e-04
## 2014-07-29 -4.538068e-03
## 2014-07-30  6.091086e-05
## 2014-07-31 -2.020193e-02
## 2014-08-01 -2.863216e-03
## 2014-08-04  7.163315e-03
## 2014-08-05 -9.732679e-03
## 2014-08-06  1.563827e-05
```

```
## 2014-08-07 -5.572115e-03
## 2014-08-08  1.146542e-02
## 2014-08-11  2.755625e-03
## 2014-08-12 -1.637982e-03
## 2014-08-13  6.684767e-03
## 2014-08-14  4.336398e-03
## 2014-08-15 -6.137475e-05
## 2014-08-18  8.495483e-03
## 2014-08-19  4.988191e-03
## 2014-08-20  2.474748e-03
## 2014-08-21  2.945547e-03
## 2014-08-22 -1.994575e-03
## 2014-08-25  4.776354e-03
## 2014-08-26  1.050529e-03
## 2014-08-27  4.998575e-05
## 2014-08-28 -1.691331e-03
## 2014-08-29  3.314914e-03
## 2014-09-02 -5.442143e-04
## 2014-09-03 -7.794445e-04
## 2014-09-04 -1.535600e-03
## 2014-09-05  5.023248e-03
## 2014-09-08 -3.077846e-03
## 2014-09-09 -6.566522e-03
## 2014-09-10  3.639444e-03
## 2014-09-11  8.815169e-04
## 2014-09-12 -5.980405e-03
## 2014-09-15 -7.104037e-04
## 2014-09-16  7.456507e-03
## 2014-09-17  1.294805e-03
## 2014-09-18  4.879257e-03
## 2014-09-19 -4.773835e-04
## 2014-09-22 -8.045602e-03
## 2014-09-23 -5.793250e-03
## 2014-09-24  7.801977e-03
## 2014-09-25 -1.630091e-02
## 2014-09-26  8.539261e-03
## 2014-09-29 -2.550051e-03
## 2014-09-30 -2.789817e-03
## 2014-10-01 -1.333711e-02
## 2014-10-02  5.143449e-06
## 2014-10-03  1.110364e-02
## 2014-10-06 -1.566386e-03
## 2014-10-07 -1.524162e-02
## 2014-10-08  1.731095e-02
## 2014-10-09 -2.087785e-02
## 2014-10-10 -1.151708e-02
## 2014-10-13 -1.660503e-02
## 2014-10-14  1.577620e-03
## 2014-10-15 -8.133301e-03
## 2014-10-16  1.449675e-04
## 2014-10-17  1.280181e-02
## 2014-10-20  9.101116e-03
## 2014-10-21  1.938537e-02
## 2014-10-22 -7.326101e-03
```

```
## 2014-10-23  1.222831e-02
## 2014-10-24  7.028690e-03
## 2014-10-27 -1.502697e-03
## 2014-10-28  1.186836e-02
## 2014-10-29 -1.386316e-03
## 2014-10-30  6.210797e-03
## 2014-10-31  1.166311e-02
## 2014-11-03 -1.189288e-04
## 2014-11-04 -2.833853e-03
## 2014-11-05  5.684311e-03
## 2014-11-06  3.768404e-03
## 2014-11-07  3.495251e-04
## 2014-11-10  3.115327e-03
## 2014-11-11  6.964517e-04
## 2014-11-12 -7.013627e-04
## 2014-11-13  5.297044e-04
## 2014-11-14  2.402412e-04
## 2014-11-17  7.350888e-04
## 2014-11-18  5.120850e-03
## 2014-11-19 -1.502287e-03
## 2014-11-20  1.965164e-03
## 2014-11-21  5.223213e-03
## 2014-11-24  2.859930e-03
## 2014-11-25 -1.150692e-03
## 2014-11-26  2.802053e-03
## 2014-11-28 -2.545664e-03
## 2014-12-01 -6.852790e-03
## 2014-12-02  6.364167e-03
## 2014-12-03  3.757674e-03
## 2014-12-04 -1.162572e-03
## 2014-12-05  1.663831e-03
## 2014-12-08 -7.283022e-03
## 2014-12-09 -2.378522e-04
## 2014-12-10 -1.648612e-02
## 2014-12-11  4.525434e-03
## 2014-12-12 -1.634647e-02
## 2014-12-15 -6.362786e-03
## 2014-12-16 -8.525260e-03
## 2014-12-17  2.014807e-02
## 2014-12-18  2.373137e-02
## 2014-12-19  4.559638e-03
## 2014-12-22  3.803223e-03
## 2014-12-23  1.744839e-03
## 2014-12-24 -1.393062e-04
## 2014-12-26  3.304110e-03
## 2014-12-29  8.614030e-04
## 2014-12-30 -4.900593e-03
## 2014-12-31 -1.036438e-02
```

```
#Stationarity test
library(tseries)
adf.test(log_ret, alternative = "stationary")
```

```
## Warning in adf.test(log_ret, alternative = "stationary"): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  log_ret
## Dickey-Fuller = -7.8121, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
#ARCH EFFECT

library(fDMA)
archtest(as.vector(log_ret))
```

```
##
##  Engle's LM ARCH Test
##
## data:  as.vector(log_ret)
## statistic = 13.505, lag = 1, p-value = 0.000238
## alternative hypothesis: ARCH effects of order 1 are present
```

```
#DUMMY VARIABLES

n=length(log_ret)
#PHASE 1 LULD
e1=which(row.names(as.matrix(log_ret))>="2013-04-08"& row.names(as.matrix(log_ret))<="2013-05-31")
d1=rep(0,n)
d1[e1]=1
#PHASE 2 LULD
e2=which(row.names(as.matrix(log_ret))>="2013-08-05"& row.names(as.matrix(log_ret))<="2013-12-09")
d2=rep(0,n)
d2[e2]=1
d1
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [260] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [297] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [334] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [371] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [408] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [445] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [482] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
d2
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [260] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
## [297] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [334] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [371] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [408] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [445] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [482] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
df_dummies=matrix(c(d1,d2), ncol = 2)
```

Fitting GARCH models:

```
# GARCH MODELS assuming a Normal distribution
#sGARCH
library(rugarch)

s_garchMod <- ugarchspec(mean.model = list(armaOrder = c(1, 0), include.mean = TRUE
),
variance.model = list(model = 'sGARCH',
                      garchOrder = c(1, 1),
                      external.regressors = df_dummies
),
distribution.model = "norm")

s_garchFit <- ugarchfit(spec=s_garchMod, data=log_ret)
s_garchFit
```

```
##
## *---------------------------------*
## *          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      0.000840    0.000066   12.7866 0.000000
## ar1    -0.053777    0.026972   -1.9938 0.046176
## omega   0.000008    0.000000   17.0431 0.000000
## alpha1  0.201098    0.018897   10.6419 0.000000
## beta1   0.624838    0.029982   20.8404 0.000000
## vxreg1  0.000008    0.000001   15.2650 0.000000
## vxreg2  0.000001    0.000000 1085.2129 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000840    0.000202   4.1497 0.000033
## ar1    -0.053777    0.048501  -1.1088 0.267524
## omega   0.000008    0.000001  11.5665 0.000000
## alpha1  0.201098    0.049469   4.0651 0.000048
## beta1   0.624838    0.030416  20.5431 0.000000
```

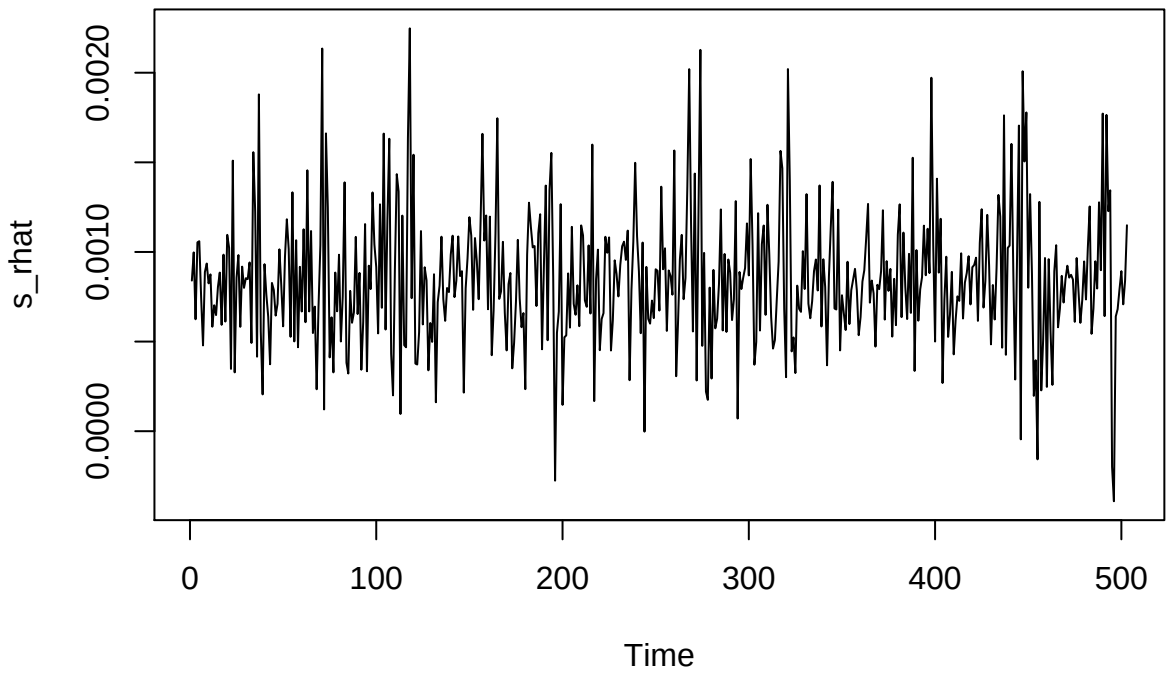
```
## vxreg1  0.000008    0.000001  10.2110 0.000000
## vxreg2  0.000001    0.000000 705.9982 0.000000
##
## LogLikelihood : 1807.367
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.1585
## Bayes        -7.0998
## Shibata      -7.1589
## Hannan-Quinn -7.1355
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.09561  0.7572
## Lag[2*(p+q)+(p+q)-1][2]   0.19632  0.9989
## Lag[4*(p+q)+(p+q)-1][5]   0.67587  0.9834
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.4936  0.4823
## Lag[2*(p+q)+(p+q)-1][5]    0.8012  0.9028
## Lag[4*(p+q)+(p+q)-1][9]    1.8493  0.9222
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1056 0.500 2.000  0.7452
## ARCH Lag[5]    0.6622 1.440 1.667  0.8349
## ARCH Lag[7]    1.1364 2.315 1.543  0.8904
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  19.2901
## Individual Statistics:
## mu     0.05189
## ar1    0.19398
## omega  3.70110
## alpha1 0.08649
## beta1  0.06999
## vxreg1 2.81711
## vxreg2 2.65670
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.69 1.9 2.35
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
```

```
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           2.4809 0.01344  **
## Negative Sign Bias  0.9585 0.33828
## Positive Sign Bias  0.2047 0.83788
## Joint Effect        9.3361 0.02514  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     32.63     0.026537
## 2    30     45.53     0.026122
## 3    40     65.35     0.005149
## 4    50     79.60     0.003710
##
##
## Elapsed time : 0.2375741
```

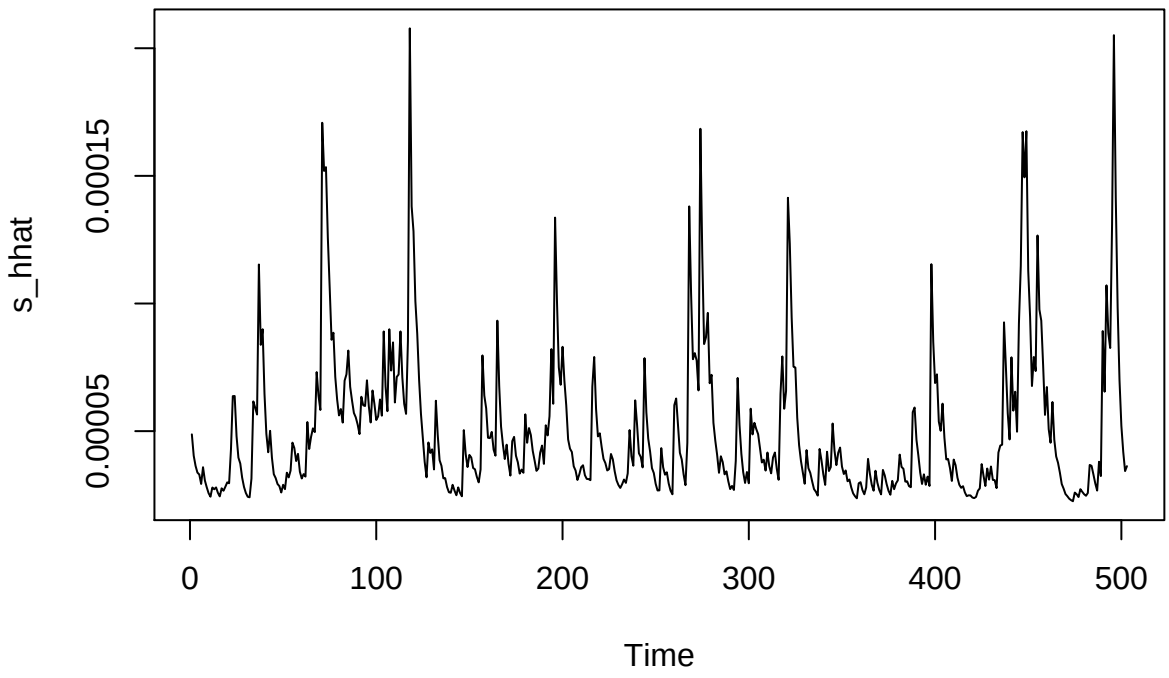
```
## Results review 1
coef(s_garchFit)
```

```
##            mu           ar1         omega        alpha1         beta1
##  8.402561e-04 -5.377668e-02  8.186756e-06  2.010979e-01  6.248384e-01
##        vxreg1        vxreg2
##  7.715679e-06  1.015541e-06
```

```
s_rhat <- s_garchFit@fit$fitted.values
plot.ts(s_rhat)
```




```
s_hhat <- ts(s_garchFit@fit$sigma^2)
plot.ts(s_hhat)
```




```
## Results review 2
fit.val     <- coef(s_garchFit)
fit.sd      <- diag(vcov(s_garchFit))
true.val = s_garchFit@fit$tval

fit.conf.lb <- fit.val + qnorm(0.025) * fit.sd
fit.conf.ub <- fit.val + qnorm(0.975) * fit.sd
print(fit.val)
```

```
##            mu           ar1         omega        alpha1         beta1
##  8.402561e-04 -5.377668e-02  8.186756e-06  2.010979e-01  6.248384e-01
##        vxreg1        vxreg2
##  7.715679e-06  1.015541e-06
```

```
print(fit.sd)
```

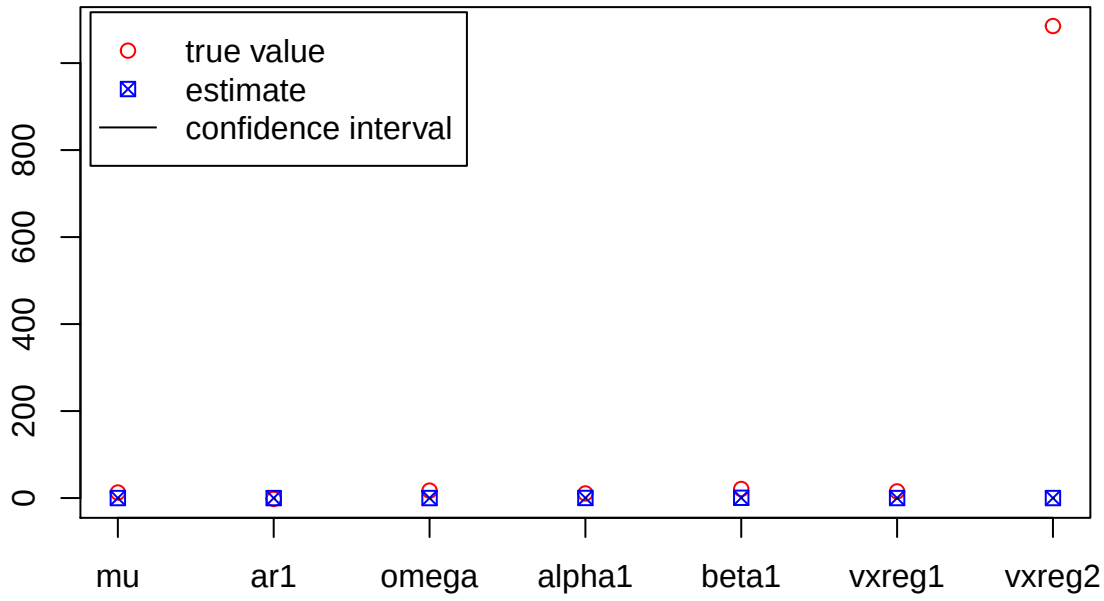
```
## [1] -4.318315e-09  7.275002e-04 -2.307418e-13 -3.570861e-04  8.989237e-04
## [6] -2.554793e-13 -8.757194e-19
```

```
print(true.val)
```

```
##          mu         ar1       omega      alpha1       beta1      vxreg1
##   12.786583   -1.993781   17.043105   10.641945   20.840414   15.264980
##      vxreg2
## 1085.212918
```

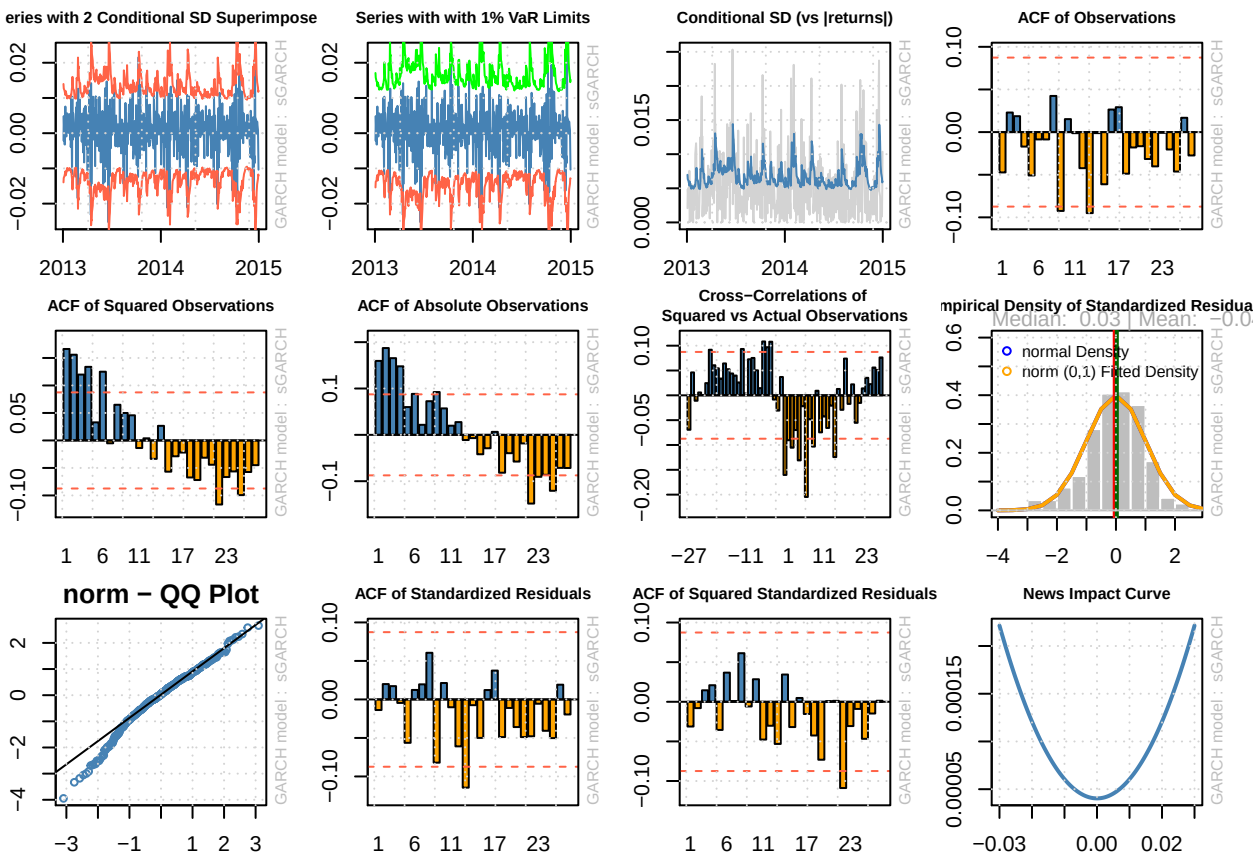
```
plot(true.val, pch = 1, col = "red",
     ylim = range(c(fit.conf.lb, fit.conf.ub, true.val)),
```

```
     xlab = "", ylab = "", axes = FALSE)
box(); axis(1, at = 1:length(fit.val), labels = names(fit.val)); axis(2)
points(coef(s_garchFit), col = "blue", pch = 7)
for (i in 1:length(fit.val)) {
  lines(c(i,i), c(fit.conf.lb[i], fit.conf.ub[i]))
}
legend( "topleft", legend = c("true value", "estimate", "confidence interval"),
        col = c("red", "blue", 1), pch = c(1, 7, NA), lty = c(NA, NA, 1), inset = 0.01)
```




```
par(mfrow=c(2, 3))
par(mar = c(2, 2, 2, 2))
plot(s_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#gjrGARCH
gjr_garchMod <- ugarchspec(
  variance.model=list(model="gjrGARCH",
                      garchOrder=c(1,1), external.regressors = df_dummies ),
  mean.model=list(armaOrder=c(1,0),
                  include.mean=TRUE
  ),
  distribution.model="norm"
)
gjr_garchFit <- ugarchfit(spec=gjr_garchMod, data=log_ret)
coef(gjr_garchFit)
```

```
##           mu           ar1         omega        alpha1         beta1
##  6.013808e-04 -5.079000e-02  5.094779e-06  1.127093e-08  7.230763e-01
##        gamma1        vxreg1        vxreg2
##  3.238522e-01  7.249355e-06  9.747307e-07
```

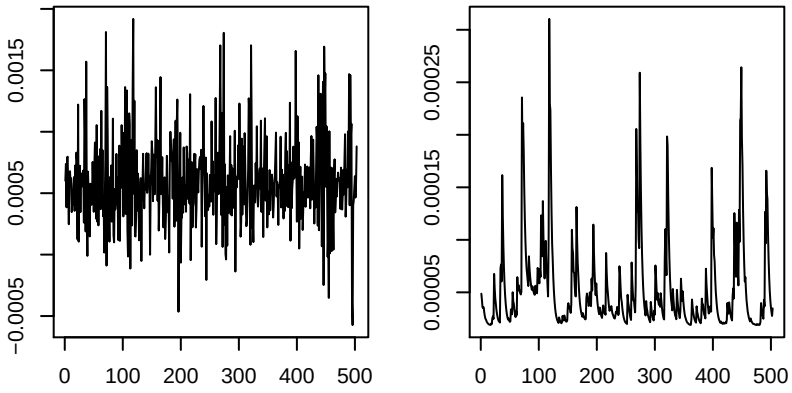
```
gjr_garchFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : gjrGARCH(1,1)
```

```
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000601    0.000117   5.16171 0.000000
## ar1    -0.050790    0.058495  -0.86828 0.385242
## omega   0.000005    0.000001   4.22847 0.000024
## alpha1  0.000000    0.023316   0.00000 1.000000
## beta1   0.723076    0.065324  11.06903 0.000000
## gamma1  0.323852    0.135816   2.38448 0.017103
## vxreg1  0.000007    0.000006   1.21559 0.224142
## vxreg2  0.000001    0.000000 152.56273 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000601    0.000198  3.04198 0.002350
## ar1    -0.050790    0.132782 -0.38251 0.702086
## omega   0.000005    0.000003  1.47590 0.139971
## alpha1  0.000000    0.052155  0.00000 1.000000
## beta1   0.723076    0.178994  4.03967 0.000054
## gamma1  0.323852    0.373783  0.86642 0.386261
## vxreg1  0.000007    0.000017  0.42914 0.667820
## vxreg2  0.000001    0.000000 27.39545 0.000000
##
## LogLikelihood : 1819.663
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.2034
## Bayes        -7.1363
## Shibata      -7.2039
## Hannan-Quinn -7.1771
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.0142  0.9051
## Lag[2*(p+q)+(p+q)-1][2]    0.2162  0.9984
## Lag[4*(p+q)+(p+q)-1][5]    0.7039  0.9812
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.025  0.3114
## Lag[2*(p+q)+(p+q)-1][5]     2.505  0.5049
## Lag[4*(p+q)+(p+q)-1][9]     3.873  0.6104
## d.o.f=2
##
## Weighted ARCH LM Tests
```

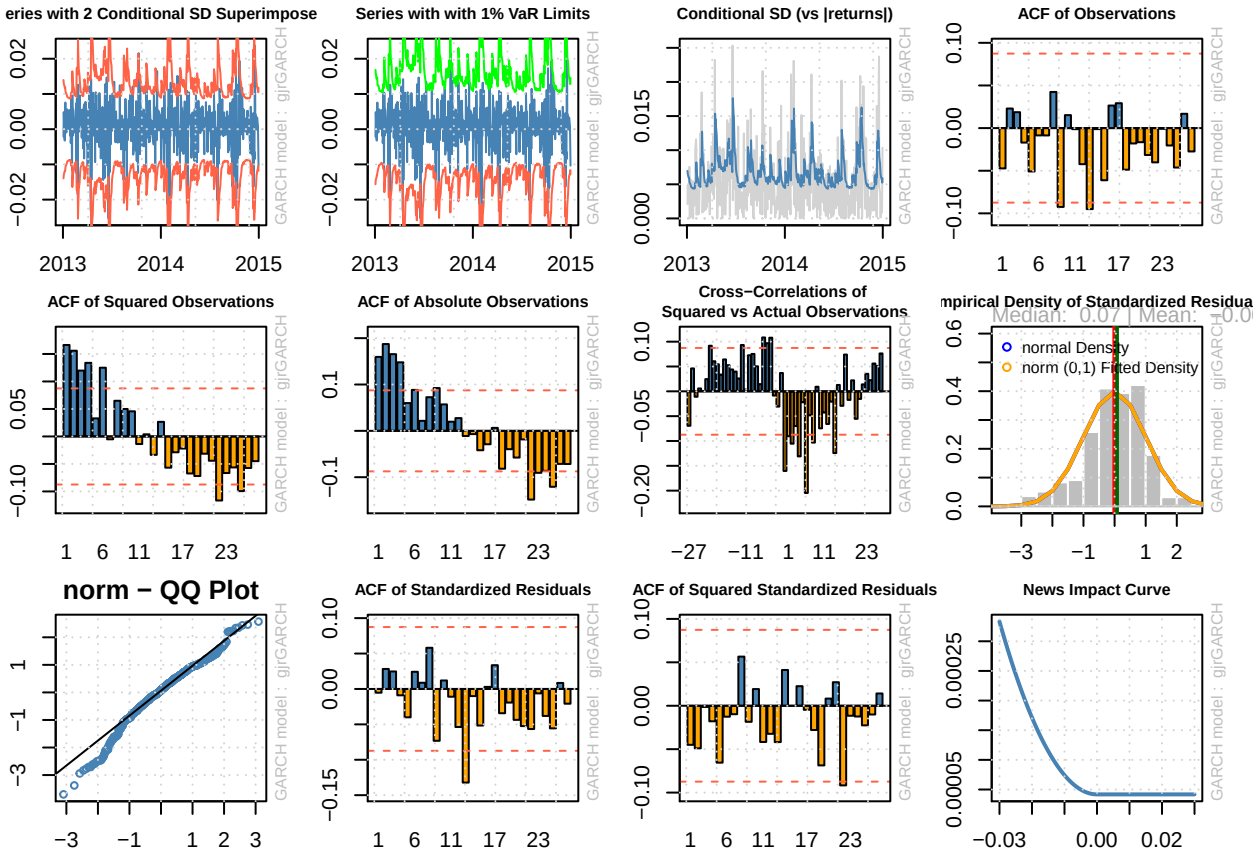
```
## -------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3] 0.0009615 0.500 2.000  0.9753
## ARCH Lag[5] 1.4261671 1.440 1.667  0.6119
## ARCH Lag[7] 1.7484958 2.315 1.543  0.7704
##
## Nyblom stability test
## -------------------------------------
## Joint Statistic:  18.4937
## Individual Statistics:
## mu     0.04812
## ar1    0.28110
## omega  4.57960
## alpha1 0.35133
## beta1  0.03505
## gamma1 0.13483
## vxreg1 2.25484
## vxreg2 1.99854
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.89 2.11 2.59
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## -------------------------------------
##                    t-value     prob sig
## Sign Bias           2.8470 0.004596 ***
## Negative Sign Bias  2.0161 0.044321  **
## Positive Sign Bias  0.7082 0.479165
## Joint Effect        9.4240 0.024153  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     42.33    0.0016002
## 2    30     55.91    0.0019389
## 3    40     69.17    0.0020635
## 4    50    100.28    0.0000219
##
##
## Elapsed time : 0.5109761
```

```
gjr_rhat <- gjr_garchFit@fit$fitted.values
plot.ts(gjr_rhat)
gjr_hhat <- ts(gjr_garchFit@fit$sigma^2)
plot.ts(gjr_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(gjr_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#eGARCH
e_garchMod <- ugarchspec(
  variance.model=list(model="eGARCH",
                      garchOrder=c(1,1), external.regressors = df_dummies ),
  mean.model=list(armaOrder=c(1,0),
                  include.mean=TRUE
  ),
  distribution.model="norm"
)
e_garchFit <- ugarchfit(spec=e_garchMod, data=log_ret)
coef(e_garchFit)
```

```
##            mu           ar1         omega        alpha1         beta1
##  0.0004433548 -0.0384901499 -0.8373357009 -0.3060358058  0.9174478258
##         gamma1        vxreg1        vxreg2
##  0.0049793591  0.0702686146  0.0155810130
```

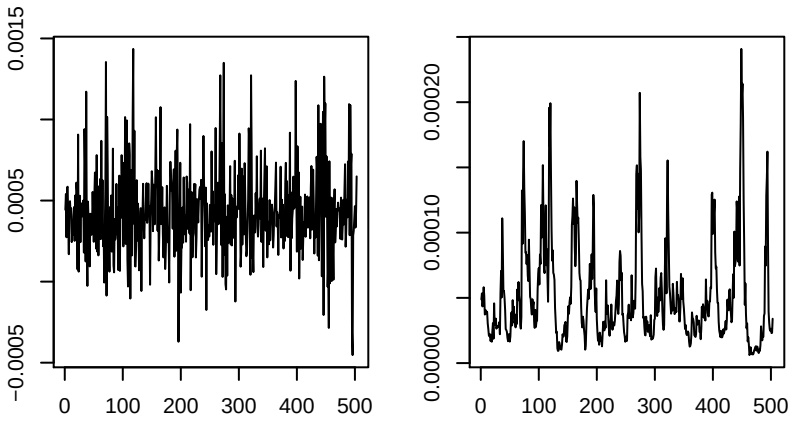
```
e_garchFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : eGARCH(1,1)
```

```
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000443    0.000212    2.09038 0.036584
## ar1    -0.038490    0.046133   -0.83432 0.404099
## omega  -0.837336    0.013444  -62.28355 0.000000
## alpha1 -0.306036    0.031828   -9.61545 0.000000
## beta1   0.917448    0.000130 7079.54844 0.000000
## gamma1  0.004979    0.036065    0.13807 0.890187
## vxreg1  0.070269    0.022581    3.11192 0.001859
## vxreg2  0.015581    0.013590    1.14652 0.251580
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000443    0.000208    2.12931 0.033229
## ar1    -0.038490    0.051070   -0.75367 0.451045
## omega  -0.837336    0.014679  -57.04207 0.000000
## alpha1 -0.306036    0.033679   -9.08689 0.000000
## beta1   0.917448    0.000130 7071.33530 0.000000
## gamma1  0.004979    0.037696    0.13209 0.894910
## vxreg1  0.070269    0.022280    3.15388 0.001611
## vxreg2  0.015581    0.013468    1.15691 0.247309
##
## LogLikelihood : 1832.049
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.2527
## Bayes        -7.1856
## Shibata      -7.2532
## Hannan-Quinn -7.2263
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.1391  0.7092
## Lag[2*(p+q)+(p+q)-1][2]    0.3472  0.9901
## Lag[4*(p+q)+(p+q)-1][5]    0.9200  0.9581
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                   0.004206  0.9483
## Lag[2*(p+q)+(p+q)-1][5]  0.876356  0.8870
## Lag[4*(p+q)+(p+q)-1][9]  1.980296  0.9073
## d.o.f=2
##
## Weighted ARCH LM Tests
```

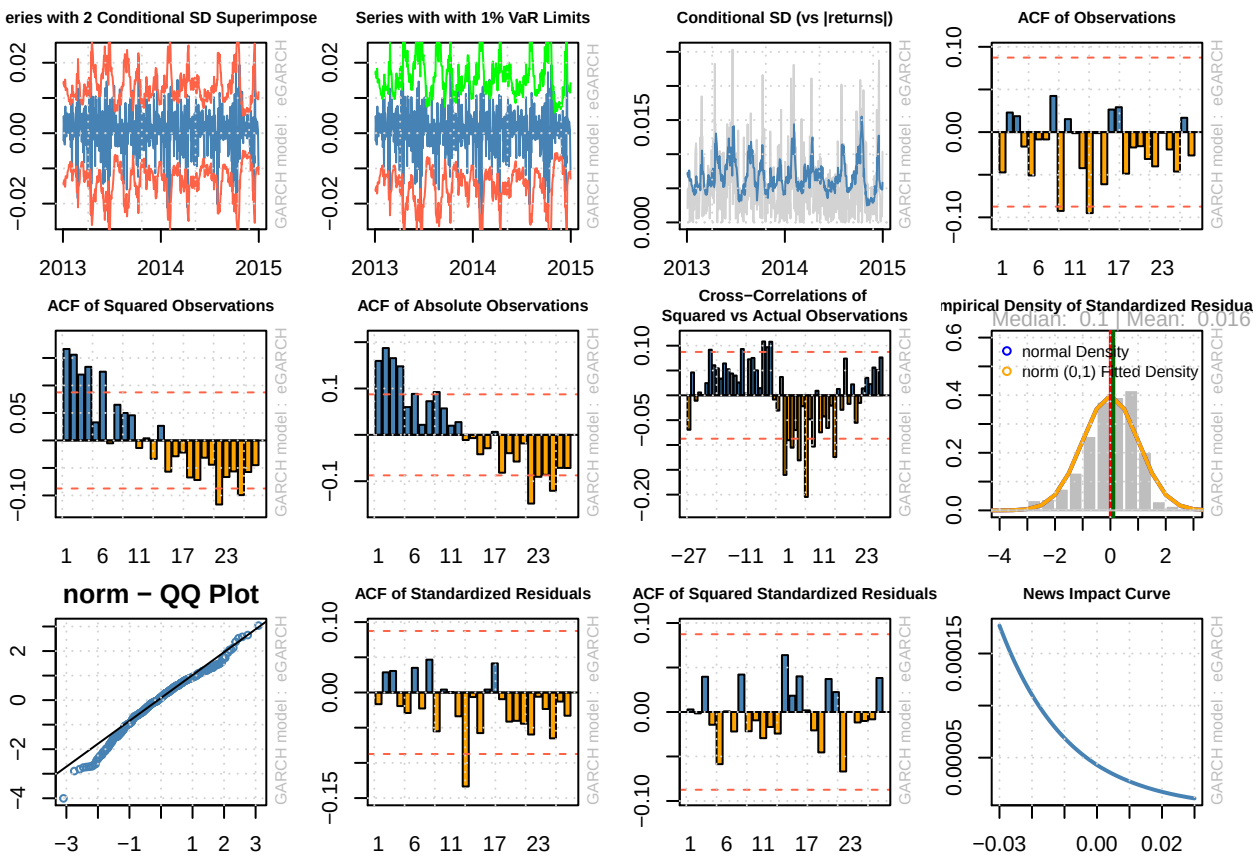
```
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.7896 0.500 2.000  0.3742
## ARCH Lag[5]    1.9101 1.440 1.667  0.4915
## ARCH Lag[7]    2.2172 2.315 1.543  0.6714
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.2258
## Individual Statistics:
## mu     0.14122
## ar1    0.10416
## omega  0.10126
## alpha1 0.28898
## beta1  0.10825
## gamma1 0.29500
## vxreg1 0.01462
## vxreg2 0.01992
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.89 2.11 2.59
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias            2.427 0.01556  **
## Negative Sign Bias   1.254 0.21043
## Positive Sign Bias   1.608 0.10845
## Joint Effect         5.999 0.11164
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     45.03    0.0006784
## 2    30     59.84    0.0006461
## 3    40     70.44    0.0015046
## 4    50     76.82    0.0067468
##
##
## Elapsed time : 0.924278
```

```
e_rhat <- e_garchFit@fit$fitted.values
plot.ts(e_rhat)
e_hhat <- ts(e_garchFit@fit$sigma^2)
plot.ts(e_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(e_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
# apARCH
apARCHMod <- ugarchspec(variance.model=list(model="apARCH",
                                            garchOrder=c(1,1),
                                            external.regressors = df_dummies ),
                        mean.model=list(armaOrder=c(1,0)),
                        distribution.model="norm")
apARCHFit <- ugarchfit(spec=apARCHMod, data= log_ret)
coef(apARCHFit)
```

```
##            mu           ar1         omega        alpha1         beta1
##  6.242414e-04 -3.546749e-02  1.875064e-08  6.259902e-02  6.520160e-01
##        gamma1         delta        vxreg1        vxreg2
##  6.260736e-01  3.111899e+00  4.802060e-08  1.265436e-08
```

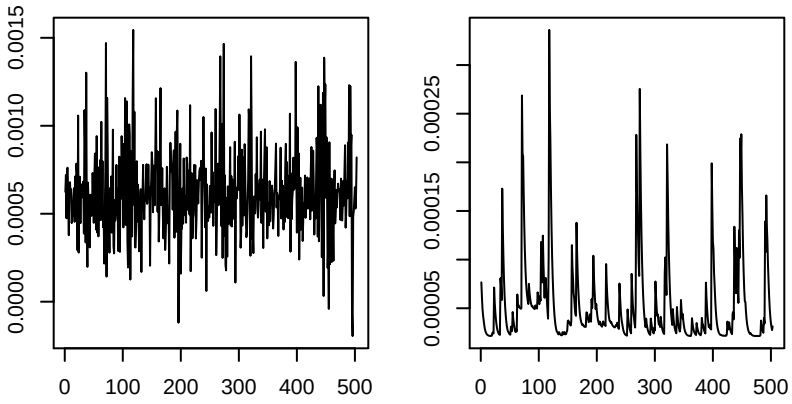
```
apARCHFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : apARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
```

```
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000624    0.000268  2.332026 0.019699
## ar1    -0.035467    0.050592 -0.701056 0.483268
## omega   0.000000    0.000000  0.071751 0.942800
## alpha1  0.062599    0.031673  1.976391 0.048110
## beta1   0.652016    0.012918 50.472089 0.000000
## gamma1  0.626074    0.229805  2.724366 0.006443
## delta   3.111899    0.033259 93.565469 0.000000
## vxreg1  0.000000    0.000001  0.069258 0.944784
## vxreg2  0.000000    0.000001  0.021642 0.982733
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000624    0.000654  0.955197  0.33948
## ar1    -0.035467    0.637195 -0.055662  0.95561
## omega   0.000000    0.000024  0.000774  0.99938
## alpha1  0.062599    0.167468  0.373796  0.70856
## beta1   0.652016    7.080015  0.092092  0.92663
## gamma1  0.626074    7.675691  0.081566  0.93499
## delta   3.111899    5.292779  0.587952  0.55656
## vxreg1  0.000000    0.000030  0.001577  0.99874
## vxreg2  0.000000    0.000028  0.000451  0.99964
##
## LogLikelihood : 1813.541
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.1751
## Bayes        -7.0996
## Shibata      -7.1757
## Hannan-Quinn -7.1455
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.2314  0.6305
## Lag[2*(p+q)+(p+q)-1][2]    0.4492  0.9752
## Lag[4*(p+q)+(p+q)-1][5]    0.9864  0.9488
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.7954  0.3725
## Lag[2*(p+q)+(p+q)-1][5]    2.0776  0.6002
## Lag[4*(p+q)+(p+q)-1][9]    3.2668  0.7142
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
```

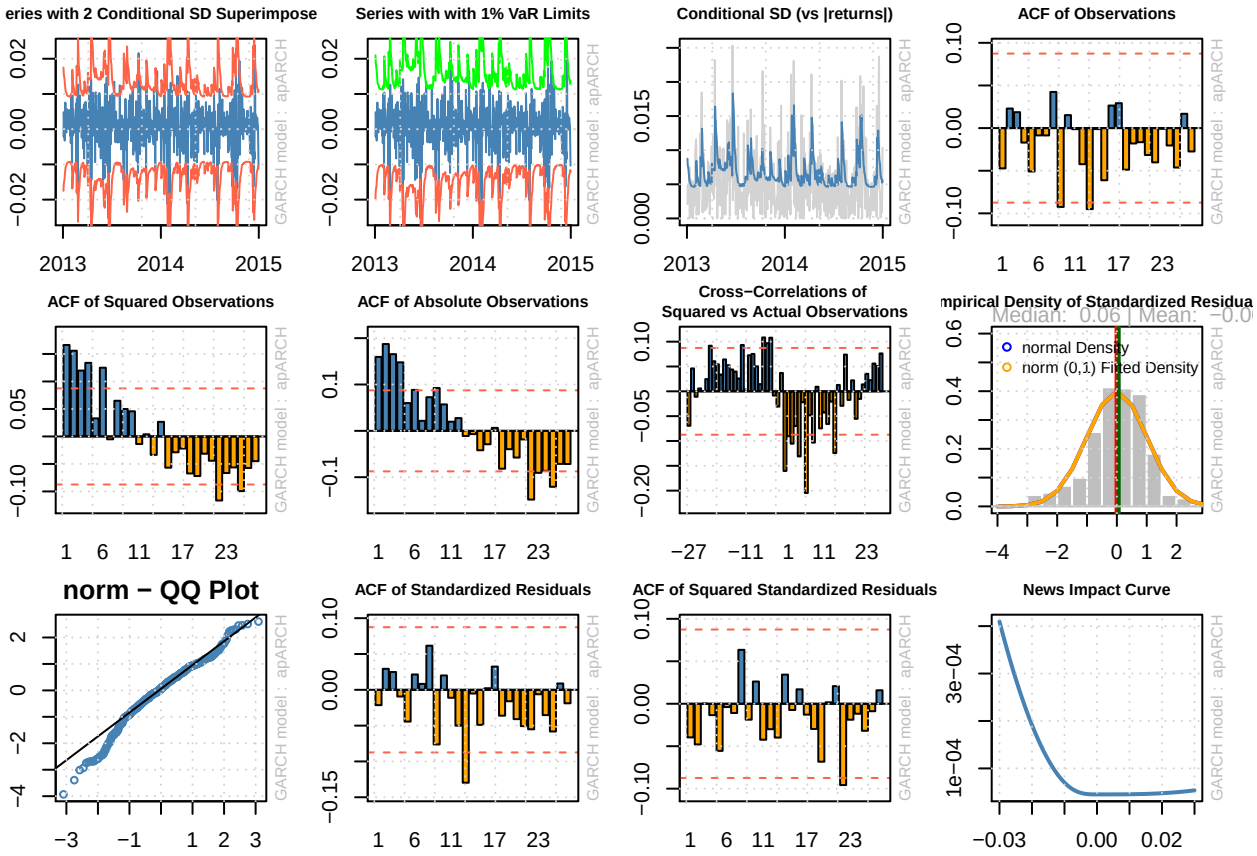
```
##             Statistic Shape Scale P-Value
## ARCH Lag[3] 1.629e-06 0.500 2.000  0.9990
## ARCH Lag[5] 1.001e+00 1.440 1.667  0.7327
## ARCH Lag[7] 1.213e+00 2.315 1.543  0.8767
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  NaN
## Individual Statistics:
## mu      0.04770
## ar1     0.50384
## omega  29.08201
## alpha1  0.13025
## beta1   0.06022
## gamma1  0.10445
## delta   0.05598
## vxreg1      NaN
## vxreg2      NaN
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.1 2.32 2.82
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value     prob sig
## Sign Bias           2.9698 0.003124 ***
## Negative Sign Bias  1.9222 0.055146   *
## Positive Sign Bias  0.8558 0.392536
## Joint Effect        9.8702 0.019702  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     46.70    3.940e-04
## 2    30     53.04    4.160e-03
## 3    40     81.89    7.046e-05
## 4    50     94.71    9.699e-05
##
##
## Elapsed time : 1.169584
```

```
apARCH_rhat <- apARCHFit@fit$fitted.values
plot.ts(apARCH_rhat)
apARCH_hhat <- ts(apARCHFit@fit$sigma^2)
plot.ts(apARCH_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(apARCHFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
# IGARCH
i_garchMod <- ugarchspec(variance.model=list(model="iGARCH",
                                             garchOrder=c(1,1),
                                             external.regressors = df_dummies ),
                         mean.model=list(armaOrder=c(1,0)),
                         distribution.model="norm")
i_garchFit <- ugarchfit(spec=i_garchMod, data= log_ret)
coef(i_garchFit)
```

```
##            mu           ar1         omega        alpha1         beta1
##  9.053321e-04 -6.029347e-02  4.083878e-06  3.199365e-01  6.800635e-01
##        vxreg1        vxreg2
##  5.139090e-06  5.802993e-07
```

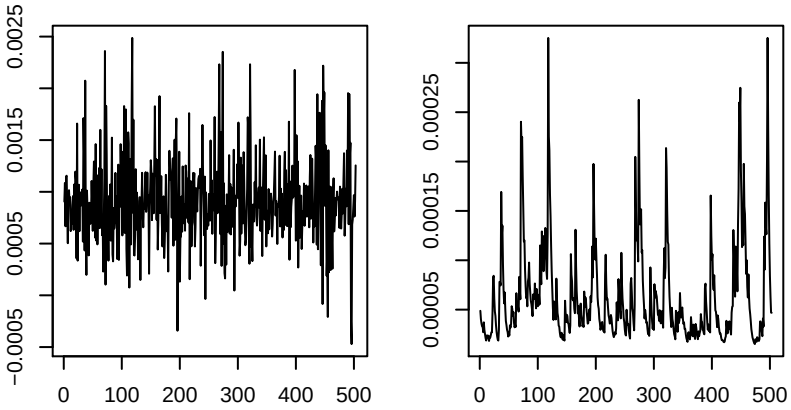
```
i_garchFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : iGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
```

```
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000905    0.000004 237.2798 0.000000
## ar1    -0.060293    0.051343  -1.1743 0.240260
## omega   0.000004    0.000001   3.1447 0.001663
## alpha1  0.319936    0.018639  17.1649 0.000000
## beta1   0.680064          NA       NA       NA
## vxreg1  0.000005    0.000000 356.8550 0.000000
## vxreg2  0.000001    0.000000 229.0615 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000905    0.000007 123.5023  0.00000
## ar1    -0.060293    0.052890  -1.1400  0.25430
## omega   0.000004    0.000003   1.5146  0.12987
## alpha1  0.319936    0.019301  16.5766  0.00000
## beta1   0.680064          NA       NA       NA
## vxreg1  0.000005    0.000000 354.0236  0.00000
## vxreg2  0.000001    0.000000  63.6517  0.00000
##
## LogLikelihood : 1800.636
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.1357
## Bayes        -7.0854
## Shibata      -7.1360
## Hannan-Quinn -7.1160
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.1921  0.6612
## Lag[2*(p+q)+(p+q)-1][2]    0.2953  0.9946
## Lag[4*(p+q)+(p+q)-1][5]    0.7580  0.9764
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.562  0.2114
## Lag[2*(p+q)+(p+q)-1][5]     2.686  0.4674
## Lag[4*(p+q)+(p+q)-1][9]     3.811  0.6210
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.08792 0.500 2.000  0.7668
## ARCH Lag[5]   1.10574 1.440 1.667  0.7018
## ARCH Lag[7]   1.41320 2.315 1.543  0.8386
```

```
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.6731
## Individual Statistics:
## mu     0.05030
## ar1    0.30104
## omega  0.05761
## alpha1 0.09532
## vxreg1 0.01264
## vxreg2 1.05998
##
## 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           2.2288 0.02627  **
## Negative Sign Bias  1.8640 0.06292   *
## Positive Sign Bias  0.8016 0.42315
## Joint Effect       11.3449 0.01000  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     49.56    0.0001519
## 2    30     61.75    0.0003721
## 3    40     75.05    0.0004587
## 4    50     78.81    0.0044131
##
##
## Elapsed time : 0.2557809
```

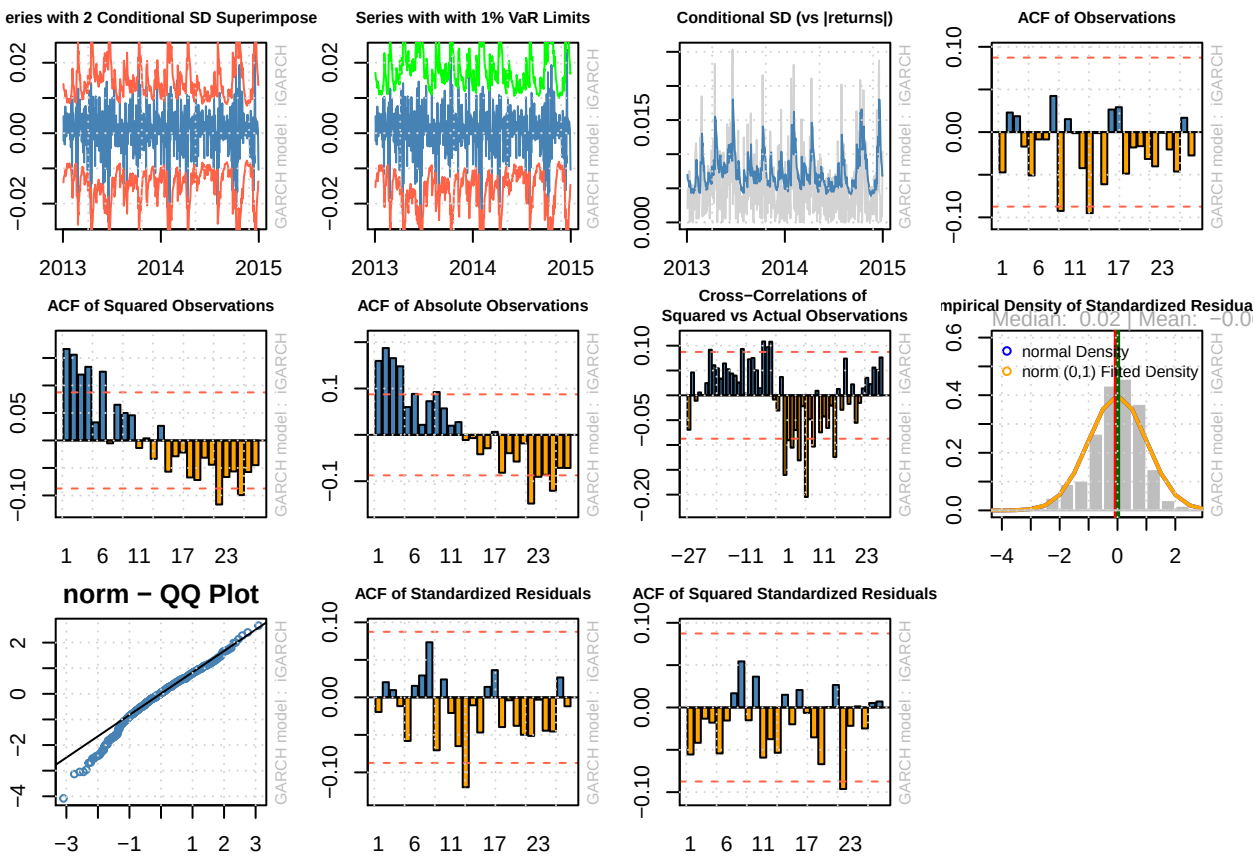
```
i_rhat <- i_garchFit@fit$fitted.values
plot.ts(i_rhat)
i_hhat <- ts(i_garchFit@fit$sigma^2)
plot.ts(i_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(i_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```

```
## Warning in FUN(x):
## plot-->: iGARCH and fiGARCH newsimpact not available
```




```
# GARCH MODELS assuming a Student's t-distribution
# Student's t-sGARCH --->  tsGARCH
library(rugarch)

ts_garchMod <- ugarchspec(mean.model = list(armaOrder = c(1, 0), include.mean = TRUE
),
variance.model = list(model = 'sGARCH',
                      garchOrder = c(1, 1),
                      external.regressors = df_dummies
),
distribution.model = "std")

ts_garchFit <- ugarchfit(spec=ts_garchMod, data=log_ret)
ts_garchFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : std
##
```

```
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001077    0.000443  2.43255 0.014993
## ar1    -0.080107    0.145467 -0.55069 0.581847
## omega   0.000007    0.000004  1.72611 0.084327
## alpha1  0.202275    0.109803  1.84215 0.065453
## beta1   0.661366    0.027788 23.80006 0.000000
## vxreg1  0.000008    0.000007  1.02688 0.304478
## vxreg2  0.000002    0.000001  1.74037 0.081794
## shape   6.142083    0.822347  7.46896 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001077    0.002446  0.44028  0.65974
## ar1    -0.080107    0.746648 -0.10729  0.91456
## omega   0.000007    0.000019  0.34617  0.72922
## alpha1  0.202275    0.553378  0.36553  0.71472
## beta1   0.661366    0.187497  3.52735  0.00042
## vxreg1  0.000008    0.000037  0.20445  0.83800
## vxreg2  0.000002    0.000007  0.34187  0.73245
## shape   6.142083    8.384014  0.73259  0.46381
##
## LogLikelihood : 1815.683
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.1876
## Bayes        -7.1205
## Shibata      -7.1881
## Hannan-Quinn -7.1613
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.01246  0.9111
## Lag[2*(p+q)+(p+q)-1][2]   0.11314  0.9999
## Lag[4*(p+q)+(p+q)-1][5]   0.61030  0.9879
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.4460  0.5042
## Lag[2*(p+q)+(p+q)-1][5]    0.7543  0.9123
## Lag[4*(p+q)+(p+q)-1][9]    1.8407  0.9231
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.0540 0.500 2.000  0.8163
```

```
## ARCH Lag[5]    0.7197 1.440 1.667  0.8175
## ARCH Lag[7]    1.1747 2.315 1.543  0.8836
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  17.8993
## Individual Statistics:
## mu     0.06043
## ar1    0.26102
## omega  1.71868
## alpha1 0.10781
## beta1  0.04842
## vxreg1 0.18054
## vxreg2 7.07380
## shape  0.04806
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.89 2.11 2.59
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias          2.06328 0.03960  **
## Negative Sign Bias 0.84033 0.40113
## Positive Sign Bias 0.03896 0.96894
## Joint Effect       7.24575 0.06446   *
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     27.30      0.09790
## 2    30     43.26      0.04305
## 3    40     46.26      0.19743
## 4    50     47.00      0.55457
##
##
## Elapsed time : 0.4289579
```

```
## Results review 1
coef(ts_garchFit)
```

```
##           mu           ar1         omega        alpha1         beta1
##  1.077123e-03 -8.010700e-02  6.700036e-06  2.022749e-01  6.613660e-01
##        vxreg1        vxreg2         shape
##  7.621360e-06  2.445218e-06  6.142083e+00
```

```
ts_rhat <- ts_garchFit@fit$fitted.values
plot.ts(ts_rhat)
ts_hhat <- ts(ts_garchFit@fit$sigma^2)
plot.ts(ts_hhat)

## Results review 2
tfit.val     <- coef(ts_garchFit)
```

```
tfit.sd      <- diag(vcov(ts_garchFit))
ttrue.val = ts_garchFit@fit$tval

tfit.conf.lb <- tfit.val + qnorm(0.025) * tfit.sd
tfit.conf.ub <- tfit.val + qnorm(0.975) * tfit.sd
print(tfit.val)
```

```
##            mu           ar1         omega        alpha1         beta1
##  1.077123e-03 -8.010700e-02  6.700036e-06  2.022749e-01  6.613660e-01
##        vxreg1        vxreg2         shape
##  7.621360e-06  2.445218e-06  6.142083e+00
```

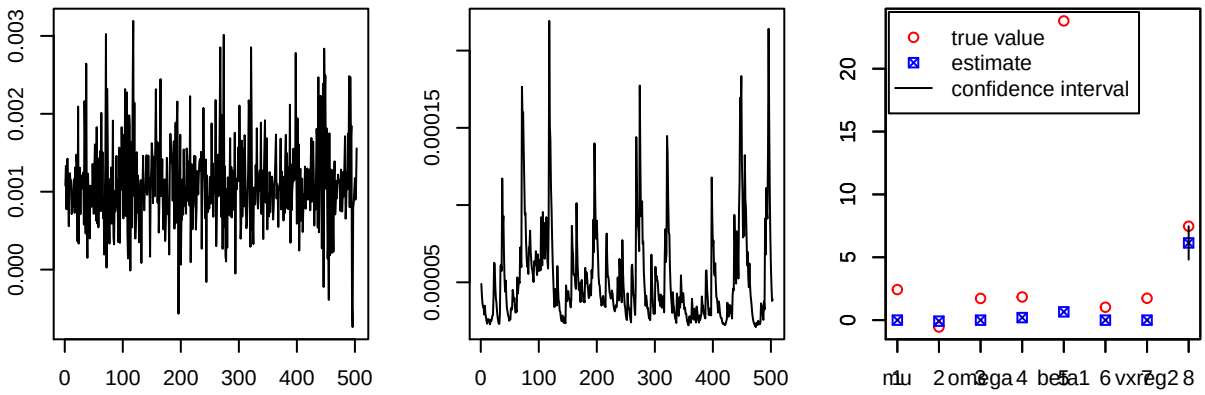
```
print(tfit.sd)
```

```
## [1] -1.960692e-07 -2.116061e-02 -1.506666e-11 -1.205680e-02 -7.721960e-04
## [6] -5.508434e-11 -1.974018e-12 -6.762552e-01
```

```
print(ttrue.val)
```

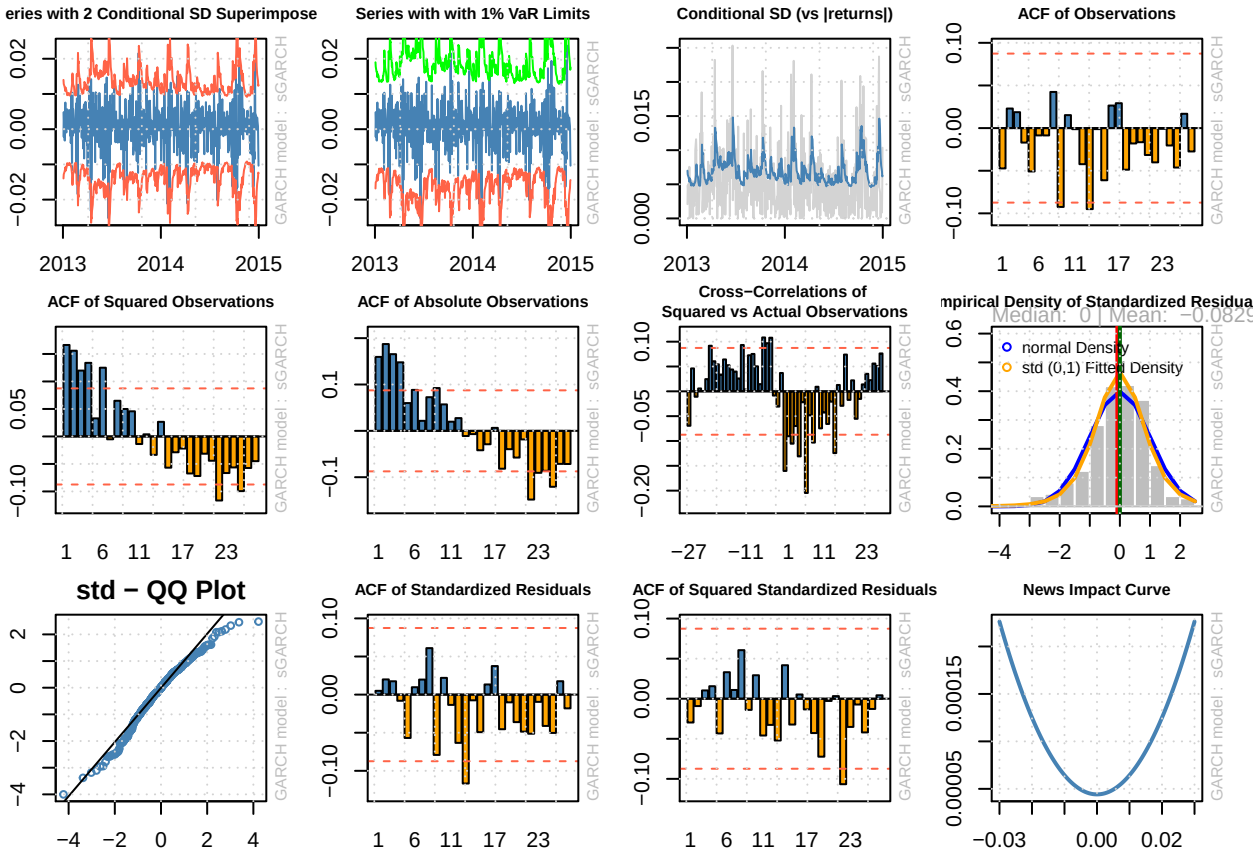
```
##        mu       ar1     omega    alpha1     beta1    vxreg1    vxreg2     shape
##  2.432545 -0.550689  1.726111  1.842154 23.800063  1.026877  1.740372  7.468964
```

```
plot(ttrue.val, pch = 1, col = "red",
     ylim = range(c(tfit.conf.lb, tfit.conf.ub, ttrue.val)),
     xlab = "", ylab = "", axes = TRUE)
box(); axis(1, at = 1:length(tfit.val), labels = names(tfit.val)); axis(2)
points(coef(ts_garchFit), col = "blue", pch = 7)
for (i in 1:length(tfit.val)) {
  lines(c(i,i), c(tfit.conf.lb[i], tfit.conf.ub[i]))
}
legend( "topleft", legend = c("true value", "estimate", "confidence interval"),
        col = c("red", "blue", 1), pch = c(1, 7, NA), lty = c(NA, NA, 1), inset = 0.01)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(ts_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#Student's t-gjrGARCH --->tgjrGARCH
tgjr_garchMod <- ugarchspec(
  variance.model=list(model="gjrGARCH",
                      garchOrder=c(1,1), external.regressors = df_dummies ),
  mean.model=list(armaOrder=c(1,0),
                  include.mean=TRUE
  ),
  distribution.model="std"
)
tgjr_garchFit <- ugarchfit(spec=tgjr_garchMod, data=log_ret)
coef(tgjr_garchFit)
```

```
##           mu           ar1         omega        alpha1         beta1
##  8.809784e-04 -6.318676e-02  4.584431e-06  2.150762e-07  7.299537e-01
##        gamma1        vxreg1        vxreg2         shape
##  3.202011e-01  6.507069e-06  2.030656e-06  6.949425e+00
```

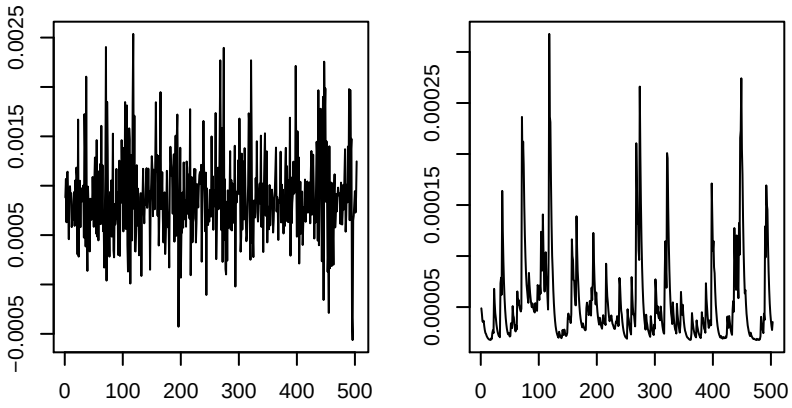
```
tgjr_garchFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : gjrGARCH(1,1)
```

```
## Mean Model   : ARFIMA(1,0,0)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000881    0.000223  3.946239 0.000079
## ar1    -0.063187    0.047453 -1.331555 0.183006
## omega   0.000005    0.000001  5.519614 0.000000
## alpha1  0.000000    0.014299  0.000015 0.999988
## beta1   0.729954    0.024061 30.337502 0.000000
## gamma1  0.320201    0.085879  3.728528 0.000193
## vxreg1  0.000007    0.000002  3.558969 0.000372
## vxreg2  0.000002    0.000000  4.294295 0.000018
## shape   6.949425    0.630171 11.027846 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000881    0.000229  3.846445 0.000120
## ar1    -0.063187    0.053955 -1.171104 0.241557
## omega   0.000005    0.000001  7.252684 0.000000
## alpha1  0.000000    0.015446  0.000014 0.999989
## beta1   0.729954    0.026885 27.151180 0.000000
## gamma1  0.320201    0.095330  3.358853 0.000783
## vxreg1  0.000007    0.000002  2.748420 0.005988
## vxreg2  0.000002    0.000001  3.449096 0.000562
## shape   6.949425    1.978854  3.511843 0.000445
##
## LogLikelihood : 1825.735
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.2236
## Bayes        -7.1481
## Shibata      -7.2242
## Hannan-Quinn -7.1940
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                  2.464e-06  0.9987
## Lag[2*(p+q)+(p+q)-1][2] 1.723e-01  0.9994
## Lag[4*(p+q)+(p+q)-1][5] 6.525e-01  0.9851
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.9704  0.3246
## Lag[2*(p+q)+(p+q)-1][5]    2.5084  0.5041
## Lag[4*(p+q)+(p+q)-1][9]    4.0526  0.5800
## d.o.f=2
```

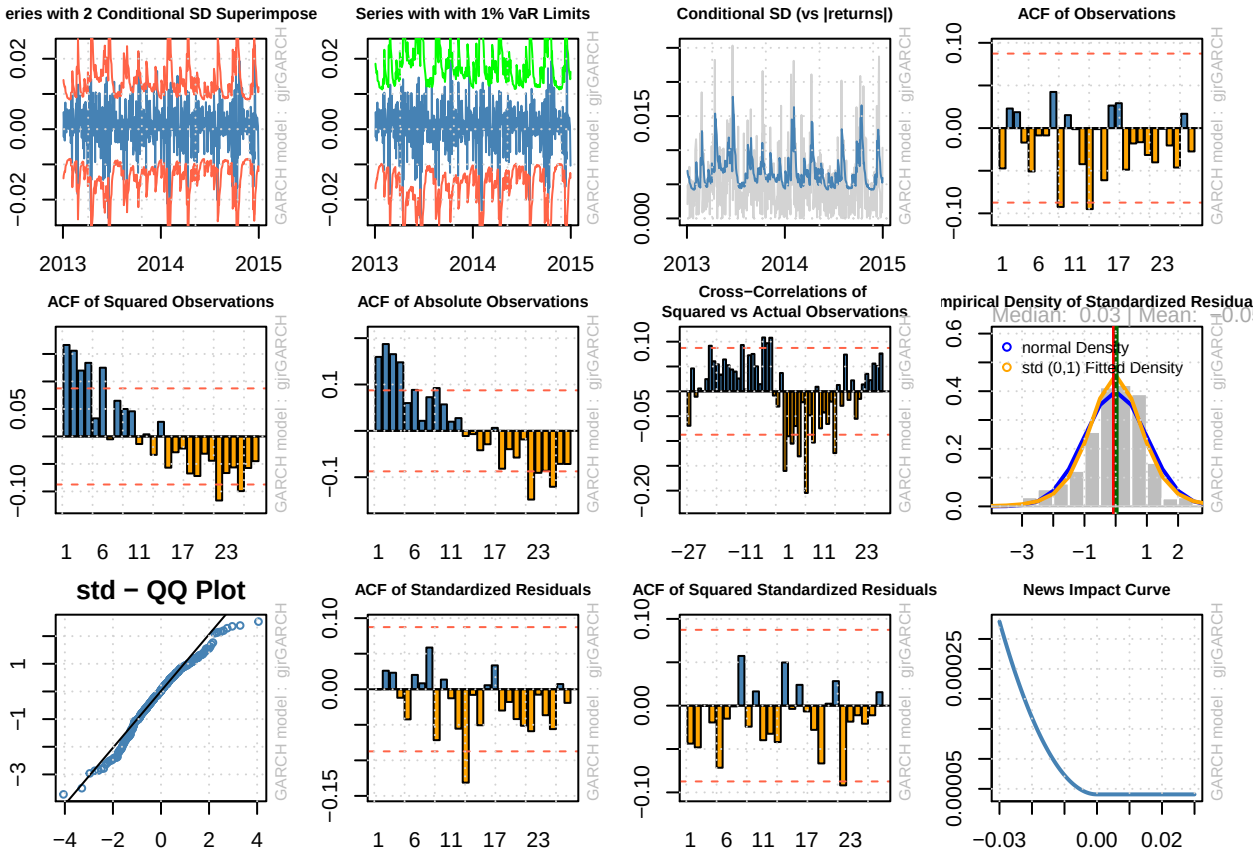
```
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3] 1.231e-05 0.500 2.000  0.9972
## ARCH Lag[5] 1.698e+00 1.440 1.667  0.5418
## ARCH Lag[7] 2.068e+00 2.315 1.543  0.7028
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  15.3759
## Individual Statistics:
## mu     0.05929
## ar1    0.28809
## omega  1.75436
## alpha1 0.39316
## beta1  0.03791
## gamma1 0.14966
## vxreg1 2.55846
## vxreg2 4.46339
## shape  0.04297
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.1 2.32 2.82
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           2.4454 0.01482  **
## Negative Sign Bias  1.8461 0.06548   *
## Positive Sign Bias  0.4741 0.63562
## Joint Effect        7.3585 0.06131   *
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     38.43     0.005228
## 2    30     37.06     0.144705
## 3    40     55.49     0.042037
## 4    50     65.49     0.057705
##
##
## Elapsed time : 0.7116909
```

```r
tgjr_rhat <- tgjr_garchFit@fit$fitted.values
plot.ts(tgjr_rhat)
tgjr_hhat <- ts(tgjr_garchFit@fit$sigma^2)
plot.ts(tgjr_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(tgjr_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#Student's t-eGARCH --->teGARCH
te_garchMod <- ugarchspec(
  variance.model=list(model="eGARCH",
                      garchOrder=c(1,1), external.regressors = df_dummies ),
  mean.model=list(armaOrder=c(1,0),
                  include.mean=TRUE
  ),
  distribution.model="std"
)
te_garchFit <- ugarchfit(spec=te_garchMod, data=log_ret)
coef(te_garchFit)
```

```
##           mu          ar1        omega       alpha1        beta1       gamma1
##  0.002012785 -0.047107533 -0.696736218 -0.300660521  0.900010871  0.601751546
##       vxreg1       vxreg2        shape
## -2.529340859 -9.172391966  3.999483351
```

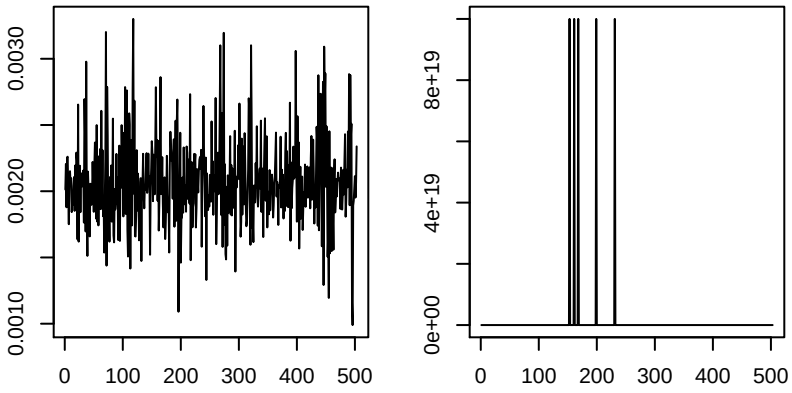
```
te_garchFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : eGARCH(1,1)
```

```
## Mean Model   : ARFIMA(1,0,0)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.002013    0.000002  1263.510        0
## ar1    -0.047108    0.000033 -1411.030        0
## omega  -0.696736    0.000615 -1133.607        0
## alpha1 -0.300661    0.000183 -1645.610        0
## beta1   0.900011    0.000299  3012.762        0
## gamma1  0.601752    0.000362  1660.753        0
## vxreg1 -2.529341    0.031686   -79.826        0
## vxreg2 -9.172392    0.008222 -1115.577        0
## shape   3.999483    0.002139  1869.912        0
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.002013    0.009895  0.20342  0.83880
## ar1    -0.047108    0.164329 -0.28667  0.77437
## omega  -0.696736    3.226641 -0.21593  0.82904
## alpha1 -0.300661    0.536478 -0.56043  0.57518
## beta1   0.900011    1.286992  0.69931  0.48436
## gamma1  0.601752    3.117495  0.19302  0.84694
## vxreg1 -2.529341   20.237763 -0.12498  0.90054
## vxreg2 -9.172392   69.834011 -0.13135  0.89550
## shape   3.999483    9.835073  0.40665  0.68426
##
## LogLikelihood : 779.7217
##
## Information Criteria
## ------------------------------------
##
## Akaike       -3.0645
## Bayes        -2.9890
## Shibata      -3.0651
## Hannan-Quinn -3.0349
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.03718  0.8471
## Lag[2*(p+q)+(p+q)-1][2]   0.09626  0.9999
## Lag[4*(p+q)+(p+q)-1][5]   0.53917  0.9919
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.02627  0.8712
## Lag[2*(p+q)+(p+q)-1][5]   0.06793  0.9991
## Lag[4*(p+q)+(p+q)-1][9]   0.21730  0.9999
## d.o.f=2
```

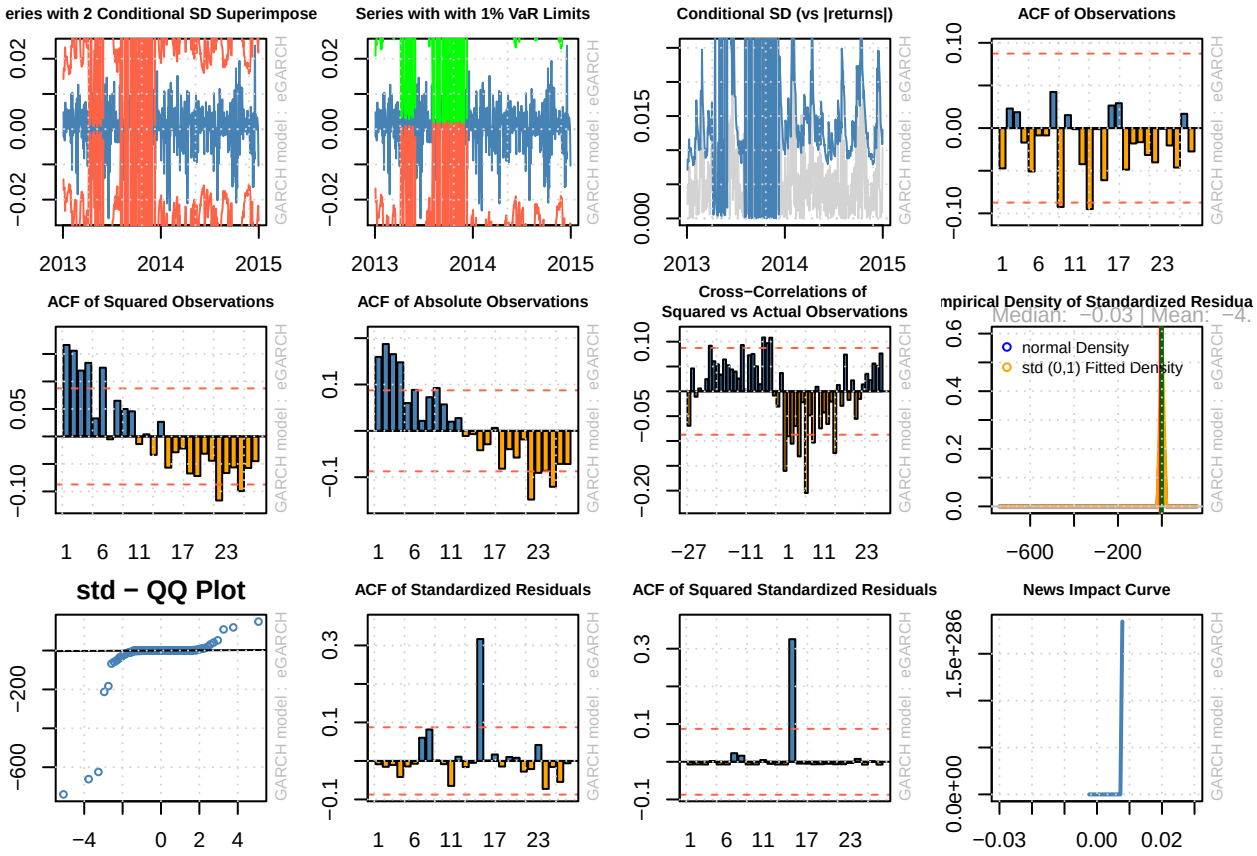
```
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.02628 0.500 2.000  0.8712
## ARCH Lag[5]   0.04269 1.440 1.667  0.9961
## ARCH Lag[7]   0.17215 2.315 1.543  0.9979
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  6.333
## Individual Statistics:
## mu     0.02455
## ar1    0.90921
## omega  0.41452
## alpha1 0.18684
## beta1  0.76394
## gamma1 0.01304
## vxreg1 0.66686
## vxreg2 0.29431
## shape  0.21633
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.1 2.32 2.82
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias            1.028 0.3043
## Negative Sign Bias   0.991 0.3222
## Positive Sign Bias   1.643 0.1011
## Joint Effect         4.291 0.2317
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     197.6    1.042e-31
## 2    30     229.2    5.178e-33
## 3    40     280.2    3.011e-38
## 4    50     304.7    1.295e-38
##
##
## Elapsed time : 0.4163711
```

```
te_rhat <- te_garchFit@fit$fitted.values
plot.ts(te_rhat)
te_hhat <- ts(te_garchFit@fit$sigma^2)
plot.ts(te_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(te_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#Student's t-apARCH ---> tapARCH
tapARCHMod <- ugarchspec(variance.model=list(model="apARCH",
                                             garchOrder=c(1,1),
                                             external.regressors = df_dummies ),
                         mean.model=list(armaOrder=c(1,0)),
                         distribution.model="std")
tapARCHFit <- ugarchfit(spec=tapARCHMod, data= log_ret)
coef(tapARCHFit)
```

```
##            mu           ar1         omega        alpha1         beta1
##  9.068272e-04 -4.473157e-02  2.271811e-08  7.601009e-02  6.835558e-01
##        gamma1         delta        vxreg1        vxreg2         shape
##  4.936238e-01  3.032251e+00  2.758024e-08  1.940875e-08  6.800218e+00
```

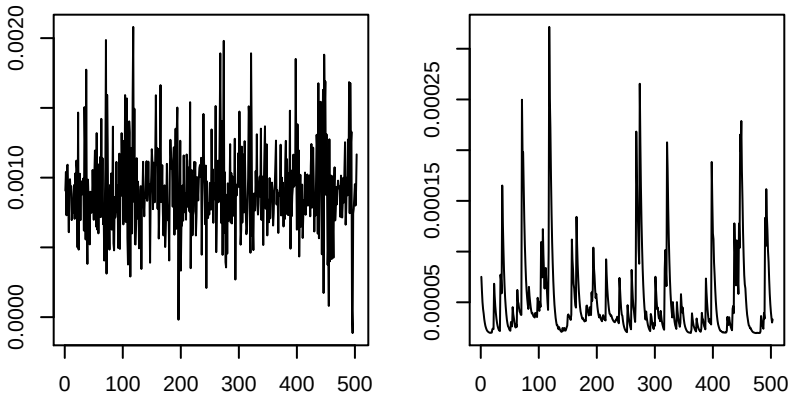
```
tapARCHFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : apARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : std
##
```

```
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000907    0.000256  3.539729 0.000401
## ar1    -0.044732    0.049692 -0.900177 0.368026
## omega   0.000000    0.000001  0.022447 0.982091
## alpha1  0.076010    0.034534  2.201011 0.027735
## beta1   0.683556    0.106456  6.420991 0.000000
## gamma1  0.493624    0.207634  2.377380 0.017436
## delta   3.032251    0.092542 32.766234 0.000000
## vxreg1  0.000000    0.000001  0.024252 0.980652
## vxreg2  0.000000    0.000001  0.025265 0.979844
## shape   6.800218    2.028400  3.352503 0.000801
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000907    0.003973  0.228225  0.81947
## ar1    -0.044732    1.822830 -0.024540  0.98042
## omega   0.000000    0.000196  0.000116  0.99991
## alpha1  0.076010    1.163708  0.065317  0.94792
## beta1   0.683556   14.759060  0.046314  0.96306
## gamma1  0.493624   11.219947  0.043995  0.96491
## delta   3.032251    8.354568  0.362945  0.71665
## vxreg1  0.000000    0.000164  0.000168  0.99987
## vxreg2  0.000000    0.000079  0.000246  0.99980
## shape   6.800218   73.106907  0.093017  0.92589
##
## LogLikelihood : 1820.434
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.1985
## Bayes        -7.1146
## Shibata      -7.1993
## Hannan-Quinn -7.1656
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.2135  0.6440
## Lag[2*(p+q)+(p+q)-1][2]    0.3887  0.9851
## Lag[4*(p+q)+(p+q)-1][5]    0.8706  0.9643
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      0.856  0.3549
## Lag[2*(p+q)+(p+q)-1][5]     2.230  0.5652
## Lag[4*(p+q)+(p+q)-1][9]     3.567  0.6628
## d.o.f=2
##
```

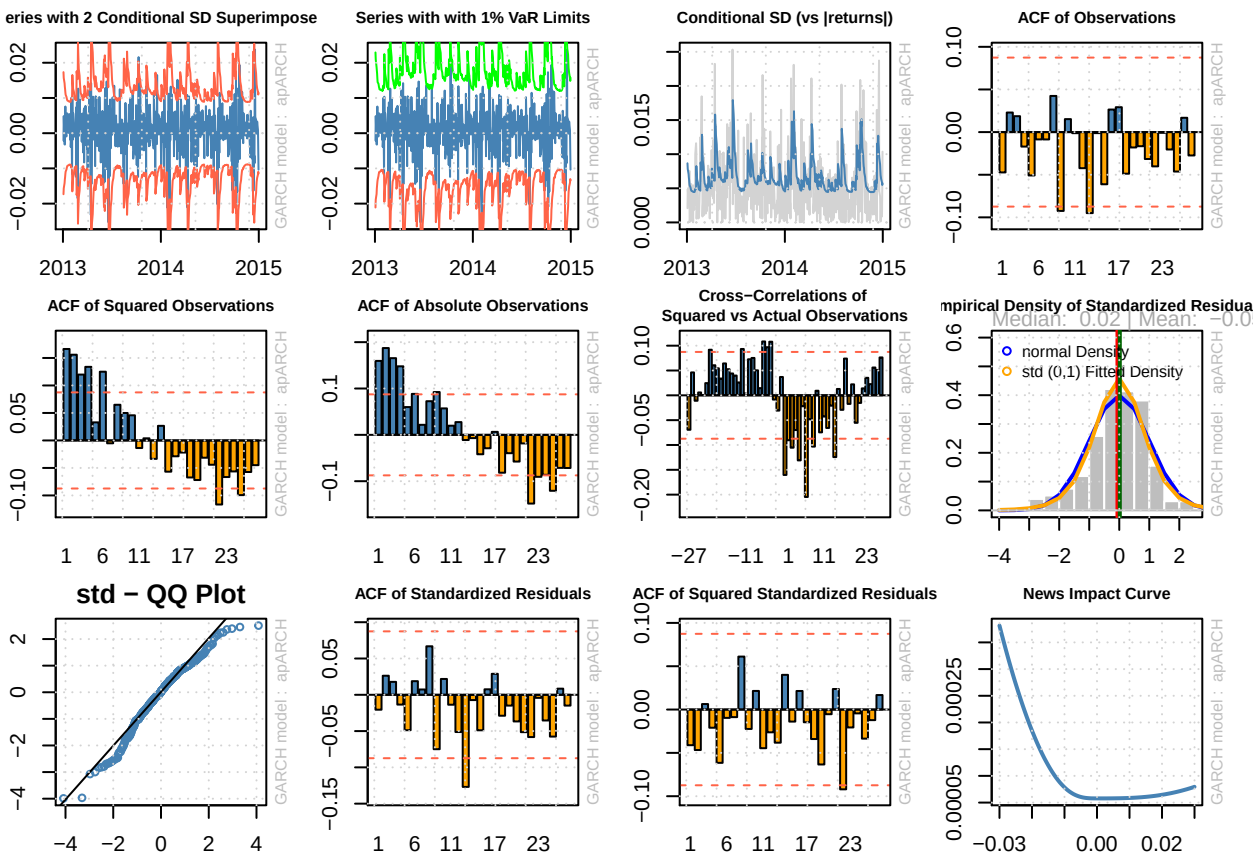
```
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.02027 0.500 2.000  0.8868
## ARCH Lag[5]   1.33998 1.440 1.667  0.6354
## ARCH Lag[7]   1.61356 2.315 1.543  0.7983
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  NaN
## Individual Statistics:
## mu      0.06654
## ar1     0.59291
## omega  59.72101
## alpha1  0.06882
## beta1   0.03319
## gamma1  0.04743
## delta   0.32302
## vxreg1      NaN
## vxreg2      NaN
## shape   0.05481
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.29 2.54 3.05
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           2.5577 0.01083  **
## Negative Sign Bias  1.7100 0.08789   *
## Positive Sign Bias  0.5431 0.58727
## Joint Effect        7.8437 0.04936  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     39.86     0.003411
## 2    30     40.88     0.070590
## 3    40     64.55     0.006188
## 4    50     78.41     0.004809
##
##
## Elapsed time : 0.6622701
```

```
tapARCH_rhat <- tapARCHFit@fit$fitted.values
plot.ts(tapARCH_rhat)
tapARCH_hhat <- ts(tapARCHFit@fit$sigma^2)
plot.ts(tapARCH_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(tapARCHFit,which="all")
```

```
##
## please wait...calculating quantiles...
```




```
#Student's t-tiGARCH ---> tiGARCH
ti_garchMod <- ugarchspec(variance.model=list(model="iGARCH",
                                              garchOrder=c(1,1),
                                              external.regressors = df_dummies ),
                          mean.model=list(armaOrder=c(1,0), include.mean=TRUE),
                          distribution.model="std")
ti_garchFit <- ugarchfit(spec=ti_garchMod, data= log_ret)
coef(ti_garchFit)
```

```
##            mu           ar1         omega        alpha1         beta1
##  1.118378e-03 -8.467020e-02  4.767178e-06  3.118613e-01  6.881387e-01
##         vxreg1        vxreg2         shape
##  7.194076e-06  2.618257e-06  4.306351e+00
```

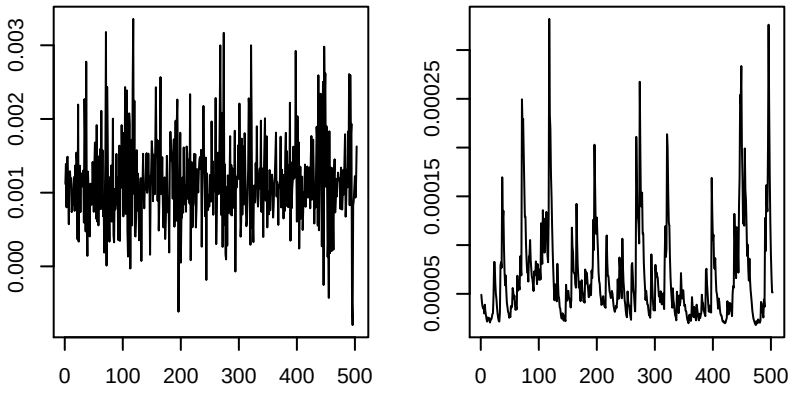
```
ti_garchFit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : iGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : std
##
```

```
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001118    0.000068  16.3305 0.000000
## ar1    -0.084670    0.035998  -2.3521 0.018668
## omega   0.000005    0.000002   2.0359 0.041757
## alpha1  0.311861    0.041158   7.5773 0.000000
## beta1   0.688139          NA       NA       NA
## vxreg1  0.000007    0.000001   9.4146 0.000000
## vxreg2  0.000003    0.000000   6.6928 0.000000
## shape   4.306351    0.813634   5.2927 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001118    0.000036  30.9085 0.000000
## ar1    -0.084670    0.036050  -2.3487 0.018841
## omega   0.000005    0.000003   1.4843 0.137735
## alpha1  0.311861    0.036843   8.4646 0.000000
## beta1   0.688139          NA       NA       NA
## vxreg1  0.000007    0.000001   8.3143 0.000000
## vxreg2  0.000003    0.000000   6.2199 0.000000
## shape   4.306351    0.730499   5.8951 0.000000
##
## LogLikelihood : 1812.892
##
## Information Criteria
## ------------------------------------
##
## Akaike       -7.1805
## Bayes        -7.1218
## Shibata      -7.1809
## Hannan-Quinn -7.1574
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                  0.0001489  0.9903
## Lag[2*(p+q)+(p+q)-1][2] 0.0988805  0.9999
## Lag[4*(p+q)+(p+q)-1][5] 0.5856371  0.9894
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.149  0.2838
## Lag[2*(p+q)+(p+q)-1][5]     1.891  0.6444
## Lag[4*(p+q)+(p+q)-1][9]     3.009  0.7574
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.03201 0.500 2.000  0.8580
```

```
## ARCH Lag[5]   0.99270 1.440 1.667  0.7351
## ARCH Lag[7]   1.30432 2.315 1.543  0.8596
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  5.5534
## Individual Statistics:
## mu     0.05802
## ar1    0.28466
## omega  0.03804
## alpha1 0.06886
## vxreg1 0.13788
## vxreg2 4.59620
## shape  0.03198
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.69 1.9 2.35
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           2.0547 0.04043  **
## Negative Sign Bias  1.5447 0.12306
## Positive Sign Bias  0.6405 0.52217
## Joint Effect        9.2030 0.02671  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     33.82      0.01930
## 2    30     40.64      0.07403
## 3    40     59.62      0.01832
## 4    50     51.37      0.38088
##
##
## Elapsed time : 0.3300061
```

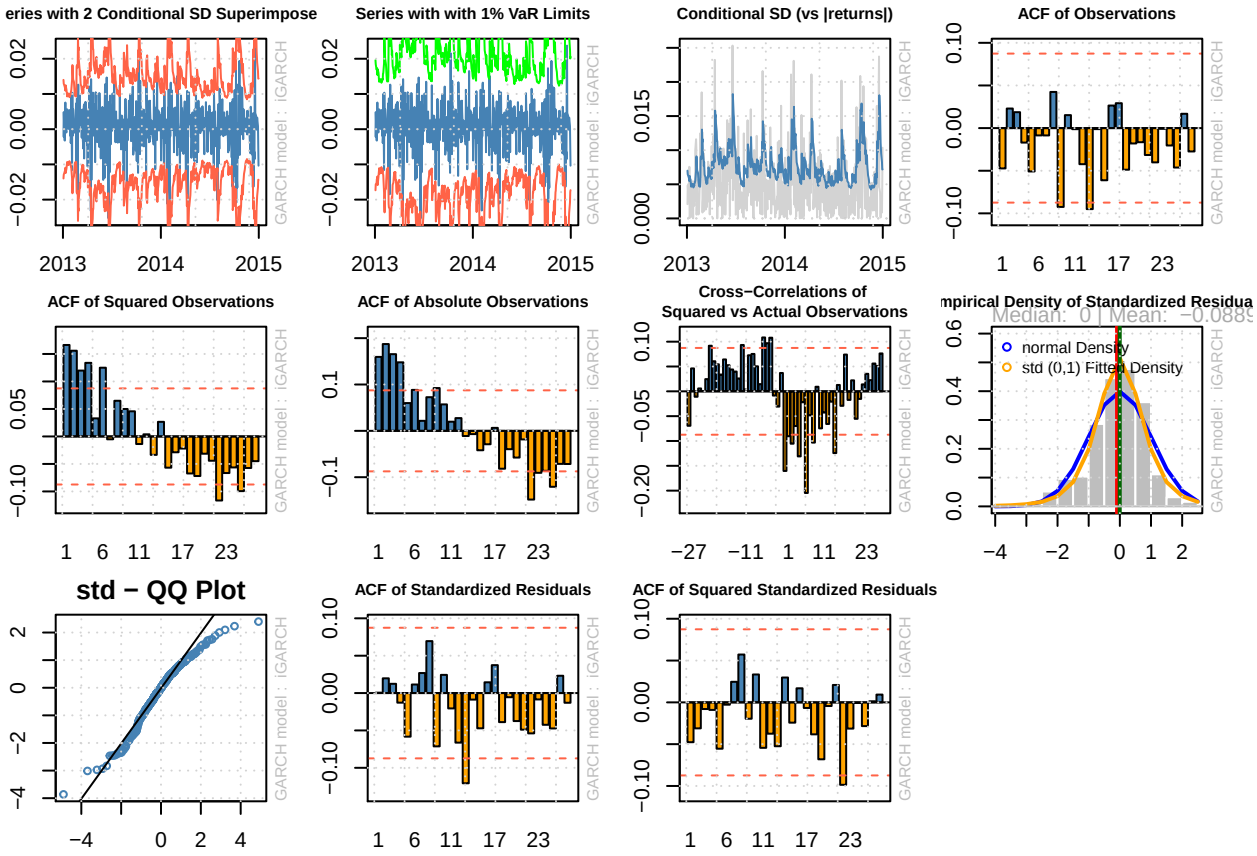
```r
ti_rhat <- ti_garchFit@fit$fitted.values
plot.ts(ti_rhat)
ti_hhat <- ts(ti_garchFit@fit$sigma^2)
plot.ts(ti_hhat)

par(mfrow=c(2, 3))
```




```
par(mar = c(2, 2, 2, 2))
plot(ti_garchFit,which="all")
```

```
##
## please wait...calculating quantiles...
```

```
## Warning in FUN(x):
## plot-->: iGARCH and fiGARCH newsimpact not available
```



Choosing the best model:

```
#Choose best GARCH model
#Normal
AIC= c(infocriteria(s_garchFit)[1], infocriteria(gjr_garchFit)[1],
       infocriteria(e_garchFit)[1], infocriteria(apARCHFit)[1],  infocriteria(i_garchFit)[1] )
rowSAIC= c("s_garch", "gjr_garchFit", "e_garchFit", "apARCHFit", "i_garchFit")


AIC_Results= data.frame(AIC, row.names = rowSAIC)
AIC_Results
```

```
##                     AIC
## s_garch      -7.158517
## gjr_garchFit -7.203430
## e_garchFit   -7.252679
## apARCHFit    -7.175112
## i_garchFit   -7.135728
```

```
BestAICNorm=AIC_Results[which.min(AIC_Results$AIC),]
BestAICNorm
```

```
## [1] -7.252679
```

```
BIC= c(infocriteria(s_garchFit)[2], infocriteria(gjr_garchFit)[2],
       infocriteria(e_garchFit)[2], infocriteria(apARCHFit)[2],  infocriteria(i_garchFit)[2] )
rowSBIC= c("s_garch", "gjr_garchFit", "e_garchFit", "apARCHFit", "i_garchFit")
BIC_Results= data.frame(BIC, row.names = rowSBIC)
```

```
BIC_Results
```

```
##                    BIC
## s_garch      -7.099782
## gjr_garchFit -7.136303
## e_garchFit   -7.185552
## apARCHFit    -7.099595
## i_garchFit   -7.085383
```

```
BestBICNorm=BIC_Results[which.min(BIC_Results$BIC),]
BestBICNorm
```

```
## [1] -7.185552
```

```
Best_Norm=c(BestAICNorm,BestBICNorm)
rowSBest_Norm= c("AICbest","BICbest")
Norm_Results= data.frame(Best_Norm, row.names =rowSBest_Norm)
Norm_Results
```

```
##         Best_Norm
## AICbest -7.252679
## BICbest -7.185552
```

```
#Student-t
tAIC= c(infocriteria(ts_garchFit)[1], infocriteria(tgjr_garchFit)[1],
        infocriteria(te_garchFit)[1], infocriteria(tapARCHFit)[1],  infocriteria(ti_garchFit)[1] )
trowSAIC= c("ts_garch", "tgjr_garchFit", "te_garchFit", "tapARCHFit", "ti_garchFit")


tAIC_Results= data.frame(tAIC, row.names = trowSAIC)
tAIC_Results
```

```
##                    tAIC
## ts_garch      -7.187608
## tgjr_garchFit -7.223597
## te_garchFit   -3.064500
## tapARCHFit    -7.198545
## ti_garchFit   -7.180487
```

```
tAIC_Results[which.min(tAIC_Results$tAIC),]
```

```
## [1] -7.223597
```

```
tBIC= c(infocriteria(ts_garchFit)[2], infocriteria(tgjr_garchFit)[2],
        infocriteria(te_garchFit)[2], infocriteria(tapARCHFit)[2],  infocriteria(ti_garchFit)[2]  )
trowSBIC= c("ts_garch", "tgjr_garchFit", "te_garchFit", "tapARCHFit", "ti_garchFit")
tBIC_Results= data.frame(tBIC, row.names = trowSBIC)
tBIC_Results
```

```
##                    tBIC
## ts_garch      -7.120481
## tgjr_garchFit -7.148079
## te_garchFit   -2.988982
## tapARCHFit    -7.114637
## ti_garchFit   -7.121751
```

```
tBIC_Results[which.min(tBIC_Results$tBIC),]
```

```
## [1] -7.148079
```

```
#Goodness of Fit
Distribution=c("Normal", "Normal", "Normal", "Normal", "Normal",
               "Student's-t", "Student's-t", "Student's-t", "Student's-t", "Student's-t" )
rowsGARCH=c("s_garchFit", "gjr_garchFit","e_garchFit","apARCHFit", "i_garchFit",
            "ts_garchFit", "tgjr_garchFit", "te_garchFit", "tapARCHFit", "ti_garchFit")
Best=c("o", "o", "*", "o", "o",
       "o", "*", "o", "o","o")
final_AIC=c(AIC,tAIC)
final_BIC=c(BIC,tBIC)
Results <- data.frame("AIC" = final_AIC, "BIC" = final_BIC , "Distribution" = Distribution,
                      "Best"= Best ,row.names = rowsGARCH)
Results
```

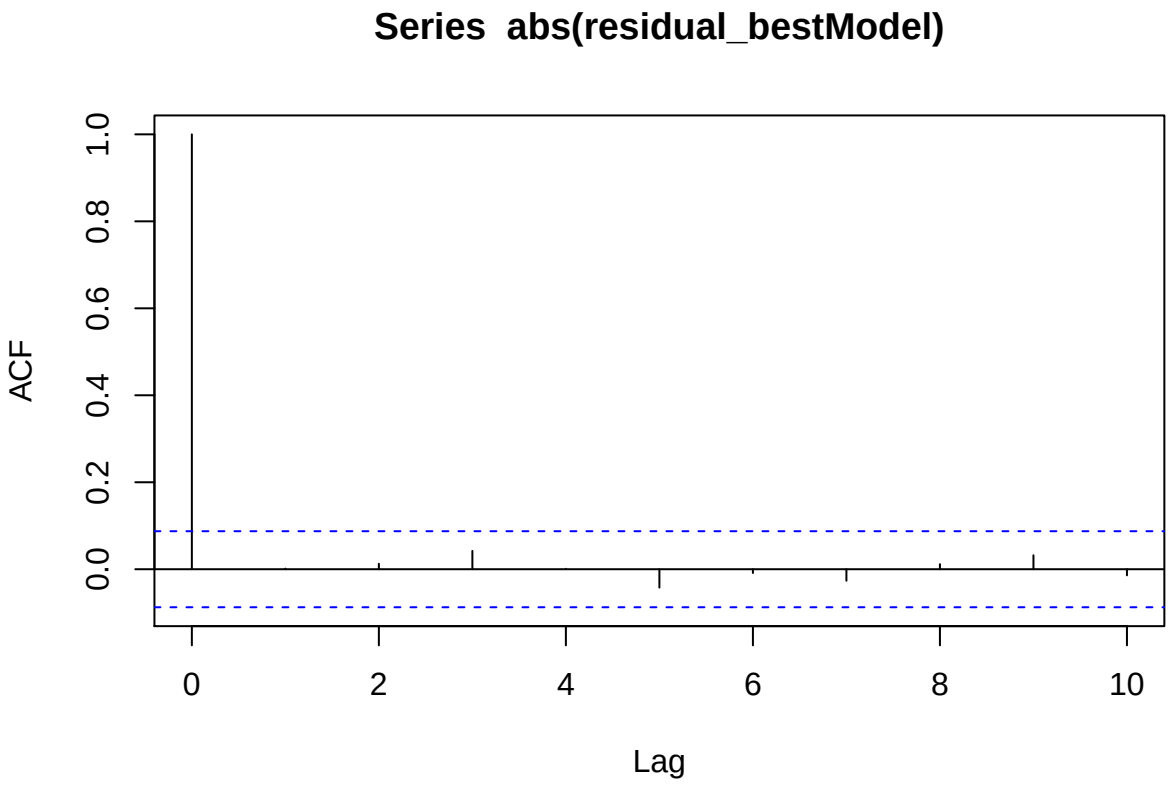
```
##                       AIC       BIC Distribution Best
## s_garchFit    -7.158517 -7.099782       Normal    o
## gjr_garchFit  -7.203430 -7.136303       Normal    o
## e_garchFit    -7.252679 -7.185552       Normal    *
## apARCHFit     -7.175112 -7.099595       Normal    o
## i_garchFit    -7.135728 -7.085383       Normal    o
## ts_garchFit   -7.187608 -7.120481  Student's-t    o
## tgjr_garchFit -7.223597 -7.148079  Student's-t    *
## te_garchFit   -3.064500 -2.988982  Student's-t    o
## tapARCHFit    -7.198545 -7.114637  Student's-t    o
## ti_garchFit   -7.180487 -7.121751  Student's-t    o
```

Validation:

```
#Validation of the model
residual_bestModel= residuals(e_garchFit, standardize = TRUE)
Box.test(abs(residual_bestModel), 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  abs(residual_bestModel)
## X-squared = 2.9893, df = 10, p-value = 0.9817
```

```
acf(abs(residual_bestModel), 10)
```


Series abs(residual_bestModel)
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
Lag