Volatility Prediction

Luca Salvador

2023-05-05

Table of Contents

1. Brief introduction to the loaded financial instruments

VT

Vanguard Total World Stock ETF (VT) aims to track the performance of an index that measures the investment returns of companies located in developed and emerging markets worldwide.

The Fund employs an indexed investment approach aimed at tracking the performance of the FTSE Global All Cap Index, a market-capitalization-weighted index adjusted for free float, designed to measure the market performance of large, medium, and small capitalization companies located around the world.

The Fund seeks to sample the target index by investing all, or substantially all, of its assets in common stocks of the Index and holding a representative sample of securities that resembles the entire Index in terms of key risk factors and other characteristics. These factors include sector weightings, country weightings, market capitalization, and other financial characteristics of the securities.

EEM

The iShares MSCI Emerging Markets ETF aims to track the investment results of an index composed of large and medium capitalization emerging markets.

The Fund aims to track the results of the MSCI Emerging Markets Index (the “Underlying Index”), which is designed to measure the stock market performance of the global emerging markets. As of August 31, 2022, the Underlying Index was composed of securities from the following 24 emerging market countries: Brazil, Chile, China, Colombia, Czech Republic, Egypt, Greece, India, Indonesia, Hungary, Kuwait, Malaysia, Mexico, Peru, Philippines, Poland, Qatar, Saudi Arabia, South Africa, South Korea, Taiwan, Thailand, Turkey, and the United Arab Emirates.

The Underlying Index includes large and medium capitalization companies and may change over time. As of August 31, 2022, a significant portion of the Benchmark Index is represented by securities of companies in the financial and technology sectors. The components of the Underlying Index may change over time. BlackRock Fund Advisors uses a “passive” or indexing approach to try to achieve the Fund’s investment objective. Unlike many investment companies, the Fund does not seek to “beat” the index it follows and does not seek temporary defensive positions when markets appear overvalued. Indexing may eliminate the possibility that the Fund substantially outperforms the Underlying Index, but can also reduce some of the risks of active management, such as poor stock selection.

2. Data loading

```
getSymbols('VT', from='2008-06-26', to='2023-03-31',warnings=FALSE,
           auto.assign=TRUE,  periodicity='daily')
```

```
## [1] "VT"
```

```
VTd = `VT` $ `VT.Adjusted`
VTd = na.omit(VTd)

getSymbols('EEM', from='2003-04-14', to='2023-03-31',warnings=FALSE,
           auto.assign=TRUE,  periodicity='daily')
```

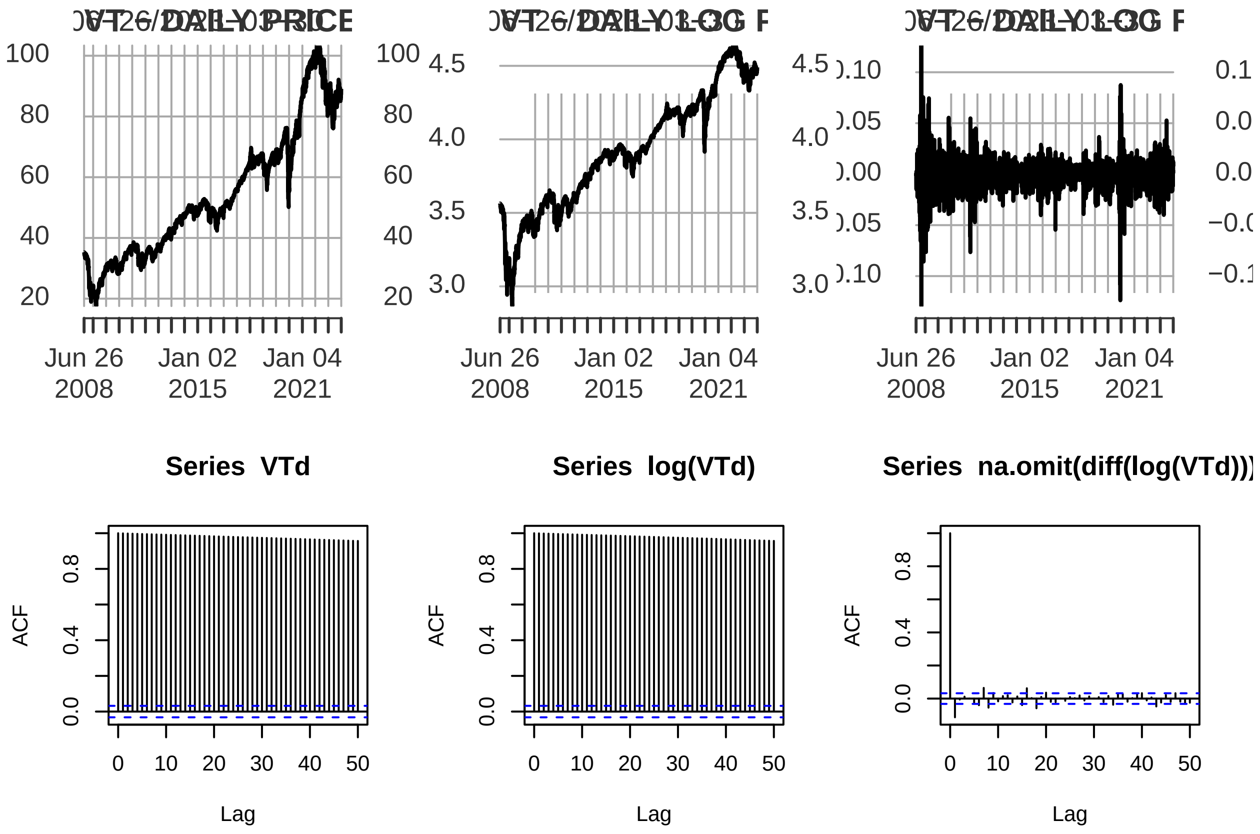
```
## [1] "EEM"
```

```
EEMd = `EEM` $ `EEM.Adjusted`
EEMd = na.omit(EEMd)
```

3. VT

3.1 "Graphically represent the returns and briefly describe the stylized facts that characterize them, highlighting any differences between the financial instruments considered. Use the statistical tools you deem most appropriate for these analyses, necessarily including the verification of the possible presence of heteroscedasticity."

```r
par(mfrow=c(2,3))
plot(VTd, main='VT - DAILY PRICES')
plot(log(VTd), main='VT - DAILY LOG PRICES')
plot(diff(log(VTd)), main='VT - DAILY LOG RETURN')
acf(VTd, lag=50)
acf(log(VTd), lag=50)
acf(na.omit(diff(log(VTd))), lag=50)
```



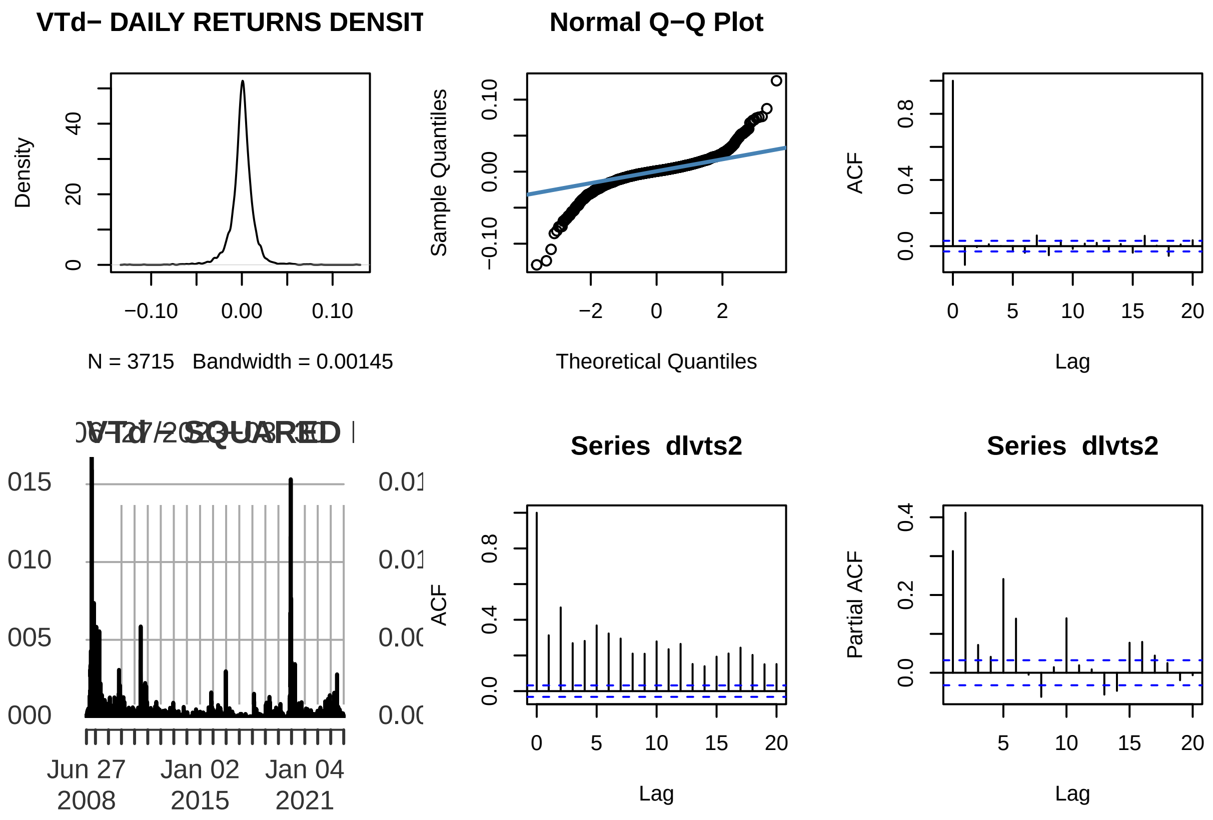
In the figure just before this, graphical analyses of the series are provided (from left to right: daily prices, daily log-prices, and log-returns) along with their corresponding correlograms. A strong bullish trend in the VT ETF can be identified within the time interval considered. One can also notice a volatility clustering behavior (more clearly observable in the series of log-returns), where periods of high volatility are followed by periods of low volatility. Regarding the global autocorrelations (ACF), as per theory, the hypotheses of a clear evidence of non-stationarity of the stochastic process are confirmed; while, in log-returns, although there are some statistically significant values (due to the large series size), these can be considered almost irrelevant from an economic perspective. In doing so, an uncorrelation of data that is at least economically non-significant is verified; however, this is not confirmed by the Ljung-Box test. The Ljung-Box test indeed checks for data uncorrelation, and with a low p-value, we reject the null hypothesis of uncorrelation and conclude that at least statistically, there is a form of serial dependence in the log returns.

```
dlvts = na.omit(diff(log(VTd)))
outB = Box.test(dlvts,1:20)
round(outB$p.value, 5)
```

```
##  [1] 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
## [10] 0.00000 0.00000 0.00000 0.00001 0.00002 0.00003 0.00005 0.00010 0.00017
## [19] 0.00029 0.00048
```

Analysis of deviation from normality

```
par(mfrow=c(2,3))
dlvts = na.omit(diff(log(VTd)))
plot(density(dlvts), main='VTd- DAILY RETURNS DENSITY')
qqnorm(dlvts,main="Normal Q-Q Plot",xlab="Theoretical Quantiles",ylab="Sample Quantiles")
qqline(dlvts,col="steelblue",lwd=2)
acf(dlvts,20,main='')
dlvts2=(dlvts)^2
plot(dlvts2, main='VTd - SQUARED LOG.RET')
acf(dlvts2, 20)
pacf(dlvts2, 20)
```



In this case, the figure represents some typical characteristics of a financial instrument that should be analyzed individually: the empirical hypothesis of non-normality of the historical series is at least graphically confirmed, particularly due to strong kurtosis visible in the top left chart, which shows very thick tails, and deviation from normality on the tails also visible in the qq-plot. The skewness, which is another empirical characteristic, for an ETF like VT that contains many companies inside, seems to approach nullity. In the second group of figures, at the bottom, a proxy for the variance (i.e., the series of squared returns) is analyzed: it is also observable in this case that the levels of variability (on which subsequently models that can estimate it will be built) are not always constant and suggest rather a sort of serial dependence, confirmed quite clearly by the ACF of the squared returns.

Testing for the presence of unit roots

```
out1 = adfTest(log(VTd), lag=21, type='c')
summary(out1@test$lm)
```

```
##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.118379 -0.005229  0.000682  0.006119  0.113117
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0016745  0.0022186   0.755 0.450443
## y.lag.1      -0.0003515  0.0005635  -0.624 0.532739
## y.diff.lag1  -0.1071502  0.0165044  -6.492 9.59e-11 ***
## y.diff.lag2  -0.0138061  0.0165903  -0.832 0.405361
## y.diff.lag3   0.0086503  0.0165875   0.521 0.602054
## y.diff.lag4  -0.0029267  0.0165592  -0.177 0.859719
## y.diff.lag5  -0.0297533  0.0165574  -1.797 0.072421 .
## y.diff.lag6  -0.0354921  0.0165427  -2.145 0.031979 *
## y.diff.lag7   0.0502455  0.0165487   3.036 0.002412 **
## y.diff.lag8  -0.0385029  0.0165687  -2.324 0.020189 *
## y.diff.lag9   0.0212807  0.0165742   1.284 0.199235
## y.diff.lag10 -0.0123989  0.0165735  -0.748 0.454440
## y.diff.lag11  0.0181168  0.0165739   1.093 0.274427
## y.diff.lag12  0.0231718  0.0165773   1.398 0.162255
## y.diff.lag13 -0.0212740  0.0165791  -1.283 0.199510
## y.diff.lag14  0.0028789  0.0165688   0.174 0.862067
## y.diff.lag15 -0.0255621  0.0165517  -1.544 0.122582
## y.diff.lag16  0.0518331  0.0165497   3.132 0.001750 **
## y.diff.lag17  0.0142780  0.0165641   0.862 0.388755
## y.diff.lag18 -0.0582209  0.0165662  -3.514 0.000446 ***
## y.diff.lag19 -0.0010489  0.0165923  -0.063 0.949597
## y.diff.lag20  0.0364474  0.0165921   2.197 0.028107 *
## y.diff.lag21 -0.0134848  0.0165084  -0.817 0.414071
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01354 on 3671 degrees of freedom
## Multiple R-squared:  0.0314, Adjusted R-squared:  0.0256
## F-statistic:  5.41 on 22 and 3671 DF,  p-value: 5.066e-15
```

By removing one lag at a time and assessing the significance of the coefficient associated with the last delay in the linear regression model constructed during the Augmented Dickey-Fuller (ADF) test for the stationarity of a time series, it was found that the appropriate number of lags to use is 20, and the intercept is not significant.

```
out1 = adfTest(log(VTd), lag=20, type='nc')
summary(out1@test$lm)
```

```
##
## Call:
```

```
## lm(formula = y.diff ~ y.lag.1 - 1 + y.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.118314 -0.005194  0.000650  0.006143  0.113246
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## y.lag.1       6.944e-05  5.682e-05   1.222 0.221698
## y.diff.lag1  -1.080e-01  1.649e-02  -6.549 6.61e-11 ***
## y.diff.lag2  -1.357e-02  1.658e-02  -0.818 0.413211
## y.diff.lag3   9.057e-03  1.655e-02   0.547 0.584328
## y.diff.lag4  -3.503e-03  1.655e-02  -0.212 0.832436
## y.diff.lag5  -3.078e-02  1.653e-02  -1.862 0.062649 .
## y.diff.lag6  -3.545e-02  1.653e-02  -2.144 0.032110 *
## y.diff.lag7   4.966e-02  1.654e-02   3.002 0.002699 **
## y.diff.lag8  -3.893e-02  1.656e-02  -2.351 0.018761 *
## y.diff.lag9   2.107e-02  1.656e-02   1.272 0.203540
## y.diff.lag10 -1.273e-02  1.657e-02  -0.768 0.442276
## y.diff.lag11  1.819e-02  1.657e-02   1.098 0.272397
## y.diff.lag12  2.275e-02  1.657e-02   1.373 0.169751
## y.diff.lag13 -2.073e-02  1.656e-02  -1.252 0.210830
## y.diff.lag14  1.949e-03  1.654e-02   0.118 0.906217
## y.diff.lag15 -2.522e-02  1.654e-02  -1.525 0.127302
## y.diff.lag16  5.197e-02  1.654e-02   3.142 0.001689 **
## y.diff.lag17  1.449e-02  1.656e-02   0.875 0.381598
## y.diff.lag18 -5.837e-02  1.656e-02  -3.525 0.000429 ***
## y.diff.lag19 -1.066e-03  1.659e-02  -0.064 0.948769
## y.diff.lag20  3.774e-02  1.649e-02   2.288 0.022177 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01354 on 3674 degrees of freedom
## Multiple R-squared:  0.0314, Adjusted R-squared:  0.02587
## F-statistic: 5.672 on 21 and 3674 DF,  p-value: 2.024e-15
```

The command is then executed without saving the output:

```
adfTest(log(VTd), lags=20, type='nc')
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 20
##   STATISTIC:
##     Dickey-Fuller: 1.2222
##   P VALUE:
##     0.9426
##
## Description:
##  Fri Apr 19 16:25:50 2024 by user:
```

As expected, the null hypothesis of non-stationarity is not rejected.

To complete this assessment, it is necessary to proceed with a further test on the first difference, to verify that there is only one unit root within the model. If this is not the case, it would imply violations of the market efficiency hypothesis.

By removing one lag at a time and evaluating the significance of the coefficient associated with the last delay in the linear regression model constructed during the ADF test for the stationarity of a time series, it was found that the lags to use are 19, and the intercept is not significant.

```
out1 = adfTest(dlvts, lag=19, type='nc')
summary(out1@test$lm)
```

```
##
## Call:
## lm(formula = y.diff ~ y.lag.1 - 1 + y.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.118008 -0.004939  0.000914  0.006408  0.113629
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## y.lag.1      -1.111e+00  8.322e-02 -13.351   <2e-16 ***
## y.diff.lag1   3.538e-03  8.105e-02   0.044   0.9652
## y.diff.lag2  -9.511e-03  7.881e-02  -0.121   0.9039
## y.diff.lag3   4.587e-05  7.673e-02   0.001   0.9995
## y.diff.lag4  -2.953e-03  7.453e-02  -0.040   0.9684
## y.diff.lag5  -3.321e-02  7.204e-02  -0.461   0.6448
## y.diff.lag6  -6.813e-02  6.957e-02  -0.979   0.3275
## y.diff.lag7  -1.791e-02  6.700e-02  -0.267   0.7893
## y.diff.lag8  -5.631e-02  6.441e-02  -0.874   0.3820
## y.diff.lag9  -3.470e-02  6.161e-02  -0.563   0.5732
## y.diff.lag10 -4.689e-02  5.857e-02  -0.801   0.4234
## y.diff.lag11 -2.817e-02  5.544e-02  -0.508   0.6114
## y.diff.lag12 -4.871e-03  5.199e-02  -0.094   0.9253
## y.diff.lag13 -2.509e-02  4.854e-02  -0.517   0.6052
## y.diff.lag14 -2.260e-02  4.446e-02  -0.508   0.6112
## y.diff.lag15 -4.730e-02  4.026e-02  -1.175   0.2401
## y.diff.lag16  5.206e-03  3.582e-02   0.145   0.8845
## y.diff.lag17  2.020e-02  3.077e-02   0.656   0.5116
## y.diff.lag18 -3.767e-02  2.460e-02  -1.531   0.1258
## y.diff.lag19 -3.821e-02  1.649e-02  -2.317   0.0205 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01354 on 3675 degrees of freedom
## Multiple R-squared:  0.5647, Adjusted R-squared:  0.5623
## F-statistic: 238.3 on 20 and 3675 DF,  p-value: < 2.2e-16
```

```
adfTest(dlvts, lags=19, type='nc')
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
```

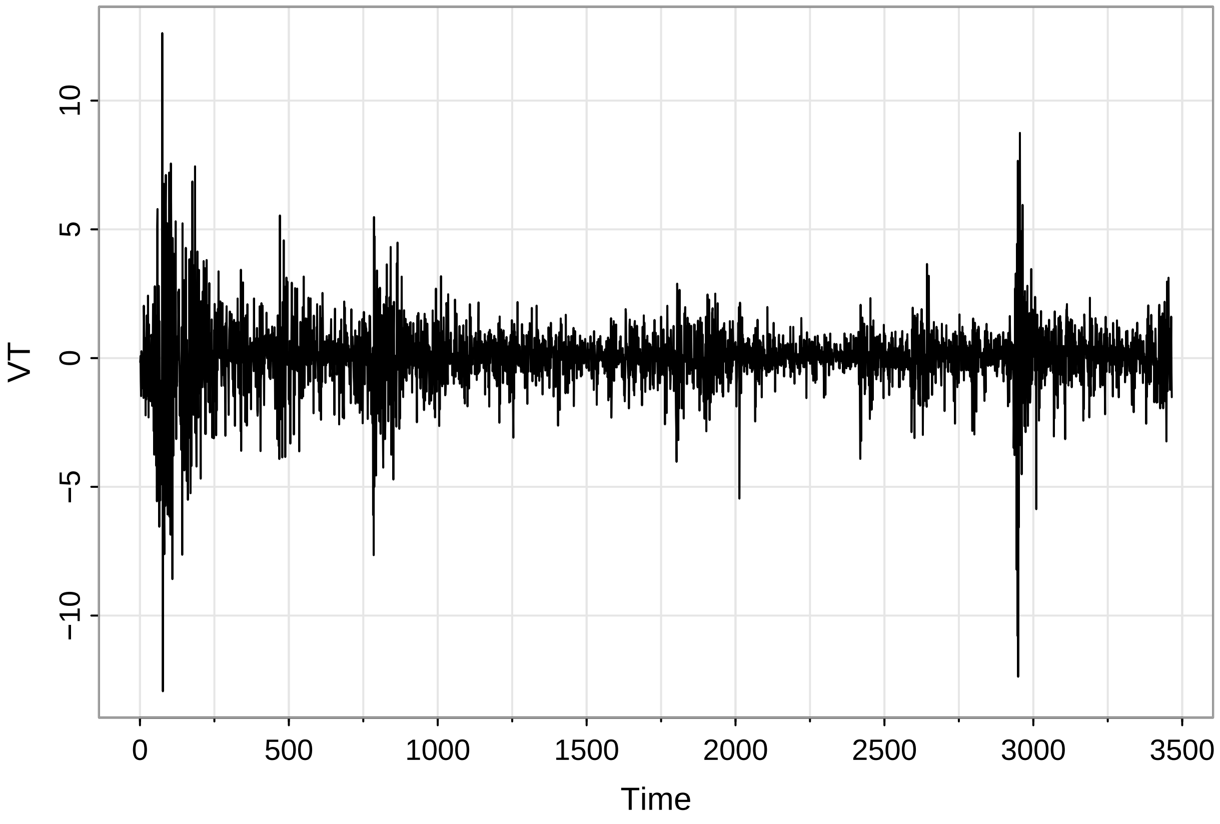
```
##     Lag Order: 19
##   STATISTIC:
##     Dickey-Fuller: -13.3505
##   P VALUE:
##     0.01
##
## Description:
##  Fri Apr 19 16:25:50 2024 by user:
```

As expected, the null hypothesis regarding the presence of an additional unit root is rejected.

The objective of the analysis is to verify whether heteroscedasticity is present or not, and if it is significant enough to warrant the application of a model subsequently.

Below, the log-prices are calculated with a window up to March 31, 2022, and the log percentage returns are computed.

```
p_VT = log(window(VTd, end="2022-03-31")) #log-prices
p_VT = as.vector(p_VT$VT.Adjusted[,1])
T = length(p_VT)
r_VT=100*(p_VT[2:T]-p_VT[1:T-1]) #log returns &
par(mfrow=c(1,1))
tsplot(r_VT, ylab='VT') # plot log returns %
```

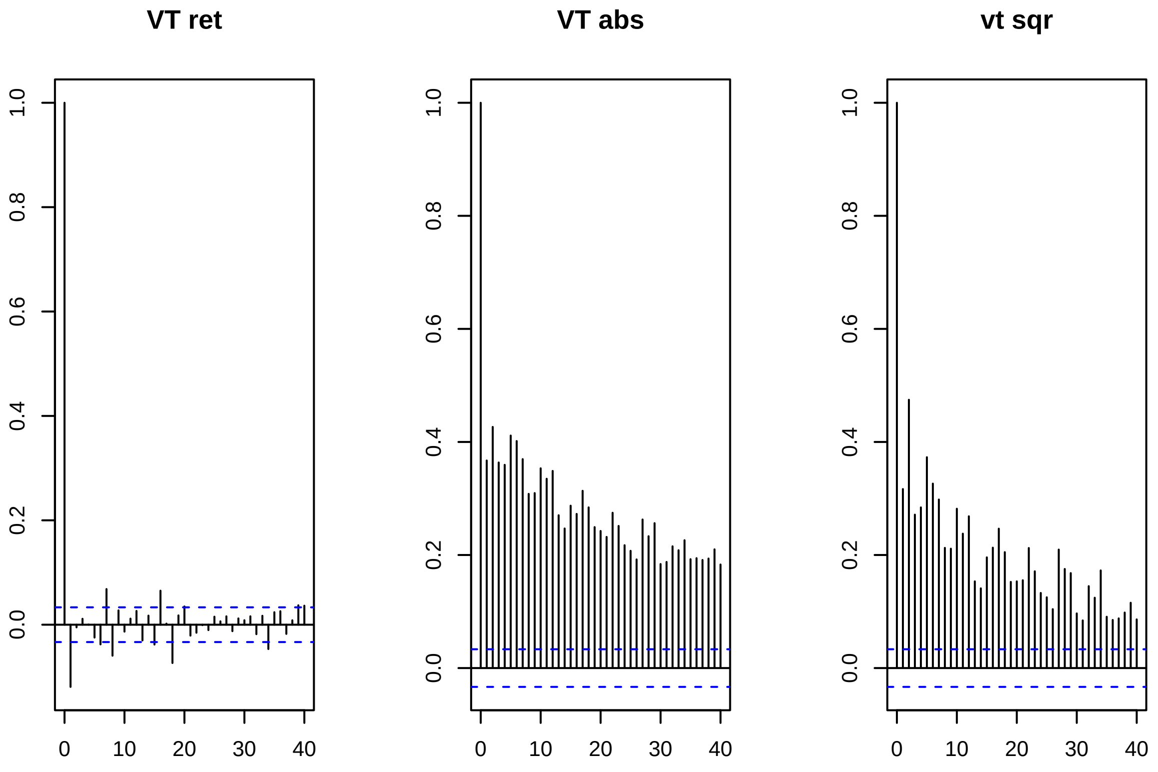


As previously noted, there is a volatility clustering effect, with phases of high volatility followed by periods of stability.

The Autocorrelation Functions (ACF) on returns, absolute values, and squares are analyzed below. There is serial dependence on absolute values and their squares. On returns, however, little is evident; the confidence bands are very narrow due to the sample dependence (with 3466 observations, they are indeed very small).

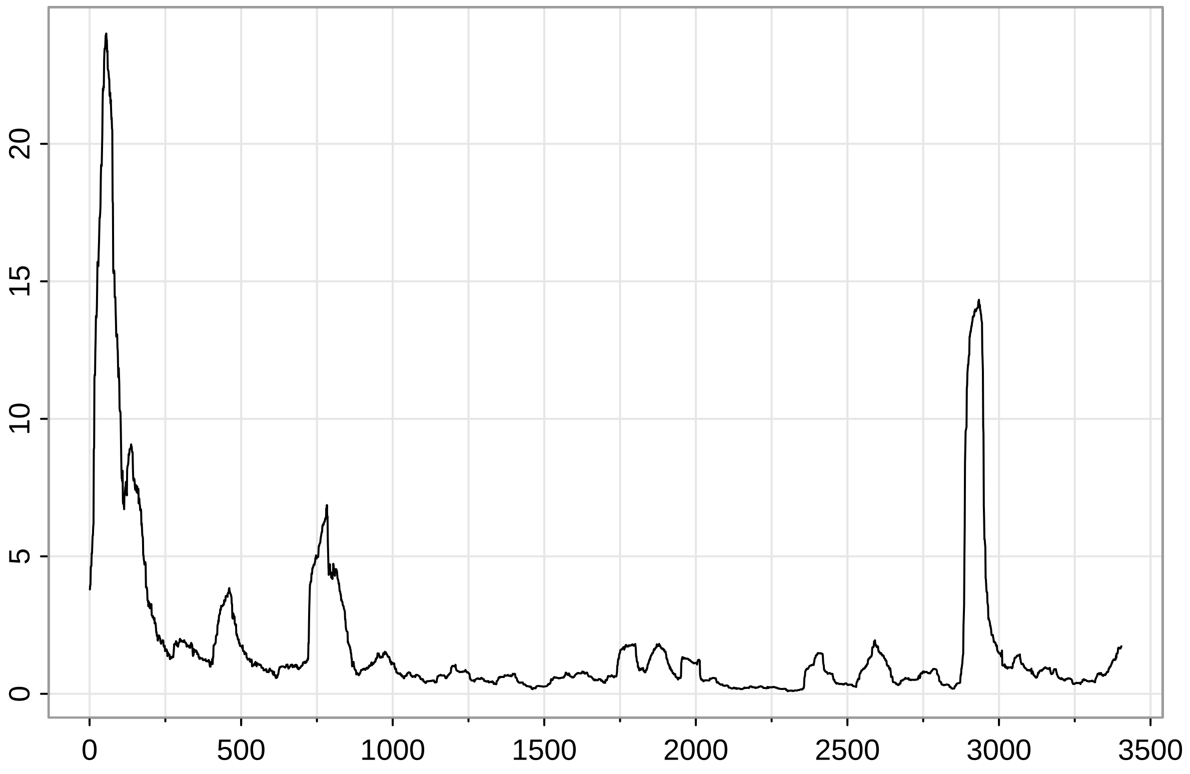
Focusing on absolute values and squares, a strong serial correlation is evident, even up to 40 lags, indicating significant persistence. This behavior is typically common in financial instruments that exhibit certain characteristics, while on the other hand, with securities that fluctuate widely or are illiquid, we might find little serial correlation.

```
par(mfrow=c(1,3))
acf(r_VT,lag.max=40,xlab="",ylab="",main="VT ret")
acf(abs(r_VT),lag.max=40,xlab="",ylab="",main="VT abs")
acf(abs(r_VT)^2,lag.max=40,xlab="",ylab="",main="vt sqr")
```

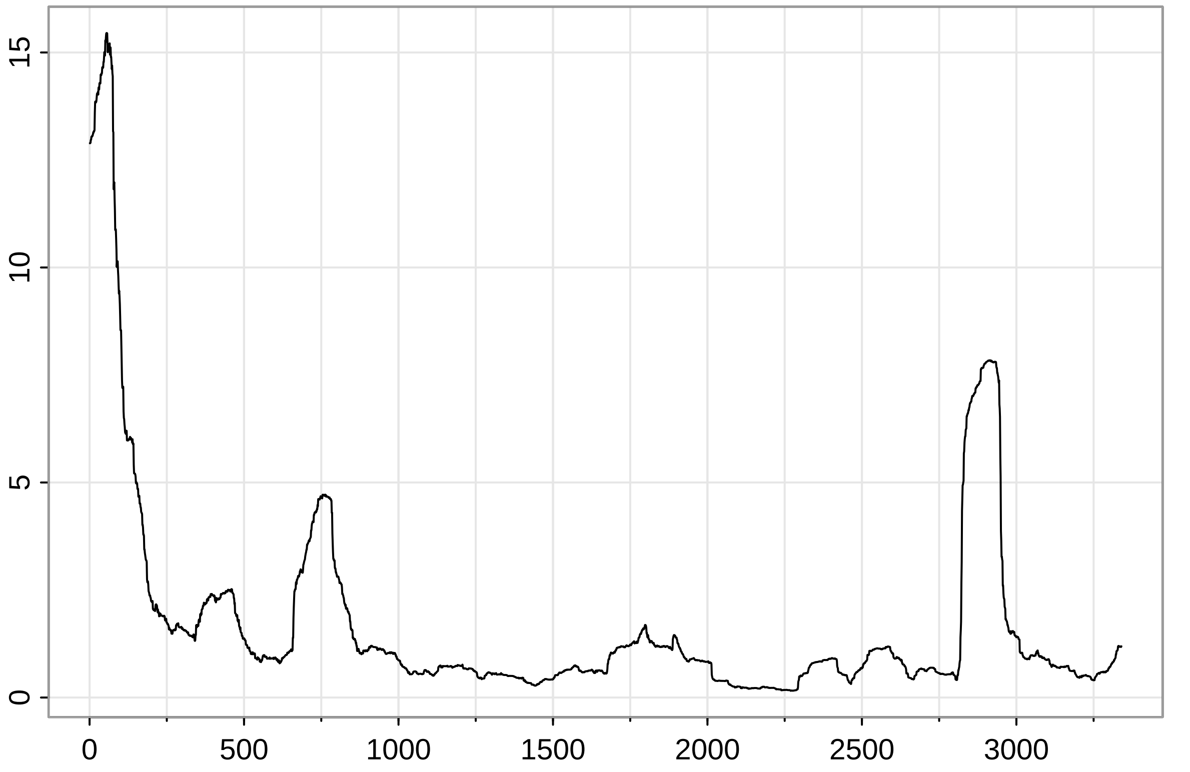


One of the first methods to assess heteroscedasticity and understand how it translates into variances is the rolling variance. Using 62 observations, we have a window of approximately one quarter, while with 126 observations, the window extends to a semester.

```
par(mfrow=c(1,1))
rollV=rollStats(as.timeSeries(r_VT),62,FUN=var)
tsplot(rollV,ylab="",xlab="")
```



```
rollV=rollStats(as.timeSeries(r_VT),126,FUN=var)
tsplot(rollV,ylab="",xlab="")
```



As expected, the historical series is characterized by heteroscedasticity; smoother trends can be observed on the graph with 126 observations due to the fact that adding one observation changes less than 1% of the sample.

We also accompany the analysis with statistical tests: Ljung-Box Test and ARCH Test

We expect to reject the null hypothesis, thus rejecting the idea that all autocorrelation functions are null up to a delay of 5 or 10.

The test statistic is a chi-square, with degrees of freedom depending on the delays considered.

```
y2 = (r_VT - mean(r_VT))^2 #returns in deviation from mean,squared
Box.test(y2, lag = 5, type = "Ljung-Box")
```

Ljung-Box Test at lags 5 and 10

```
##
##  Box-Ljung test
##
## data:  y2
## X-squared = 2149.8, df = 5, p-value < 2.2e-16
```

```
Box.test(y2, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  y2
## X-squared = 3416.1, df = 10, p-value < 2.2e-16
```

Following the results from the Ljung-Box test, where a low p-value led to the rejection of the null hypothesis of non-correlation, we now proceed with the LM-ARCH test.

Test LM-ARCH This test involves conducting an auxiliary regression to evaluate the null hypothesis of homoscedasticity against the alternative hypothesis of heteroscedasticity. If the test statistic exceeds a certain critical threshold, the null hypothesis of no heteroscedasticity is rejected in favor of the alternative hypothesis indicating the presence of heteroscedasticity.

The code below considers 1, 2, and 3 lags for the test:

```
# consider 1, 2, and 3 lags for the LM test
y2=as.timeSeries(y2)
y2L1 <- lag(y2,1)
y2L2 <- lag(y2,2)
y2L3 <- lag(y2,3)
# three test statistics and corresponding P-values
#
# First lag:
out1 <- lm(y2 ~ y2L1)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
# using T-2 because some observations are lost in lags 1,2,3 and to compare I choose 2
pchisq(LMARCH1,1,lower.tail=FALSE) # test only on the upper tail
```

```
## [1] 4.821489e-77
```

```
# Second lag:
out2 <- lm(y2 ~ y2L1 + y2L2)
sum2 <- summary(out2)
```

```r
LMARCH2 <- sum2$r.squared*(T-2)
pchisq(LMARCH2,2,lower.tail=FALSE)
```

```
## [1] 2.300224e-193
```

```r
# Third lag:
out3 <- lm(y2 ~ y2L1 + y2L2 + y2L3)
sum3 <- summary(out3)
LMARCH3 <- sum3$r.squared*(T-2)
pchisq(LMARCH3,3,lower.tail=FALSE)
```

```
## [1] 7.929598e-195
```

Given the rejection of the null hypothesis in all three ARCH tests, we conclude that there is heteroscedasticity present in the historical series considered.

3.2 "Fit a GARCH(1,1) model with normal distribution to the yields. The sample you use for estimation should run until the end of March 2022, thus leaving one year for later analysis. Comment on the differences in the fit to the data in the two stocks considered, assessing the presence of heteroschedasticity in the standardised residuals, the presence of skewness, and the consistency of the choice of distribution for innovations."

GARCH(1,1) Model

We now aim to fit a GARCH(1,1) model capable of capturing the heteroscedasticity present in the historical series.

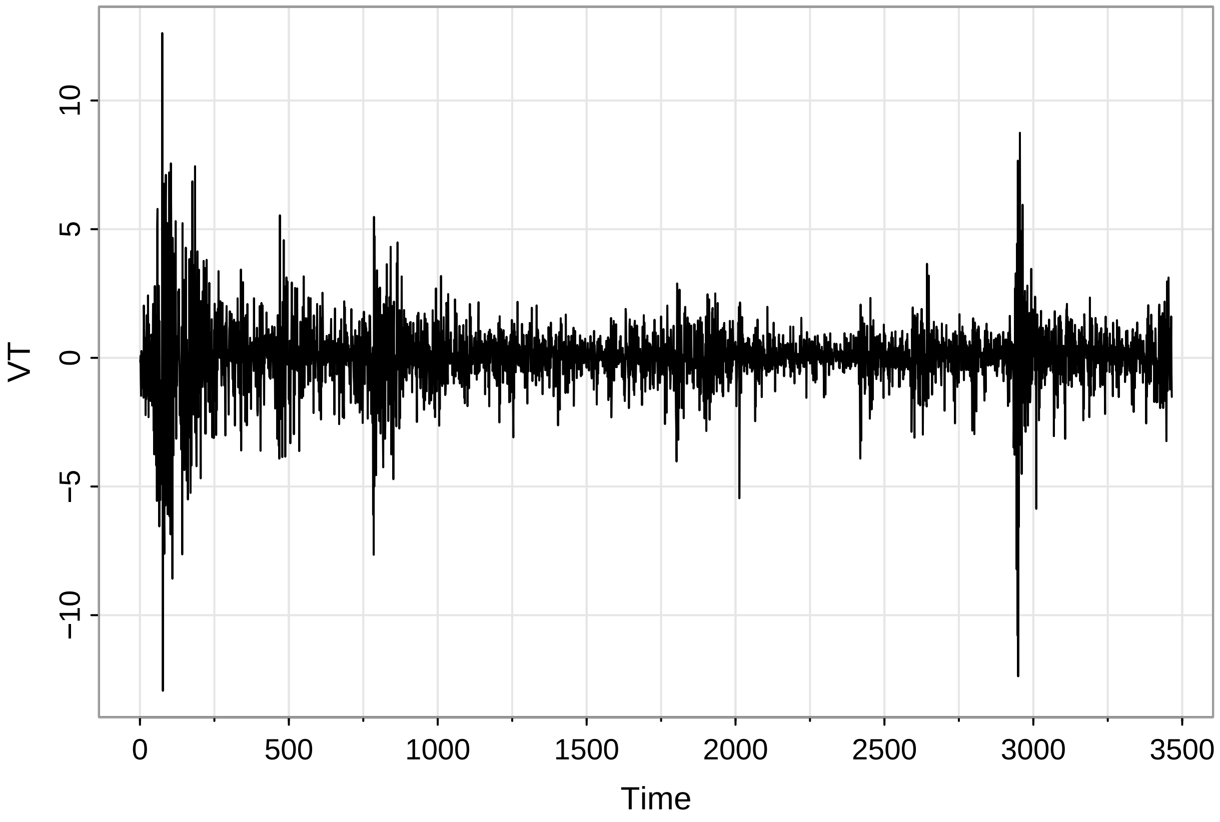
The GARCH(1,1) model posits that the volatility of a financial asset depends partly on the volatility from the previous time interval and a shock term. Specifically, the formula for conditional volatility at time t is:

$$\sigma_t^2 = \omega + \alpha \cdot r_{t-1}^2 + \beta \cdot \sigma_{t-1}^2$$

where σ_t^2 is the conditional volatility at time t, ω represents the baseline level of volatility, α is the weight of the square of the residual at time t-1, and β is the weight of the conditional volatility at time t-1.

Model specification:

```
par(mfrow=c(1,1))
tsplot(r_VT, ylab='VT') # plot of log percentage returns
```



```
spec0 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)), mean.model = list(ar
# estimate model saving output
fit0 <- ugarchfit(spec0, r_VT)
```

```
# estimate model with output on screen
fit0@fit$robust.matcoef
```

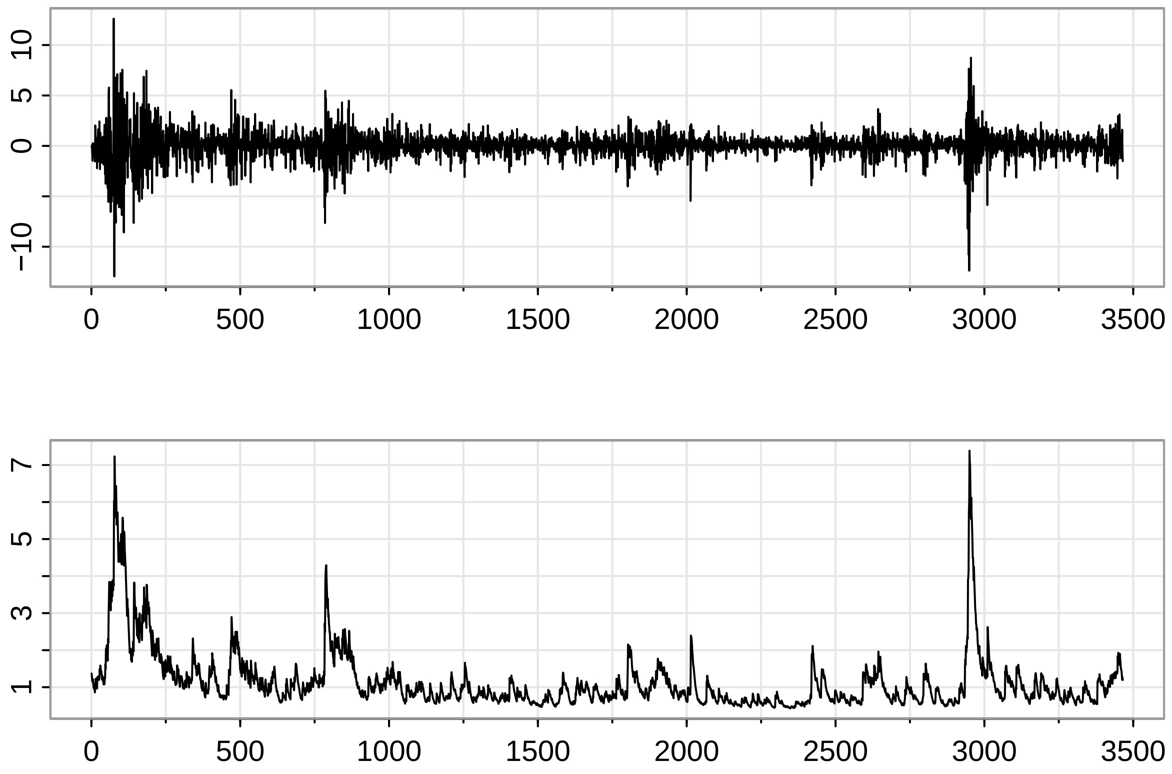
```
##           Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.06660303  0.01183198  5.629067 1.811870e-08
## omega  0.02410625  0.00623667  3.865245 1.109779e-04
## alpha1 0.15594476  0.02203340  7.077654 1.466161e-12
## beta1  0.83534027  0.01976134 42.271446 0.000000e+00
```

Observing the p-values calculated with the quasi-maximum likelihood method, it is noted that all parameters are statistically significant.

The beta parameter assumes a value of 0.83 and is within the range of values that are expected for this parameter.

```
par(mfrow=c(2,1))
tsplot(r_VT, type='l', ylab="", xlab="")
tsplot(fit0@fit$sigma, type='l',ylab="",xlab="")
```

Does the estimated model provide dynamic variances?



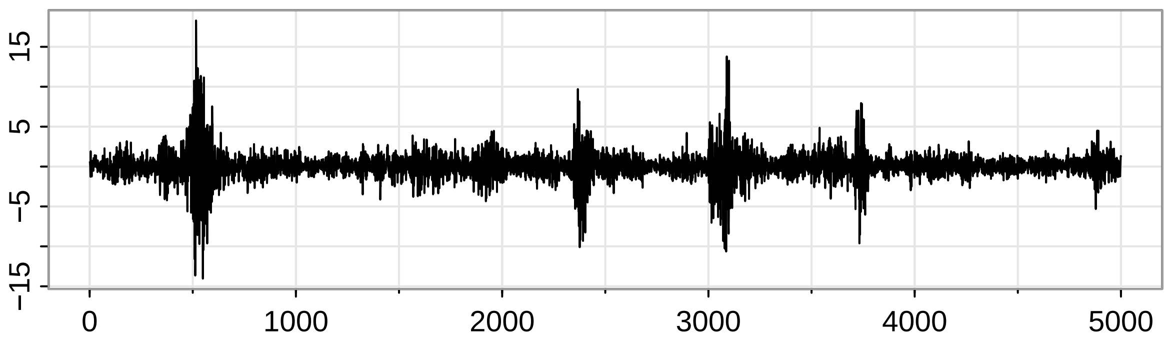
This is a graphical representation that combines the returns with the sequence of estimated variances. The chart resembles the variances obtained through rolling methods.

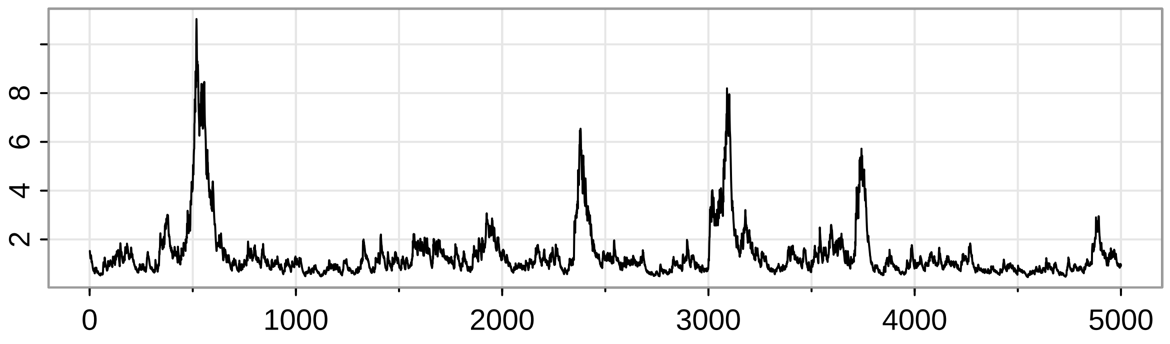
Does this choice replicate the empirical characteristics of my series?

Simulation of a GARCH model and charts I will check if the estimated model replicates the stylized facts.

AGAINST:

```
sim0<-ugarchsim(fit0, n.sim = 5000)
par(mfrow=c(2,1))
tsplot(sim0@simulation$seriesSim,type="l",ylab="",xlab="")
tsplot(sim0@simulation$sigmaSim,type="l",ylab="",xlab="")
```





We notice the presence of volatility clustering despite high persistence in volatility, but we have a frequency of extreme events which, although in line with our historical series, does not deviate too much from the many other peaks of volatility present.

The model is likely unable to replicate the presence of fat tails in the distributions, therefore we must rethink the specification, changing the choice made for z and go on to analyze the results later.

We will probably need to replace the normal distribution with another distribution.

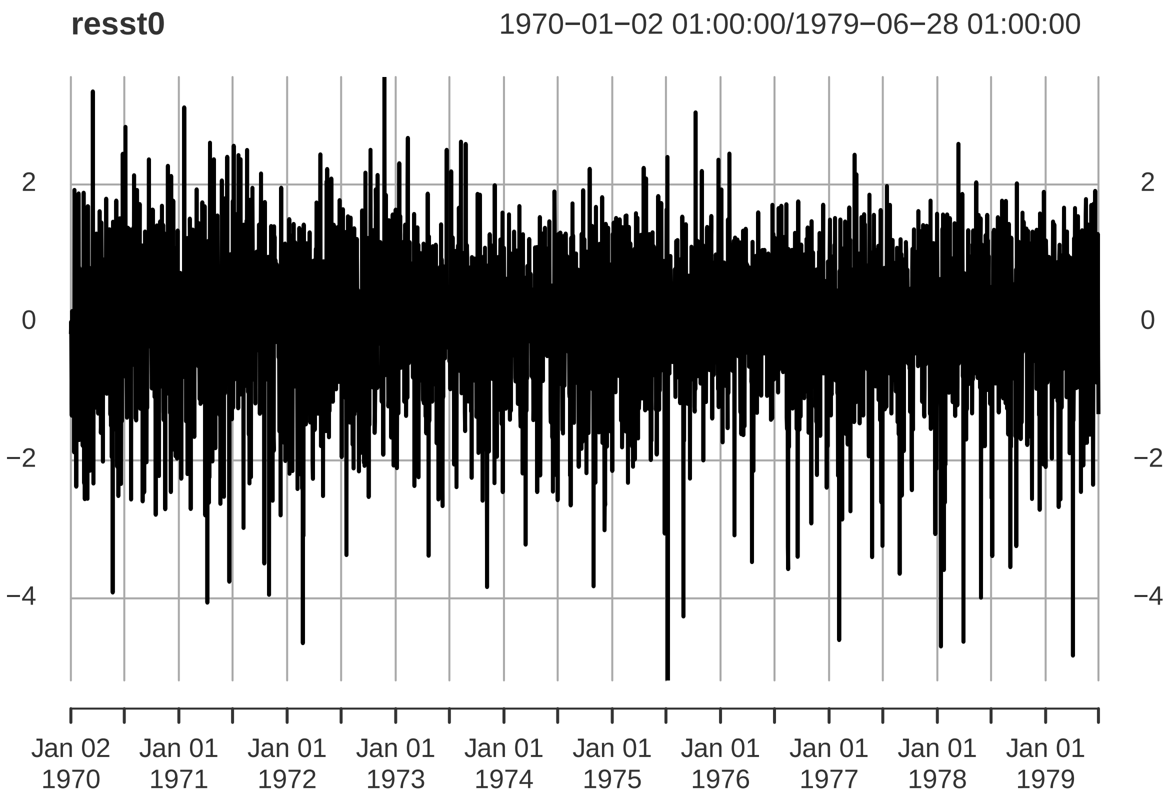
Analysis of Standardized Residuals: Having estimated a model, a variance dynamics and a distribution are hypothesized, which translates into the distribution of the quantity z.

I define the quantity z as the returns deviating from the mean and standardized by the conditional variance.

In the first case, the z's are independent and homoscedastic with unit variance, distributed as a normal.

I can perform analysis on the standardized residuals:

```
resst0 <- residuals(fit0,standardize=TRUE)
plot(resst0)
```



```
# analysis for GARCH effects
y2<-(as.timeSeries(resst0))^2
# consider 1, 2, and 3 lags for the LM test
y2L1 <- lag(y2,1)
y2L2 <- lag(y2,2)
y2L3 <- lag(y2,3)

# Evaluate the regression to understand if zt ^ 2 depends on its lags.
# If alpha = 0, zt is homoscedastic.
# If alpha is different from zero, zt is heteroscedastic.
# Test with 1,2,3 delays, the test statistic changes.
# The null hypothesis is that the lags are jointly zero.
# If the test is correctly specified, I expect to have
# no further evidence of ARCH effects.
# three test statistics and their P-values
out1 <- lm(y2 ~ y2L1)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1,1,lower.tail=FALSE)
```

```
## [1] 0.2757858
```

```
out1 <- lm(y2 ~ y2L1 + y2L2)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1,2,lower.tail=FALSE)
```

```
## [1] 0.04348037
```

```
out1 <- lm(y2 ~ y2L1 + y2L2 + y2L3)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1,3,lower.tail=FALSE)
```

```
## [1] 0.0958508
```

```
# In this case, we do not reject the null hypothesis, the coefficients are zero,
# in the case of the first, only one coefficient.
```

Observing the heteroscedasticity that remains may be captured by variance skewness, and hence there might be an incorrect specification of variance dynamics.

A Ljung-Box test is also conducted:

```
# Ljung-Box test (stats) at 5 and 10 lags
Box.test(as.numeric(y2), lag = 5, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  as.numeric(y2)
## X-squared = 7.8358, df = 5, p-value = 0.1655
```

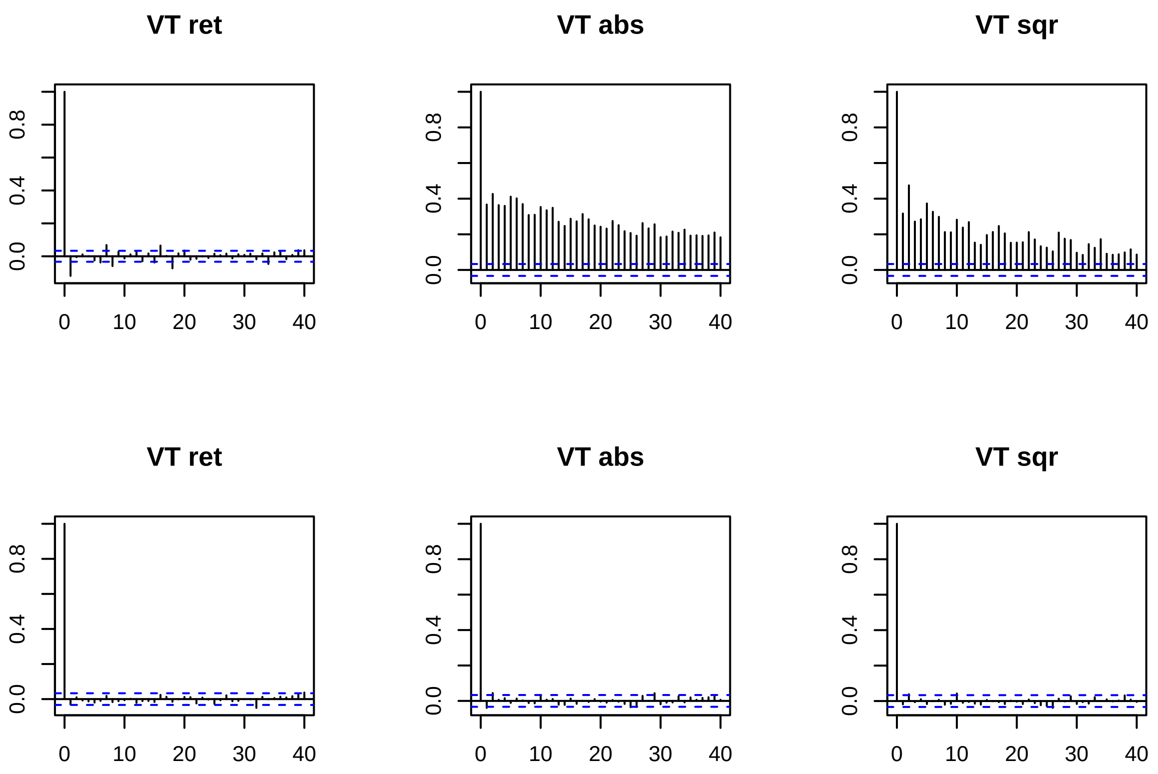
```
Box.test(as.numeric(y2), lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  as.numeric(y2)
## X-squared = 16.587, df = 10, p-value = 0.08402
```

I do not reject the null hypothesis of the Ljung-Box test, and I can hypothesize that the modeling of heteroscedasticity is sufficient.

The ACFs of returns, absolute values, squares, and of standardized residuals, absolute values, and squares are also evaluated from a graphical perspective.

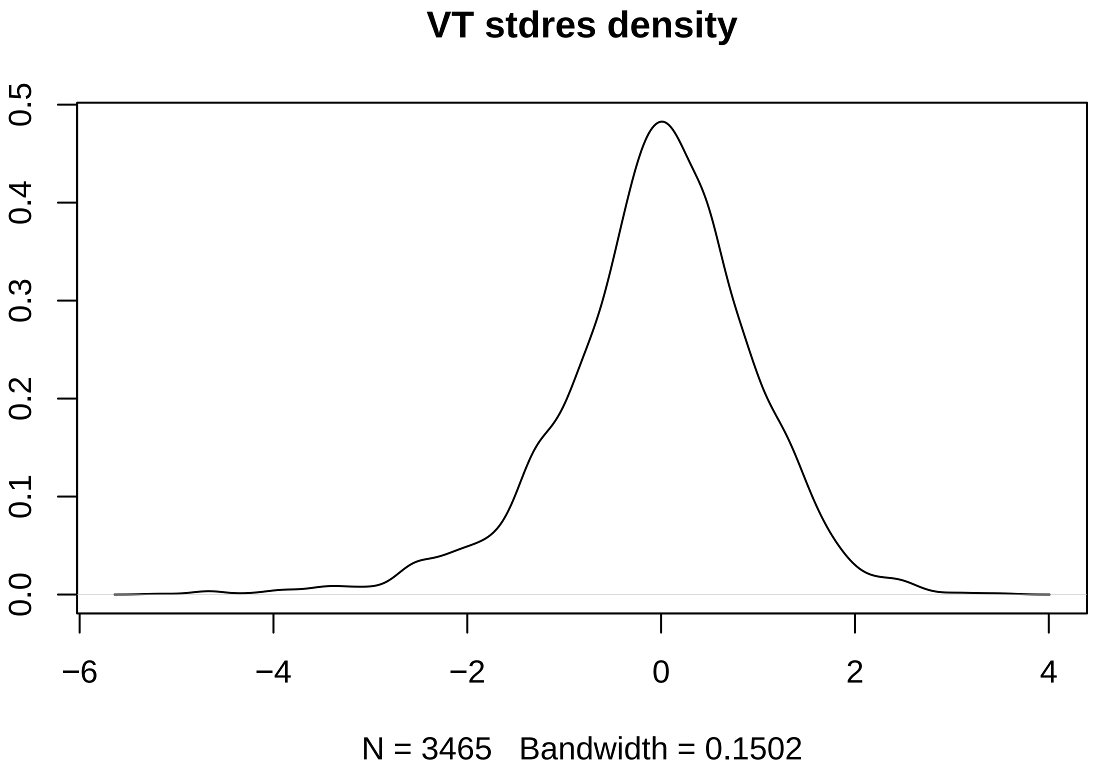
```
# Correlograms
par(mfrow=c(2,3))
acf(r_VT,lag.max=40,xlab="",ylab="",main="VT ret")
acf(abs(r_VT),lag.max=40,xlab="",ylab="",main="VT abs")
acf(abs(r_VT)^2,lag.max=40,xlab="",ylab="",main="VT sqr")
acf(as.numeric(resst0),lag.max=40,xlab="",ylab="",main="VT ret")
acf(abs(as.numeric(resst0)),lag.max=40,xlab="",ylab="",main="VT abs")
acf(abs(as.numeric(resst0))^2,lag.max=40,xlab="",ylab="",main="VT sqr")
```



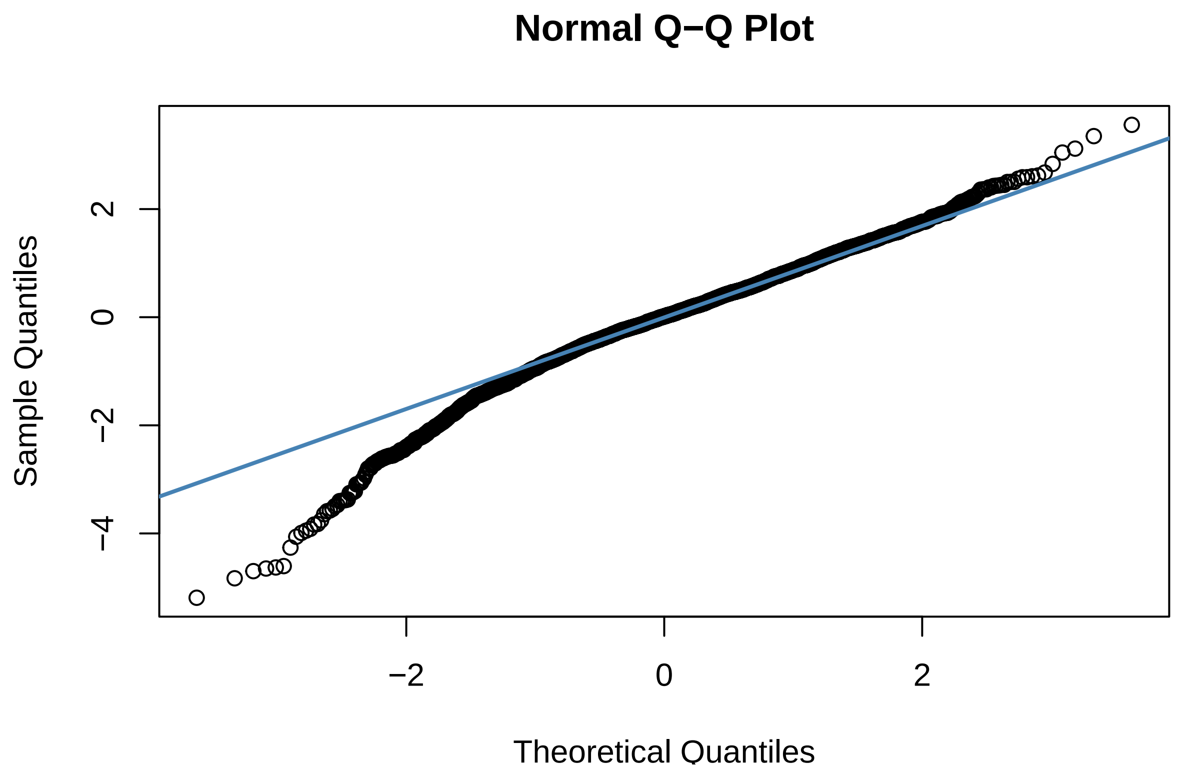
As seen previously, there is structure on squared returns and absolute values, while nothing notable on the standardized residuals of the estimated model.

#### Assessing if the assumed distribution is correct.

```
# Evaluating the density of residuals
# QQ-plot against alternative distributions
d0<-density(resst0)
# Residual density
par(mfrow=c(1,1))
plot (d0,main ="VT stdres density",ylab ="")
```



```
# QQ-plot against normal
qqnorm(resst0,main="Normal Q-Q Plot",xlab="Theoretical Quantiles",ylab ="Sample Quantiles")
qqline(resst0,col="steelblue",lwd=2)
```



The tails are fat and there seems to be some skewness present as well.

It does not seem appropriate to use a normal density given the heaviness of the tails and the deviation from normality at these tails.

```
y=r_VT-mean(r_VT)
nnr=y*(y<0)# per negative sign bias
ppr=y*(y>0)# per positive signi bias
dn=y<0# per sign bias
nnr=as.timeSeries(nnr)
ppr=as.timeSeries(ppr)
dn=as.timeSeries(dn)
nnr=lag(nnr,1)
ppr=lag(ppr,1)
dn=lag(dn,1)

out.sbt=lm(resst0^2 ~ dn)
summary(out.sbt)
```

Tests for variance skewness

```
##
## Call:
## lm(formula = resst0^2 ~ dn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1201 -0.8799 -0.6534  0.1250 26.0410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.89255    0.04501  19.830  < 2e-16 ***
## dnTRUE       0.22758    0.06563   3.467 0.000532 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.928 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.003461,   Adjusted R-squared:  0.003173
## F-statistic: 12.02 on 1 and 3462 DF,  p-value: 0.0005319
```

```
out.nsbt=lm(resst0^2 ~ nnr)
summary(out.nsbt)
```

```
##
## Call:
## lm(formula = resst0^2 ~ nnr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2794 -0.9327 -0.6740  0.1410 25.9469
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.98663    0.03616  27.283   <2e-16 ***
## nnr         -0.03019    0.03545  -0.852    0.394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.931 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0002095,  Adjusted R-squared:  -7.934e-05
## F-statistic: 0.7253 on 1 and 3462 DF,  p-value: 0.3945
```

```
out.psbt=lm(resst0^2 ~ ppr)
summary(out.psbt)
```

```
##
## Call:
## lm(formula = resst0^2 ~ ppr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0580 -0.9200 -0.6518  0.1394 26.1418
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.05799    0.03711  28.511  < 2e-16 ***
## ppr         -0.13604    0.04059  -3.352 0.000811 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.928 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.003235,   Adjusted R-squared:  0.002947
## F-statistic: 11.24 on 1 and 3462 DF,  p-value: 0.0008112
```

```
out.jsbt=lm(resst0^2 ~ dn + nnr + ppr)
summary(out.jsbt)
```

```
##
## Call:
## lm(formula = resst0^2 ~ dn + nnr + ppr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1558 -0.8926 -0.6455  0.1376 26.1413
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.96347    0.05889  16.361   <2e-16 ***
## dnTRUE       0.19293    0.08439   2.286   0.0223 *
## nnr          0.03976    0.04063   0.979   0.3278
## ppr         -0.08750    0.04688  -1.867   0.0620 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.927 on 3460 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.004739,   Adjusted R-squared:  0.003876
## F-statistic: 5.491 on 3 and 3460 DF,  p-value: 0.0009226
```

We reject the null hypothesis (both with the sign bias test and the positive sign bias) that there is no asymmetry in the model and conclude that we likely need to model variance asymmetry using an appropriate model.

3.3 "Adapt to the returns at least three different specifications from the GARCH model family (thus GARCH(1,1) plus at least two others, e.g., EGARCH, APARCH, or other possibilities provided by the library you are using). For each specification use at least two different distributions (the same for all chosen GARCH models, thus Normal and at least one other distribution). Always estimate using data up to the end of March 2022. Evaluate whether the fit to the data improves, supporting your observations with appropriate indicators, tests or graphical analysis. Among the chosen models, identify the best specification using information criteria."

The following will be used:

1)GARCH(1,1)

2)GJR-GARCH (1,1)

3)APARCH (1,1)

And as distributions, the Normal and the skewed t-distribution will be used.

The GARCH(1,1) model is re-specified below.

```
p_VT = log(window(VTd, end="2022-03-31")) #log-prices
p_VT = as.vector(p_VT$VT.Adjusted[,1])
T = length(p_VT)
r_VT=100*(p_VT[2:T]-p_VT[1:T-1]) #log returns %
y = r_VT - mean(r_VT)

spec0n<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
mean.model = list(armaOrder = c(0, 0),
include.mean = TRUE), distribution.model="norm")
fit0n<-ugarchfit(spec0n,r_VT)
resst0n <- residuals(fit0n,standardize=TRUE)
fit0n@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.06660303  0.01183198  5.629067 1.811870e-08
## omega  0.02410625  0.00623667  3.865245 1.109779e-04
## alpha1 0.15594476  0.02203340  7.077654 1.466161e-12
## beta1  0.83534027  0.01976134 42.271446 0.000000e+00
```

The GJR-GARCH is specified below.

The asymmetry parameter is γ_1.

```
specgn <- ugarchspec(variance.model = list(model="gjrGARCH", garchOrder = c(1, 1)), mean.model = list(a
fitgn <- ugarchfit(specgn,r_VT)
resstgn <- residuals(fitgn,standardize=TRUE)
fitgn@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.03364732 0.012265015  2.743357 6.081446e-03
## omega  0.02517779 0.005934567  4.242566 2.209789e-05
## alpha1 0.04880759 0.021970704  2.221485 2.631812e-02
## beta1  0.84591714 0.019804106 42.714230 0.000000e+00
## gamma1 0.18139297 0.039182517  4.629436 3.666623e-06
```

Looking at the robust p-values, obtained through quasi-maximum likelihood, of the estimated coefficients, it can be seen that they are all significant.

In particular, the beta, the coefficient related to the persistence of past variances, has a high value (0.84) as expected, and it is noted that the gamma1 coefficient, related to the asymmetry in variance, is strongly significant.

If the shock is positive, the effect is alpha1.

If the shock is negative, the effect is $\alpha_1 + \gamma_1$.

Below, the APARCH model is specified.

```
specan <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)), mean.model = list(arma
fitan <- ugarchfit(specan,r_VT)
resstan <- residuals(fitan,standardize=TRUE)
fitan@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.01770283 0.012740994  1.389438 1.646995e-01
## omega  0.02730570 0.005781998  4.722538 2.329196e-06
## alpha1 0.11970982 0.016317166  7.336435 2.193801e-13
## beta1  0.88848557 0.016440822 54.041433 0.000000e+00
## gamma1 0.74812003 0.112265611  6.663840 2.667644e-11
## delta  0.88050822 0.117467454  7.495763 6.594725e-14
```

All parameters are significant:

As usual, β assumes a high value: 0.89, and it multiplies the lagged variance indicating how much of the past volatility affects current volatility.

This is the context in which we look at the asymmetry in variance:

$$\alpha_1 \cdot (|\epsilon_{t-1}| - \gamma_1 \cdot \epsilon_{t-1})^{\delta_1}$$

The parameter γ_1 ranges between -1 and 1, and the presence of asymmetry, understood as the greater impact of negative shocks compared to positive shocks, requires that γ be greater than zero.

Indeed, if ϵ_{t-1} is greater than zero, then α_1 multiplies:

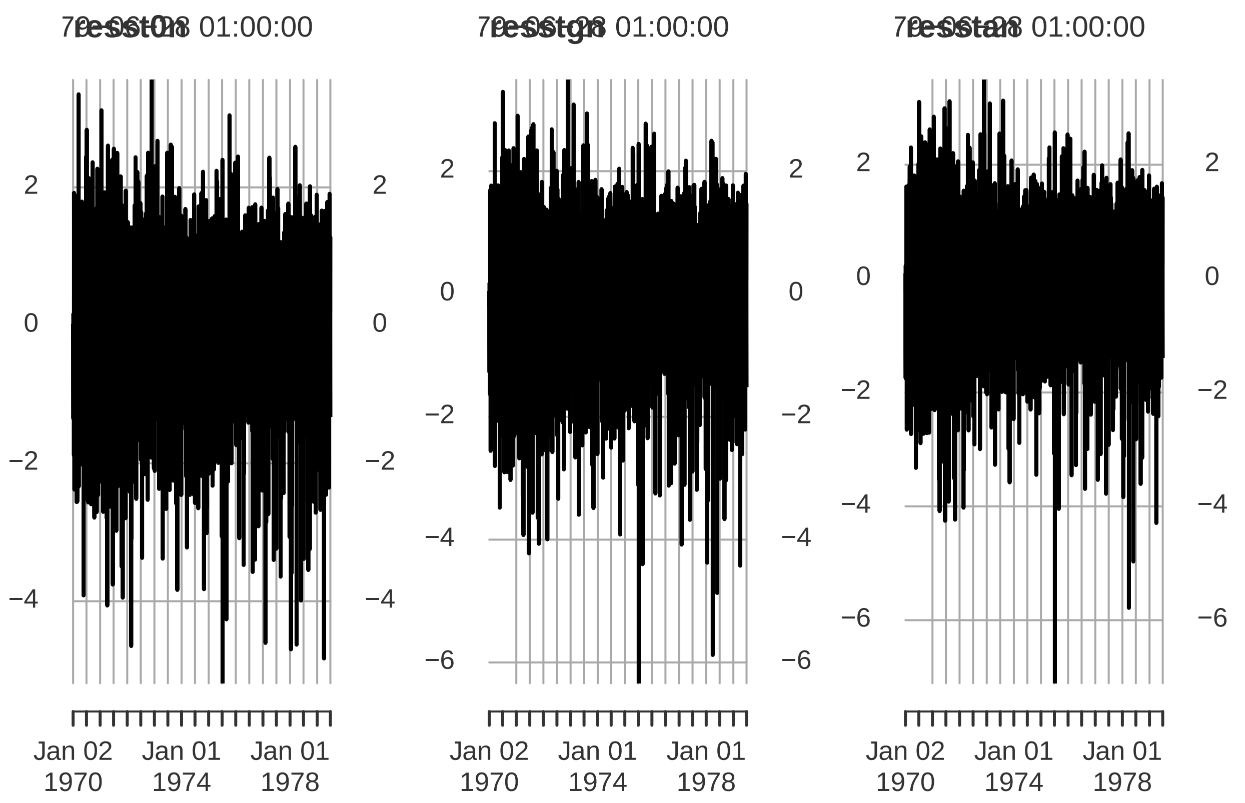
$$((1 - \gamma) \cdot \epsilon_{t-1})^{\delta_1}$$

Instead, if $\epsilon_{t-1} < 0$, α_1 multiplies:

$$((1 + \gamma) \cdot (-\epsilon_{t-1}))^{\delta_1}$$

Thus, as theory suggests, the impact is greater for negative shocks as γ_1 is estimated to be greater than zero.

```
par(mfrow=c(1,3))
plot(resst0n)
plot(resstgn)
plot(resstan)
```



```r
par(mfrow=c(1,1))

ks.test(as.matrix(resst0n),"pnorm",0,1)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  as.matrix(resst0n)
## D = 0.037457, p-value = 0.0001198
## alternative hypothesis: two-sided
```

The null hypothesis of normality is rejected.

Specification of models by changing the distribution and using asymmetric t.

```
spec0ta <- ugarchspec(variance.model = list(model="sGARCH", garchOrder = c(1, 1)), mean.model = list(arm
fit0ta <- ugarchfit(spec0ta,r_VT)
resst0ta <- residuals(fit0ta,standardize=TRUE)

specgta <- ugarchspec(variance.model = list(model="gjrGARCH", garchOrder = c(1, 1)), mean.model = list(a
fitgta <- ugarchfit(specgta,r_VT)
resstgta <- residuals(fitgta,standardize=TRUE)

specata <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)), mean.model = list(arm
fitata <- ugarchfit(specata,r_VT)
resstata <- residuals(fitata,standardize=TRUE)
```

The significance of the skew and shape coefficients associated with the choice of the asymmetric t-distribution is assessed below.

```
fit0ta@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.05997024 0.011081530  5.411730 6.241892e-08
## omega  0.01734842 0.004536613  3.824091 1.312553e-04
## alpha1 0.14628577 0.017448204  8.384001 0.000000e+00
## beta1  0.85001646 0.016236876 52.350986 0.000000e+00
## skew   0.85962749 0.020882832 41.164316 0.000000e+00
## shape  6.68994091 0.690063394  9.694676 0.000000e+00
```

```
fitgta@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.02842987 0.012143743  2.341112 1.922638e-02
## omega  0.02221374 0.005444534  4.080007 4.503439e-05
## alpha1 0.01679190 0.012706547  1.321516 1.863295e-01
## beta1  0.85970322 0.018602603 46.214136 0.000000e+00
## gamma1 0.22523914 0.037052806  6.078869 1.210335e-09
## skew   0.83539097 0.020479208 40.792153 0.000000e+00
## shape  6.84542149 0.750176208  9.125085 0.000000e+00
```

```
fitata@fit$robust.matcoef
```

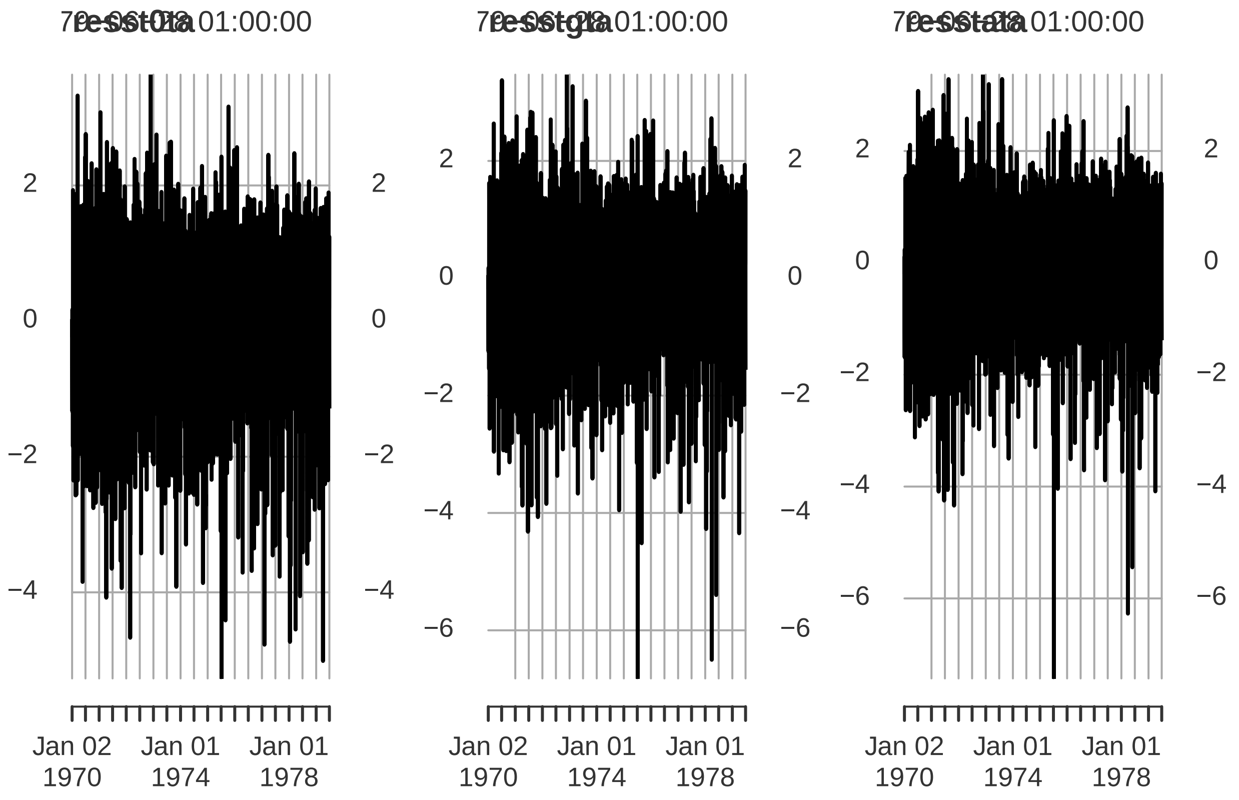
```
##          Estimate  Std. Error    t value     Pr(>|t|)
## mu     0.01064383 0.012921489  0.8237306 4.100926e-01
## omega  0.02541971 0.005505266  4.6173455 3.886799e-06
## alpha1 0.11040457 0.011492001  9.6070799 0.000000e+00
## beta1  0.90167073 0.012506721 72.0948939 0.000000e+00
## gamma1 0.94919492 0.074848773 12.6815028 0.000000e+00
## delta  0.87345622 0.084212577 10.3720400 0.000000e+00
## skew   0.81713820 0.021213537 38.5196581 0.000000e+00
## shape  7.09408067 0.808570050  8.7736130 0.000000e+00
```

Despite the α_1 coefficient being non-significant in the GJR model with skewed-t distribution, it is customary to leave it in, whereas looking at the APARCH, one might consider removing the mean and re-estimating:

```
specata <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)), mean.model = list(arm
fitata <- ugarchfit(specata,r_VT)
resstata <- residuals(fitata,standardize=TRUE)
fitata@fit$robust.matcoef
```

```
##           Estimate  Std. Error   t value     Pr(>|t|)
## omega  0.02669834 0.009027517  2.957440 3.102055e-03
## alpha1 0.11076401 0.017278210  6.410618 1.449310e-10
## beta1  0.90213752 0.021629229 41.709185 0.000000e+00
## gamma1 0.95895044 0.023665962 40.520239 0.000000e+00
## delta  0.85988819 0.114654895  7.499795 6.394885e-14
## skew   0.81250755 0.018760531 43.309412 0.000000e+00
## shape  7.06588178 0.828753301  8.525917 0.000000e+00
```

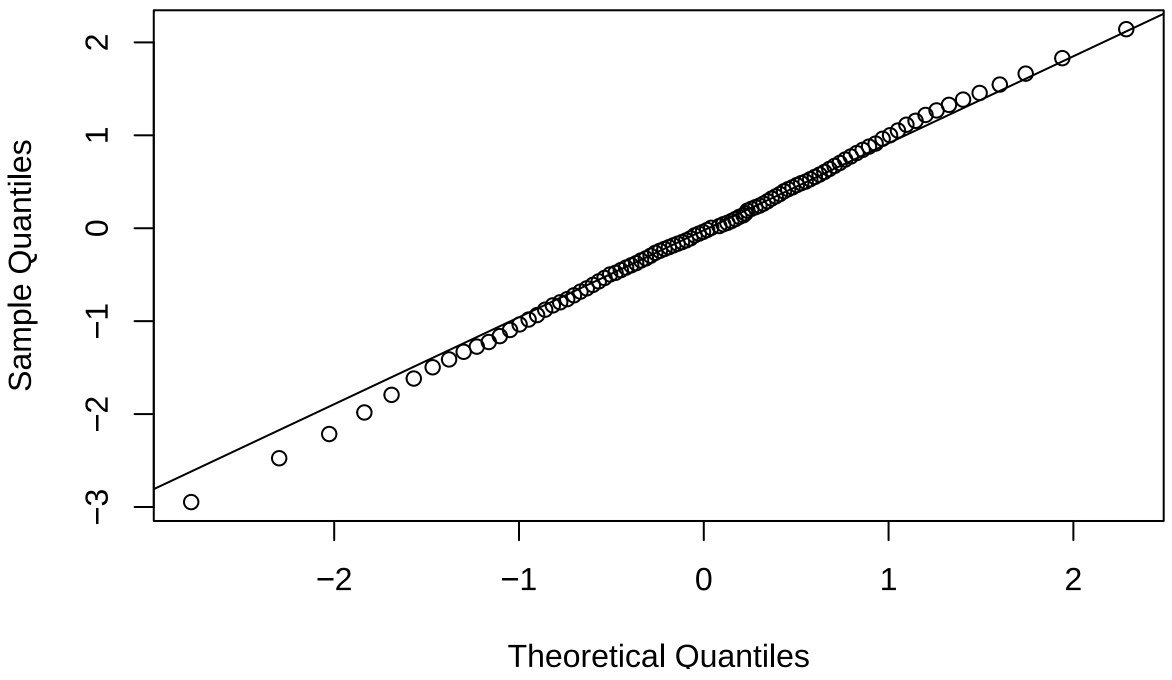
```
par(mfrow=c(1,3))
plot(resst0ta)
plot(resstgta)
plot(resstata)
```



```
par(mfrow=c(1,1))
```

This code is used to create a Q-Q plot to compare the theoretical distribution of the skewed-t with the empirical distribution of the standardized residuals *resst0ta* from the previously specified GARCH(1,1) model with skewed-t.

```
p=0.01*(1:99)
qemp=quantile(resst0ta,probs=p)
qteor=qdist(distribution="sstd",p,mu=0,sigma=1,skew=fit0ta@fit$coef[5],shape=fit0ta@fit$coef[6])
qqplot(qteor,qemp,xlab="Theoretical Quantiles",ylab ="Sample Quantiles")
qqline(qemp, distribution = function(p)
   qdist(distribution="sstd",p,mu=0,sigma=1,
                  skew=fit0ta@fit$coef[5],shape=fit0ta@fit$coef[6]))
```



It is noted a deviation in the tails, which means that the theoretical distribution is not fully capable of capturing the variation in the data.However, this deviation is definitely less pronounced than what was seen with the normal distribution.

Following is the Kolmogorov-Smirnov test to check if an empirical distribution (residuals) follows a specified theoretical distribution (skew-t distribution):

```
# H0: Skew-T
y=as.matrix(resst0ta)
y=as.matrix(sort(y))
ks.test(as.matrix(resst0ta),pdist(distribution="sstd",y,mu=0,sigma=1,lambda=-0.5,
skew=fit0ta@fit$coef[5],shape=fit0ta@fit$coef[6]))
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  as.matrix(resst0ta) and pdist(distribution = "sstd", y, mu = 0, sigma = 1, lambda = -0.5, ske
## D = 0.48831, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

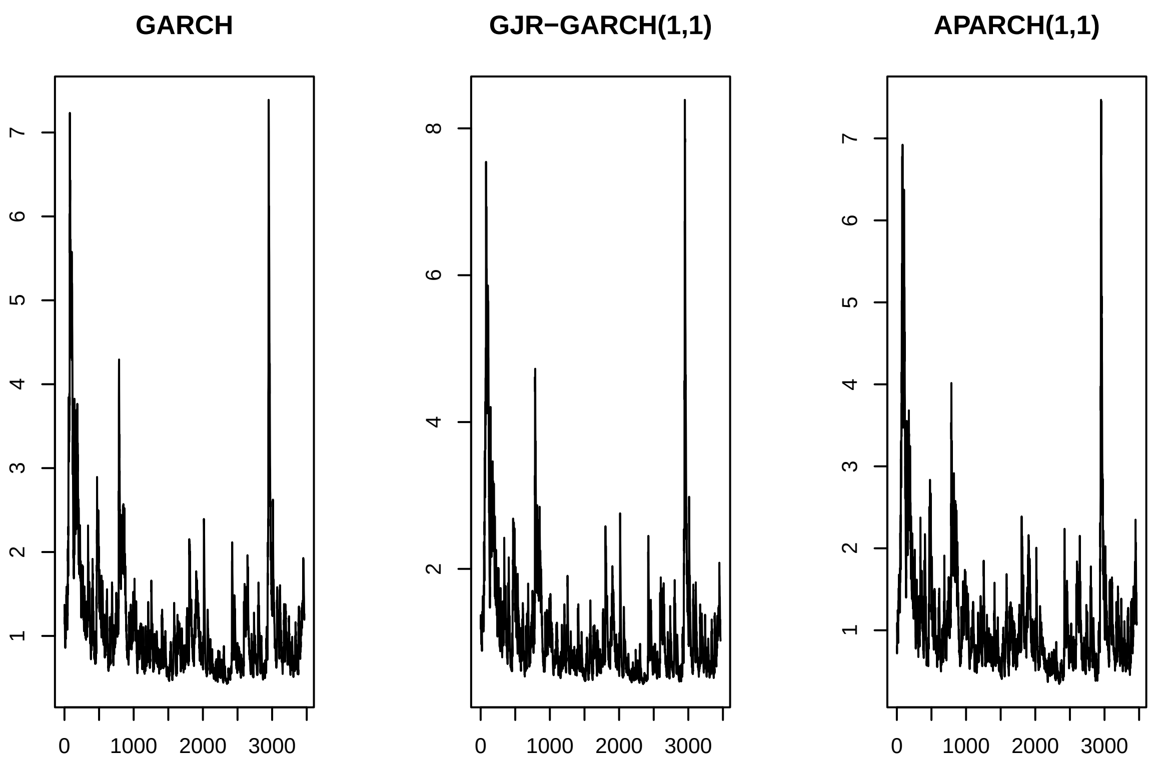
I reject the hypothesis of the distribution as a skewed-t probably due to the length of the series.

However, the fit to the data seems to improve as the skew and shape parameters are always significant and as noted earlier from a graphical standpoint, the deviation in the tails seems lesser.

Plots of estimated variances for the three models using the function sigma:

```
# grafici varianze
par(mfrow=c(1,3))
sigma0n <- sigma(fit0n)
sigmagn <- sigma(fitgn)
sigmaan <- sigma(fitan)
par(mfrow=c(1,3))
plot(as.numeric(sigma0n),type="l",ylab="",xlab="")
title("GARCH")
plot(as.numeric(sigmagn),type="l",ylab="",xlab="")
```

```
title("GJR-GARCH(1,1)")
plot(as.numeric(sigmaan),type="l",ylab="",xlab="")
title("APARCH(1,1)")
```



```
par(mfrow=c(1,1))
```

In the three different graphs, it is difficult to notice differences.

Test Sign Bias

```
y=r_VT-mean(r_VT)
nnr=y*(y<0)# for negative sign bias
ppr=y*(y>0)# for positive signi bias
dn=y<0# for sign bias
nnr=as.timeSeries(nnr)
ppr=as.timeSeries(ppr)
dn=as.timeSeries(dn)
nnr=lag(nnr,1)
ppr=lag(ppr,1)
dn=lag(dn,1)

out.sbt=lm(resst0ta^2 ~ dn)
summary(out.sbt)
```

```
##
## Call:
## lm(formula = resst0ta^2 ~ dn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1379 -0.8891 -0.6612  0.1223 26.8684
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.90077    0.04587  19.638  < 2e-16 ***
## dnTRUE       0.23709    0.06689   3.545 0.000398 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.965 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.003616,   Adjusted R-squared:  0.003328
## F-statistic: 12.56 on 1 and 3462 DF,  p-value: 0.0003983
```

```
out.nsbt=lm(resst0ta^2 ~ nnr)
summary(out.nsbt)
```

```
##
## Call:
## lm(formula = resst0ta^2 ~ nnr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2957 -0.9438 -0.6815  0.1401 26.7701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.99906    0.03686  27.108   <2e-16 ***
## nnr         -0.03078    0.03612  -0.852    0.394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.968 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0002096,  Adjusted R-squared:  -7.919e-05
## F-statistic: 0.7258 on 1 and 3462 DF,  p-value: 0.3943
```

```
out.psbt=lm(resst0ta^2 ~ ppr)
summary(out.psbt)
```

```
##
## Call:
## lm(formula = resst0ta^2 ~ ppr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0728 -0.9320 -0.6596  0.1508 26.9723
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.07280    0.03782  28.368  < 2e-16 ***
## ppr         -0.14100    0.04136  -3.409  0.00066 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.965 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.003345,   Adjusted R-squared:  0.003058
## F-statistic: 11.62 on 1 and 3462 DF,  p-value: 0.0006597
```

```
out.jsbt=lm(resst0ta^2 ~ dn + nnr + ppr)
summary(out.jsbt)
```

```
##
## Call:
## lm(formula = resst0ta^2 ~ dn + nnr + ppr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1755 -0.9026 -0.6511  0.1352 26.9718
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.97386    0.06001  16.228   <2e-16 ***
## dnTRUE       0.20258    0.08599   2.356   0.0185 *
## nnr          0.04231    0.04140   1.022   0.3069
## ppr         -0.09019    0.04777  -1.888   0.0591 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.964 on 3460 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.004942,   Adjusted R-squared:  0.004079
## F-statistic: 5.728 on 3 and 3460 DF,  p-value: 0.0006606
```

```
out.sbt=lm(resstgta^2 ~ dn)
summary(out.sbt)
```

```
##
```

```
## Call:
## lm(formula = resstgta^2 ~ dn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.076 -0.915 -0.660  0.148 45.573
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.94363    0.04907  19.230   <2e-16 ***
## dnTRUE       0.13250    0.07156   1.852   0.0642 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.102 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0009894,  Adjusted R-squared:  0.0007008
## F-statistic: 3.429 on 1 and 3462 DF,  p-value: 0.06416
```

```
out.nsbt=lm(resstgta^2 ~ nnr)
summary(out.nsbt)
```

```
##
## Call:
## lm(formula = resstgta^2 ~ nnr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.023 -0.929 -0.665  0.148 45.493
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.02344    0.03937  25.992   <2e-16 ***
## nnr          0.04080    0.03859   1.057     0.29
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.103 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0003228,  Adjusted R-squared:  3.403e-05
## F-statistic: 1.118 on 1 and 3462 DF,  p-value: 0.2905
```

```
out.psbt=lm(resstgta^2 ~ ppr)
summary(out.psbt)
```

```
##
## Call:
## lm(formula = resstgta^2 ~ ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.040 -0.925 -0.659  0.145 45.630
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.03953    0.04045  25.697   <2e-16 ***
## ppr         -0.07823    0.04425  -1.768   0.0771 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.102 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0009023,  Adjusted R-squared:  0.0006137
## F-statistic: 3.126 on 1 and 3462 DF,  p-value: 0.07712
```

```
out.jsbt=lm(resstgta^2 ~ dn + nnr + ppr)
summary(out.jsbt)
```

```
##
## Call:
## lm(formula = resstgta^2 ~ dn + nnr + ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.165 -0.906 -0.650  0.145 45.630
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.98387    0.06419  15.328   <2e-16 ***
## dnTRUE       0.18359    0.09198   1.996   0.0460 *
## nnr          0.10014    0.04428   2.262   0.0238 *
## ppr         -0.04965    0.05110  -0.972   0.3313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.101 on 3460 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.002736,   Adjusted R-squared:  0.001871
## F-statistic: 3.164 on 3 and 3460 DF,  p-value: 0.02355
```

```
out.sbt=lm(resstata^2 ~ dn)
summary(out.sbt)
```

```
##
## Call:
## lm(formula = resstata^2 ~ dn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.006 -0.916 -0.626  0.155 54.240
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.97961    0.04894  20.016   <2e-16 ***
## dnTRUE       0.02595    0.07137   0.364    0.716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.096 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  3.819e-05,  Adjusted R-squared:  -0.0002506
## F-statistic: 0.1322 on 1 and 3462 DF,  p-value: 0.7161
```

```
out.nsbt=lm(resstata^2 ~ nnr)
summary(out.nsbt)
```

```
##
## Call:
## lm(formula = resstata^2 ~ nnr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.013 -0.913 -0.630  0.162 54.207
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.01319    0.03925  25.814   <2e-16 ***
## nnr          0.04985    0.03847   1.296    0.195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.096 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0004848,  Adjusted R-squared:  0.0001961
## F-statistic: 1.679 on 1 and 3462 DF,  p-value: 0.1951
```

```
out.psbt=lm(resstata^2 ~ ppr)
summary(out.psbt)
```

```
##
## Call:
## lm(formula = resstata^2 ~ ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.005 -0.914 -0.625  0.158 54.276
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.00540    0.04034  24.921   <2e-16 ***
## ppr         -0.03165    0.04412  -0.717    0.473
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.096 on 3462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0001486,  Adjusted R-squared:  -0.0001402
## F-statistic: 0.5146 on 1 and 3462 DF,  p-value: 0.4732
```

```
out.jsbt=lm(resstata^2 ~ dn + nnr + ppr)
summary(out.jsbt)
```

```
##
## Call:
## lm(formula = resstata^2 ~ dn + nnr + ppr)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.072 -0.909 -0.623  0.167 54.276
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.00516    0.06404  15.695   <2e-16 ***
## dnTRUE       0.06863    0.09177   0.748   0.4546
## nnr          0.07482    0.04418   1.693   0.0905 .
## ppr         -0.03153    0.05098  -0.618   0.5363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.096 on 3460 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0009767,  Adjusted R-squared:  0.0001105
## F-statistic: 1.128 on 3 and 3460 DF,  p-value: 0.3365
```

It is noticeable that using the GJR I reject in the joint test while looking at the aparch I never reject and therefore seem to capture the asymmetry in variance well.

```
ICall<-cbind(infocriteria(fit0n),infocriteria(fitgn),infocriteria(fitan), infocriteria(fit0ta),infocrit
colnames(ICall)<-c("GARCH N","GJR N","APARCH N","GARCH TA","GJR TA","APARCH TA")
ICall_df <- data.frame(ICall)
print(ICall_df)
```

```
##                GARCH.N    GJR.N APARCH.N GARCH.TA   GJR.TA APARCH.TA
## Akaike        2.811885 2.788392 2.773225 2.750603 2.721766  2.705060
## Bayes         2.818985 2.797267 2.783875 2.761253 2.734191  2.717485
## Shibata       2.811882 2.788388 2.773219 2.750597 2.721758  2.705052
## Hannan-Quinn 2.814420 2.791561 2.777028 2.754406 2.726203  2.709497
```

The APARCH with skewed-t distribution appears to be the best according to information criteria.

3.4 "Proceed to build one-step-ahead forecasts for the period from April 2022 to March 2023 (one year) with all the GARCH specifications from the previous point (at least 6). Evaluate any differences in the obtained forecasts by conducting a Diebold-Mariano test among all possible pairs of models. Are you able to achieve a complete ranking among the models?"

```
p_VT = log(window(VTd, end="2023-03-31")) #log-prices
p_VT = as.vector(p_VT$VT.Adjusted[,1])
T = length(p_VT)
r_VT=100*(p_VT[2:T]-p_VT[1:T-1]) #log returns %
y = r_VT - mean(r_VT)

# GARCH(1,1) Normal
for0n = ugarchroll(spec0n,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="movi
for0nd = as.data.frame(for0n)
s0nf = for0nd$Sigma

# GARCH(1,1) skew-t
for0ta = ugarchroll(spec0ta,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="mo
for0tad = as.data.frame(for0ta)
s0taf = for0tad$Sigma

# GJRGARCH(1,1) Normal
forgn = ugarchroll(specgn,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="movi
forgnd = as.data.frame(forgn)
sgnf = forgnd$Sigma

# GJRGARCH(1,1) skew-t
forgta = ugarchroll(specgta,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="mo
forgtad = as.data.frame(forgta)
sgtaf = forgtad$Sigma

# APARCH(1,1) Normal
foran = ugarchroll(specan,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="movi
forand = as.data.frame(foran)
sanf = forand$Sigma

# APARCH(1,1) skew-t
forata = ugarchroll(specata,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="mo
foratad = as.data.frame(forata)
sataf = foratad$Sigma
```

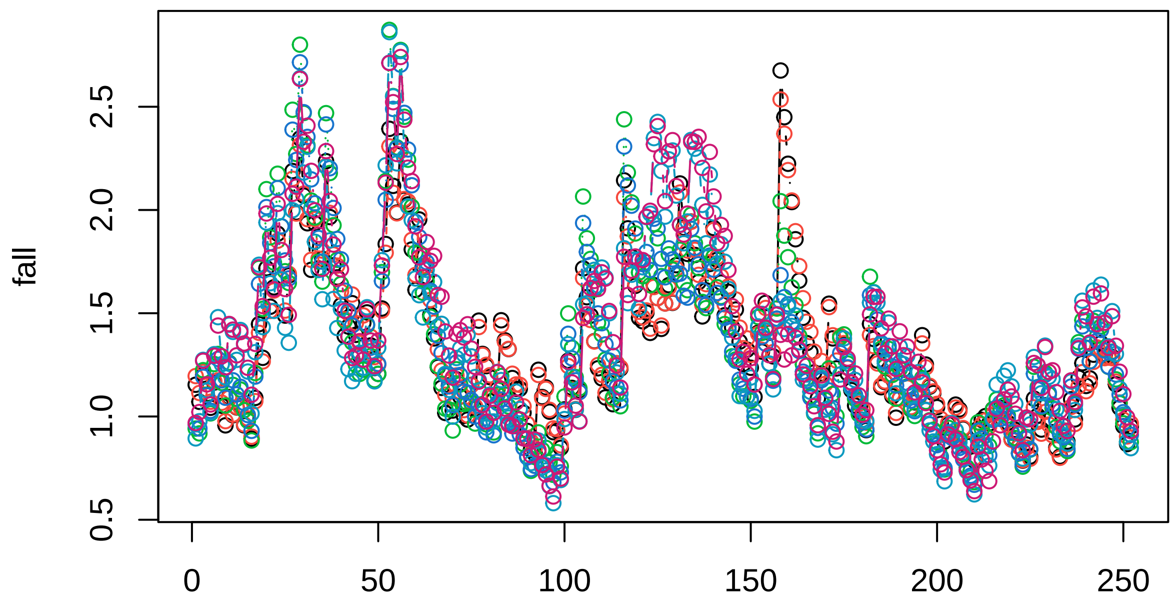
`ugarchroll` performs a rolling-window forecast, meaning it repeatedly makes a forecast by "scrolling" through the observation window, with a length specified in `window.size` and a scrolling interval specified in `refit.every`.

The `refit.every` parameter specifies the number of observations after which the model is refitted to the data. For example, if `refit.every = 5` and the window length (`window.size`) is 1000, the model will be refitted to the data each time the window moves by 5 observations, i.e., each time the window contains observations 1-1000, 6-1005, 11-1010, and so on.

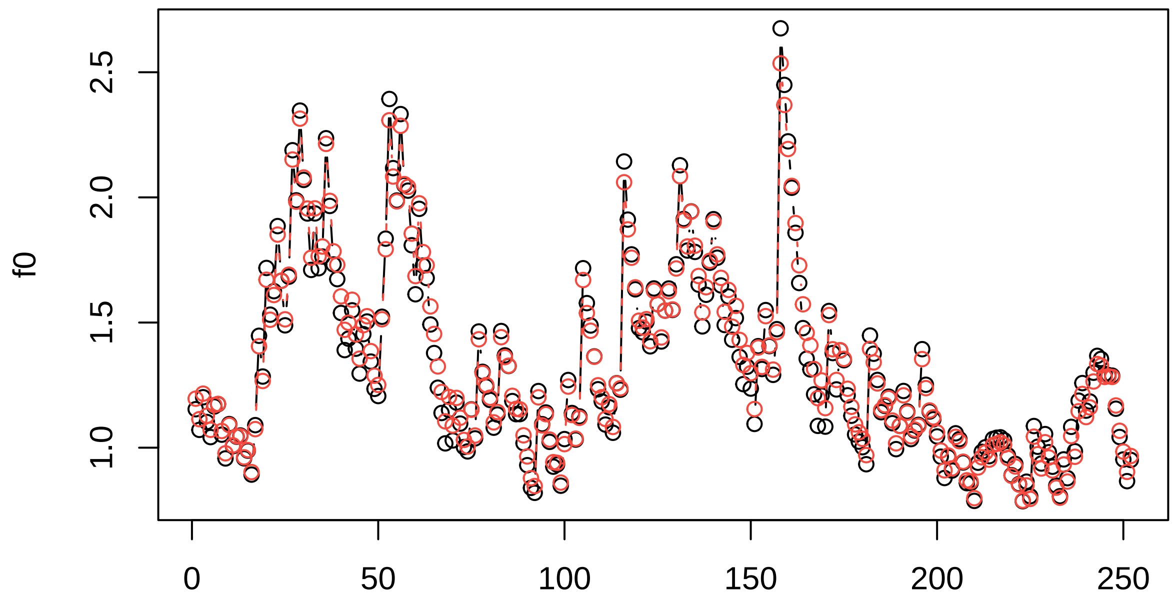
The forecast for a forecast horizon of 252 periods (one year) is specified in `forecast.length`.

This code generates several graphs that show the variance forecasts obtained from the different specified models, comparing the various models with each other. This graphical comparison is crucial for visually assessing the predictive performance and stability of each model across the forecasting horizon.

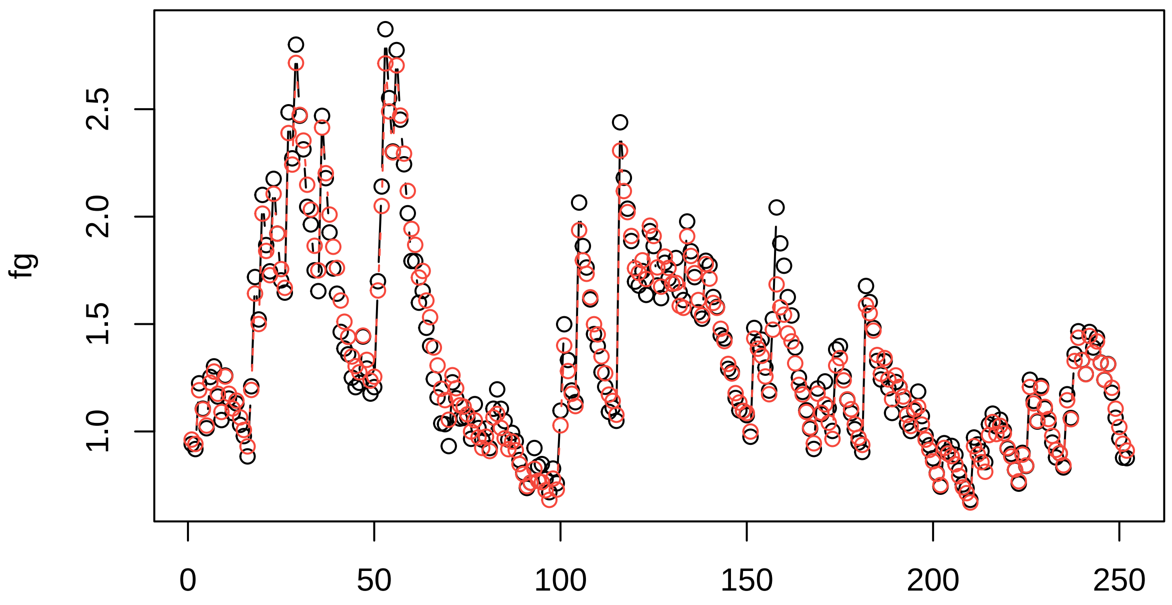
```
fall = cbind(s0nf,s0taf,sgnf,sgtaf,sanf,sataf)
par(mfrow=c(1,1))
matplot(fall, type = c("b"),pch=1,col = 1:6)
```



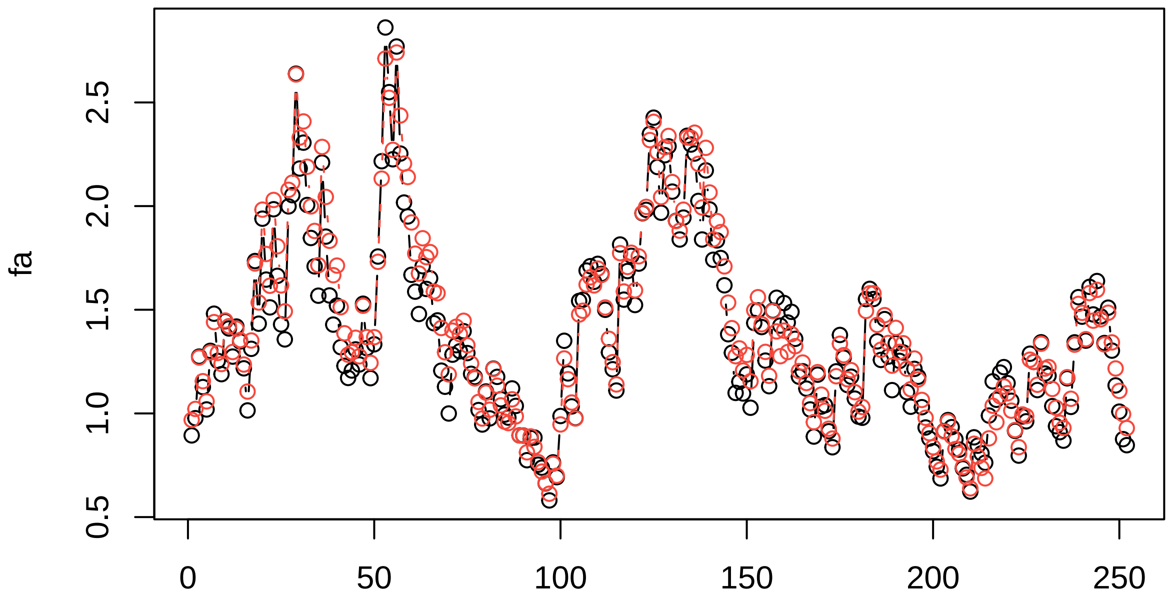
```
f0 = cbind(s0nf,s0taf)
fg = cbind(sgnf,sgtaf)
fa = cbind(sanf,sataf)
matplot(f0, type=('b'),pch=1,col=1:2)
```



```
matplot(fg, type=c('b'),pch=1,col=1:2)
```



```
matplot(fa, type=c('b'),pch=1,col=1:2)
```



Let $s0nf$ be the historical series of forecasted variances from the GARCH(1,1) model with a normal distribution. Let rf be the historical series of realized volatilities.

The calculated measures such as lN and others are a historical series calculated as the square of the difference between the forecasted variance and the realized volatility, raised to the square.

In other words, these are the squared errors between the model's forecasts and the realized volatilities. This method quantifies the accuracy of the forecasts by measuring how closely the model's variance predictions match the actual observed volatilities in the market, providing a clear metric to assess model performance.

```
rf = for0nd$Realized

lN = (s0nf^2 -rf^2)^2 # GARCH
lTA= (s0taf^2 -rf^2)^2 # GARCH skew-t
lGN = (sgnf^2 -rf^2)^2 # GJRGARCH
lGTA = (sgtaf^2 -rf^2)^2 # GJRGARCH skew-t
```

```
lAN = (sanf^2 -rf^2)^2 # GJRGARCH t
lATA = (sataf^2 -rf^2)^2 # APARCH
```

I calculate the differences:

```
dNTA = lN-lTA
dNGN = lN-lGN
dNGTA = lN-lGTA
dNAN = lN-lAN
dNATA = lN-lATA

dTAGN = lTA-lGN
dTAGTA = lTA-lGTA
dTAAN = lTA-lAN
dTAATA = lTA-lATA

dGNGTA = lGN-lGTA
dGNAN = lGN-lAN
dGNATA = lGN-lATA

dGTAAN = lGTA-lAN
dGTAATA = lGTA-lATA

dANATA = lAN-lATA
```

The function “NeweyWest” estimates the covariance matrices of model parameters using the Newey-West method with a rolling window, with a lag equal to the variable `m` defined earlier.

In this case, the outputs given by the function are scalar values and not matrices because we are estimating only one parameter.

```
m<-floor(0.75*((NROW(rf))^(1/3)))

x1<-as.vector(matrix(1,nrow=NROW(rf)))

VNTA = NeweyWest(lm(dNTA~x1-1),lag=m,prewhite=0)
VNGN = NeweyWest(lm(dNGN~x1-1),lag=m,prewhite=0)
VNGTA = NeweyWest(lm(dNGTA~x1-1),lag=m,prewhite=0)
VNAN = NeweyWest(lm(dNAN~x1-1),lag=m,prewhite=0)
VNATA = NeweyWest(lm(dNATA~x1-1),lag=m,prewhite=0)

VTAGN = NeweyWest(lm(dTAGN~x1-1),lag=m,prewhite=0)
VTAGTA = NeweyWest(lm(dTAGTA~x1-1),lag=m,prewhite=0)
VTAAN = NeweyWest(lm(dTAAN~x1-1),lag=m,prewhite=0)
VTAATA = NeweyWest(lm(dTAATA~x1-1),lag=m,prewhite=0)

VGNGTA = NeweyWest(lm(dGNGTA~x1-1),lag=m,prewhite=0)
VGNAN = NeweyWest(lm(dGNAN~x1-1),lag=m,prewhite=0)
VGNATA = NeweyWest(lm(dGNATA~x1-1),lag=m,prewhite=0)

VGTAAN = NeweyWest(lm(dGTAAN~x1-1),lag=m,prewhite=0)
VGTAATA = NeweyWest(lm(dGTAATA~x1-1),lag=m,prewhite=0)

VANATA = NeweyWest(lm(dANATA~x1-1),lag=m,prewhite=0)

DM<-matrix(0,nrow=6,ncol=6)
```

```
# Output -> robust variances regressors

# loss = model in row minus model in column
colnames(DM)<-c("GARCHnorm","GARCH TA","GJRnorm","GJR ta","APARCHnorm","APARCH TA")
rownames(DM)<-c("GARCHnorm","GARCH TA","GJRnorm","GJR ta","APARCHnorm","APARCH TA")

DM[1, 2] <- mean(dNTA) / sqrt(VNTA)
DM[1, 3] <- mean(dNGN) / sqrt(VNGN)
DM[1, 4] <- mean(dNGTA) / sqrt(VNGTA)
DM[1, 5] <- mean(dNAN) / sqrt(VNAN)
DM[1, 6] <- mean(dNATA) / sqrt(VNATA)
DM[2, 3] <- mean(dTAGN) / sqrt(VTAGN)
DM[2, 4] <- mean(dTAGTA) / sqrt(VTAGTA)
DM[2, 5] <- mean(dTAAN) / sqrt(VTAAN)
DM[2, 6] <- mean(dTAATA) / sqrt(VTAATA)
DM[3, 4] <- mean(dGNGTA) / sqrt(VGNGTA)
DM[3, 5] <- mean(dGNAN) / sqrt(VGNAN)
DM[3, 6] <- mean(dGNATA) / sqrt(VGNATA)
DM[4, 5] <- mean(dGTAAN) / sqrt(VGTAAN)
DM[4, 6] <- mean(dGTAATA) / sqrt(VGTAATA)
DM[5, 6] <- mean(dANATA) / sqrt(VANATA)
DM
```

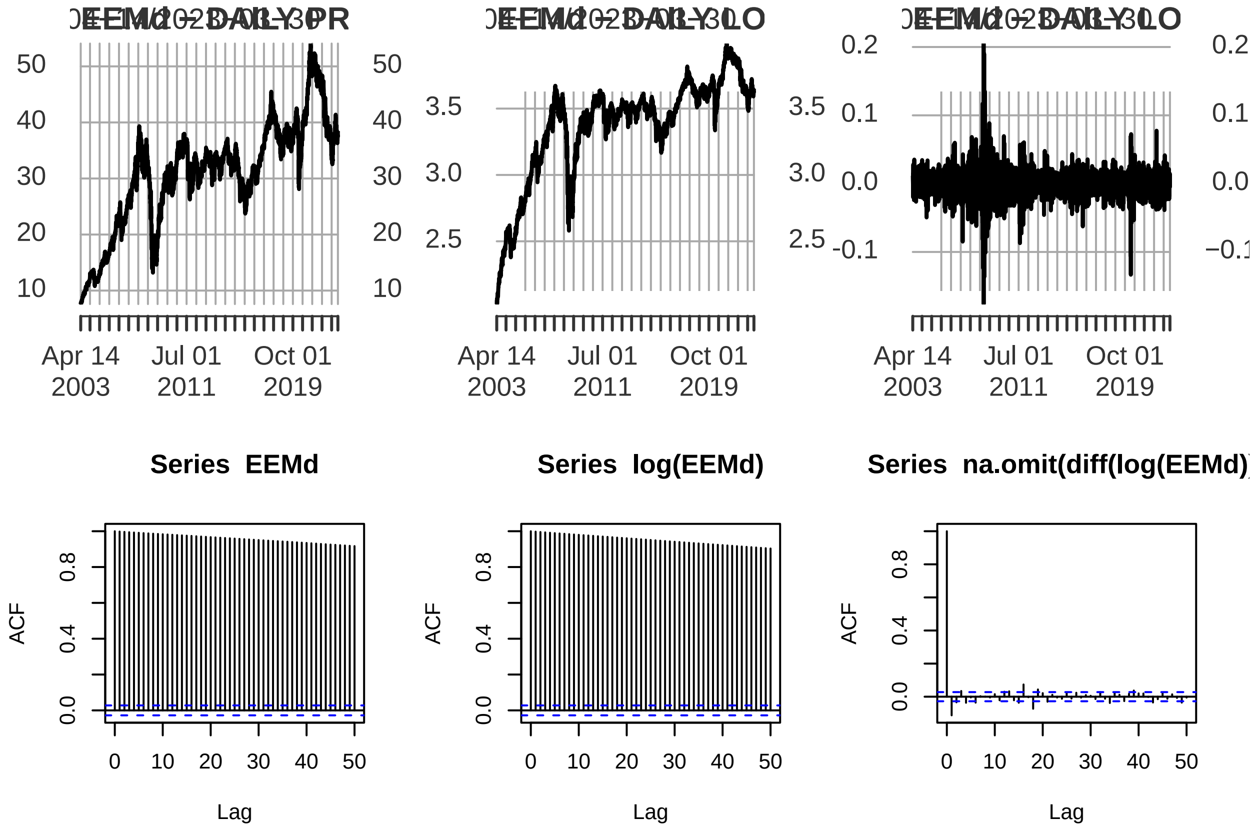
```
##            GARCHnorm GARCH TA   GJRnorm     GJR ta APARCHnorm  APARCH TA
## GARCHnorm          0  1.81984 -0.894206 -0.2934257 -0.2771301 -0.2281993
## GARCH TA           0  0.00000 -1.114441 -0.5442184 -0.5006893 -0.4511082
## GJRnorm            0  0.00000  0.000000  2.6412092  0.8022290  0.8282550
## GJR ta             0  0.00000  0.000000  0.0000000 -0.0307680  0.0391854
## APARCHnorm         0  0.00000  0.000000  0.0000000  0.0000000  0.1971203
## APARCH TA          0  0.00000  0.000000  0.0000000  0.0000000  0.0000000
```

The APARCH model with skewed-t distribution is preferable over 3 of the other models but not from a statistical standpoint. The only value that seems to be statistically significant is the superiority of the GJR with skewed-t distribution compared to the GJR with a normal distribution.

4. EEM

4.1 “Graphically represent the returns and briefly describe the stylized facts that characterize them, highlighting any differences between the financial instruments considered. Use the statistical tools you deem most appropriate for these analyses, necessarily including the verification of the possible presence of heteroscedasticity.”

```
par(mfrow=c(2,3))
plot(EEMd, main='EEMd - DAILY PRICES')
plot(log(EEMd), main='EEMd - DAILY LOG PRICES')
plot(diff(log(EEMd)), main='EEMd - DAILY LOG RETURN')
acf(EEMd, lag=50)
acf(log(EEMd), lag=50)
acf(na.omit(diff(log(EEMd))), lag=50)
```



In the figure just above, the graphical analyses of the series are provided (from left: daily prices, daily log-prices, and daily log-returns) and the corresponding correlograms. A strong upward trend of the EEM ETF can be identified in the first part of the time interval considered, while subsequently, although present, the upward trend is limited.

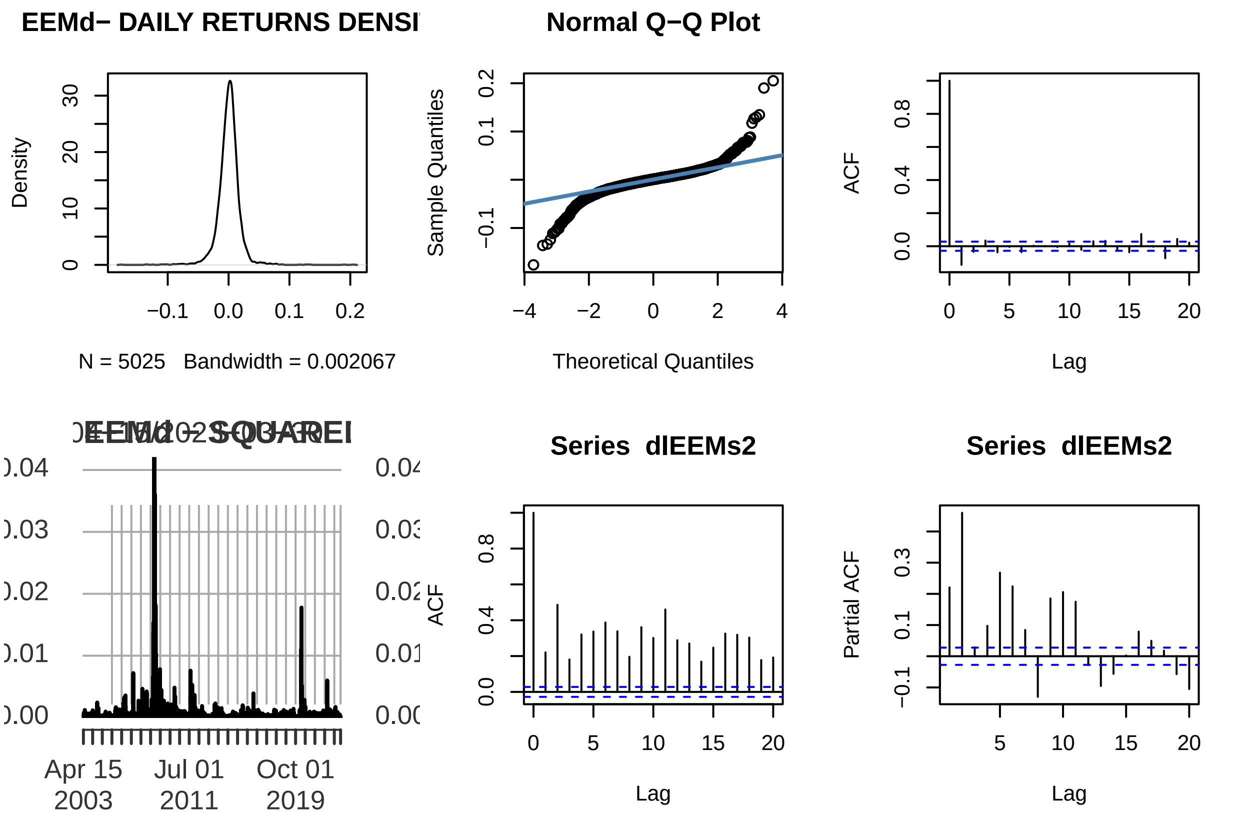
It can also be noted a clustering behavior of volatility (observable, more clearly, on the series of log-returns), where periods of high volatility are followed by periods of low volatility. As for the global autocorrelations (ACF), as per theory, the hypotheses of a clear evidence of non-stationarity of the stochastic process are confirmed; while, in the log-returns, although there are some statistically significant values (due to the large amplitude of the series) these can be considered almost irrelevant from an economic standpoint. Thus, an uncorrelation of the data is verified, at least economically insignificant; which is however not confirmed by the Ljung-Box test. The Ljung-Box test indeed tests the uncorrelation of the data, and with a low p-value, we reject the null hypothesis of uncorrelation and conclude that at least statistically there is a form of serial dependence in the log returns.

```
dlEEMs = na.omit(diff(log(EEMd)))
outB = Box.test(dlEEMs,1:20)
round(outB$p.value, 5)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Analysis of deviation from normality

```
par(mfrow=c(2,3))
dlEEMs = na.omit(diff(log(EEMd)))
plot(density(dlEEMs), main='EEMd- DAILY RETURNS DENSITY')
qqnorm(dlEEMs,main="Normal Q-Q Plot",xlab="Theoretical Quantiles",ylab="Sample Quantiles")
qqline(dlEEMs,col="steelblue",lwd=2)
acf(dlEEMs,20,main='')
dlEEMs2=(dlEEMs)^2
plot(dlEEMs2, main='EEMd - SQUARED LOG.RET')
acf(dlEEMs2, 20)
pacf(dlEEMs2, 20)
```



In this case, some typical characteristics of a financial instrument that should be analyzed individually are represented in the figure: the empirical hypothesis of non-normality of the historical series is at least graphically confirmed, particularly due to strong kurtosis visible in the top-left graph showing very thick tails, and deviation from normality on the tails also visible in the qq-plot. The asymmetry, which is another empirical characteristic, for an ETF like EEM that contains many companies within it, seems almost approaching nullity. In the second group of figures, below, a proxy of the variance (i.e., the series of squared returns) is analyzed: also in this case, it can be observed that the levels of variability (on which subsequently models that can estimate it are to be built) are not always constant and indeed suggest a sort of serial dependence, quite clearly confirmed by the ACF of squared returns.

Test for the presence of unit roots

```
out1 = adfTest(log(EEMd), lag=21, type='c')
summary(out1@test$lm)
```

```
##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.167183 -0.007973  0.000878  0.008868  0.183188
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0079473  0.0022560   3.523 0.000431 ***
## y.lag.1      -0.0022450  0.0006638  -3.382 0.000725 ***
## y.diff.lag1  -0.1135892  0.0141501  -8.027 1.23e-15 ***
## y.diff.lag2  -0.0367008  0.0142369  -2.578 0.009970 **
## y.diff.lag3   0.0150796  0.0142412   1.059 0.289709
## y.diff.lag4  -0.0399370  0.0142166  -2.809 0.004986 **
## y.diff.lag5  -0.0107144  0.0142274  -0.753 0.451438
## y.diff.lag6  -0.0406461  0.0141996  -2.862 0.004221 **
## y.diff.lag7  -0.0074476  0.0142045  -0.524 0.600084
## y.diff.lag8  -0.0020000  0.0142034  -0.141 0.888023
## y.diff.lag9  -0.0052577  0.0141948  -0.370 0.711101
## y.diff.lag10  0.0097797  0.0141887   0.689 0.490692
## y.diff.lag11 -0.0153540  0.0141884  -1.082 0.279237
## y.diff.lag12  0.0293884  0.0141868   2.072 0.038360 *
## y.diff.lag13  0.0321080  0.0141915   2.262 0.023711 *
## y.diff.lag14 -0.0137184  0.0141977  -0.966 0.333969
## y.diff.lag15 -0.0319705  0.0141968  -2.252 0.024369 *
## y.diff.lag16  0.0639469  0.0141949   4.505 6.79e-06 ***
## y.diff.lag17  0.0042616  0.0142230   0.300 0.764476
## y.diff.lag18 -0.0608355  0.0142139  -4.280 1.90e-05 ***
## y.diff.lag19  0.0269323  0.0142384   1.892 0.058612 .
## y.diff.lag20  0.0252950  0.0142333   1.777 0.075602 .
## y.diff.lag21 -0.0207968  0.0141470  -1.470 0.141611
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01749 on 4981 degrees of freedom
## Multiple R-squared:  0.03523,    Adjusted R-squared:  0.03097
## F-statistic: 8.268 on 22 and 4981 DF,  p-value: < 2.2e-16
```

By removing one lag at a time and assessing the significance of the coefficient associated with the last delay in the linear regression model built during the ADF (Augmented Dickey-Fuller) test for stationarity of a time series, it is found that the lags to use are 20 and the intercept is significant.

```
out1 = adfTest(log(EEMd), lag=18, type='c')
summary(out1@test$lm)
```

```
##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.165890 -0.007968  0.000907  0.008802  0.180311
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0080258  0.0022482   3.570 0.000361 ***
## y.lag.1      -0.0022646  0.0006617  -3.422 0.000626 ***
## y.diff.lag1  -0.1150204  0.0141144  -8.149 4.59e-16 ***
## y.diff.lag2  -0.0390521  0.0142078  -2.749 0.006006 **
## y.diff.lag3   0.0180416  0.0141904   1.271 0.203646
## y.diff.lag4  -0.0389655  0.0141855  -2.747 0.006039 **
## y.diff.lag5  -0.0131690  0.0141941  -0.928 0.353567
## y.diff.lag6  -0.0395755  0.0141881  -2.789 0.005302 **
## y.diff.lag7  -0.0056590  0.0141930  -0.399 0.690120
## y.diff.lag8  -0.0023073  0.0141920  -0.163 0.870856
## y.diff.lag9  -0.0057763  0.0141875  -0.407 0.683924
## y.diff.lag10  0.0103640  0.0141864   0.731 0.465085
## y.diff.lag11 -0.0160980  0.0141866  -1.135 0.256543
## y.diff.lag12  0.0290975  0.0141875   2.051 0.040327 *
## y.diff.lag13  0.0311644  0.0141823   2.197 0.028036 *
## y.diff.lag14 -0.0148628  0.0141879  -1.048 0.294891
## y.diff.lag15 -0.0324182  0.0141793  -2.286 0.022278 *
## y.diff.lag16  0.0636181  0.0141869   4.484 7.48e-06 ***
## y.diff.lag17  0.0046034  0.0142038   0.324 0.745876
## y.diff.lag18 -0.0650525  0.0141122  -4.610 4.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0175 on 4987 degrees of freedom
## Multiple R-squared:  0.03348,    Adjusted R-squared:  0.02979
## F-statistic: 9.091 on 19 and 4987 DF,  p-value: < 2.2e-16
```

We then proceed to run the command without saving the output:

```r
adfTest(log(EEMd), lag=18, type='c')
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 18
##   STATISTIC:
##     Dickey-Fuller: -3.4223
##   P VALUE:
##     0.01084
##
## Description:
##  Fri Apr 19 16:27:55 2024 by user:
```

It is noted that, unlike the observed results, the null hypothesis of non-stationarity of the historical series, or that it is affected by a unit root, is rejected. Statistically, therefore, it cannot be proven that the prices are non-stationary.

By removing one lag at a time and assessing the significance of the coefficient associated with the last delay in the linear regression model built during the ADF (Augmented Dickey-Fuller) test for stationarity of a time series, it is found that the lags to use are 17 and the intercept is not significant.

```
out1 = adfTest(dlEEMs, lag=17, type='nc')
summary(out1@test$lm)
```

```
##
## Call:
## lm(formula = y.diff ~ y.lag.1 - 1 + y.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.164439 -0.007640  0.001203  0.009166  0.182150
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## y.lag.1      -1.211057   0.070711 -17.127  < 2e-16 ***
## y.diff.lag1   0.096531   0.068887   1.401  0.16119
## y.diff.lag2   0.058287   0.066940   0.871  0.38394
## y.diff.lag3   0.077317   0.064688   1.195  0.23206
## y.diff.lag4   0.039266   0.062479   0.628  0.52973
## y.diff.lag5   0.027131   0.060239   0.450  0.65244
## y.diff.lag6  -0.011306   0.057778  -0.196  0.84486
## y.diff.lag7  -0.015688   0.055070  -0.285  0.77576
## y.diff.lag8  -0.016718   0.052292  -0.320  0.74921
## y.diff.lag9  -0.021138   0.049301  -0.429  0.66813
## y.diff.lag10 -0.009372   0.046138  -0.203  0.83905
## y.diff.lag11 -0.024130   0.042745  -0.565  0.57244
## y.diff.lag12  0.006301   0.039082   0.161  0.87192
## y.diff.lag13  0.038676   0.035306   1.095  0.27338
## y.diff.lag14  0.024929   0.031170   0.800  0.42389
## y.diff.lag15 -0.006372   0.026719  -0.238  0.81153
## y.diff.lag16  0.058488   0.021153   2.765  0.00571 **
## y.diff.lag17  0.064131   0.014125   4.540 5.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01752 on 4989 degrees of freedom
## Multiple R-squared:  0.5647, Adjusted R-squared:  0.5631
## F-statistic: 359.5 on 18 and 4989 DF,  p-value: < 2.2e-16
```

```
adfTest(dlEEMs, lag=17, type='nc')
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 17
##   STATISTIC:
##     Dickey-Fuller: -17.1269
##   P VALUE:
##     0.01
```

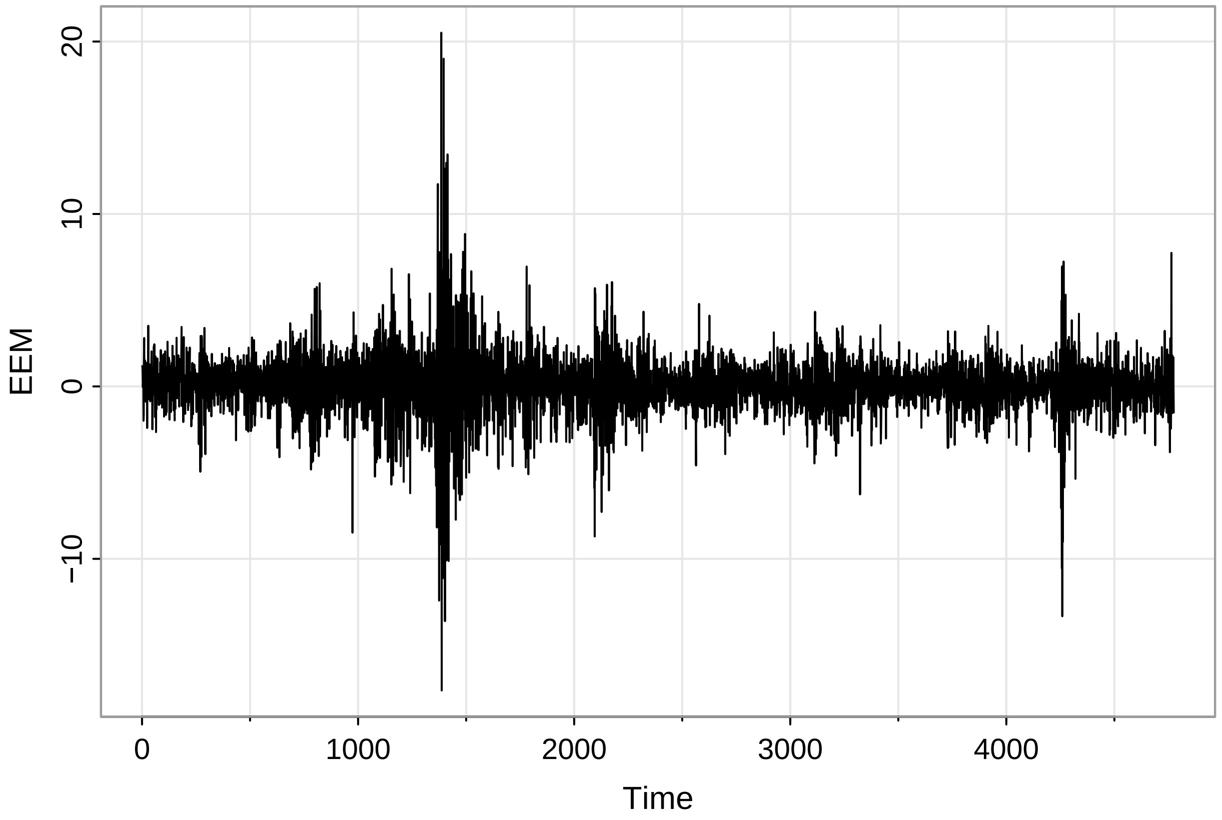
```
##
## Description:
##  Fri Apr 19 16:27:56 2024 by user:
```

As expected in this case, the null hypothesis is rejected.

The goal of the analysis is to verify whether heteroscedasticity is present or not, and if it allows for the application of a model subsequently.

The log-prices are calculated below up to March 31, 2022, and the percentage log returns.

```
p_EEM = log(window(EEMd, end="2022-03-31")) #log-prices
p_EEM = as.vector(p_EEM$EEM.Adjusted[,1])
T = length(p_EEM)
r_EEM=100*(p_EEM[2:T]-p_EEM[1:T-1]) #log percentage returns
par(mfrow=c(1,1))
tsplot(r_EEM, ylab='EEM') # plot of log percentage returns
```

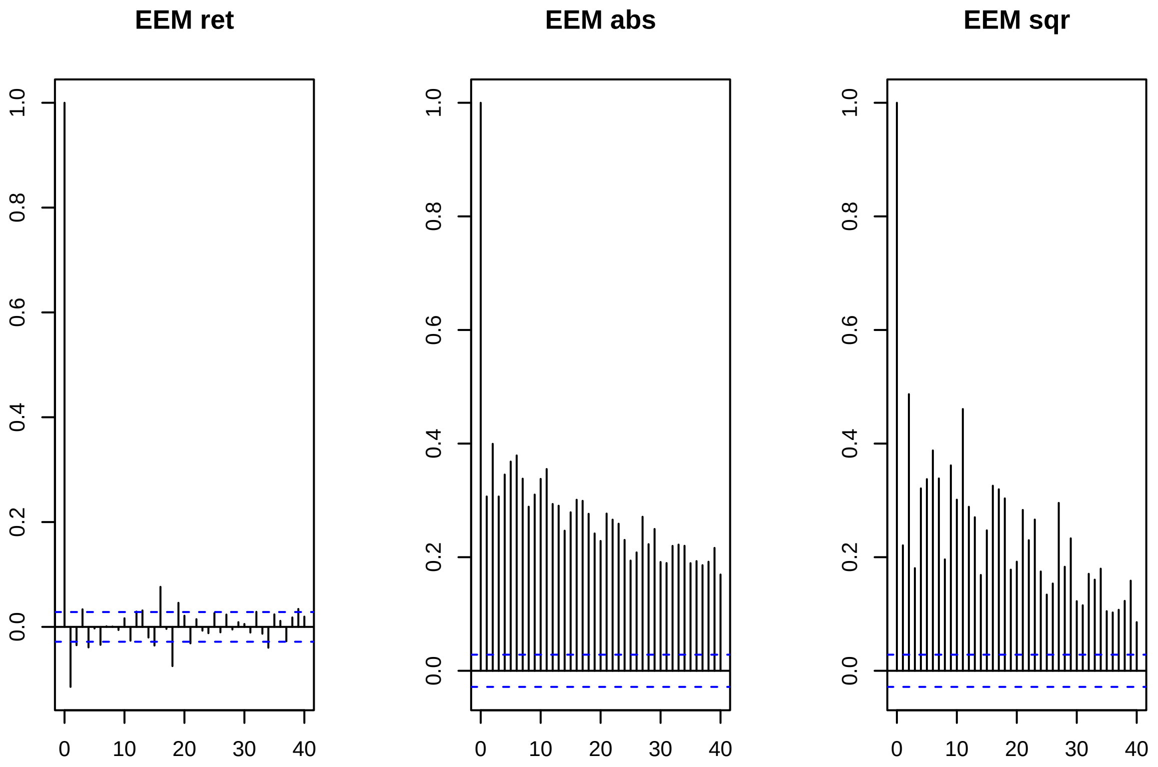


As already noted earlier, the effect of volatility clustering is present, with phases of high volatility followed by periods of stability.

The ACFs on returns, absolute values, and squares are then shown. Serial dependence is noted on the absolute values and their squares. On the returns, it is observed that some values exceed the bands which however suffer from sample dependence (with 4776 observations they are indeed very small).

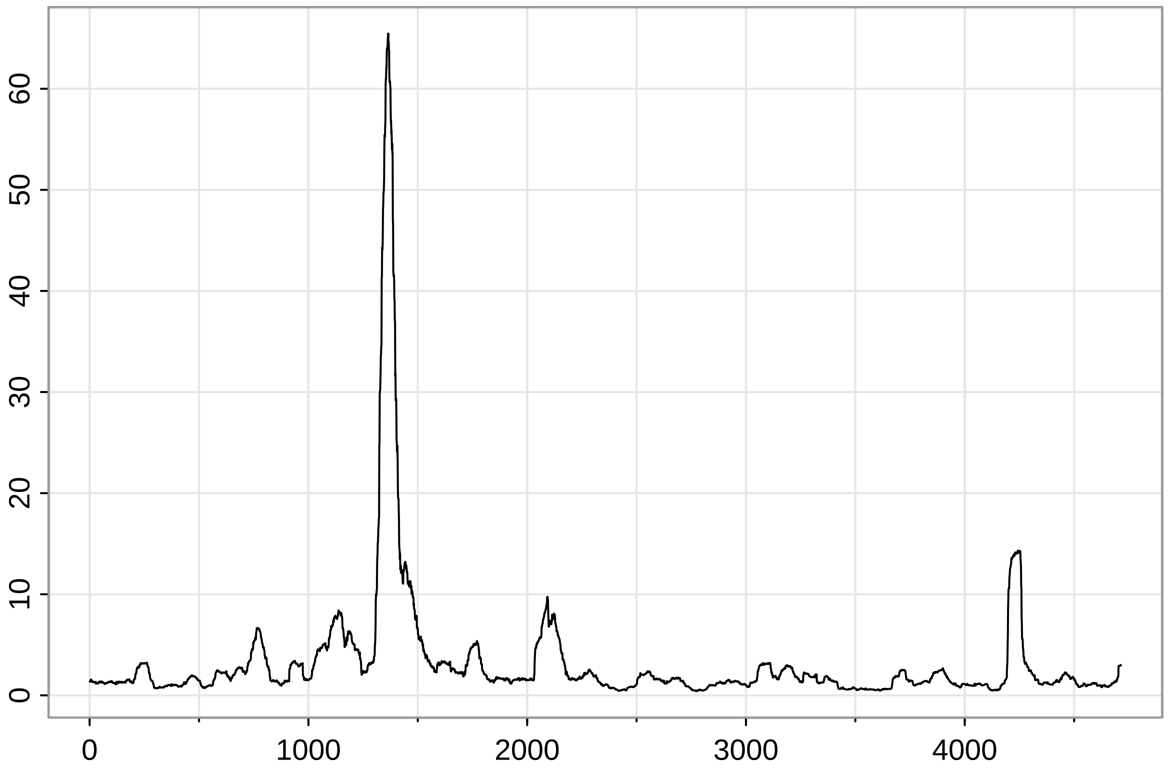
If we focus on absolute values and squares, a strong serial correlation emerges, strong even up to 40 lags, indicating significant persistence. This behavior is typically common to observe in financial instruments that exhibit certain characteristics, while instead with securities that fluctuate a lot, or that are less liquid, we might find little serial correlation.

```
par(mfrow=c(1,3))
acf(r_EEM,lag.max=40,xlab="",ylab="",main="EEM ret")
acf(abs(r_EEM),lag.max=40,xlab="",ylab="",main="EEM abs")
acf(abs(r_EEM)^2,lag.max=40,xlab="",ylab="",main="EEM sqr")
```

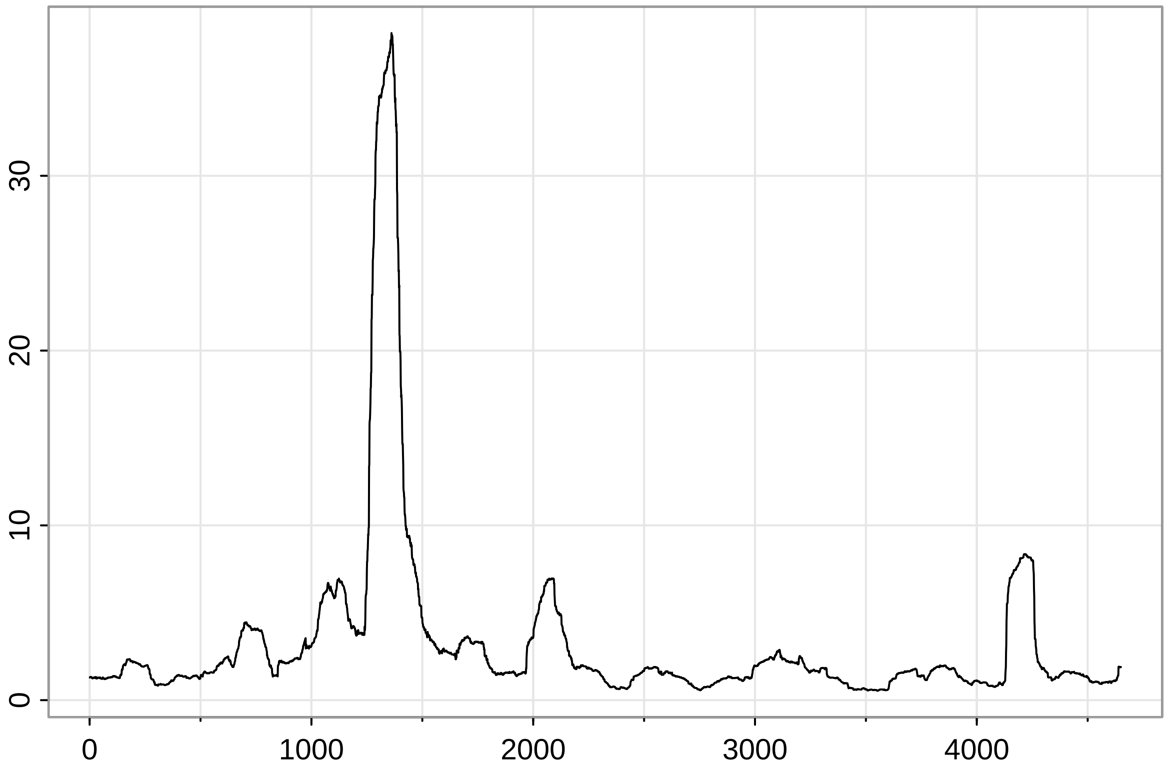


One of the first elements to assess heteroscedasticity and understand how it translates in terms of variances is the rolling variance. Using 62 observations we have a window of more or less the duration of a quarter while with 126 of a semester.

```
par(mfrow=c(1,1))
rollV=rollStats(as.timeSeries(r_EEM),62,FUN=var)
tsplot(rollV,ylab="",xlab="")
```



```
rollV=rollStats(as.timeSeries(r_EEM),126,FUN=var)
tsplot(rollV,ylab="",xlab="")
```



We notice that the historical series as expected is characterized by heteroscedasticity; more smoothed trends can be seen on the graph with 126 observations due to the fact that the addition of an observation changes less than 1% of the sample.

We also accompany the analysis with statistical nature tests: LJUNG BOX AND ARCH TESTS

We expect to reject the null hypothesis and thus the rejection that the autocorrelation functions are all zero up to lag 5 or 10.

The test statistic is a chi-square, with degrees of freedom depending on the lags considered.

```r
y2 = (r_EEM - mean(r_EEM))^2 # returns squared, deviation from the mean
Box.test(y2, lag = 5, type = "Ljung-Box")
```

Ljung-Box Test at lags 5 and 10

```
##
##  Box-Ljung test
##
## data:  y2
## X-squared = 2578.1, df = 5, p-value < 2.2e-16
```

```r
Box.test(y2, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  y2
## X-squared = 5103.3, df = 10, p-value < 2.2e-16
```

As expected, the p-value is low and the null hypothesis of uncorrelation is rejected.

LM-ARCH Test We now proceed with the LM-ARCH test implementing the auxiliary regression. The test assesses the null hypothesis of homoscedasticity against the alternative hypothesis of heteroscedasticity. If the test statistic exceeds a certain critical threshold, the null hypothesis of no heteroscedasticity is rejected and we move towards the alternative hypothesis of the presence of heteroscedasticity.

The code below considers 1, 2, and 3 lags for the test:

```r
# consider 1, 2, and 3 lags for the LM test
y2=as.timeSeries(y2)
y2L1 <- lag(y2,1)
y2L2 <- lag(y2,2)
y2L3 <- lag(y2,3)
# three test statistics and corresponding P-values
#
# First lag:
out1 <- lm(y2 ~ y2L1)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
# using T-2 because we lose some observations in lags 1,2,3 and for comparison, I choose 2
pchisq(LMARCH1,1,lower.tail=FALSE)# test only on the upper tail
```

```
## [1] 6.035835e-53
```

```r
# Second lag:
out2 <- lm(y2 ~ y2L1 + y2L2)
sum2 <- summary(out2)
```

```
LMARCH2 <- sum2$r.squared*(T-2)
pchisq(LMARCH2,2,lower.tail=FALSE)
```

```
## [1] 5.707643e-262
```

```
# Third lag:
out3 <- lm(y2 ~ y2L1 + y2L2 + y2L3)
sum3 <- summary(out3)
LMARCH3 <- sum3$r.squared*(T-2)
pchisq(LMARCH3,3,lower.tail=FALSE)
```

```
## [1] 4.21478e-261
```

In all three cases, the null hypothesis is rejected and it is concluded that there is heteroscedasticity in the historical series considered.

Similarities/Differences with the VT instrument

For both instruments, the tails are heavy, so the normal distribution seems inappropriate. Moreover, asymmetry seems present but not very high, indicative of the fact that ETFs are composed of many stocks.

In both instruments, the phenomenon of volatility clustering is present. For both instruments, the presence of homoscedasticity is rejected, and thus evaluations will proceed that model the present heteroscedasticity.

4.2 "Fit a GARCH(1,1) model with a normal distribution to the returns. The sample used for estimation should go up to the end of March 2022, leaving a year for subsequent analysis. Comment on the differences in fit to the data between the two considered securities, evaluating the presence of heteroscedasticity in the standardized residuals, the presence of asymmetry, and the appropriateness of the distribution choice for the innovations."

GARCH(1,1) Model

We now want to fit a GARCH(1,1) model capable of capturing the heteroscedasticity present in the historical series.

The GARCH(1,1) model assumes that the volatility of the financial asset partly depends on the volatility of the previous time interval and on an error term.

Specifically, the formula for conditional volatility at time t is:

$$\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2$$

where σ_t^2 is the volatility conditional on time t, ω represents the baseline level of volatility, α is the weight of the squared residual at time $t-1$, and β is the weight of the conditional volatility at time $t-1$.

Model specification:

```
spec0<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)), mean.model = list(arma
# estimate model saving output
fit0<-ugarchfit(spec0,r_EEM)
# estimate model with output on screen
fit0@fit$robust.matcoef
```

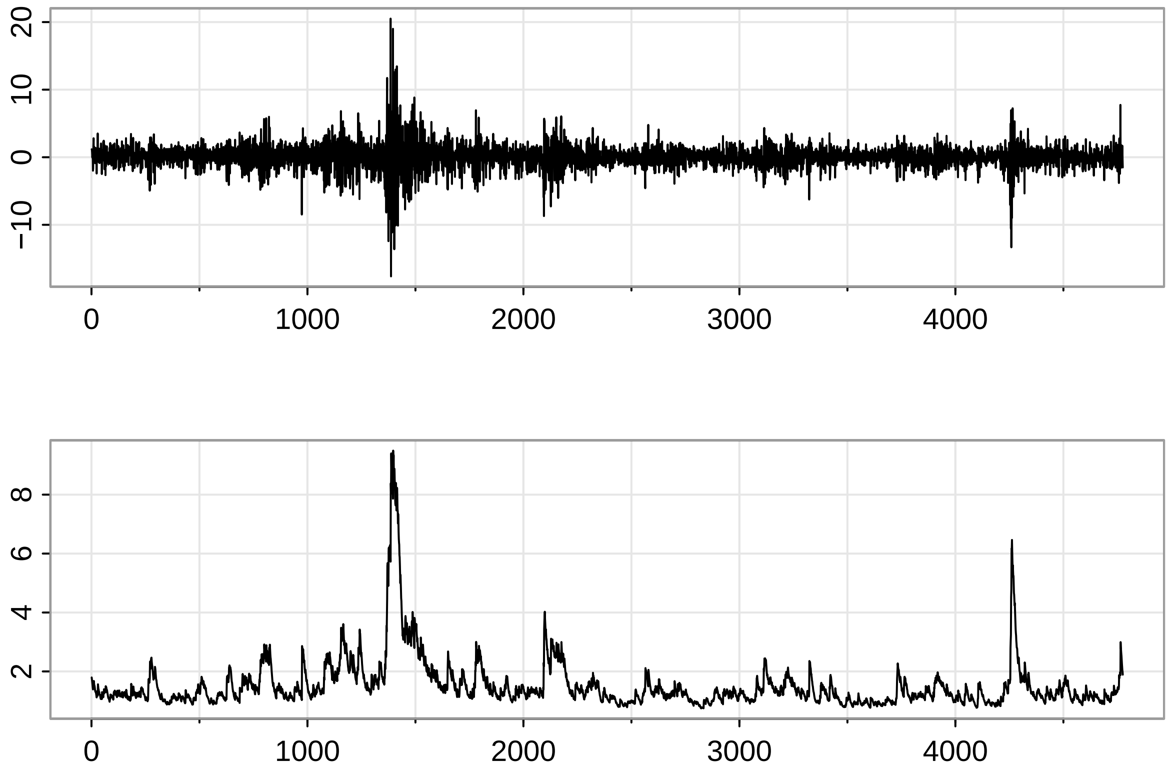
```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.06690372  0.01691650  3.954940 7.655397e-05
## omega  0.04355217  0.01051885  4.140395 3.467089e-05
## alpha1 0.09873422  0.01567990  6.296867 3.037226e-10
## beta1  0.88467088  0.01676817 52.758934 0.000000e+00
```

From the p-values calculated with the quasi-maximum likelihood method, it is noted that all parameters are statistically significant.

Beta assumes a value of 0.88 and is in the range of values expected for this parameter.

```
par(mfrow=c(2,1))
tsplot(r_EEM, type='l', ylab="", xlab="")
tsplot(fit0@fit$sigma, type='l',ylab="",xlab="")
```

Does the estimated model provide variances that have dynamics?



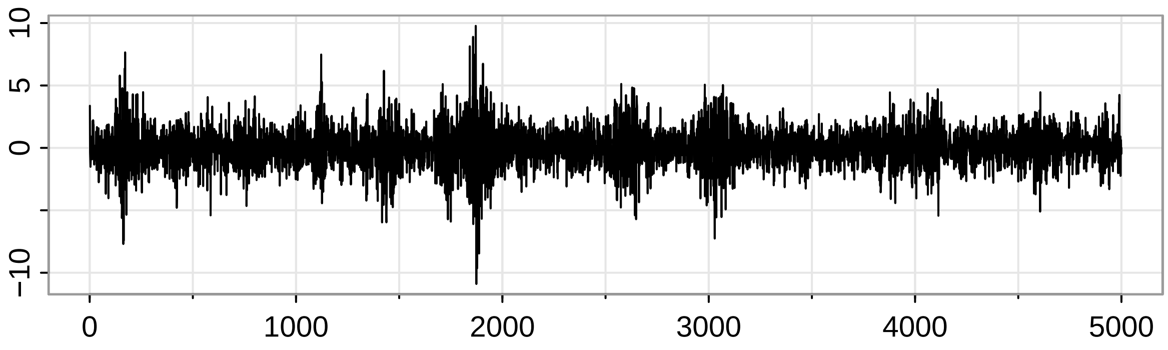
Graphic representation that combines returns with the sequence of estimated variances. The chart resembles variances obtained through rolling methods.

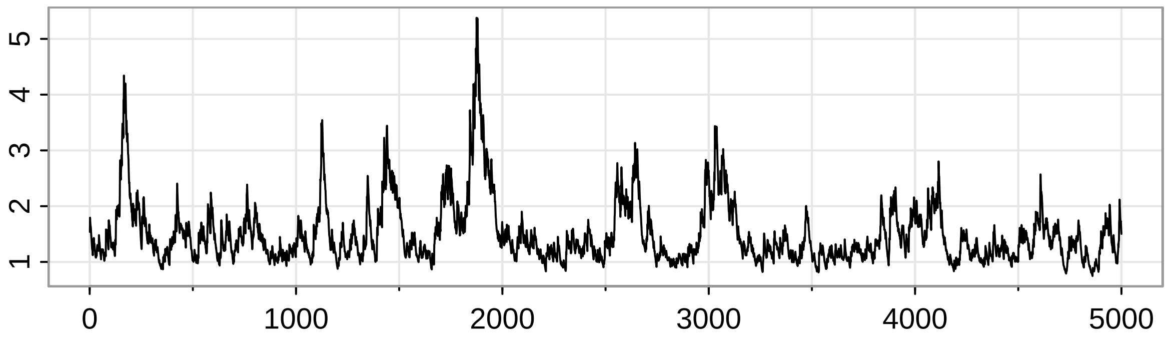
Does the chosen specification replicate the empirical characteristics of my series?

Simulation of a GARCH model and graphs Let's see if the estimated model replicates the stylized facts.

CONS:

```
sim0<-ugarchsim(fit0, n.sim = 5000)
par(mfrow=c(2,1))
tsplot(sim0@simulation$seriesSim,type="l",ylab="",xlab="")
tsplot(sim0@simulation$sigmaSim,type="l",ylab="",xlab="")
```





There is no noticeable presence of volatility clustering as there is visible high persistence of volatility. We have a frequency of extreme events that is quite high and does not deviate too much from the many other peaks of volatility present.

The model probably cannot replicate the presence of thick tails on the distributions, thus we need to reconsider the specification, changing the choice made for z and analyzing the results subsequently.

We will likely need to replace the normal distribution with another distribution.

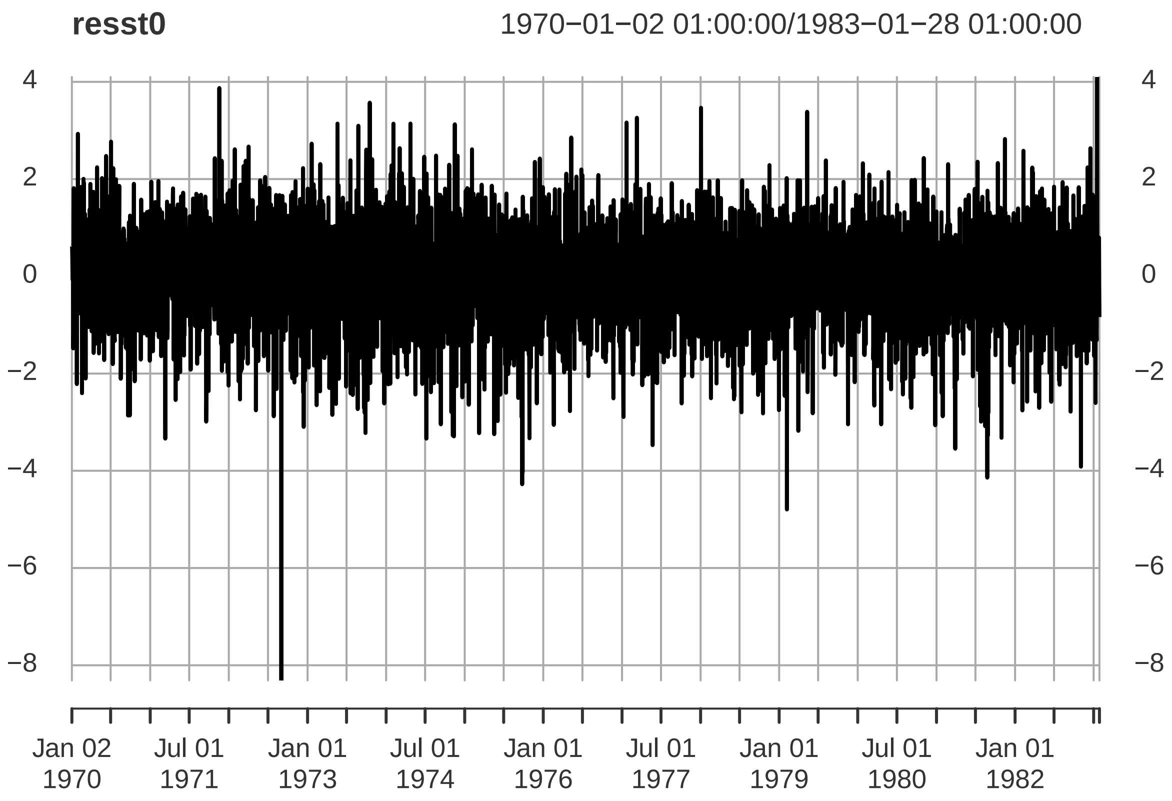
Analysis of Standardized Residuals: Having estimated a model, a variance dynamics and a distribution are hypothesized, which translates into the distribution of the quantity z.

I define the quantity z as the returns deviated from the mean and standardized by the conditional variance.

In the first case, the z's are independent and homoscedastic with unit variance, distributed as a normal.

I can perform analysis on standardized residuals:

```
resst0 <- residuals(fit0,standardize=TRUE)
plot(resst0)
```



```
# analysis for GARCH effects
y2 <- (as.timeSeries(resst0))^2
# consider 1, 2, and 3 lags for the LM test
y2L1 <- lag(y2,1)
y2L2 <- lag(y2,2)
y2L3 <- lag(y2,3)

# Evaluate the regression to understand if z_t^2 depends on its lags.
# If alpha = 0 then z_t is homoschedastic.
# If alpha is not 0 then z_t is heteroschedastic.
# Try with 1,2,3 lags, changing the test statistic.
# The null hypothesis is that the lags are jointly zero.
# If the test is correctly specified, I expect to have
# no further evidence of ARCH effects.
# three test statistics and corresponding P-values
out1 <- lm(y2 ~ y2L1)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1,1,lower.tail=FALSE)
```

```
## [1] 0.2585661
```

```
out1 <- lm(y2 ~ y2L1 + y2L2)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1,2,lower.tail=FALSE)
```

```
## [1] 0.01354163
```

```
out1 <- lm(y2 ~ y2L1 + y2L2 + y2L3)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1,3,lower.tail=FALSE)
```

```
## [1] 0.0322964
```

```
# In this case, we do not reject the null hypothesis, the coefficients are zero.
```

It's possible that the remaining heteroscedasticity can be captured by variance asymmetry, and therefore there may be an incorrect specification of the variance dynamics. Furthermore, it might also be that the GARCH(1,1) model is not suitable and an increase in the order of coefficients should be considered.

This choice is not explored in the following analyses but it is shown that the LM test never rejects the null hypothesis when increasing the order of coefficients.

However, it is noted that the β_1 coefficient in this case would be not significant.

```
spec22<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2, 2)), mean.model = list(arma
# estimate model saving output
fit22<-ugarchfit(spec22,r_EEM)
# estimate model with output on screen
fit22@fit$robust.matcoef
```

```
##          Estimate  Std. Error  t value     Pr(>|t|)
## mu     0.06809851  0.01677541 4.059424 4.919391e-05
## omega  0.07618370  0.02008091 3.793837 1.483373e-04
## alpha1 0.05972016  0.02000094 2.985868 2.827745e-03
## alpha2 0.10152065  0.02800839 3.624651 2.893516e-04
## beta1  0.37277125  0.23041506 1.617825 1.057003e-01
## beta2  0.43637500  0.21206108 2.057780 3.961128e-02
```

```
resst22 <- residuals(fit22,standardize=TRUE)
# analysis for GARCH effects
y2 <- (as.timeSeries(resst22))^2
# consider 1, 2, and 3 lags for the LM test
y2L1 <- lag(y2,1)
y2L2 <- lag(y2,2)
y2L3 <- lag(y2,3)

out1 <- lm(y2 ~ y2L1)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1,1,lower.tail=FALSE)
```

```
## [1] 0.7417816
```

```
out1 <- lm(y2 ~ y2L1 + y2L2)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
```

```
pchisq(LMARCH1,2,lower.tail=FALSE)
```

```
## [1] 0.5046876
```

```
out1 <- lm(y2 ~ y2L1 + y2L2 + y2L3)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1,3,lower.tail=FALSE)
```

```
## [1] 0.6125211
```

It is possible that the remaining heteroscedasticity can be captured by variance asymmetry, suggesting a mis-specification in the variance dynamics. Furthermore, the GARCH(1,1) model might not be suitable, and an increase in the order of coefficients may be needed.

This choice is not explored in the subsequent analyses, but it is shown that the LM test never rejects the null hypothesis when the order of coefficients is increased.

However, it is noted that the β_1 coefficient in this case would be not significant.

```
spec22<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2, 2)), mean.model = list(arma
# estimate model saving output
fit22<-ugarchfit(spec22, r_EEM)
# estimate model with output on screen
fit22@fit$robust.matcoef
```

```
##           Estimate  Std. Error  t value     Pr(>|t|)
## mu      0.06809851  0.01677541 4.059424 4.919391e-05
## omega   0.07618370  0.02008091 3.793837 1.483373e-04
## alpha1  0.05972016  0.02000094 2.985868 2.827745e-03
## alpha2  0.10152065  0.02800839 3.624651 2.893516e-04
## beta1   0.37277125  0.23041506 1.617825 1.057003e-01
## beta2   0.43637500  0.21206108 2.057780 3.961128e-02
```

```
resst22 <- residuals(fit22, standardize=TRUE)
# analysis for GARCH effects
y2 <- (as.timeSeries(resst22))^2
# consider 1, 2, and 3 lags for the LM test
y2L1 <- lag(y2, 1)
y2L2 <- lag(y2, 2)
y2L3 <- lag(y2, 3)

out1 <- lm(y2 ~ y2L1)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1, 1, lower.tail=FALSE)
```

```
## [1] 0.7417816
```

```
out1 <- lm(y2 ~ y2L1 + y2L2)
sum1 <- summary(out1)
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1, 2, lower.tail=FALSE)
```

```
## [1] 0.5046876
```

```
out1 <- lm(y2 ~ y2L1 + y2L2 + y2L3)
sum1 <- summary(out1)
```

```r
LMARCH1 <- sum1$r.squared*(T-2)
pchisq(LMARCH1, 3, lower.tail=FALSE)
```

```
## [1] 0.6125211
```

Returning to the GARCH(1,1), a Ljung Box test is also conducted:

```r
y2 <- (as.timeSeries(resst0))^2
# Ljung-Box test (stats) at 5 and 10 lags
Box.test(as.numeric(y2), lag = 5, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  as.numeric(y2)
## X-squared = 9.635, df = 5, p-value = 0.08626
```

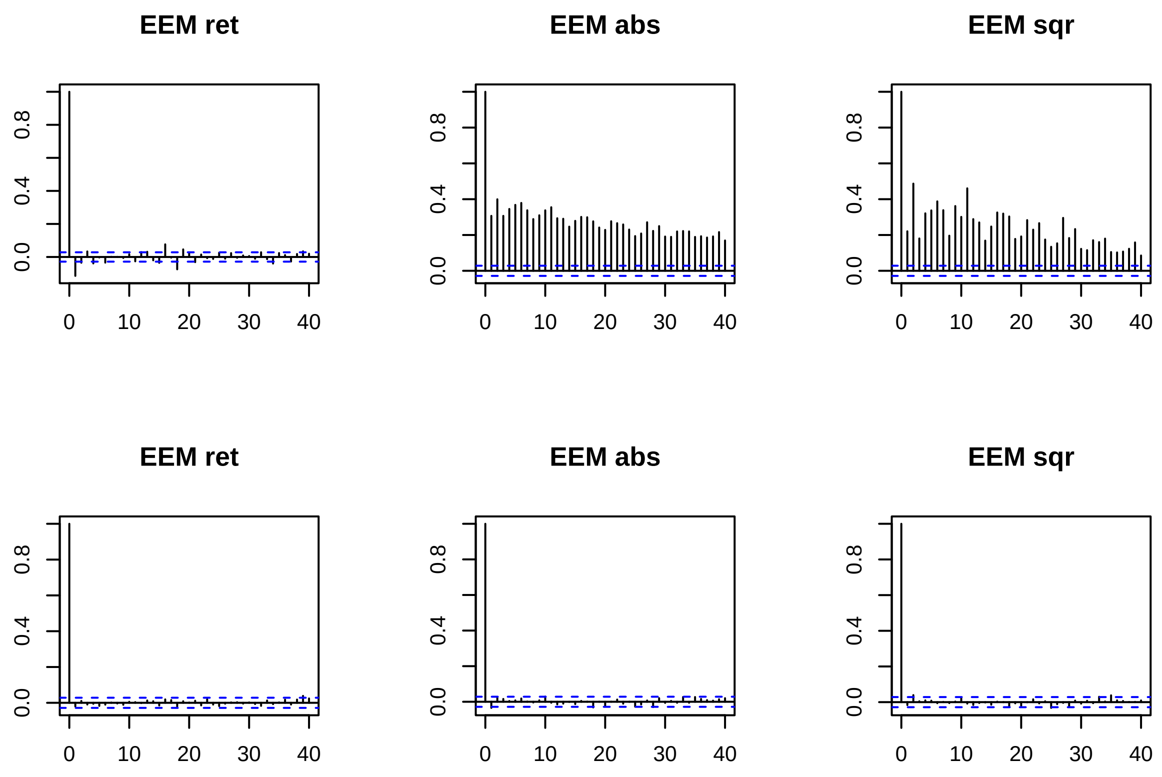
```r
Box.test(as.numeric(y2), lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  as.numeric(y2)
## X-squared = 13.426, df = 10, p-value = 0.2008
```

I do not reject the null hypothesis of the Ljung Box, and it can be hypothesized that the modeling of heteroscedasticity is sufficient according to this test.

The ACF of returns, absolute values, and squares and of standardized residuals, absolute values, and squares are also evaluated graphically.

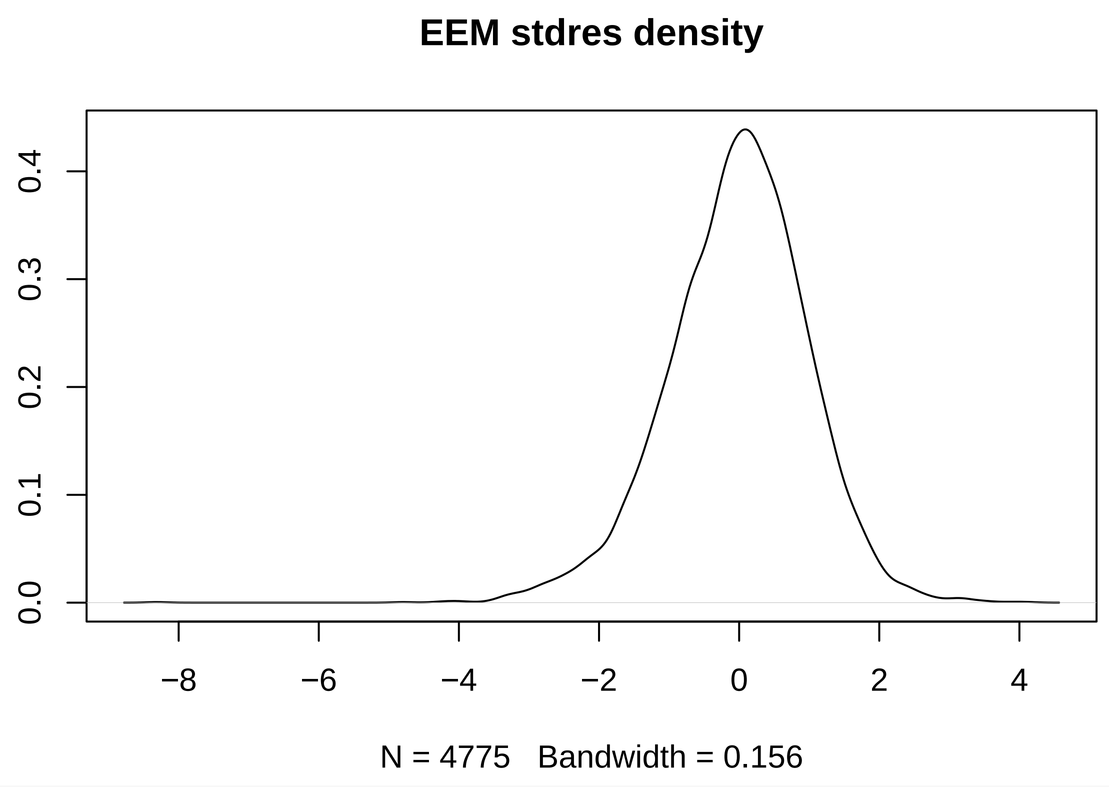
```r
# correlograms
par(mfrow=c(2,3))
acf(r_EEM, lag.max=40, xlab="", ylab="", main="EEM ret")
acf(abs(r_EEM), lag.max=40, xlab="", ylab="", main="EEM abs")
acf(abs(r_EEM)^2, lag.max=40, xlab="", ylab="", main="EEM sqr")
acf(as.numeric(resst0), lag.max=40, xlab="", ylab="", main="EEM ret")
acf(abs(as.numeric(resst0)), lag.max=40, xlab="", ylab="", main="EEM abs")
acf(abs(as.numeric(resst0))^2, lag.max=40, xlab="", ylab="", main="EEM sqr")
```



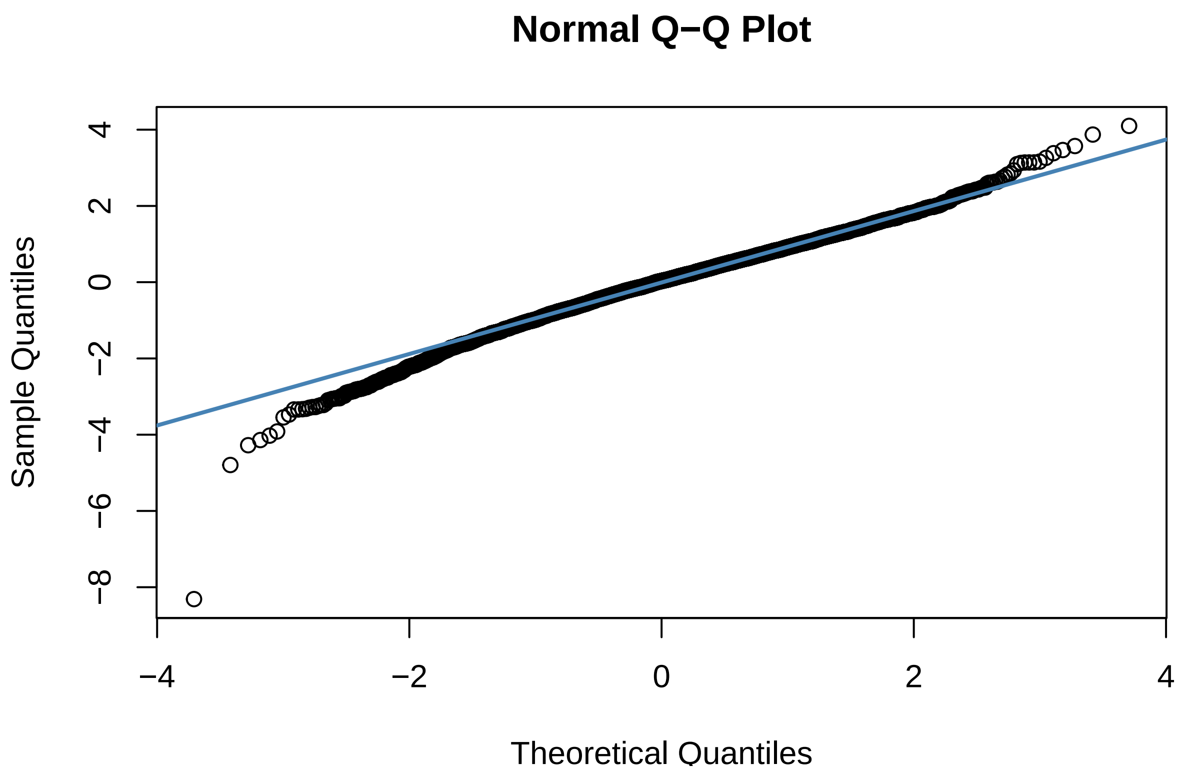
As seen before, there is structure on squared returns and absolute values, while nothing notable is observed on the standardized residuals of the estimated model.

```
# Evaluation of residual density
# QQ-plot against alternative distributions
d0 <- density(resst0)
# Residual density
par(mfrow=c(1,1))
plot(d0, main="EEM stdres density", ylab="")
```

We evaluate if the assumed distribution is correct.



```
# Thick tails, density of standardized residuals
# QQ-plot against normal distribution
qqnorm(resst0, main="Normal Q-Q Plot", xlab="Theoretical Quantiles", ylab="Sample Quantiles")
qqline(resst0, col="steelblue", lwd=2)
```



The tails are thick and there appears to be some asymmetry.

Therefore, it does not seem appropriate to use a normal density given the heaviness of the tails and the deviation from normality at those points.

```
y = r_EEM - mean(r_EEM)
nnr = y * (y < 0) # for negative sign bias
ppr = y * (y > 0) # for positive sign bias
dn = y < 0 # for sign bias
nnr = as.timeSeries(nnr)
ppr = as.timeSeries(ppr)
dn = as.timeSeries(dn)
nnr = lag(nnr, 1)
ppr = lag(ppr, 1)
dn = lag(dn, 1)

out.sbt = lm(resst0^2 ~ dn)
summary(out.sbt)
```

Test for variance asymmetry

```
##
## Call:
## lm(formula = resst0^2 ~ dn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.095 -0.885 -0.587  0.181 67.998
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.91249    0.03827  23.845  < 2e-16 ***
## dnTRUE       0.18246    0.05545   3.291  0.00101 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.913 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.002264,   Adjusted R-squared:  0.002055
## F-statistic: 10.83 on 1 and 4772 DF,  p-value: 0.001007
```

```
out.nsbt = lm(resst0^2 ~ nnr)
summary(out.nsbt)
```

```
##
## Call:
## lm(formula = resst0^2 ~ nnr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.212 -0.914 -0.599  0.177 68.107
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.98367    0.03117  31.557   <2e-16 ***
## nnr         -0.02654    0.02405  -1.104     0.27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.915 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0002552,  Adjusted R-squared:  4.569e-05
## F-statistic: 1.218 on 1 and 4772 DF,  p-value: 0.2698
```

```
out.psbt = lm(resst0^2 ~ ppr)
summary(out.psbt)
```

```
##
## Call:
## lm(formula = resst0^2 ~ ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.042 -0.903 -0.589  0.179 68.051
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.04200    0.03153  33.049  < 2e-16 ***
## ppr         -0.07183    0.02539  -2.829  0.00469 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.914 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.001674,   Adjusted R-squared:  0.001465
## F-statistic: 8.003 on 1 and 4772 DF,  p-value: 0.004689
```

```
out.jsbt = lm(resst0^2 ~ dn + nnr + ppr)
summary(out.jsbt)
```

```
##
## Call:
## lm(formula = resst0^2 ~ dn + nnr + ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.122 -0.884 -0.583  0.183 67.972
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.95667    0.05093  18.783   <2e-16 ***
## dnTRUE       0.16651    0.07392   2.252   0.0243 *
## nnr          0.02268    0.02853   0.795   0.4266
## ppr         -0.03901    0.02968  -1.314   0.1888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.913 on 4770 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.002757,   Adjusted R-squared:  0.00213
## F-statistic: 4.396 on 3 and 4770 DF,  p-value: 0.004279
```

We reject the null hypothesis (both with the sign bias test and the positive sign bias, and partly also with the joint test) that there is no asymmetry in the model and conclude that we probably need to model variance asymmetry using an appropriate model.

Similarities/Differences with the VT instrument

In both cases, all parameters of the GARCH(1,1) are significant.

Regarding VT, I do not reject the hypothesis of homoscedasticity in the residuals more clearly. While for the EEM instrument, it is noted that higher orders might be necessary in the GARCH model.

In both instruments, it is concluded that there is a need to model variance asymmetry following the sign bias tests.

Moreover, in both instruments, the normal distribution seems unsuitable when looking at the QQ plot of the model's residuals (more evidently in VT than in EEM).

4.3 "Fit at least three different specifications from the GARCH model family (thus GARCH(1,1) plus at least two others, e.g., EGARCH, APARCH, or other options provided by the library you are using). For each specification use at least two different distributions (the same for all chosen GARCH models, thus Normal and at least one other distribution). Always estimate using data up to the end of March 2022. Evaluate whether the adaptation to the data improves, supporting your observations with appropriate indicators, tests, or graphical analysis. Among the chosen models, identify the best specification using information criteria."

We will use:

1)GARCH(1,1) 2)GJR-GARCH (1,1) 3) APARCH (1,1)

And the distributions will be Normal and skewed-t.

Below, the GARCH(1,1) model is re-specified:

```r
p_EEM = log(window(EEMd, end="2022-03-31")) # log-prices
p_EEM = as.vector(p_EEM$EEM.Adjusted[,1])
T = length(p_EEM)
r_EEM = 100 * (p_EEM[2:T] - p_EEM[1:T-1]) # log percentage returns
y = r_EEM - mean(r_EEM)

spec0n <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)), mean.model = list(a
fit0n <- ugarchfit(spec0n, r_EEM)
resst0n <- residuals(fit0n, standardize=TRUE)
fit0n@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.06690372  0.01691650  3.954940 7.655397e-05
## omega  0.04355217  0.01051885  4.140395 3.467089e-05
## alpha1 0.09873422  0.01567990  6.296867 3.037226e-10
## beta1  0.88467088  0.01676817 52.758934 0.000000e+00
```

Next, the GJR-GARCH is specified.

The parameter related to asymmetry is γ_1:

```r
specgn <- ugarchspec(variance.model = list(model="gjrGARCH", garchOrder = c(1, 1)), mean.model = list(a
fitgn <- ugarchfit(specgn, r_EEM)
resstgn <- residuals(fitgn, standardize=TRUE)
fitgn@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.02873173  0.01837821  1.563358 1.179684e-01
## omega  0.04753528  0.01139761  4.170635 3.037521e-05
## alpha1 0.02873893  0.01013443  2.835772 4.571511e-03
## beta1  0.89542578  0.01659872 53.945456 0.000000e+00
## gamma1 0.10809231  0.02113984  5.113203 3.167410e-07
```

Looking at the robust p-values, obtained through quasi-maximum likelihood, the estimated coefficients are all significant except for the mean μ, so it can be re-estimated setting include.mean=FALSE.

```r
specgn <- ugarchspec(variance.model = list(model="gjrGARCH", garchOrder = c(1, 1)), mean.model = list(a
fitgn <- ugarchfit(specgn, r_EEM)
resstgn <- residuals(fitgn, standardize=TRUE)
fitgn@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## omega  0.04871373 0.011470711  4.246793 2.168521e-05
## alpha1 0.02715347 0.009639972  2.816759 4.851098e-03
## beta1  0.89581739 0.016562657 54.086575 0.000000e+00
## gamma1 0.11246607 0.020936318  5.371817 7.794712e-08
```

In particular, β, the coefficient related to the persistence of past variances, assumes a high value (0.89) as expected, and it is noted that the coefficient γ_1, related to variance asymmetry, is strongly significant.

If the shock is positive, the effect is α_1. If the shock is negative, the effect is $\alpha_1 + \gamma_1$.

Next, the APARCH model is specified. All coefficients are significant except for the mean μ.

```r
specan <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)), mean.model = list(arma
fitan <- ugarchfit(specan, r_EEM)
fitan@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## mu     0.02756967  0.01840973  1.497560 1.342476e-01
## omega  0.04291942  0.01079755  3.974923 7.040212e-05
## alpha1 0.07760406  0.01344061  5.773852 7.747967e-09
## beta1  0.90062761  0.01619707 55.604369 0.000000e+00
## gamma1 0.42558299  0.09225767  4.612982 3.969323e-06
## delta  1.69903159  0.23774009  7.146593 8.895107e-13
```

It is noted that the mean is not significant and the model is re-estimated removing it by setting include.mean = FALSE.

```r
specan <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)), mean.model = list(arma
fitan <- ugarchfit(specan, r_EEM)
resstan <- residuals(fitan, standardize=TRUE)
fitan@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## omega  0.04392279  0.01073194  4.092716 4.263489e-05
## alpha1 0.07749995  0.01345402  5.760356 8.393692e-09
## beta1  0.90113546  0.01603998 56.180588 0.000000e+00
## gamma1 0.44458313  0.09071646  4.900799 9.544776e-07
## delta  1.68892610  0.23243483  7.266235 3.697043e-13
```

All parameters are significant:

β as usual assumes a high value: 0.90 and multiplies the delayed variance indicating how much of the past volatility impacts the current volatility.

This is the context in which we find ourselves for looking at the part of the asymmetry in variance:

$$\alpha_1 \times (|\epsilon_{t-1}| - \gamma_1 \epsilon_{t-1})^{\delta_1}$$

The parameter γ_1 assumes values between -1 and 1 and the presence of asymmetry, understood as the greater impact of negative shocks compared to positive shocks, requires that γ be greater than zero.

If ϵ_{t-1} is greater than zero, then it is found that α_1 multiplies:

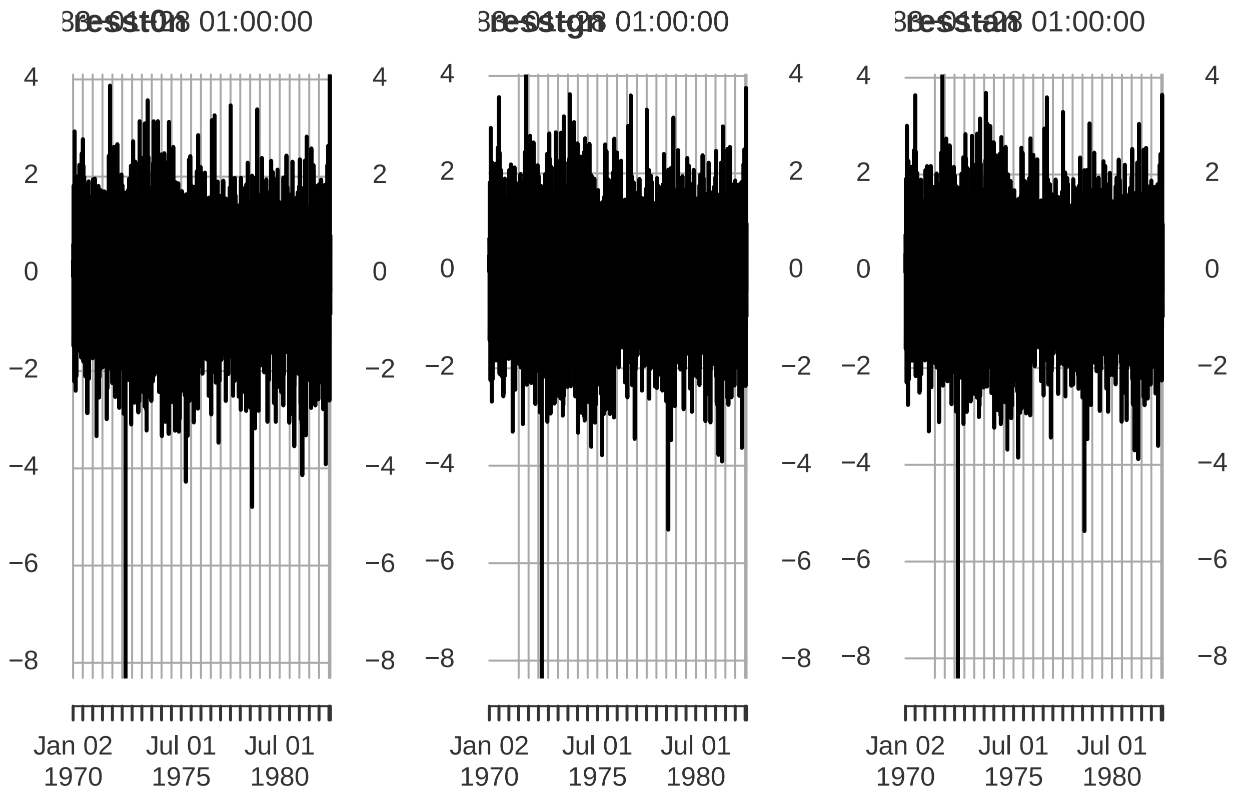
$$((1-\gamma) \times \epsilon_{t-1})^{\delta_1}$$

Instead, if $\epsilon_{t-1} < 0$ it is found that α_1 multiplies:

$$((1+\gamma)\times(-\epsilon_{t-1}))^{\delta_1}$$

And thus, as per the theory, the impact is greater for negative shocks given γ_1 greater than zero.

```r
par(mfrow=c(1,3))
plot(resst0n)
plot(resstgn)
plot(resstan)
```



```r
par(mfrow=c(1,1))

ks.test(as.matrix(resst0n),"pnorm",0,1)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  as.matrix(resst0n)
## D = 0.024753, p-value = 0.005752
## alternative hypothesis: two-sided
```

The null hypothesis of normality is rejected.

Specification of Models Changing the Distribution and Using the Skewed-t.

```
spec0ta <- ugarchspec(variance.model = list(model="sGARCH", garchOrder = c(1, 1)), mean.model = list(arm
fit0ta <- ugarchfit(spec0ta,r_EEM)
resst0ta <- residuals(fit0ta,standardize=TRUE)

specgta <- ugarchspec(variance.model = list(model="gjrGARCH", garchOrder = c(1, 1)), mean.model = list(a
fitgta <- ugarchfit(specgta,r_EEM)
resstgta <- residuals(fitgta,standardize=TRUE)

specata <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)), mean.model = list(arm
fitata <- ugarchfit(specata,r_EEM)
resstata <- residuals(fitata,standardize=TRUE)
```

Below, we evaluate the significance of the skew and shape coefficients related to the choice of the skewed-t distribution:

```
fit0ta@fit$robust.matcoef
```

```
##            Estimate  Std. Error   t value     Pr(>|t|)
## mu       0.06280737 0.016683863  3.764558 1.668440e-04
## omega    0.04073260 0.009027133  4.512241 6.414619e-06
## alpha1   0.09925601 0.012483684  7.950859 1.776357e-15
## beta1    0.88480453 0.013271686 66.668584 0.000000e+00
## skew     0.87445792 0.019231908 45.469119 0.000000e+00
## shape   10.03873153 1.413952544  7.099766 1.249667e-12
```

```
fitgta@fit$robust.matcoef
```

```
##            Estimate  Std. Error   t value     Pr(>|t|)
## mu       0.02847781  0.01823777  1.561475 1.184118e-01
## omega    0.04898358  0.01096966  4.465369 7.993075e-06
## alpha1   0.02666486  0.00836172  3.188920 1.428053e-03
## beta1    0.89048750  0.01369233 65.035484 0.000000e+00
## gamma1   0.12198882  0.02049070  5.953374 2.626698e-09
## skew     0.86472878  0.01915447 45.145005 0.000000e+00
## shape   10.43579686  1.54907308  6.736801 1.619127e-11
```

```
fitata@fit$robust.matcoef
```

```
##            Estimate  Std. Error   t value     Pr(>|t|)
## mu       0.02651764 0.018355379  1.444679 1.485480e-01
## omega    0.04215186 0.009823359  4.290983 1.778840e-05
## alpha1   0.08268733 0.010394636  7.954807 1.776357e-15
## beta1    0.89845674 0.013178302 68.176972 0.000000e+00
## gamma1   0.48400155 0.087419137  5.536563 3.084651e-08
## delta    1.57346478 0.190995102  8.238247 2.220446e-16
## skew     0.86402554 0.019152392 45.113192 0.000000e+00
## shape   10.39431183 1.529130483  6.797531 1.064282e-11
```

Looking at the GJR and APARCH, one might consider removing the mean and re-estimating:

```
specgta <- ugarchspec(variance.model = list(model="gjrGARCH", garchOrder = c(1, 1)), mean.model = list(a
fitgta <- ugarchfit(specgta,r_EEM)
resstgta <- residuals(fitgta,standardize=TRUE)

specata <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)), mean.model = list(arm
fitata <- ugarchfit(specata,r_EEM)
resstata <- residuals(fitata,standardize=TRUE)
```

```
fit0ta@fit$robust.matcoef
```

```
##            Estimate  Std. Error   t value     Pr(>|t|)
## mu       0.06280737 0.016683863  3.764558 1.668440e-04
## omega    0.04073260 0.009027133  4.512241 6.414619e-06
## alpha1   0.09925601 0.012483684  7.950859 1.776357e-15
## beta1    0.88480453 0.013271686 66.668584 0.000000e+00
## skew     0.87445792 0.019231908 45.469119 0.000000e+00
## shape   10.03873153 1.413952544  7.099766 1.249667e-12
```

```
fitgta@fit$robust.matcoef
```

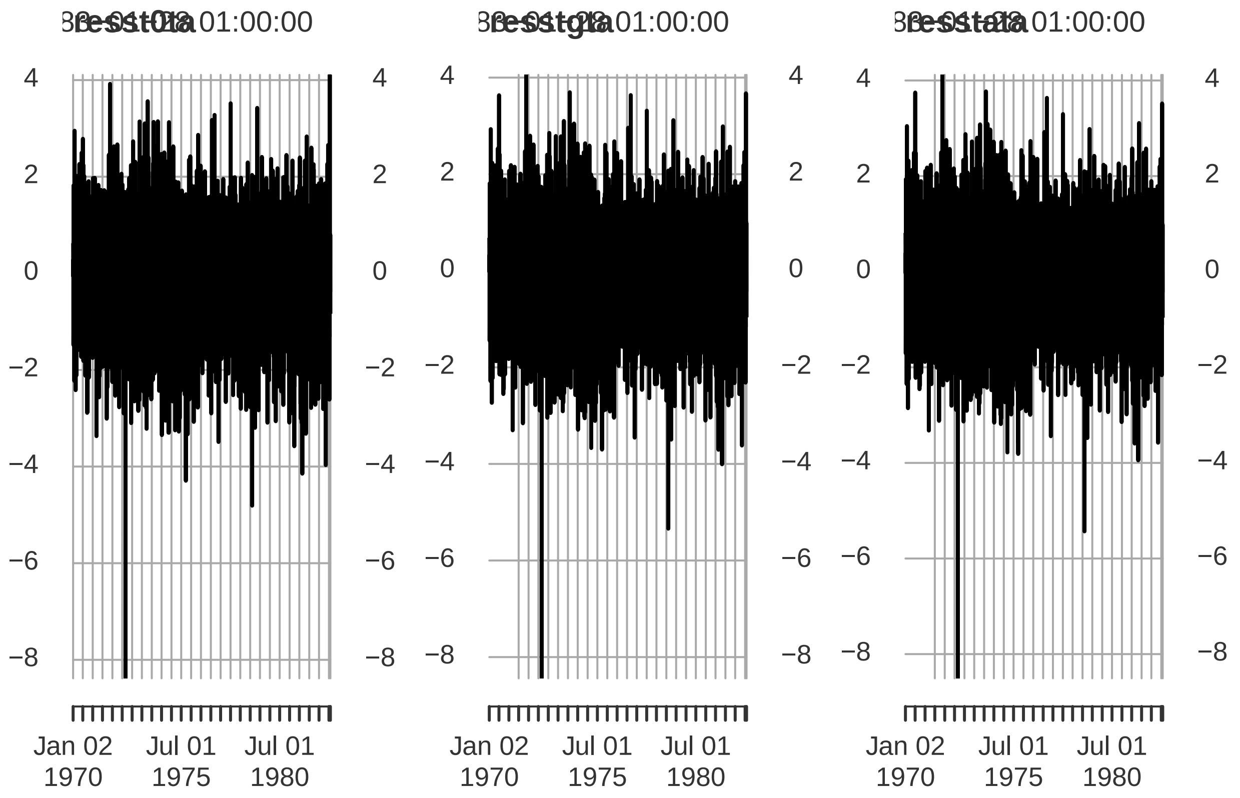
```
##            Estimate  Std. Error   t value     Pr(>|t|)
## omega    0.05127175 0.011140874  4.602130 4.181918e-06
## alpha1   0.02506583 0.008047989  3.114546 1.842282e-03
## beta1    0.89069056 0.013709470 64.968999 0.000000e+00
## gamma1   0.12764385 0.019996561  6.383290 1.733229e-10
## skew     0.85879709 0.018475926 46.481951 0.000000e+00
## shape   10.44638973 1.568148692  6.661607 2.708500e-11
```

```
fitata@fit$robust.matcoef
```

```
##            Estimate  Std. Error   t value     Pr(>|t|)
## omega    0.04397385 0.009830508  4.473202 7.705703e-06
## alpha1   0.08294787 0.010409071  7.968806 1.554312e-15
## beta1    0.89872604 0.013075399 68.734121 0.000000e+00
## gamma1   0.50486719 0.085840329  5.881468 4.066425e-09
## delta    1.56248569 0.185899856  8.404986 0.000000e+00
## skew     0.85846568 0.018510698 46.376731 0.000000e+00
## shape   10.40121987 1.545629232  6.729440 1.703171e-11
```

The comments made previously regarding the estimated parameters remain valid for the models even in this case. The evaluation now includes the significance of the data's asymmetry and the degrees of freedom of the t.

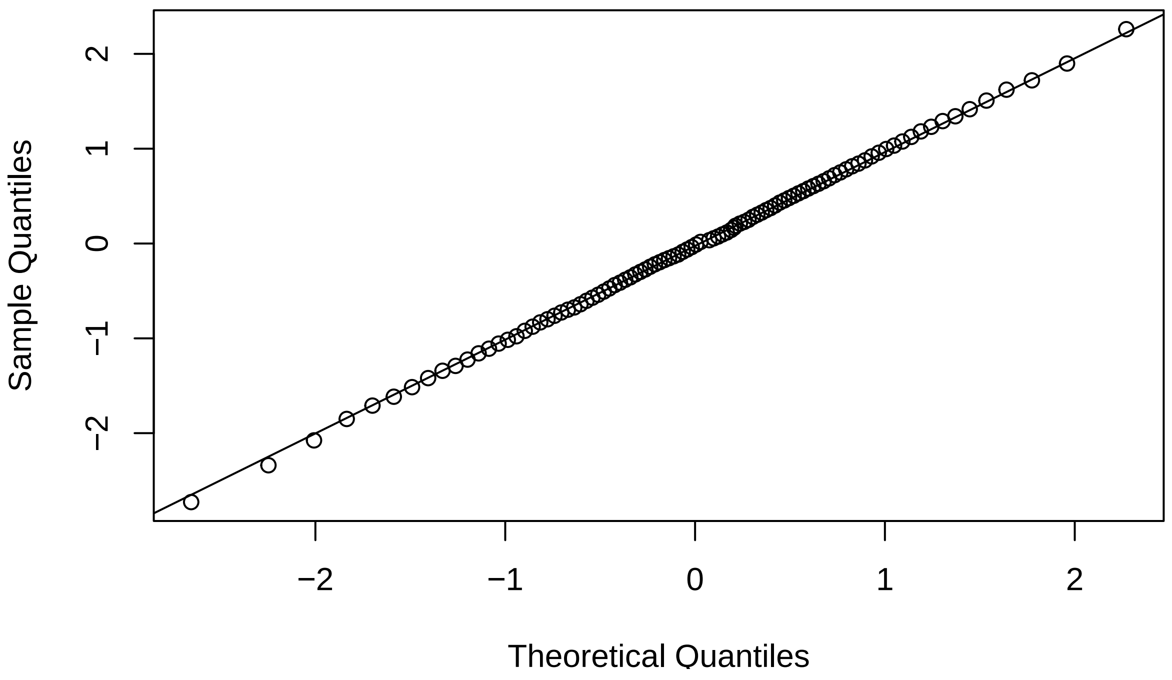
```
par(mfrow=c(1,3))
plot(resst0ta)
plot(resstgta)
plot(resstata)
```



```
par(mfrow=c(1,1))
```

This code is used to create a Q-Q plot to compare the theoretical distribution of the skewed Student's t with the empirical distribution of the standardized residuals resst0ta from the GARCH(1,1) model with skewed-t specified earlier.

```
p=0.01*(1:99)
qemp=quantile(resst0ta,probs=p)
qteor=qdist(distribution="sstd",p,mu=0,sigma=1,skew=fit0ta@fit$coef[5],shape=fit0ta@fit$coef[6])
qqplot(qteor,qemp,xlab="Theoretical Quantiles",ylab ="Sample Quantiles")
qqline(qemp, distribution = function(p) qdist(distribution="sstd",p,mu=0,sigma=1,skew=fit0ta@fit$coef[5
```



A slight deviation on the tails is observed, which means that the theoretical distribution is not fully capable of capturing the variation in the data. However, this deviation is certainly less pronounced than what was seen with the normal distribution.

Below is the Kolmogorov-Smirnov test to verify if an empirical distribution (residuals) follows a specified theoretical distribution (skewed-t distribution).

```
# H0: Skew-T
y=as.matrix(resst0ta)
y=as.matrix(sort(y))
ks.test(as.matrix(resst0ta),pdist(distribution="sstd",y,mu=0,sigma=1,lambda=-0.5,
skew=fit0ta@fit$coef[5],shape=fit0ta@fit$coef[6]))
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  as.matrix(resst0ta) and pdist(distribution = "sstd", y, mu = 0, sigma = 1, lambda = -0.5, ske
## D = 0.48293, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

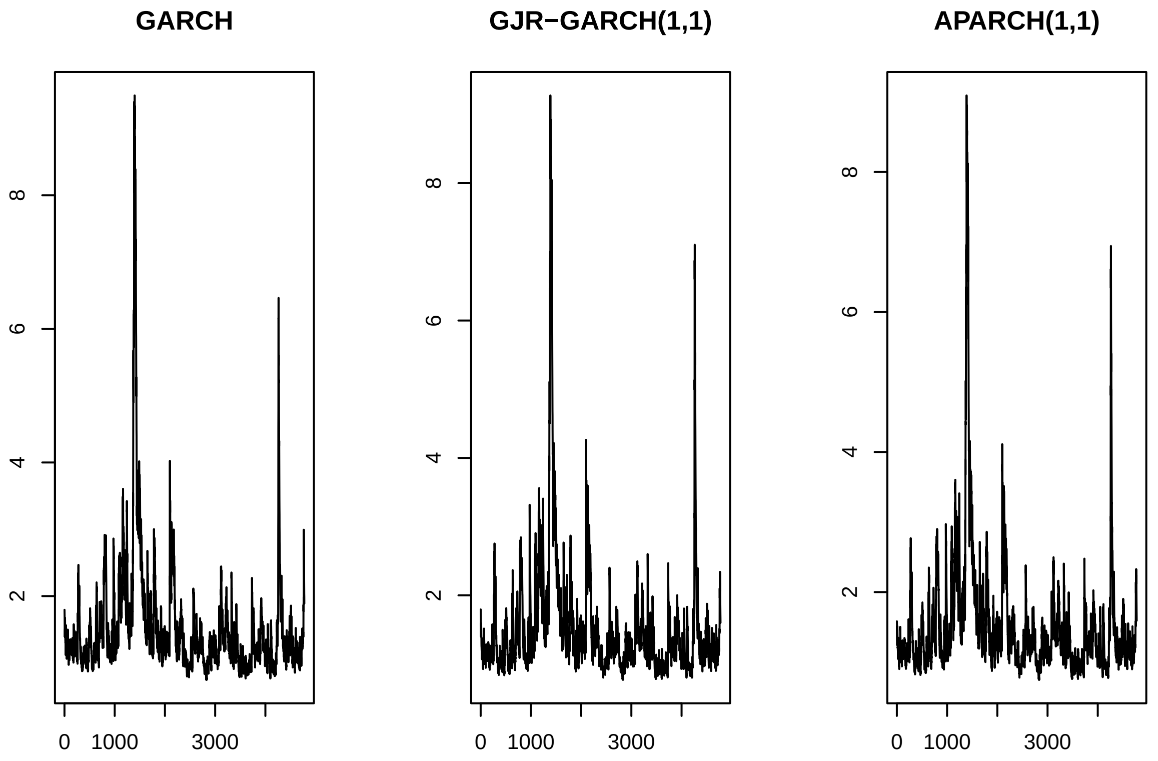
I reject the hypothesis of the distribution as a skewed-t, likely due to the length of the series.

However, the fit to the data seems to improve as the skew and shape parameters are always significant, and as noted earlier, the deviation at the tails appears to be less.

Graphs of the estimated variances for the three models using the sigma function:

```
# variance plots
par(mfrow=c(1,3))
sigma0n <- sigma(fit0n)
sigmagn <- sigma(fitgn)
sigmaan <- sigma(fitan)
par(mfrow=c(1,3))
plot(as.numeric(sigma0n),type="l",ylab="",xlab="")
title("GARCH")
plot(as.numeric(sigmagn),type="l",ylab="",xlab="")
```

```
title("GJR-GARCH(1,1)")
plot(as.numeric(sigmaan),type="l",ylab="",xlab="")
title("APARCH(1,1)")
```



```
par(mfrow=c(1,1))
```

In the 3 plots it is difficult to notcie diffrences.

Test Sign Bias

```
y=r_EEM-mean(r_EEM)
nnr=y*(y<0)# for negative sign bias
ppr=y*(y>0)# for positive signi bias
dn=y<0# for sign bias
nnr=as.timeSeries(nnr)
ppr=as.timeSeries(ppr)
dn=as.timeSeries(dn)
nnr=lag(nnr,1)
ppr=lag(ppr,1)
dn=lag(dn,1)

out.sbt=lm(resst0ta^2 ~ dn)
summary(out.sbt)
```

```
##
## Call:
## lm(formula = resst0ta^2 ~ dn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.107 -0.894 -0.591  0.176 69.173
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.92129    0.03871   23.80  < 2e-16 ***
## dnTRUE       0.18566    0.05609    3.31 0.000939 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.935 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.002291,   Adjusted R-squared:  0.002082
## F-statistic: 10.96 on 1 and 4772 DF,  p-value: 0.0009389
```

```
out.nsbt=lm(resst0ta^2 ~ nnr)
summary(out.nsbt)
```

```
##
## Call:
## lm(formula = resst0ta^2 ~ nnr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.208 -0.923 -0.604  0.184 69.283
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.99499    0.03153  31.555   <2e-16 ***
## nnr         -0.02487    0.02433  -1.022    0.307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.937 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0002189,  Adjusted R-squared:  9.371e-06
## F-statistic: 1.045 on 1 and 4772 DF,  p-value: 0.3068
```

```
out.psbt=lm(resst0ta^2 ~ ppr)
summary(out.psbt)
```

```
##
## Call:
## lm(formula = resst0ta^2 ~ ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.054 -0.912 -0.594  0.183 69.226
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.05401    0.03189  33.050  < 2e-16 ***
## ppr         -0.07466    0.02568  -2.907  0.00366 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.936 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.001768,   Adjusted R-squared:  0.001559
## F-statistic: 8.452 on 1 and 4772 DF,  p-value: 0.003663
```

```
out.jsbt=lm(resst0ta^2 ~ dn + nnr + ppr)
summary(out.jsbt)
```

```
##
## Call:
## lm(formula = resst0ta^2 ~ dn + nnr + ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.138 -0.894 -0.590  0.184 69.143
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.96868    0.05152  18.802   <2e-16 ***
## dnTRUE       0.17075    0.07477   2.284   0.0224 *
## nnr          0.02610    0.02885   0.905   0.3658
## ppr         -0.04185    0.03002  -1.394   0.1634
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.935 on 4770 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.002868,   Adjusted R-squared:  0.002241
## F-statistic: 4.574 on 3 and 4770 DF,  p-value: 0.003339
```

```
out.sbt=lm(resstgta^2 ~ dn)
summary(out.sbt)
```

```
##
```

```
## Call:
## lm(formula = resstgta^2 ~ dn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.053 -0.897 -0.586  0.193 70.176
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.94248    0.03808  24.752   <2e-16 ***
## dnTRUE       0.11002    0.05517   1.994   0.0462 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.904 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0008327,  Adjusted R-squared:  0.0006233
## F-statistic: 3.977 on 1 and 4772 DF,  p-value: 0.04618
```

```
out.nsbt=lm(resstgta^2 ~ nnr)
summary(out.nsbt)
```

```
##
## Call:
## lm(formula = resstgta^2 ~ nnr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.001 -0.909 -0.585  0.181 70.228
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.00133    0.03100  32.305   <2e-16 ***
## nnr          0.01087    0.02391   0.454     0.65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.905 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  4.327e-05,  Adjusted R-squared:  -0.0001663
## F-statistic: 0.2065 on 1 and 4772 DF,  p-value: 0.6495
```

```
out.psbt=lm(resstgta^2 ~ ppr)
summary(out.psbt)
```

```
##
## Call:
## lm(formula = resstgta^2 ~ ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.020 -0.903 -0.588  0.182 70.208
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.01968    0.03137  32.509   <2e-16 ***
## ppr         -0.04180    0.02526  -1.655    0.098 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.904 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0005736,  Adjusted R-squared:  0.0003641
## F-statistic: 2.739 on 1 and 4772 DF,  p-value: 0.09801
```

```
out.jsbt=lm(resstgta^2 ~ dn + nnr + ppr)
summary(out.jsbt)
```

```
##
## Call:
## lm(formula = resstgta^2 ~ dn + nnr + ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.116 -0.890 -0.581  0.185 70.116
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.96678    0.05067  19.079   <2e-16 ***
## dnTRUE       0.14989    0.07354   2.038   0.0416 *
## nnr          0.05156    0.02838   1.817   0.0693 .
## ppr         -0.02146    0.02953  -0.727   0.4675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.903 on 4770 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.001634,   Adjusted R-squared:  0.001006
## F-statistic: 2.603 on 3 and 4770 DF,  p-value: 0.05028
```

```
out.sbt=lm(resstata^2 ~ dn)
summary(out.sbt)
```

```
##
## Call:
## lm(formula = resstata^2 ~ dn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.044 -0.901 -0.588  0.187 71.314
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.95189    0.03838   24.80   <2e-16 ***
## dnTRUE       0.09177    0.05561    1.65   0.0989 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.919 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0005704,  Adjusted R-squared:  0.000361
## F-statistic: 2.724 on 1 and 4772 DF,  p-value: 0.09895
```

```
out.nsbt=lm(resstata^2 ~ nnr)
summary(out.nsbt)
```

```
##
## Call:
## lm(formula = resstata^2 ~ nnr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.003 -0.909 -0.586  0.194 71.355
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.00348    0.03124  32.124   <2e-16 ***
## nnr          0.01328    0.02410   0.551    0.582
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.919 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  6.362e-05,  Adjusted R-squared:  -0.0001459
## F-statistic: 0.3036 on 1 and 4772 DF,  p-value: 0.5817
```

```
out.psbt=lm(resstata^2 ~ ppr)
summary(out.psbt)
```

```
##
## Call:
## lm(formula = resstata^2 ~ ppr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.017 -0.904 -0.590  0.189 71.341
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.01654    0.03161  32.155   <2e-16 ***
## ppr         -0.03530    0.02546  -1.386    0.166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.919 on 4772 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0004027,  Adjusted R-squared:  0.0001932
## F-statistic: 1.922 on 1 and 4772 DF,  p-value: 0.1657
```

```
out.jsbt=lm(resstata^2 ~ dn + nnr + ppr)
summary(out.jsbt)
```

```
##
## Call:
## lm(formula = resstata^2 ~ dn + nnr + ppr)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.104 -0.894 -0.583  0.194 71.257
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.97283    0.05108  19.047   <2e-16 ***
## dnTRUE       0.13175    0.07413   1.777   0.0756 .
## nnr          0.04895    0.02861   1.711   0.0871 .
## ppr         -0.01849    0.02976  -0.621   0.5346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.919 on 4770 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.001264,   Adjusted R-squared:  0.0006362
## F-statistic: 2.013 on 3 and 4770 DF,  p-value: 0.1099
```

It is noted that using the GJR, I reject in the sign bias test and the joint test, while looking at the APARCH, I never reject, and thus it seems to capture the asymmetry in variance well.

```
ICall<-cbind(infocriteria(fit0n),infocriteria(fitgn),infocriteria(fitan),
infocriteria(fit0ta),infocriteria(fitgta),infocriteria(fitata))
colnames(ICall)<-c("GARCH N","GJR N","APARCH N","GARCH TA","GJR TA","APARCH TA")
ICall_df <- data.frame(ICall)
print(ICall_df)
```

```
##               GARCH.N    GJR.N APARCH.N GARCH.TA   GJR.TA APARCH.TA
## Akaike       3.506946 3.492475 3.492369 3.474212 3.460589  3.460040
## Bayes        3.512367 3.497896 3.499145 3.482343 3.468720  3.469526
## Shibata      3.506945 3.492473 3.492367 3.474208 3.460586  3.460035
## Hannan-Quinn 3.508851 3.494379 3.494750 3.477068 3.463445  3.463372
```

The APARCH with the skewed-t distribution appears to be the best according to three information criteria, while according to the Bayes criterion, the GJR with the skewed-t is better.

4.4 "Proceed to construct one-step-ahead forecasts for the period from April 2022 to March 2023 (one year) with all the GARCH specifications from the previous point (at least 6). Evaluate any differences in the obtained forecasts by conducting a Diebold-Mariano test among all possible pairs of models. Are you able to obtain a complete ranking among the models?"

```r
p_EEM = log(window(EEMd, end="2023-03-31")) #log-prices
p_EEM = as.vector(p_EEM$EEM.Adjusted[,1])
T = length(p_EEM)
r_EEM=100*(p_EEM[2:T]-p_EEM[1:T-1]) #log percentage returns
y = r_EEM - mean(r_EEM)

# GARCH(1,1) Normal
for0n = ugarchroll(spec0n,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="movi
for0nd = as.data.frame(for0n)
s0nf = for0nd$Sigma

# GARCH(1,1) Skewed-Student t
for0ta = ugarchroll(spec0ta,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="mo
for0tad = as.data.frame(for0ta)
s0taf = for0tad$Sigma

# GJRGARCH(1,1) Normal
forgn = ugarchroll(specgn,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="movi
forgnd = as.data.frame(forgn)
sgnf = forgnd$Sigma

# GJRGARCH(1,1) Skewed-Student t
forgta = ugarchroll(specgta,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="mo
forgtad = as.data.frame(forgta)
sgtaf = forgtad$Sigma

# APARCH(1,1) Normal
foran = ugarchroll(specan,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="movi
forand = as.data.frame(foran)
sanf = forand$Sigma

# APARCH(1,1) Skewed-Student t
forata = ugarchroll(specata,data=y,forecast.length=252,refit.every=5,window.size=1000, refit.window="mo
foratad = as.data.frame(forata)
sataf = foratad$Sigma
```

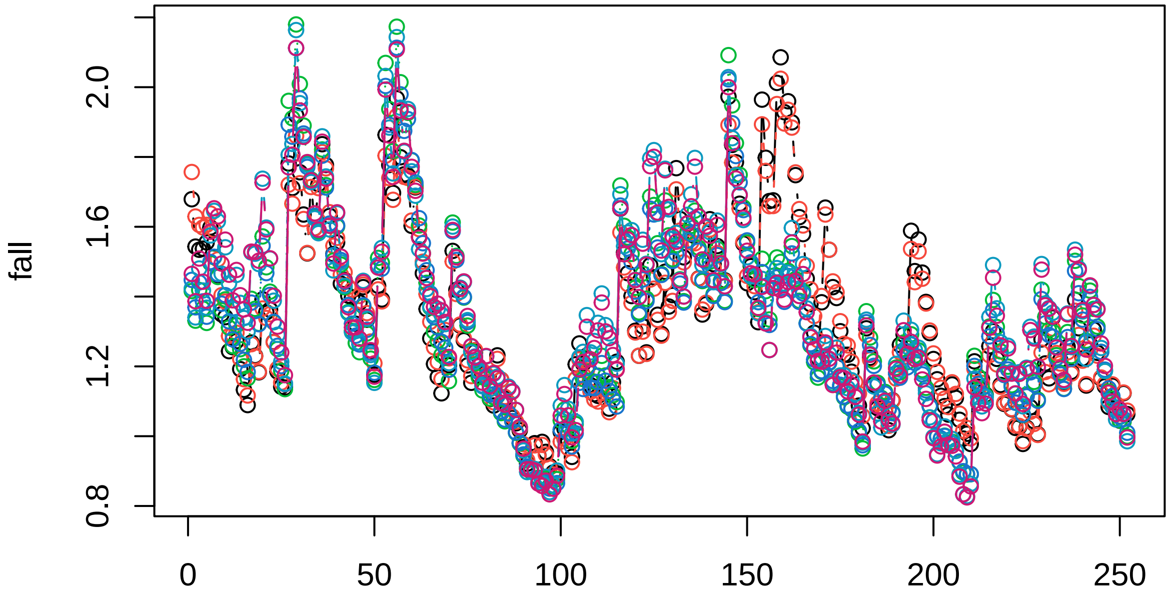
Ugarchroll performs a rolling-window forecast, repeatedly making a forecast by "sliding" the observation window, with a length specified in window.size and a sliding interval specified in refit.every.

The refit.every parameter specifies the number of observations after which the model is refitted to the data. For example, if refit.every=5 and the window length (window.size) is 1000, the model will be refitted to the data each time the window moves by 5 observations, i.e., each time the window contains observations 1-1000, 6-1005, 11-1010, and so on.

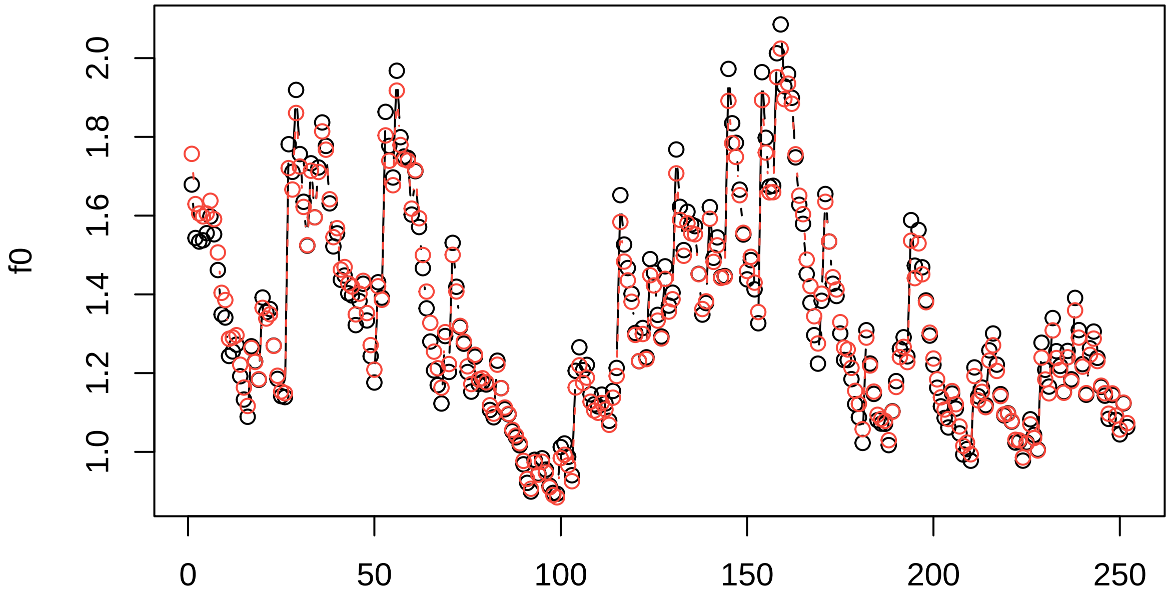
The forecast for a forecast horizon of 252 periods (one year) is specified in forecast.length.

This code creates several graphs showing the variance forecasts obtained from the different specified models, comparing the different models to each other.

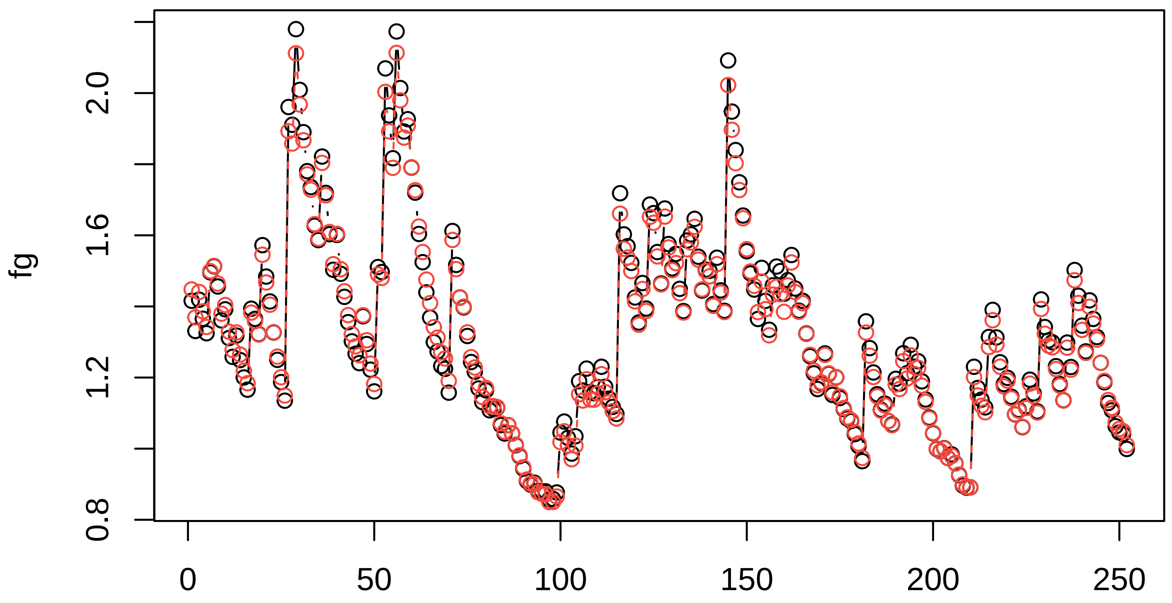
```r
fall = cbind(s0nf,s0taf,sgnf,sgtaf,sanf,sataf)
par(mfrow=c(1,1))
matplot(fall, type = c("b"),pch=1,col = 1:6)
```



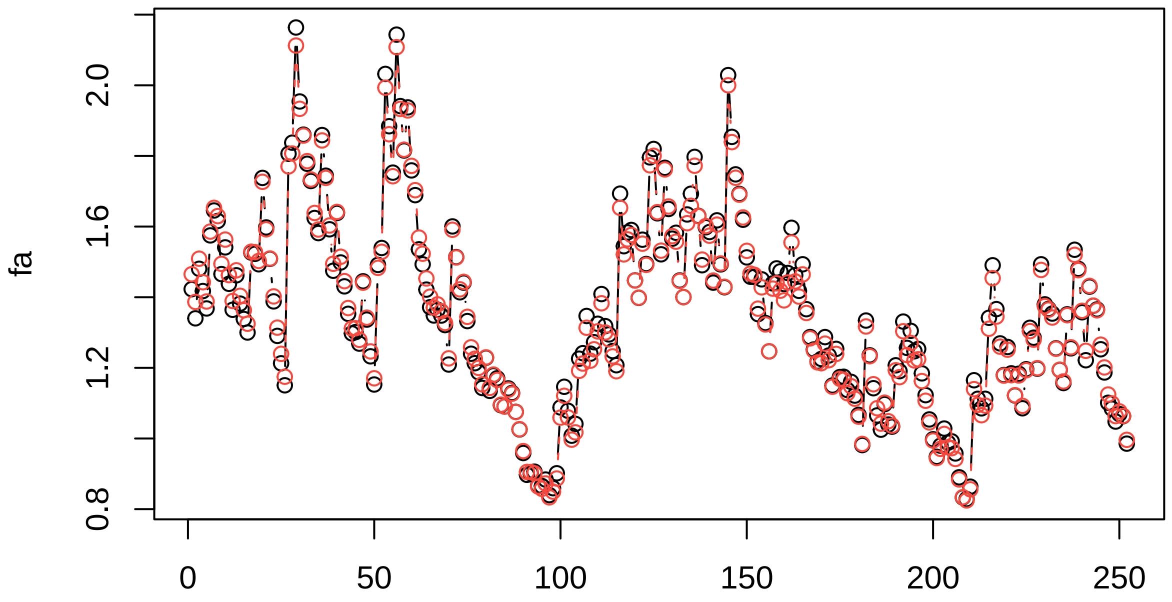
```r
f0 = cbind(s0nf,s0taf)
fg = cbind(sgnf,sgtaf)
fa = cbind(sanf,sataf)
matplot(f0, type=('b'),pch=1,col=1:2)
```



```r
matplot(fg, type=c('b'),pch=1,col=1:2)
```



```r
matplot(fa, type=c('b'),pch=1,col=1:2)
```



s0nf is the historical series of variances forecasted by the GARCH(1,1) model with a normal distribution. rf is the historical series of realized volatilities.

lN and the other calculated measures are a historical series calculated as the square of the difference between the variance forecast and the realized volatility, squared.

In other words, the quadratic errors between the model's forecasts and the realized volatilities are being calculated.

```r
rf = for0nd$Realized

lN = (s0nf^2 -rf^2)^2 # GARCH
lTA= (s0taf^2 -rf^2)^2 # GARCH skewed-t
lGN = (sgnf^2 -rf^2)^2 # GJRGARCH
lGTA = (sgtaf^2 -rf^2)^2 # GJRGARCH skewed-t
lAN = (sanf^2 -rf^2)^2 # GJRGARCH t
```

```
lATA = (sataf^2 -rf^2)^2 # APARCH
```

Calculation of the differences

```
dNTA = lN-lTA
dNGN = lN-lGN
dNGTA = lN-lGTA
dNAN = lN-lAN
dNATA = lN-lATA

dTAGN = lTA-lGN
dTAGTA = lTA-lGTA
dTAAN = lTA-lAN
dTAATA = lTA-lATA

dGNGTA = lGN-lGTA
dGNAN = lGN-lAN
dGNATA = lGN-lATA

dGTAAN = lGTA-lAN
dGTAATA = lGTA-lATA

dANATA = lAN-lATA
```

The “NeweyWest” function estimates the covariance matrices of model parameters using the Newey-West method with a rolling window, with a lag equal to the variable “m” previously defined.

In this case, the outputs given by the function are scalar values and not matrices because we are estimating only one parameter.

```
m<-floor(0.75*((NROW(rf))^(1/3)))

x1<-as.vector(matrix(1,nrow=NROW(rf)))

VNTA = NeweyWest(lm(dNTA~x1-1),lag=m,prewhite=0)
VNGN = NeweyWest(lm(dNGN~x1-1),lag=m,prewhite=0)
VNGTA = NeweyWest(lm(dNGTA~x1-1),lag=m,prewhite=0)
VNAN = NeweyWest(lm(dNAN~x1-1),lag=m,prewhite=0)
VNATA = NeweyWest(lm(dNATA~x1-1),lag=m,prewhite=0)

VTAGN = NeweyWest(lm(dTAGN~x1-1),lag=m,prewhite=0)
VTAGTA = NeweyWest(lm(dTAGTA~x1-1),lag=m,prewhite=0)
VTAAN = NeweyWest(lm(dTAAN~x1-1),lag=m,prewhite=0)
VTAATA = NeweyWest(lm(dTAATA~x1-1),lag=m,prewhite=0)

VGNGTA = NeweyWest(lm(dGNGTA~x1-1),lag=m,prewhite=0)
VGNAN = NeweyWest(lm(dGNAN~x1-1),lag=m,prewhite=0)
VGNATA = NeweyWest(lm(dGNATA~x1-1),lag=m,prewhite=0)

VGTAAN = NeweyWest(lm(dGTAAN~x1-1),lag=m,prewhite=0)
VGTAATA = NeweyWest(lm(dGTAATA~x1-1),lag=m,prewhite=0)

VANATA = NeweyWest(lm(dANATA~x1-1),lag=m,prewhite=0)

DM<-matrix(0,nrow=6,ncol=6)
# Output -> robust variances of regressor
# loss = model in row minus model il column
colnames(DM)<-c("GARCHnorm","GARCH TA","GJRnorm","GJR ta","APARCHnorm","APARCH TA")
rownames(DM)<-c("GARCHnorm","GARCH TA","GJRnorm","GJR ta","APARCHnorm","APARCH TA")

DM[1, 2] <- mean(dNTA) / sqrt(VNTA)
DM[1, 3] <- mean(dNGN) / sqrt(VNGN)
DM[1, 4] <- mean(dNGTA) / sqrt(VNGTA)
DM[1, 5] <- mean(dNAN) / sqrt(VNAN)
DM[1, 6] <- mean(dNATA) / sqrt(VNATA)
DM[2, 3] <- mean(dTAGN) / sqrt(VTAGN)
DM[2, 4] <- mean(dTAGTA) / sqrt(VTAGTA)
DM[2, 5] <- mean(dTAAN) / sqrt(VTAAN)
DM[2, 6] <- mean(dTAATA) / sqrt(VTAATA)
DM[3, 4] <- mean(dGNGTA) / sqrt(VGNGTA)
DM[3, 5] <- mean(dGNAN) / sqrt(VGNAN)
DM[3, 6] <- mean(dGNATA) / sqrt(VGNATA)
DM[4, 5] <- mean(dGTAAN) / sqrt(VGTAAN)
DM[4, 6] <- mean(dGTAATA) / sqrt(VGTAATA)
DM[5, 6] <- mean(dANATA) / sqrt(VANATA)
DM
```

```
##            GARCHnorm GARCH TA    GJRnorm     GJR ta APARCHnorm  APARCH TA
## GARCHnorm          0 2.092726 -0.2669687  0.1950476 -0.4869688 -0.1818041
## GARCH TA           0 0.000000 -0.6498220 -0.1974929 -0.8789436 -0.5639721
## GJRnorm            0 0.000000  0.0000000  2.4266200 -0.5807075  0.1637822
## GJR ta             0 0.000000  0.0000000  0.0000000 -1.8441019 -1.0562102
## APARCHnorm         0 0.000000  0.0000000  0.0000000  0.0000000  2.2936546
## APARCH TA          0 0.000000  0.0000000  0.0000000  0.0000000  0.0000000
```

The APARCH model with skewed-t distribution is preferable over 2 of the other models, specifically it is statistically better than the APARCH with a normal distribution. The GJR with a skewed-t is statistically better than both the GJR with a normal distribution and the APARCH with a normal distribution. Finally, the GARCH with a skewed-t is statistically better than the GARCH with a normal distribution.

5. Correlations among Standardized Residuals

“Retrieve the standardized returns of the two financial instruments, based on the best model identified in section 4. Use an estimation up to the end of March 2023. Calculate the correlations between the standardized residuals over a rolling window of 62 observations, and over a rolling window of 252 observations. Based on graphical analysis, do the correlations appear constant?”

The best model in both the first and second cases turns out to be the APARCH with a skewed-t distribution.

Below are analyses comparing the correlations with the residuals of the second financial instrument used.

Consider data up to March 2023 and re-estimate the two models:

```
p_VT = log(window(VTd, end="2023-03-31")) #log-prices
p_VT = as.vector(p_VT$VT.Adjusted[,1])
T = length(p_VT)
r_VT=100*(p_VT[2:T]-p_VT[1:T-1]) #log percentage returns
yVT = r_VT - mean(r_VT)

specataVT <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)),
                        mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
                        distribution.model="sstd")
fitataVT <- ugarchfit(specataVT, r_VT)
resstataVT <- residuals(fitataVT, standardize=TRUE)
fitataVT@fit$robust.matcoef
```

```
##          Estimate  Std. Error   t value     Pr(>|t|)
## omega  0.02517499 0.008807398  2.858391 4.257950e-03
## alpha1 0.10372672 0.017266453  6.007413 1.885065e-09
## beta1  0.90778374 0.021527435 42.168690 0.000000e+00
## gamma1 0.97097904 0.024723952 39.272809 0.000000e+00
## delta  0.86824747 0.113713215  7.635414 2.242651e-14
## skew   0.81871764 0.018643377 43.914664 0.000000e+00
## shape  7.44777512 0.883084074  8.433823 0.000000e+00
```

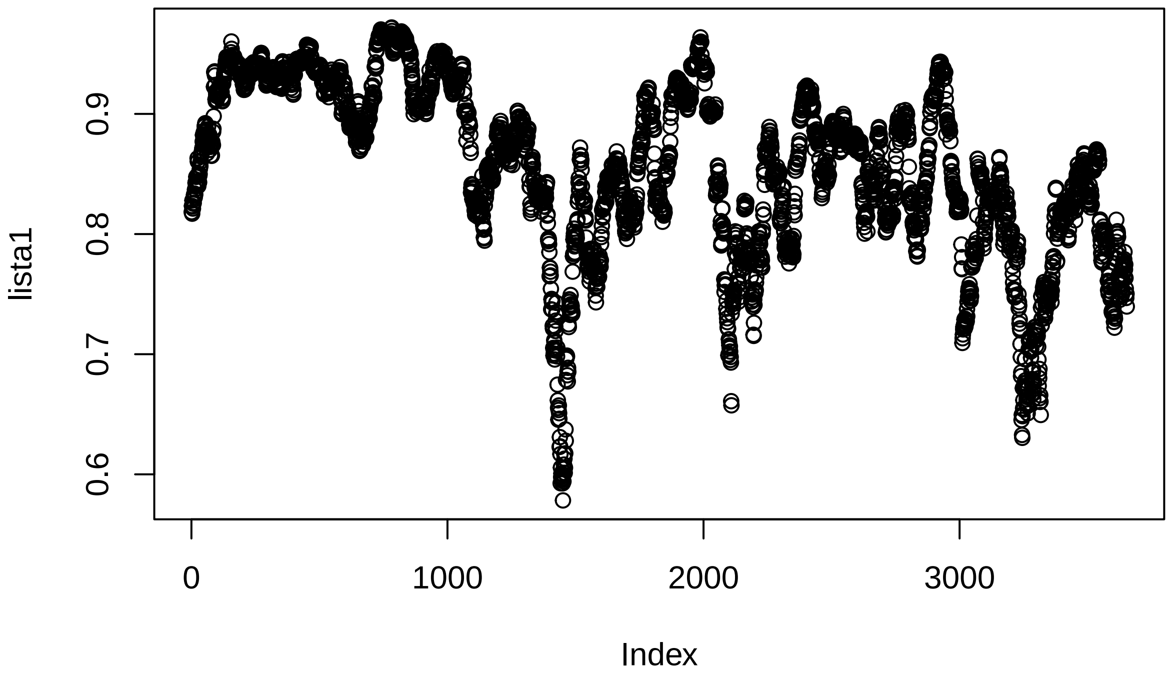
```
p_EEM = log(window(EEMd, end="2023-03-31")) #log-prices
p_EEM = as.vector(p_EEM$EEM.Adjusted[,1])
T = length(p_EEM)
r_EEM=100*(p_EEM[2:T]-p_EEM[1:T-1]) #log percentage returns
yEEM = r_EEM - mean(r_EEM)

specataEEM <- ugarchspec(variance.model = list(model="apARCH", garchOrder = c(1, 1)),
                         mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
                         distribution.model="sstd")
fitataEEM <- ugarchfit(specataEEM, r_EEM)
resstataEEM <- residuals(fitataEEM, standardize=TRUE)
fitataEEM@fit$robust.matcoef
```

```
##           Estimate  Std. Error   t value     Pr(>|t|)
## omega   0.04300448  0.00939816  4.575841 4.743115e-06
## alpha1  0.08010815  0.01000174  8.009419 1.110223e-15
## beta1   0.90092739  0.01252200 71.947578 0.000000e+00
## gamma1  0.50018290  0.08365126  5.979382 2.239851e-09
## delta   1.57862260  0.18376935  8.590239 0.000000e+00
## skew    0.86463210  0.01831313 47.213770 0.000000e+00
## shape  10.67398989  1.57623662  6.771820 1.271738e-11
```

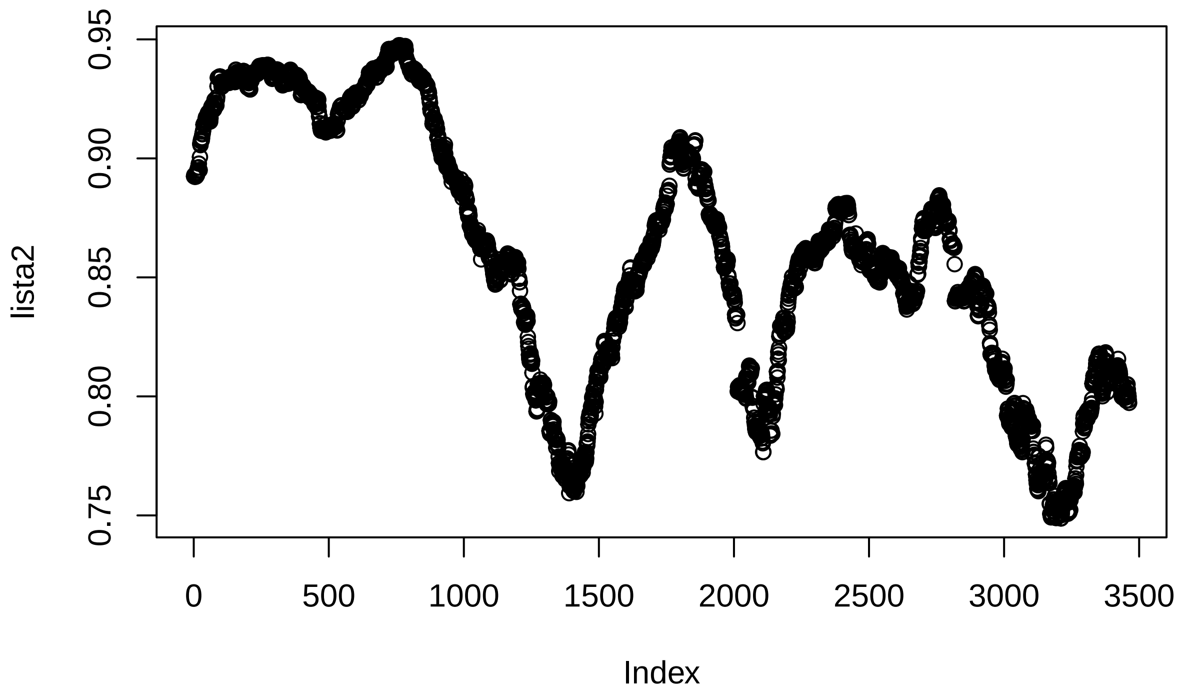
Below are the for loops that calculate the correlations between the standardized residuals of the two financial instruments, reciprocally with windows of 62 and 252 observations.

Thus about a quarter of data and a year of data.

```
lista1=c()
for (i in 1:(min(length(yEEM),length(yVT))-62)){
  lista1[i]=cor(as.vector(residuals(fitataVT,standardize=T)[i:(i+61)]),
                as.vector(residuals(fitataEEM,standardize=T)[(i+1310):(i+1310+61)]))
}
plot(lista1)
```



```
lista2=c()
for (i in 1:(min(length(yEEM),length(yVT))-252)){
  lista2[i]=cor(as.vector(residuals(fitataVT,standardize=T)[i:(i+251)]),
                as.vector(residuals(fitataEEM,standardize=T)[(i+1310):(i+1310+251)]))
}
plot(lista2)
```

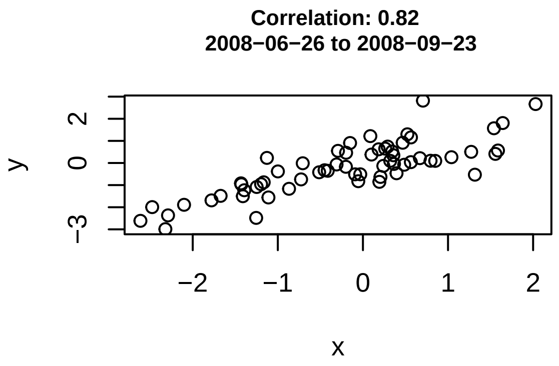


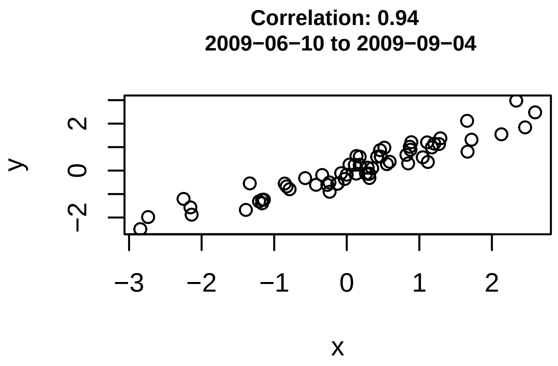
As expected when using the larger window, the trend is smoother.

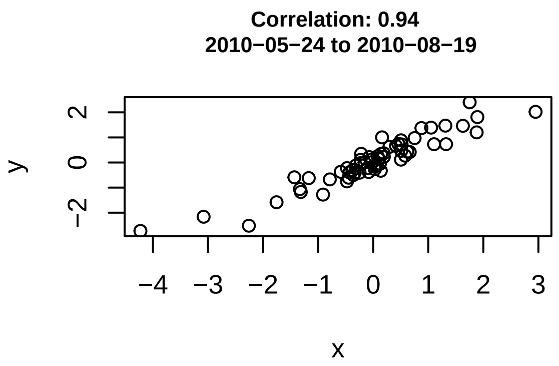
It may also be interesting to look at the plots between the residuals at different time points.

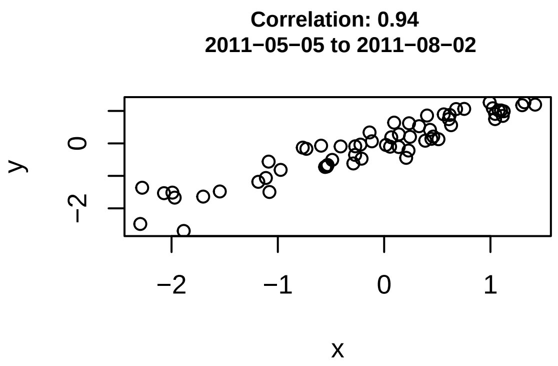
We will take as an example the window of 62 observations where more differences can be noted.

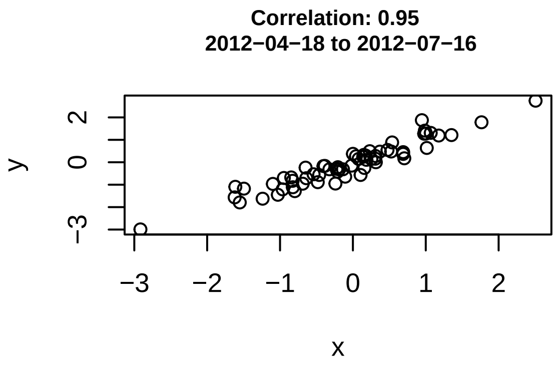
```
par(mfrow = c(2,2))
for (i in seq(from = 1, to = 3653, by = 240)) {
  x <- as.vector(residuals(fitataVT, standardize = TRUE)[i:(i + 61)])
  y <- as.vector(residuals(fitataEEM, standardize = TRUE)[(i + 1310):(i + 1310 + 61)])
    start_date <- as.character(time(VTd)[i])
  end_date <- as.character(time(VTd)[i + 61])
    titolo <- paste("Correlation:", round(cor(x, y), 2), "\n", start_date, "to", end_date)
    plot(x, y, main = titolo, cex.main = 0.8)
}
```

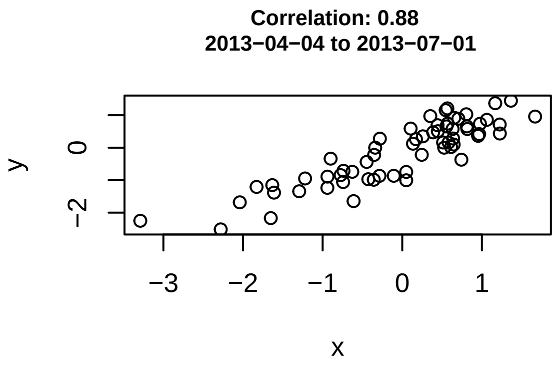

Correlation: 0.82
2008−06−26 to 2008−09−23
y
−3
0
2
−2
−1
0
1
2
x

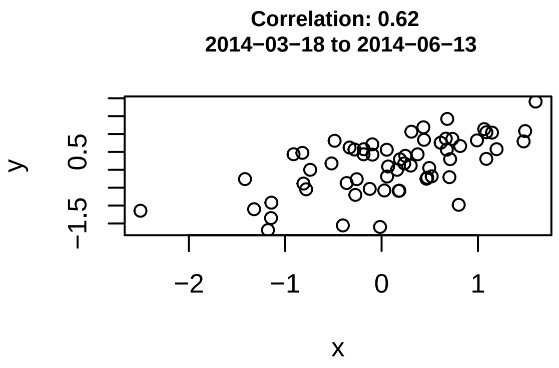

Correlation: 0.94
2009−06−10 to 2009−09−04
y
−2
0
2
−3
−2
−1
0
1
2
x

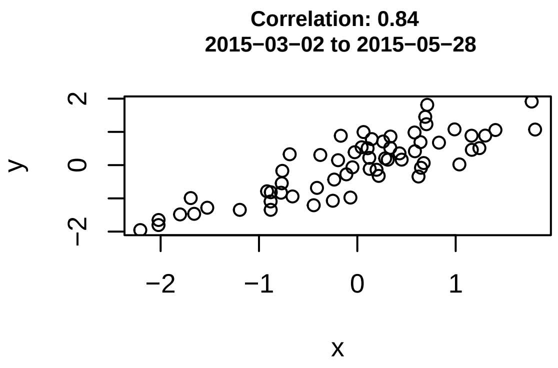

Correlation: 0.94
2010−05−24 to 2010−08−19
y
−2
0
2
−4
−3
−2
−1
0
1
2
3
x

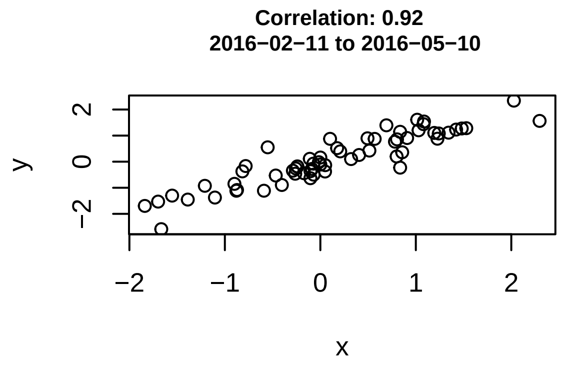

Correlation: 0.94
2011−05−05 to 2011−08−02
y
−2
0
−2
−1
0
1
x

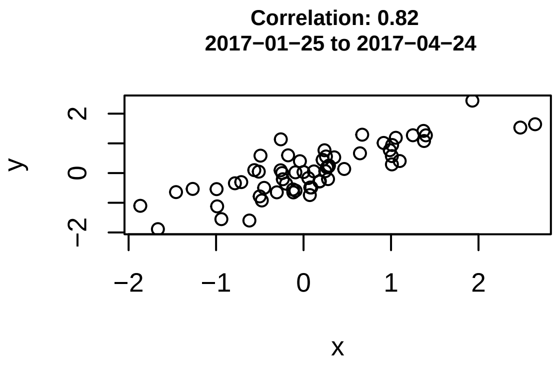

Correlation: 0.95
2012−04−18 to 2012−07−16
y
−3
0
2
−3
−2
−1
0
1
2
x

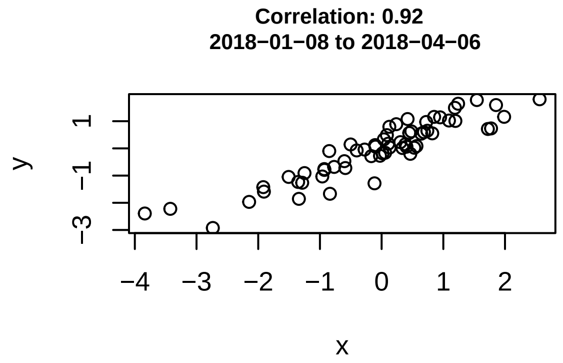

Correlation: 0.88
2013−04−04 to 2013−07−01
y
−2
0
−3
−2
−1
0
1
x

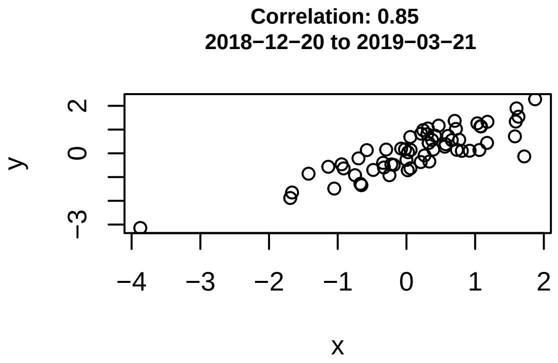

Correlation: 0.62
2014−03−18 to 2014−06−13
y
−1.5
0.5
−2
−1
0
1
x

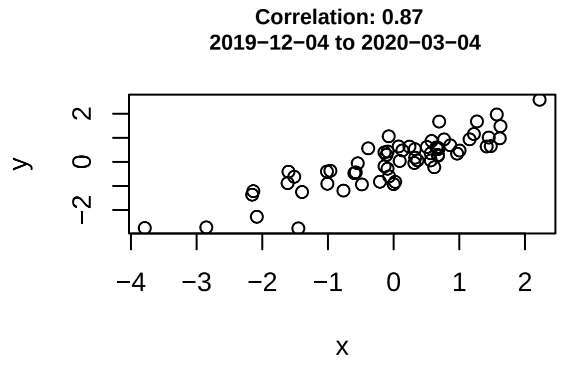

Correlation: 0.84
2015−03−02 to 2015−05−28
y
−2
0
2
−2
−1
0
1
x

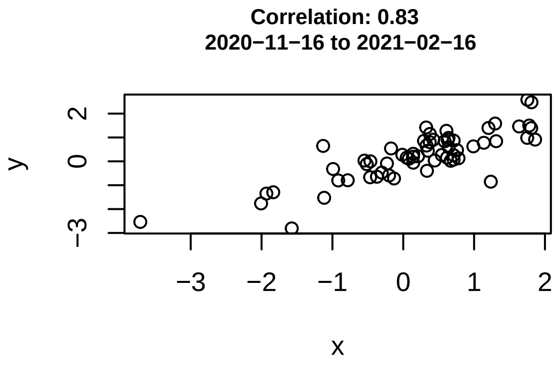


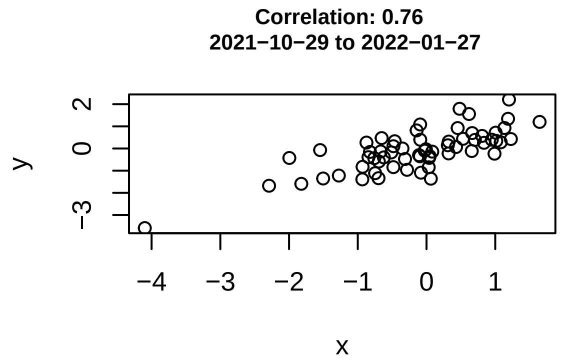


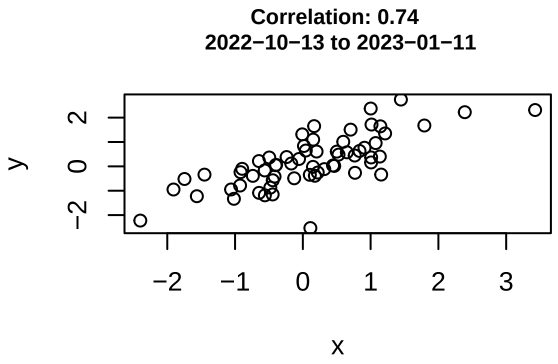






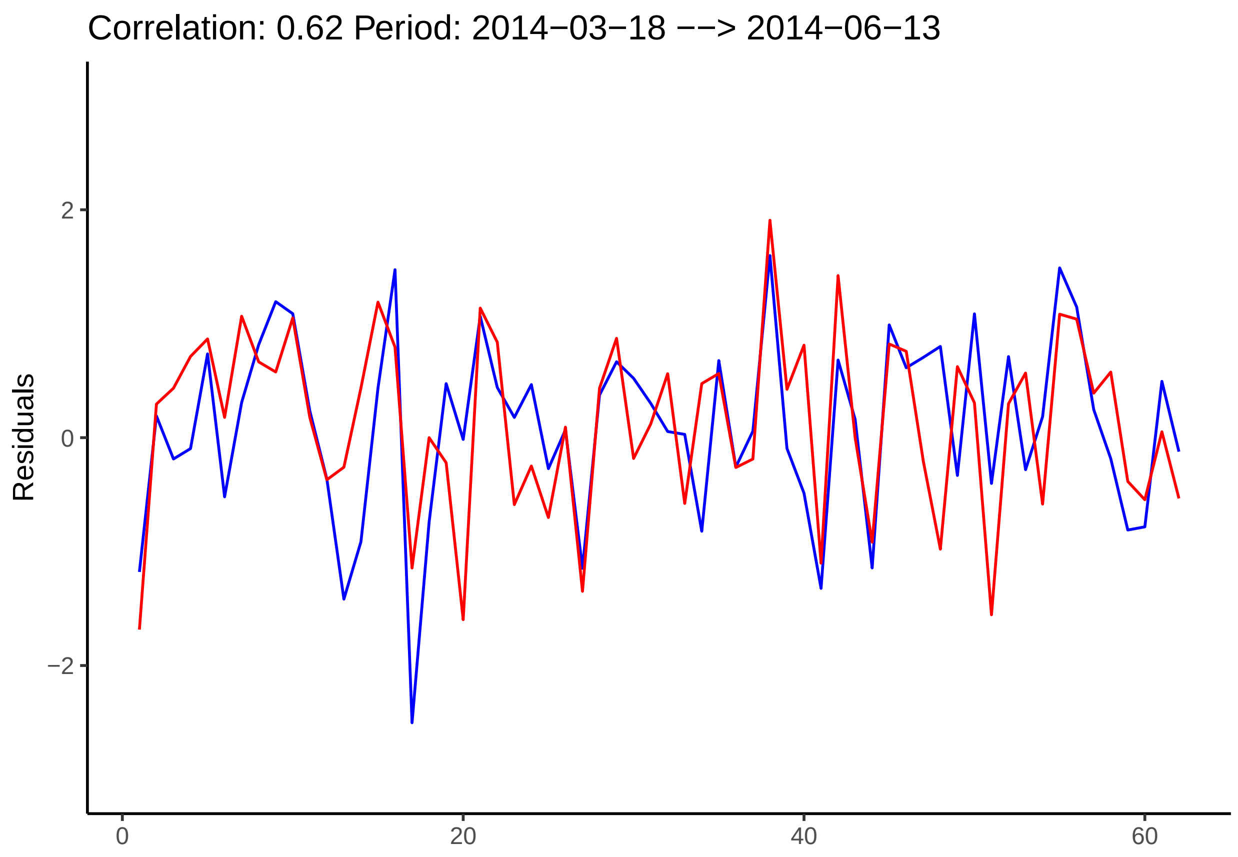






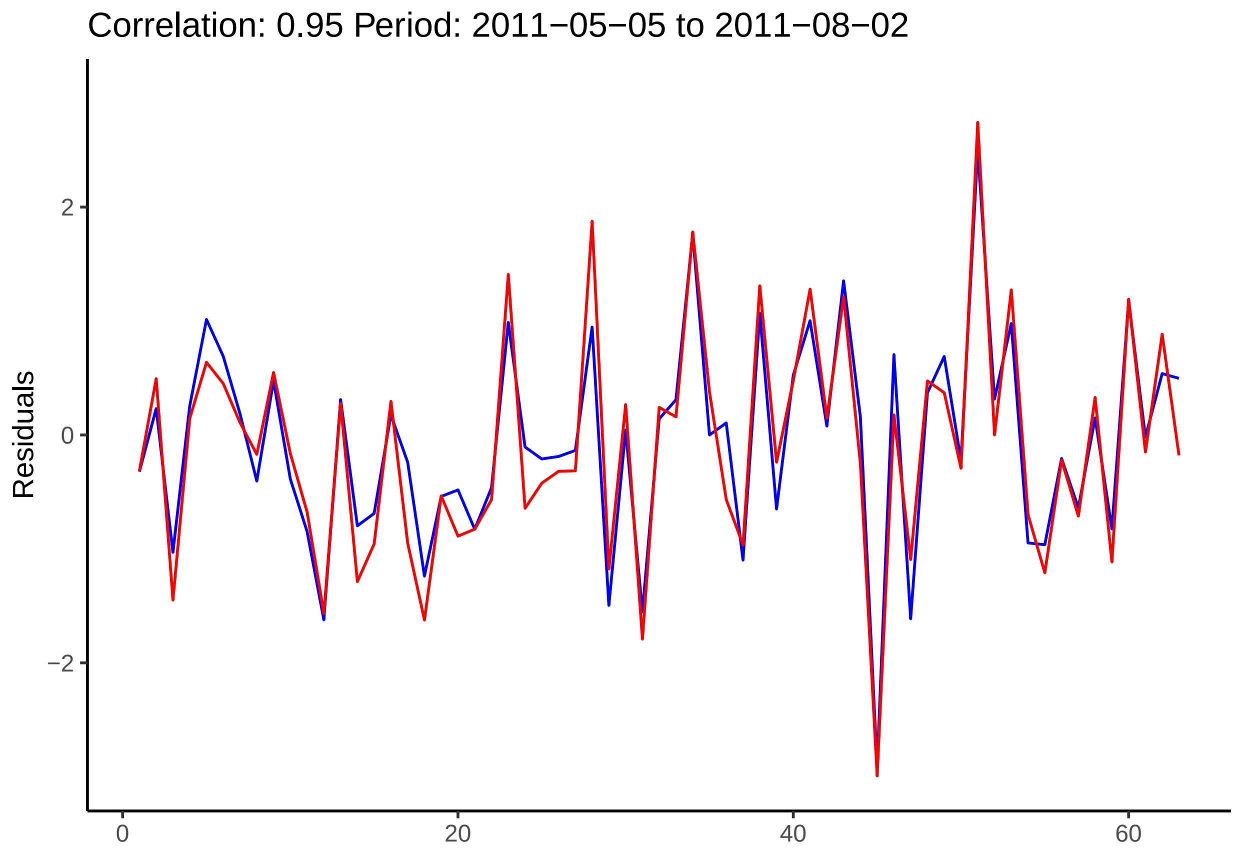
Below are the residuals compared for the lower correlation of those seen in the graphs above, namely the one that goes from 2014-03-18 to 2014-06-13:

```
par(mfrow=c(1,1))
library(ggplot2)
residuals_VT <- residuals(fitataVT, standardize = TRUE)[1441:1502]
residuals_EEM <- residuals(fitataEEM, standardize = TRUE)[2751:2812]
data <- data.frame(residuals_VT, residuals_EEM)
ggplot(data) +
  geom_path(aes(x = seq_along(residuals_VT), y = residuals_VT), color = "blue") +
  geom_path(aes(x = seq_along(residuals_EEM), y = residuals_EEM), color = "red") +
  labs(y = "Residuals", title = "Correlation: 0.62 Period: 2014-03-18 --> 2014-06-13") +
  ylim(-3, 3) +
  theme_classic() +
  theme(axis.title.x = element_blank())
```



While to conclude, the residuals compared for the highest correlation seen previously, covering the period from May 5, 2011, to August 2, 2011:

```
par(mfrow=c(1,1))
library(ggplot2)
residuals_VT <- residuals(fitataVT, standardize = TRUE)[961:1023]
residuals_EEM <- residuals(fitataEEM, standardize = TRUE)[2271:2333]
data <- data.frame(residuals_VT, residuals_EEM)
ggplot(data) +
  geom_path(aes(x = seq_along(residuals_VT), y = residuals_VT), color = "blue") +
  geom_path(aes(x = seq_along(residuals_EEM), y = residuals_EEM), color = "red") +
  labs(y = "Residuals", title = "Correlation: 0.95 Period: 2011-05-05 to 2011-08-02") +
  ylim(-3, 3) +
  theme_classic() +
  theme(axis.title.x = element_blank())
```



And it is observed that the two series of residuals are practically superimposed.