ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI SCIENZE STATISTICHE

Corso di Laurea in Statistica, Finanza e Assicurazioni

Tesi di laurea Triennale

Time series and volatility analysis on Nasdaq100 with ARIMA-GARCH approach and Markov-switching approach

Relatore:

Prof. Andrea Guizzardi

Candidato:

Matteo Ferniani

matricola 941863

Anno Accademico 2021-2022

Index

Introduction

Volatility plays an important role both for those who operate in financial markets and for those who study their characteristics and trends. The GARCH model describing the conditional variance of a financial process has been particularly popular among those who operate in the markets and need to attribute a value to volatility. Its theoretical approach, combined with an easy implementation, is at the basis of its frequent use in operational realities. Nevertheless, research activity has highlighted the limits of GARCH in modeling structural breaks often present in financial time series. The literature has ended up suggesting several solutions to the problem of excessive persistence of GARCH, both proposing variations and specifications, and new more complex solutions. Markov Switching GARCHs, with the introduction of a latent variable for the process describing volatility but preserving the classical structure of GARCHs within states, in this sense are models on the border between these two classes. The theoretical development and applications proposed in the literature have confirmed the better performance of MS-GARCH over GARCH in terms of volatility forecasting capability and persistence in parametric terms (source). The empirical analysis proposed in the second part of the thesis proposes, instead, a verification of the ability to specify shocks present in the time series of MS-GARCH models. The hypothesis is that structural breaks occur in the phases in which the model jumps from a regime of standard volatility to one with higher volatility. Further generalization of these models is also proposed by considering the assumption of specifying different density distributions for each volatility regime.

(Source for introduction) Modelli MS-GARCH e Break strutturali: sviluppi teorici ed evidenze empiriche Dott. Guido Caporilli Razza

Comparisons will be made on the various classical methods of volatility analysis, and we will try to specify the best model for both the mean and the variance conditional.

The main goal of this paper is to explain the behaviour of financial time series and the relationship between returns and conditional volatility.

Do ARIMA and GARCH processes provide parsimonious approximations to mean and volatility dynamics?

Theory

Before you start modelling the Nasdaq100 time series, we need to know the theoretical basis of time series analysis

<u>Asset returns</u>:

• Simple return:

$$R_t \triangleq \frac{p_t - p_{t-1}}{p_{t-1}} = \frac{p_t}{p_{t-1}} - 1.$$

• Log-return:

$$r_t \triangleq y_t - y_{t-1} = \log \frac{p_t}{p_{t-1}} = \log (1 + R_t)$$

<u>**Stylized facts of financial data**</u>:

◊ Lack of stationarity: past returns do not necessarily reflect future performance

◊ Absence of autocorrelations: autocorrelations of returns are often insignificant

◊ Heavy tails: Gaussian distributions generally do not hold in financial data

◊ Gain/loss asymmetry

◊ Aggregational Gaussianity: for lower frequencies, the distribution tends to become more Gaussian.

◊ Volatility clustering: high-volatility evens tend to cluster in time

we must take these stylized facts into account when modeling financial time series

For the time series modeling and forecasting (mean models) I used ARIMA models.

They aim to describe the autocorrelations in the data.

<u>VAR</u> (Vector Auto-Regressive) model of order p is

$$\mathbf{r}_t = \phi_0 + \sum_{i=1}^{p} \mathbf{\Phi}_i \mathbf{r}_{t-i} + \mathbf{w}_t,$$

where p is a nonnegative integer and

◊ the vector $\varphi_0 \in R^N$ and the matrices $\Phi_i \in R^{N \times N}$ are parameters,

◊ w_t is a white noise series with zero mean and constant covariance matrix Σ_w.

‘* VAR (vector autoregression) is a generalization of AR (autoregressive model) for multiple time series, identifying the linear relationship between them. The AR can be seen as a particular case of VAR for only one serie.
The hypothesis necessary to apply the VAR is the series is just that one influence other in a intertemporal way
Source: StackExchange

VARMA (Vector Auto-Regressive and Moving Average) model is

$$\mathbf{r}_t = \phi_0 + \sum_{i=1}^{p} \mathbf{\Phi}_i \mathbf{r}_{t-i} + \mathbf{w}_t - \sum_{j=1}^{q} \mathbf{\Theta}_j \mathbf{w}_{t-j},$$

where p and q are nonnegative integers and

◊ the vector $\varphi 0 \in$ RN and the matrices Φi, $\Theta j \in$ RN×N are parameters,

◊ wt is a white noise series with zero mean and constant covariance matrix Σw.

Order selection of models

two common approaches:

◊ cross-validation: splitting the data into a training part and a cross-validation part, the latter being used to test the model trained with the training data for different combinations of orders.

◊ penalized estimation methods: penalizing the number of parameters of the model with a penalty term like: AIC, BIC.

ARIMA(p,d,q) model

◊ A multivariate time series yt is said to be a ARIMA(p,1,q) process if it is nonstationary but after differencing the series times xt = yt − yt−1 then xt follows a stationary ARMA(p,q) model.

◊ More generally, a ARIMA(p,d,q) process has to be differenced d times to obtain a stationary ARMA(p,q) process.

◊ In finance, price series pt (or log-prices yt = log (pt)) are believed to be nonstationary, but the log-return series rt = yt − yt−1 = log (pt) − log (pt−1) is stationary.

◊ Thus, it is the same to talk about a ARIMA(p,1,q) log-price series and about a ARMA(p,q) log-return series.

Volatility clustering

As we know from financial stylized facts, the volatility (i.e., the square root of conditional variance) is clustered

Before starting with complicated models, I considered simpler ways.

- Rolling means (moving average) of the squared returns:

$$\sigma_t^2 = \frac{1}{m}\sum_{i=1}^{m} w_{t-i}^2$$

- EWMA of the squared returns:

$$\sigma_t^2 = \alpha w_{t-1}^2 + (1-\alpha)\sigma_{t-1}^2$$

ARCH models for dealing with volatility clustering

$$w_t = \sigma_t z_t,$$

where zt is a white noise series with zero mean and constant unit variance, and the conditional variance σ2t is modeled by

$$\sigma_t^2 = \omega + \sum_{i=1}^{m} \alpha_i w_{t-i}^2$$

$\omega > 0,\ \alpha_i \geq 0$ for all $i > 0$.

A limitation of the ARCH model is that the high volatility is not persistent enough. This can be overcame by the Generalized ARCH (GARCH) model.

$$w_t = \sigma_t z_t,$$

where zt is a white noise series with zero mean and constant unit variance, and the conditional variance σ2t is modeled by

$$\sigma_t^2 = \omega + \sum_{i=1}^{m} \alpha_i w_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2.$$

$\omega > 0,\ \alpha_i \geq 0,\ \beta_j \geq 0$ for all $i > 0$ and $j > 0$ and $\sum_{i=1}^{m} \alpha_i + \sum_{j=1}^{s} \beta_j \leq 1.$

Source: Daniel P. Palomar slides

A problem with GARCH model is that they tend to overestimate volatility forecasts, particularly during the periods of high volatility. The reason behind this is the high degree of persistence implied by the GARCH model.

To overcome this issue, a Markov-Switching process can be employed to model changes in the parameters.

The **Markov-Switching model** has the advantage of being used in the information estimation about the varying regime switching probabilities of being in a particular regime instead of a linear model that would have to be estimated for each regime completely and separately.

Markov Regime-Switching (MRS) models are characterized by the possibility for some or all of the parameters to change across several states according to a Markov process, which is governed by a state variable S_t .

The transition probability represents the probability of switching from state i at t − 1 to state j at t:

$\Pr(s_t = j \mid s_{t-1} = i) = P_{ij}$

If we consider, for simplicity, that there are only 2 states, the transition matrix can be written as:

$$P = \begin{pmatrix} \rho_{11} & \rho_{21} \\ \rho_{12} & \rho_{22} \end{pmatrix} = \begin{pmatrix} \rho & 1-q \\ 1-\rho & q \end{pmatrix}$$

In our study we let $\{S_t\}$ follow a two-state first-order Markov process with timevarying transition probabilities and write this as

$$\begin{aligned} \Pr(S_t = 1 \mid S_{t-1} = 1,\ r_t) &= P_t, \\ \Pr(S_t = 2 \mid S_{t-1} = 1,\ r_t) &= 1 - P_t, \\ \Pr(S_t = 1 \mid S_{t-1} = 2,\ r_t) &= 1 - q_t, \\ \Pr(S_t = 2 \mid S_{t-1} = 2,\ r_t) &= q_t. \end{aligned}$$

The probability of being in regime i for i = 1, 2 depends on realizations in $\tilde{r}_t$ and $\{S_t\}$ only through S_{t-1}. For the time-varying transition probabilities we assume

$$P_t = \Phi\left(d_1 + e_1.r_t\right),$$
$$q_t = \Phi\left(d_2 + e_2.r_t\right).$$

With Φ (0) denoting the cumulative distribution function and $d_1,\ d_2,\ e_1,\ e_2$ representing parameters to be estimated from the data. A general form of the MRS-GARCH models can be written as:

$$r_t\,|\phi_{t-1} \sim \begin{cases} f\left(\theta_t^{(1)}\right) & \textit{with probability } p_{1t} \\ f\left(\theta_t^{(2)}\right) & \textit{with probability } (1 - p_{1t}) \end{cases}$$

where, $f(*)$ stands for one of the possible conditional distributions used in the model (normal, Student-t, GED), $\theta_t^{(i)}$ is the vector of parameters in the *ith* regime that characterizes the distribution, $p_{1t} = \Pr(s_t = 1\,|\phi_{t-1})$ is the ex-ante probability and ϕ_{t-1} is the information set at $t-1$.

The vector of time-varying parameters θ, can be written as a function of three components:

$$\theta_t^{(i)} = \left(\mu_t^{(i)},\ h_t^{(i)},\ v_t^{(i)}\right)$$

where, $\mu_t^{(i)} = E(r_t\,|\phi_{t-1})$ is the conditional mean, $h_t^{(i)} = Var(r_t\,|\phi_{t-1})$ is the conditional variance, and $v_t^{(i)}$ is the shape parameter of the conditional distribution.

we consider the two-regime case.

The corresponding probability density function(of conditional probability distribution of rt+1) has the form

$$\begin{aligned} f\left(r_{t+1}\,|\,\phi_t\right) &= \sum_{i=1}^{2} f\left(r_{t+1}, S_t = i\,|\,\phi_t\right) \\ &= \sum_{i=1}^{2} \Pr\left(S_t = i\,|\,\phi_t\right).f\left(r_{t+1}\,|\,S_t = i,\ \phi_t\right) \\ &= \sum_{i=1}^{2} p_{i,t}.f\left(r_{t+1}\,|\,S_t = i,\ \phi_t\right), \end{aligned}$$

where, $p_{i,t} \equiv \Pr\left(S_t = i\,|\,\phi_t\right)$ denotes the ex-ante regime-i probability. The information set ϕ_t consists of the entire history of $\tilde{r}_t = \{r_t,\ r_{t-1},\ldots\}$.

Since the regime indicator St follows a first-order Markov process the ex-ante probability pi,t depends only on St−1 and rt . Using the Theorem of Total Probabilities:

$$p_{i,t} = \sum_{j=1}^{2} \Pr\left(S_t = i\,|\,S_{t-1} = j,\ \tilde{r}_t\right)\Pr\left(S_{t-1} = j\,|\ \tilde{r}_t\right)$$

As we are assuming conditional normality f is given as follows:

$$f\left(r_{t+1}\,|\,S_t = i,\ \phi_t\right) = \frac{1}{\sqrt{2\pi h_{i,t}}}\exp\left\{-\frac{\left[r_{t+1} - \left(\lambda_i + \gamma_i\sqrt{h_{i,t}}\right)\right]^2}{2h_{i,t}}\right\}.$$

The variance $h_{i,t}$ depends on the explicit functional form of the GARCH equation.

$$h_{i,t} = \begin{cases} \left[w_i + \alpha_i\sqrt{h_{t-1}^{(i)}}^{\mu_i} f_i^{v_i}\left(\delta_t^{(i)}\right) + \beta_i\sqrt{h_{t-1}^{(i)}}^{\mu_i}\right]^{2/\mu_i} & \textit{for } \mu_i > 0 \\ \left[\exp\left\{w_i + \alpha_i f_i^{v_i}\left(\delta_t^{(i)}\right) + \beta_i \ln\left(\sqrt{h_{t-1}^{(i)}}\right)\right\}\right] & \textit{for } \mu_i = 0 \end{cases}$$

with appropriately defined parameters $w_i,\ \alpha_i,\ \beta_i$.

As is the case with the standard GARCH models. Student-t and GED are widely used alternatives in financial studies using GARCH-type models. Both of those two distributions can capture leptokurtic and heavy-tail behaviors. When they are applied to the MRS-GARCH model, the corresponding density functions are changed as follows:

$$\text{Student's } t : f\left(\varepsilon_t \mid s_t = j, \theta, \phi_{t-1}\right) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\Gamma\left(\frac{v}{2}\right)\sqrt{\pi(v-2)h_{j,t}}}\left[1 + \frac{\varepsilon_t^2}{(v-2)h_{j,t}}\right]^{\frac{v+1}{2}}$$

$$GED : f\left(\varepsilon_t \mid s_t = j, \theta, \phi_{t-1}\right) = \frac{v e^{-\frac{1}{2}\left|\frac{\varepsilon_t}{\lambda\sqrt{h_{j,t}}}\right|^{v}}}{\lambda 2^{\frac{v+1}{v}}\Gamma\left(\frac{1}{v}\right)}$$

where, $\lambda = \left[\frac{2^{\left(-\frac{2}{v}\right)}\Gamma(1/v)}{\Gamma(3/v)}\right]^{1/2}$.

Source: Mehdi Zolfaghari, Bahram Sahabi

Quantitative risk management:

Value-at-risk (VaR)

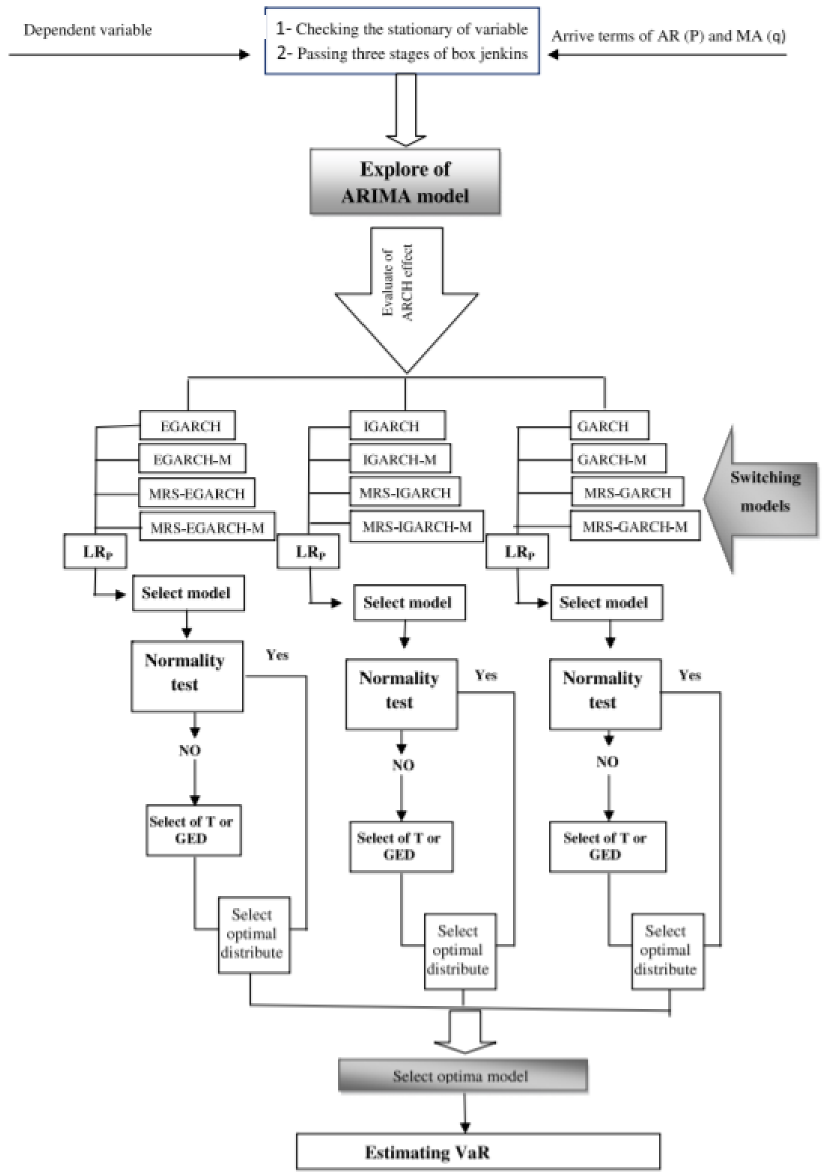
The VaR measures the threshold value such that the probability of observing a loss larger or equal to it in a given time horizon is equal to α.

Formally, the VaR forecast in $T+1$ at risk level α (given the information set up to at time T) is defined as:

$$\text{VaR}_{T+1}^{\alpha} \equiv \inf\left\{y_{T+1} \in \mathbb{R} \mid F(y_{T+1} \mid \mathcal{I}_T) = \alpha\right\} ,$$

Source: Journal of Statistical Software

Methodology



This image taken from Mehdi Zolfaghari, Bahram Sahabi perfectly summarizes the metdology used epr this work,

In the first part in fact, I tried to make a fit of a hybrid ARIMA-GARCH model, trying different combinations of GARCH, having already specified the mean model.

Hybrid ARIMA-GARCH Model

First stage, we use the best ARIMA model that fits on stationary and linear time series data while the residuals of the linear model will contain the non-linear part of the data. Second stage, we use the GARCH model in order to contain non-linear residuals patterns.

Based on ADF test, we check if the series is stationary. Modelling the series consists of two phases according to the Box and Jenkins process. The first phase involves the specification of the ARMA (p, q) model for mean returns, and the diagnostic tests of their residuals. The second phase is the specification of our strategy for modelling the conditional mean to search over alternative ARMA (p, q) models by varying the p and q parameters using the autocorrelation (AC) and partial autocorrelation (PAC) functions corresponding to the stages described by Box and Jenkins and identifying the best fitted model using the Schwartz Information Criterion (SIC) proposed by Schwarz (1978). SIC uses a likelihood function to select the best fitted model. This criterion represents a trade-off between “fit” measured by the log likelihood value and “parsimony” as measured by the number of free parameters. After estimating the ARMA model, the “ARCH effect” is tested on it based on F and $\chi 2$ statistics. If there is an ARCH effect we model the conditional variance of the series with GARCH models under the normal, Student t, and GED distributions. I used AIC, BIC criteria to choose the best fitted model among the ARMA-GARCH family models.

Diagnostic Checking of Hybrid ARIMA-GARCH Model

The diagnostic tests of hybrid ARIMA-GARCH models are based on residuals.

Residuals’ normality test is employed with Jarque and Bera (1980) test. Ljung and Box (1978) (Q-statistics) statistic for all time lags of autocorrelation is used for the serial correlation test.

Forecast Evaluation

$$\hat{Y}_t(l) = E\left(Y_{t+l} \mid Y_t, Y_{t-1}, \dots\right) = \phi_0 + \sum_{i=1}^{p} \phi_i Y_{t+l-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t+l-j}$$

Where the εs follow the stated GARCH model

To evaluate the forecast efficiency, we use two statistical measures

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{pi} - y_{ai}\right)^2},$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left(\left|\frac{y_{pi} - y_{ai}}{y_{ai}}\right| \times 100\right),$$

where yai and ypi represent the actual and predicted values, respectively. The evaluation criteria of prediction accuracy (RMSE and MAPE) are used to examine the prediction accuracy of different models.

In the second part I made the fit of an MSGARCH model, helping with the functions of packages MSGARCH and SBAGM .

Then I back tested the VaR 5% and seen if switching between periods really improved the model's quality.

After that I built* a trading strategy to prove that the ARIMA-GARCH models beat the buy and hold strategy.

Another to test the superiority of the MSGARCH models.

‘* Actually, due to coding difficulties I taken all the code, pics and results from R-bloggers

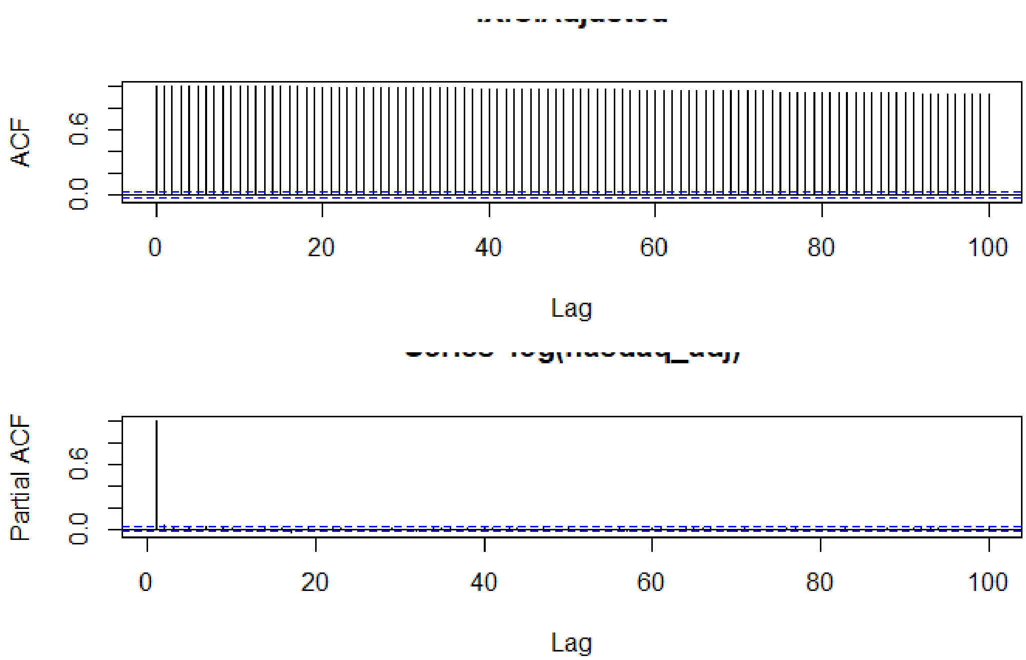
Data

exploratory analysis

Chart of Nasdaq100 series:

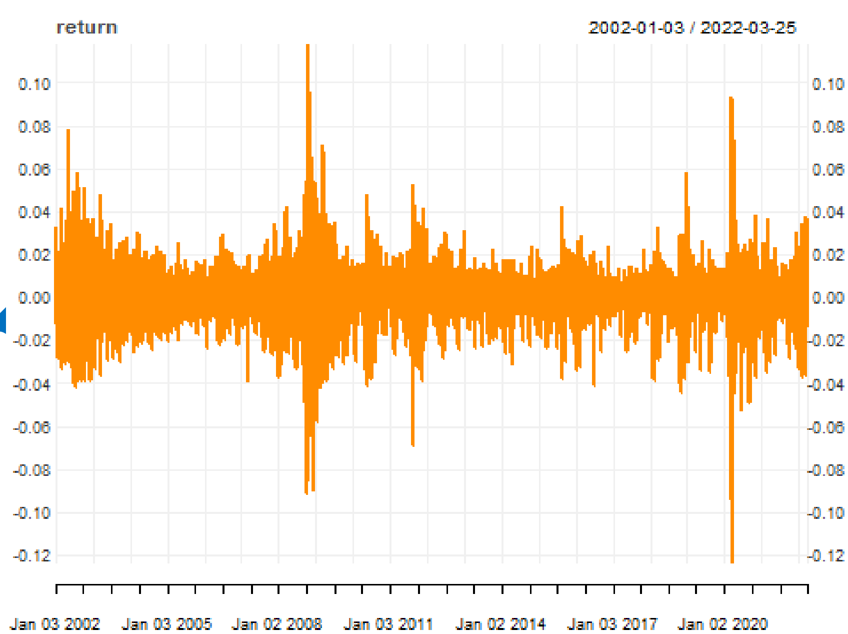
source: Yahoo Finance from 2002-01-01 to 2022-03-27

```
ACF and PACF plot for 100 lags of the log adjusted returns of Nasdaq100:
```



From the ACF plot, we observe that the plot decays to zero slowly, meaning the shock affects the process permanently. We can conclude that we need to perform time series analysis on the daily return (log return) of the stock prices.

```
Chart of the log returns:
```

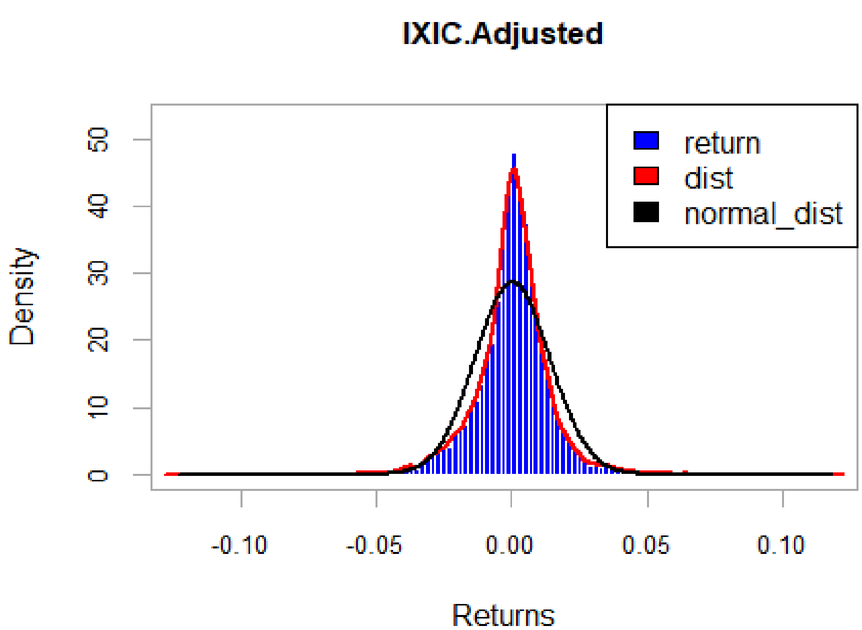


The log returns of the stock prices have zero mean but high volatility, standard stationarity won't work.

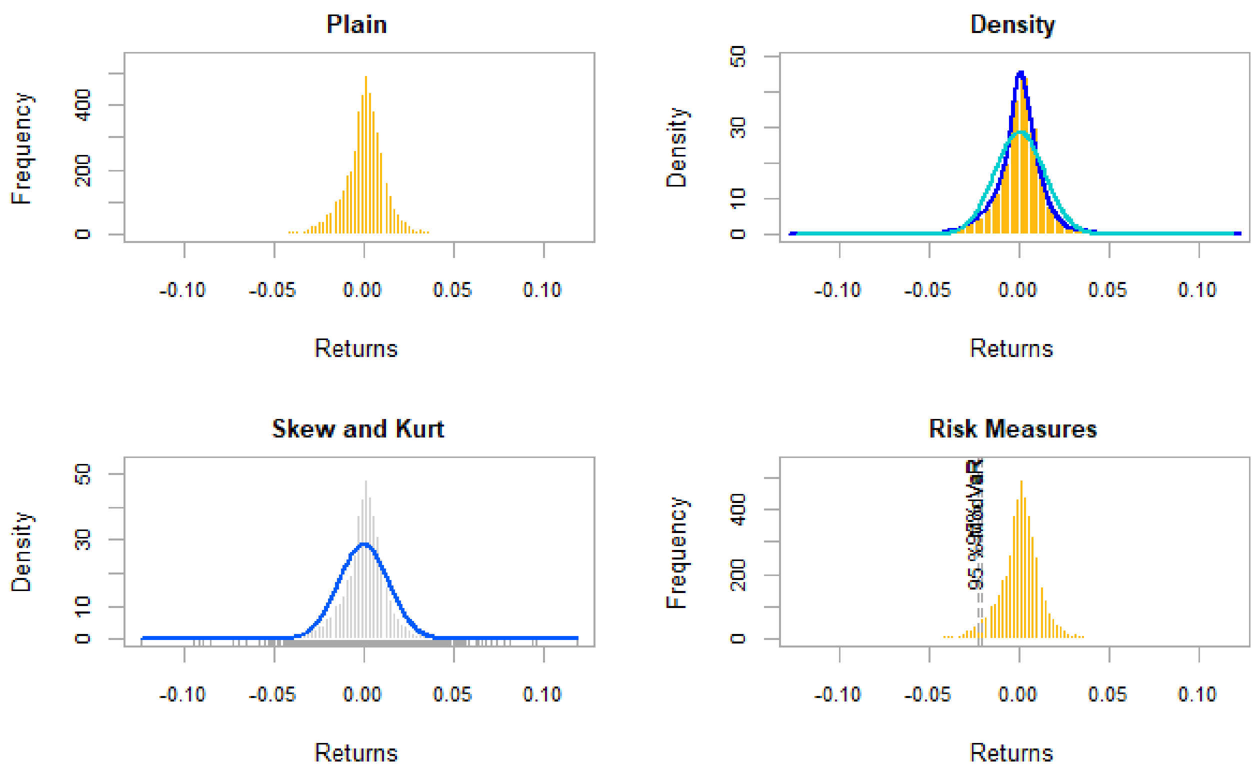
```
Basic stats of the returns:

##               IXIC.Adjusted
## nobs            5093.000000
## NAs                0.000000
## Minimum           -0.123213
## Maximum            0.118059
## 1. Quartile       -0.005530
## 3. Quartile        0.007297
## Mean               0.000483
## Median             0.000987
## Sum                2.461913
## SE Mean            0.000195
## LCL Mean           0.000101
## UCL Mean           0.000865
## Variance           0.000193
## Stdev              0.013906
## Skewness          -0.133247
## Kurtosis           6.808118
```

From the basic statistics of the log return of the stock prices we observe that the mean is 0 and the distribution of log returns has large kurtosis (fat tails).

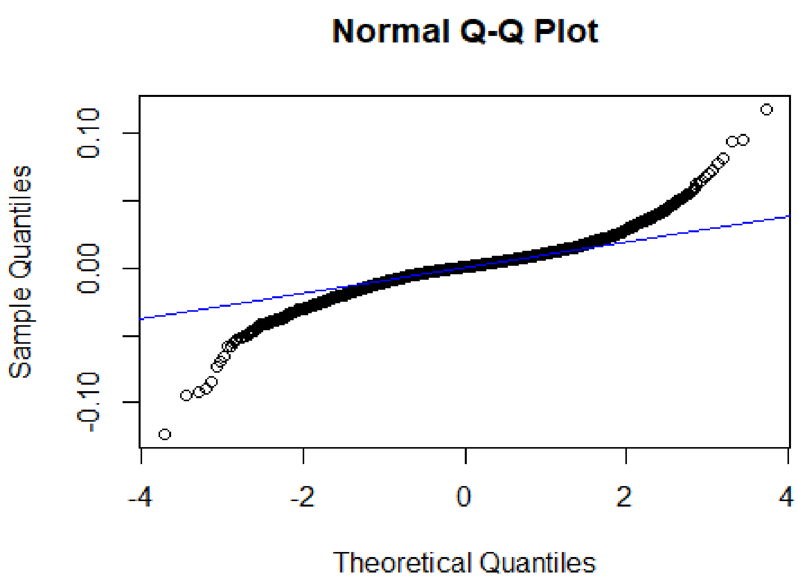
```
Histogram of the returns:
```



Charts with density (normal -vs- actual), skewness, kurtosis and a risk measure:

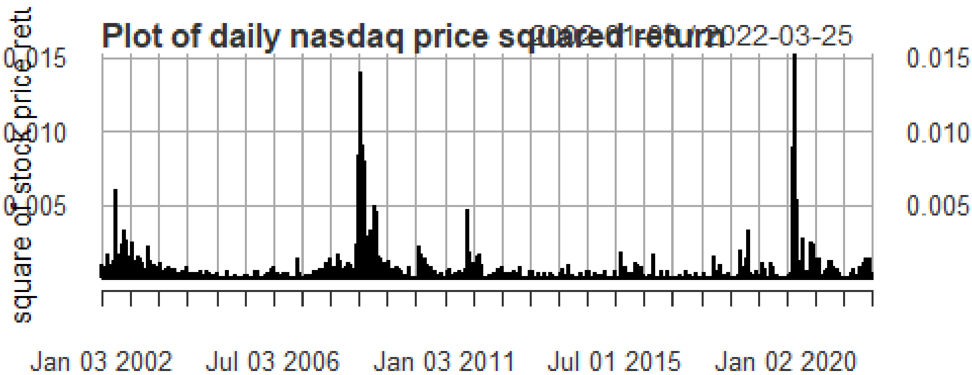


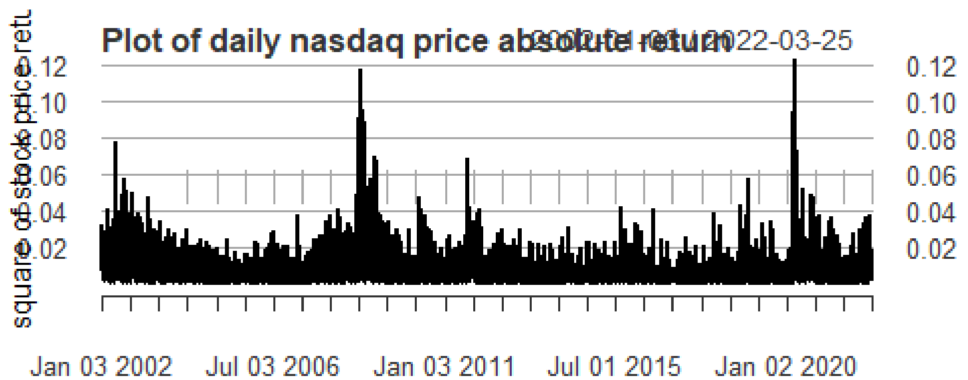
Returns’ QQplot:



The graphs confirm that seems to be more skewed than the normal distribution. The series has a somewhat normal distribution with fat tails at both ends

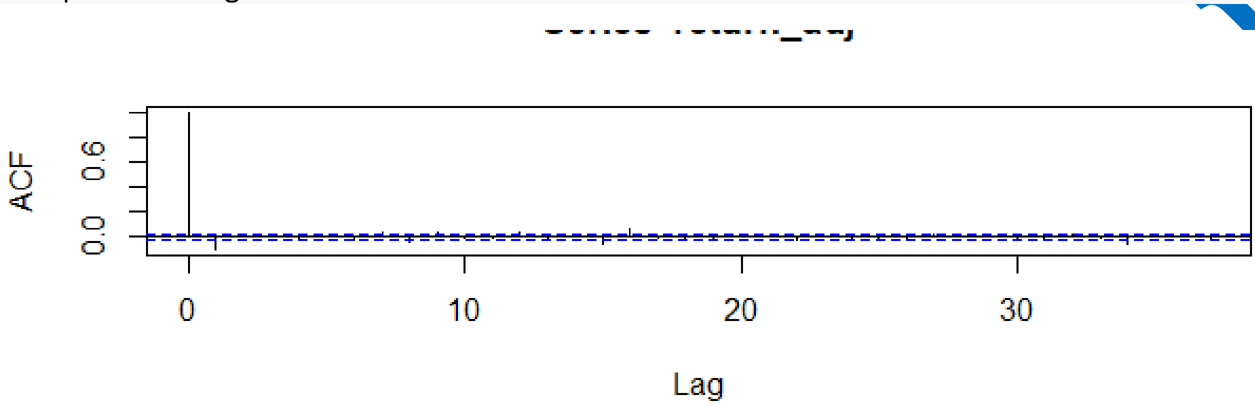
```
plot of returns squared and absolute:
```

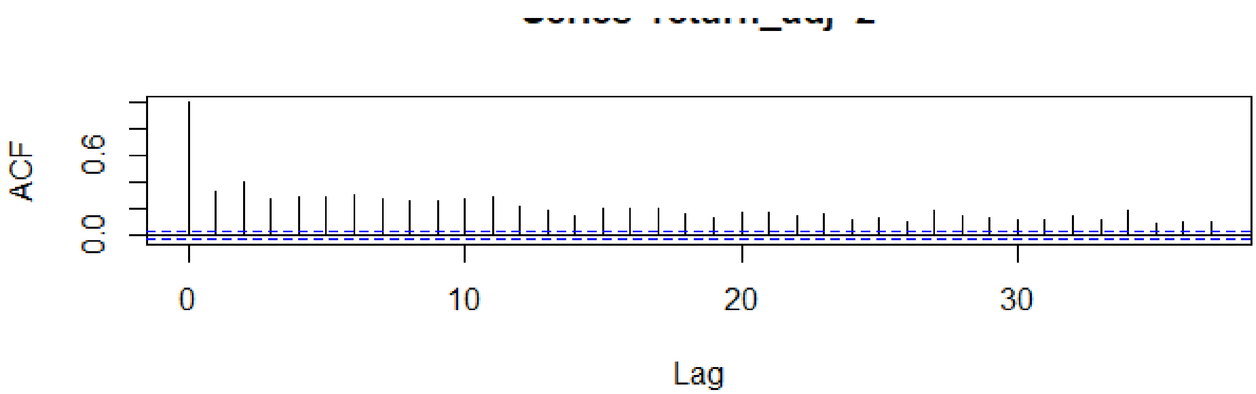




During the years 2002, 2008-2009, 2013 and 2020, there is spike in volatility indicating non-constant conditional volatility. The high volatility doesn't decrease fast because of the negative shocks have an effect on the process.

```
ACF plot of log returns:
```





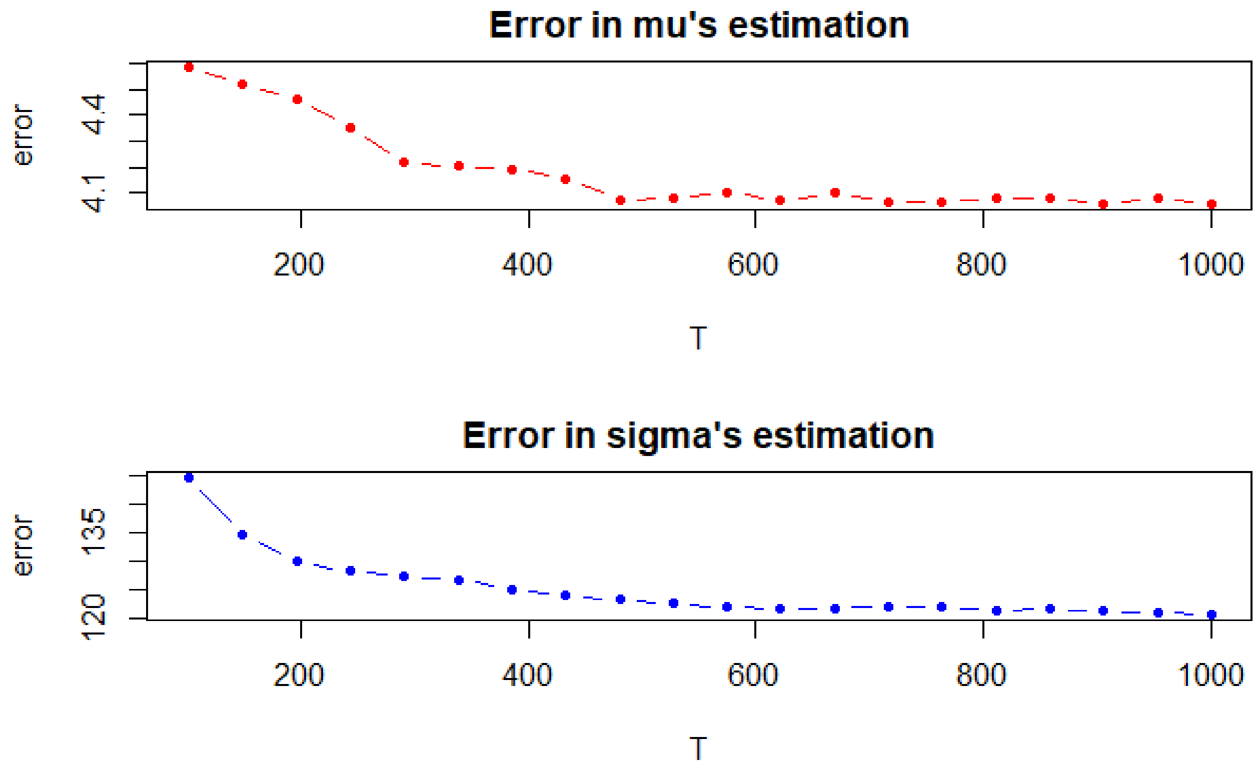
The statistics showed that the mean was constant and nearly 0. This is further confirmed by the time series plot. The ACF plot further shows that since, the log stock price returns are not correlated, the mean is constant for the time series. However, the squared stock price return values have high correlation. Thus, we may conclude that the log returns process has a strong non-linear dependence.

theoretical background means models

I'd like to demonstrate, through the use of R, some of the theoretical concepts of mean models

```
I.I.D.

I started by generating Gaussian synthetic return data, looped over subsets of the
samples, estimate them, and compute the errors.

Then make sure the estimation process gives the correct results (sanity check)
```

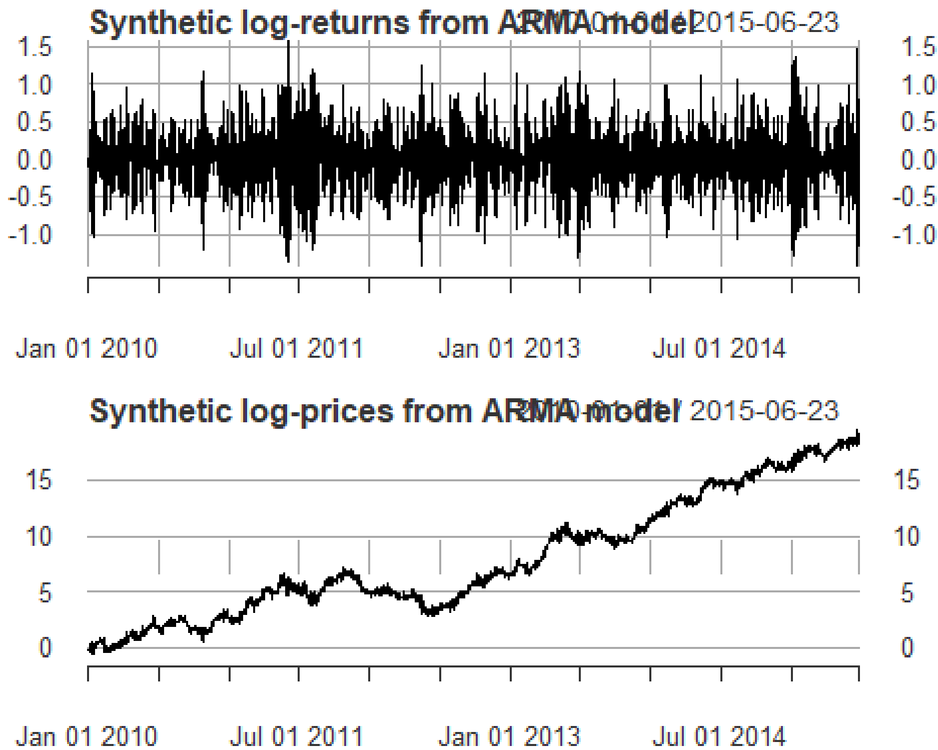
```
plot of error in estimations:
```



The sanity check passed; the plot shows a decreasing error for increasing T

Now I simulate Univariate ARMA model

Plot of synthetic log returns and log prices:

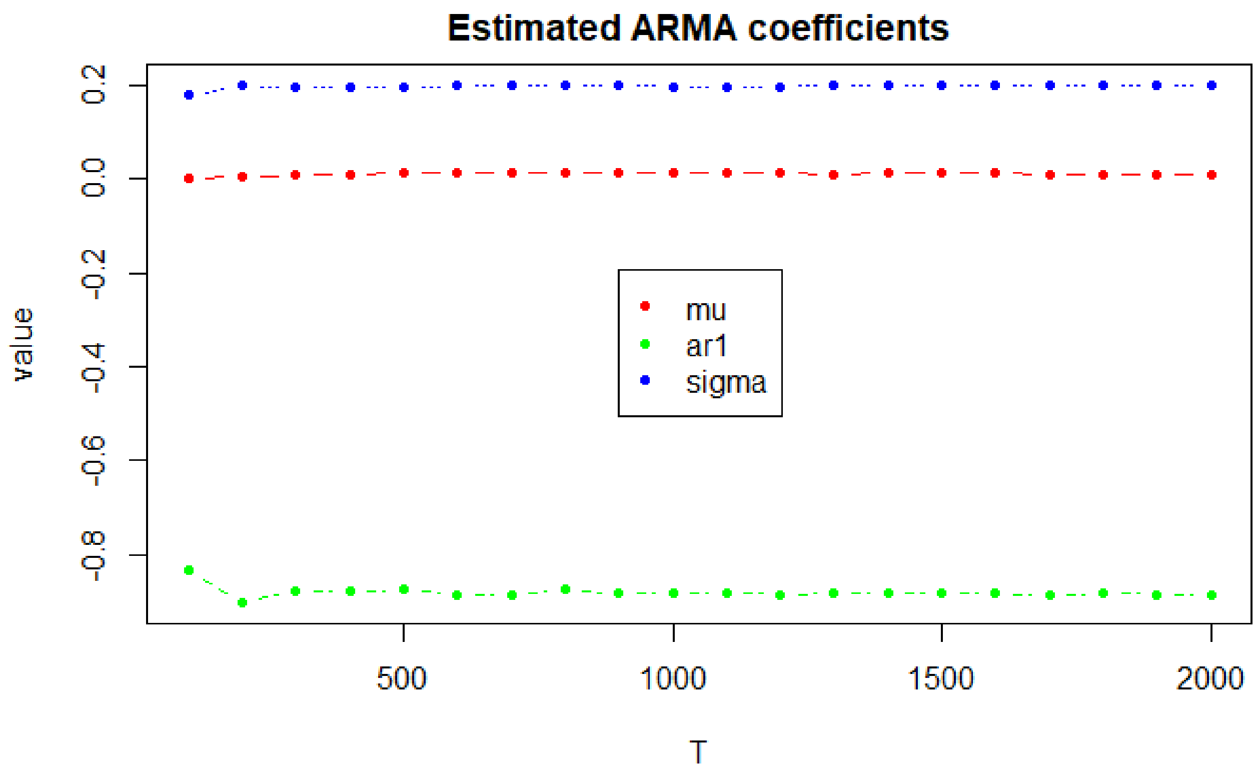


specify an AR(1) model with given coefficients and parameters (where mu, ar1, sigma = 0.01, -0.90, 0.20) are the true parameters and the plot above is the simulated path

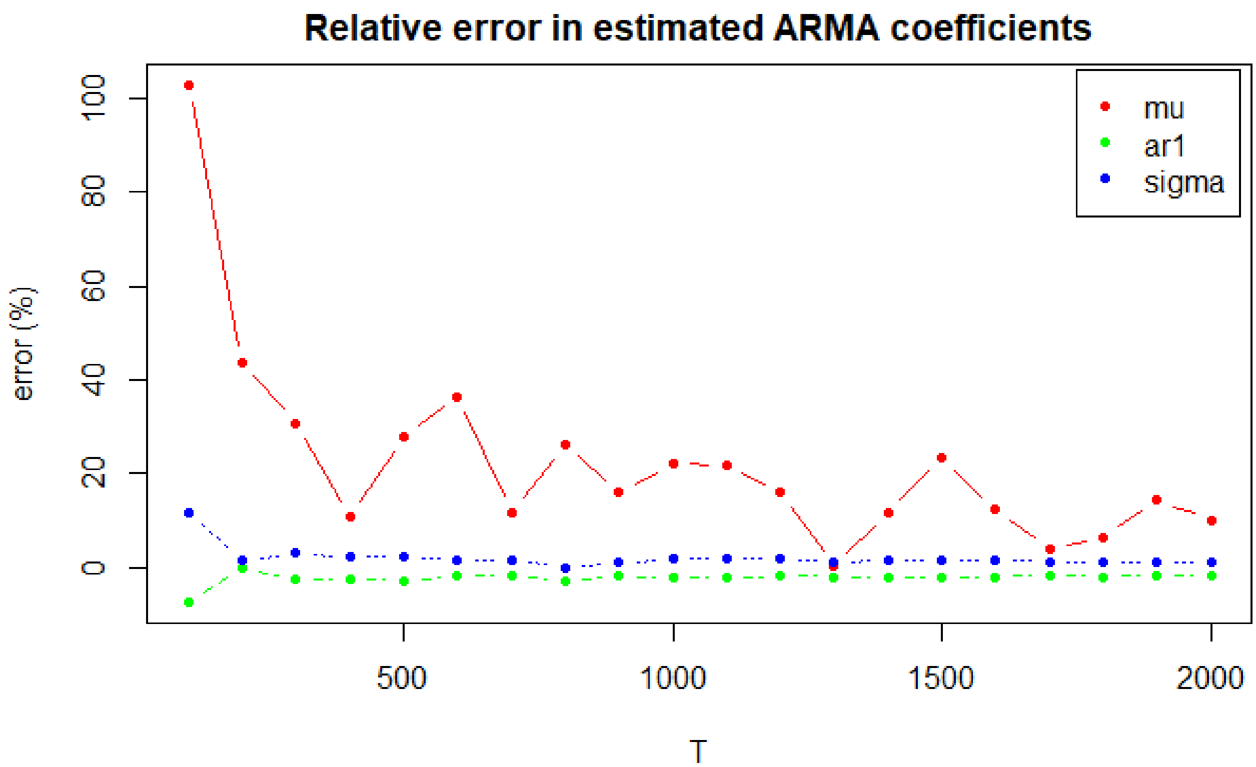
After that I estimated a model

```
##           mu          ar1        sigma
## 0.0006186942 0.0059595185 0.0025511270
```

And plot the error:



they're aligned with the true parameters



The estimation of mu is very noisy; however the coefficients seem to be well estimated after 800 samples.

In practice, the order is unknown, and one must try different combinations of orders. The higher the order, the better the fit, but this will inevitably produce overfitting. Many methods have been developed to penalize the increase of the order complexity to avoid overfitting (AIC, BIC).

Now I created a fit with the function “autoarfima” wit the synth log returnsand looked at the ranking of the combinations

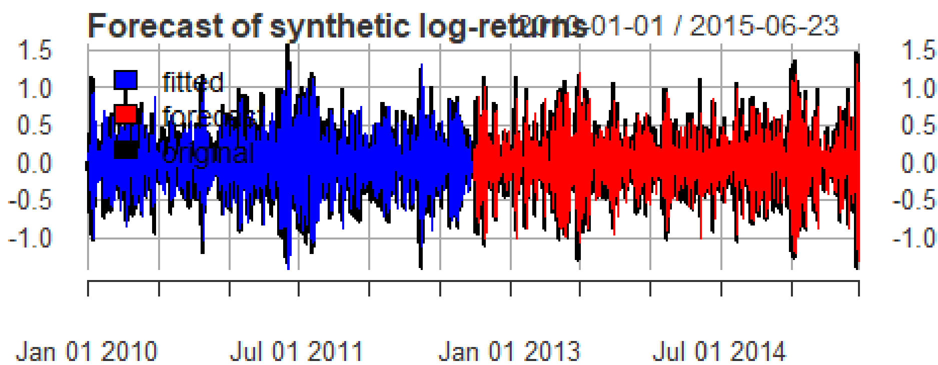
```
arma_fit$rank.matrix

##     AR MA Mean ARFIMA         BIC converged
## 1    1  0    1      0 -0.34424952         1
## 2    2  0    1      0 -0.34212182         1
## 3    1  1    1      0 -0.34193858         1
## 4    2  1    1      0 -0.33859351         1
## 5    3  0    1      0 -0.33842746         1
## 6    1  2    1      0 -0.33824313         1
## 7    3  1    1      0 -0.33594955         1
## 8    1  3    1      0 -0.33484541         1
## 9    2  2    1      0 -0.33466143         1
## 10   3  2    1      0 -0.33251617         1
## 11   2  3    1      0 -0.33132812         1
## 12   3  3    1      0 -0.32981539         1
## 13   0  3    1      0 -0.02446239         1
## 14   0  2    1      0  0.17446465         1
```

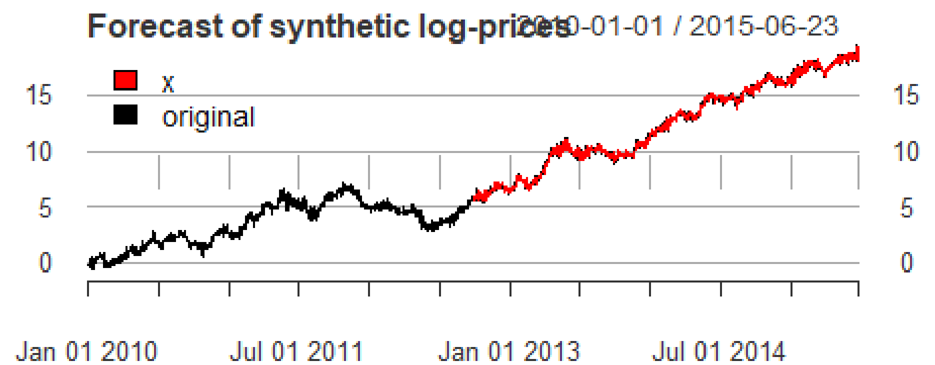
```
## 15  0  1    1      0  0.56412197         1
## 16  0  0    1      0  1.36496316         1
```

and choose the best

```
armaOrder <- arma_fit$rank.matrix[1, c("AR","MA")]
```

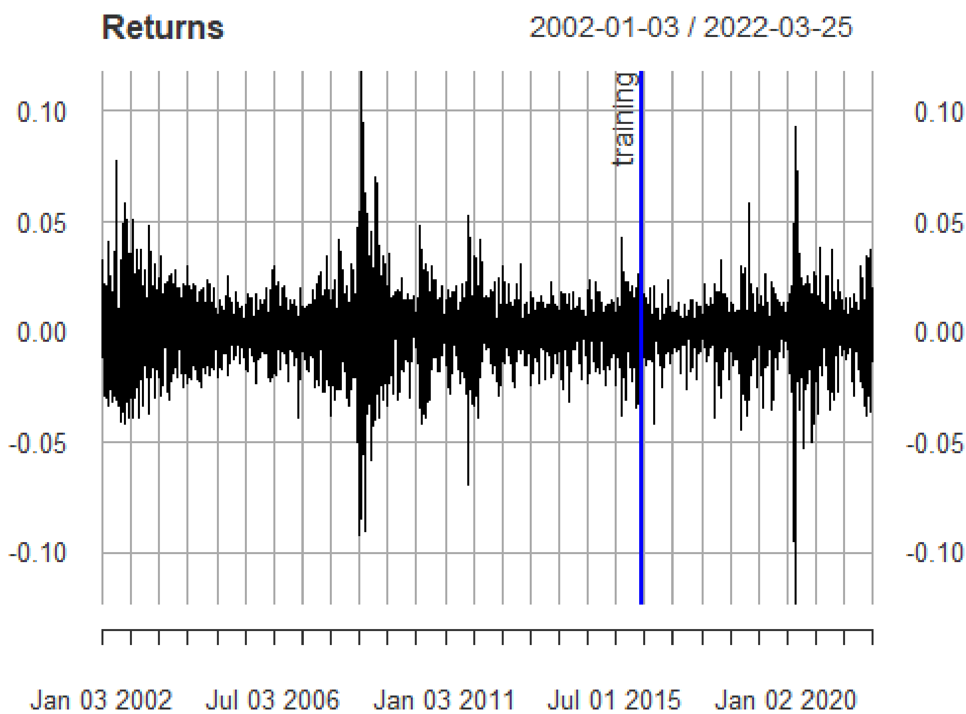
```
Then estimate the model excluding the out of sample, once the ARMA model
parameters have been estimated φ^i and θ^j, can we use the model to
forecast the values ahead.
```





Real data: prepare training and test data

I divided the data in training and test set, and used the training data (for t=1,…,Ttrn) to fit different models. Fit AR(1), ARMA(2,2), ARMA(2,1), MA(1) and use the different models to forecast the log-returns. Then compute the forecast errors



Static -vs- rolling window comparison

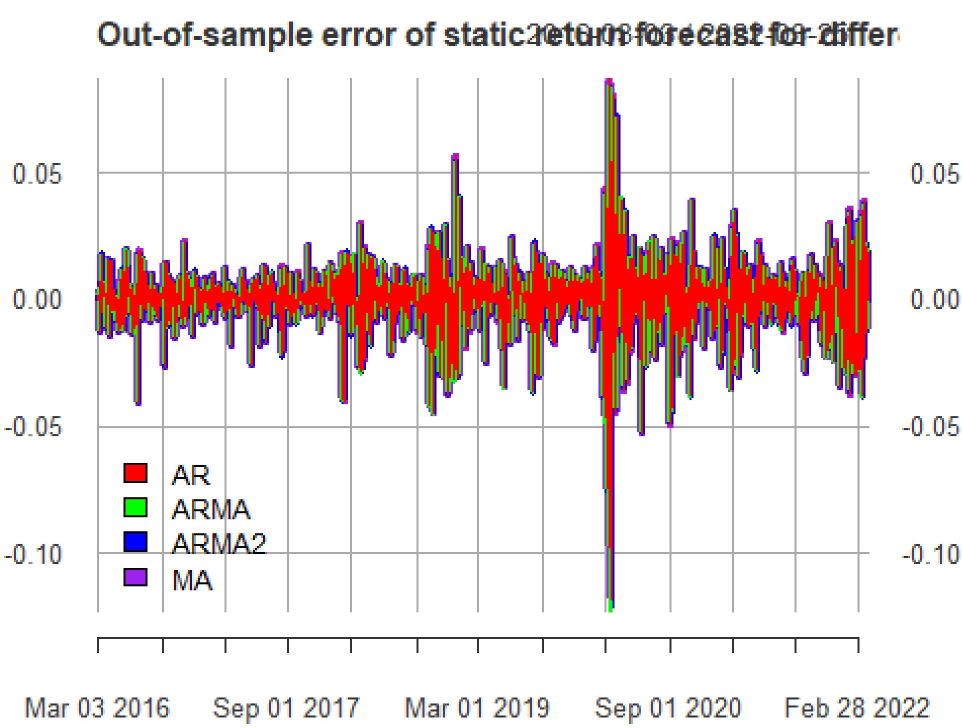
I fit AR(1), ARMA(2,1), ARMA(2,2) and MA(1) models

Then use the different models to forecast the log-returns and compute the forecast errors:

```
Print of forecast errors:

##               in-sample out-of-sample
## AR(1)     0.0001989092  0.0001752712
## ARMA(2,2) 0.0001984664  0.0001767856
## ARMA(2,1) 0.0001985510  0.0001767177
## MA(1,0)   0.0001988444  0.0001751717
```

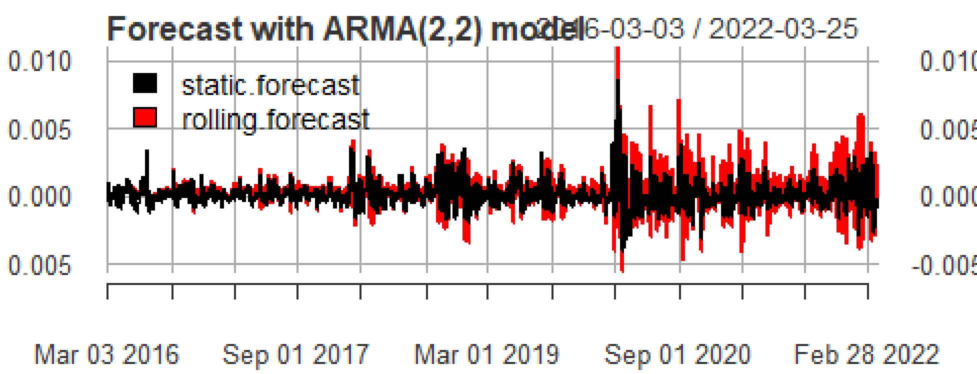
```
The important quantity is the out-of-sample error: i can see that
increasing the model complexity may give few results. It seems that the
simplest AR or MA model is good enough in terms of error of forecast
returns.
```

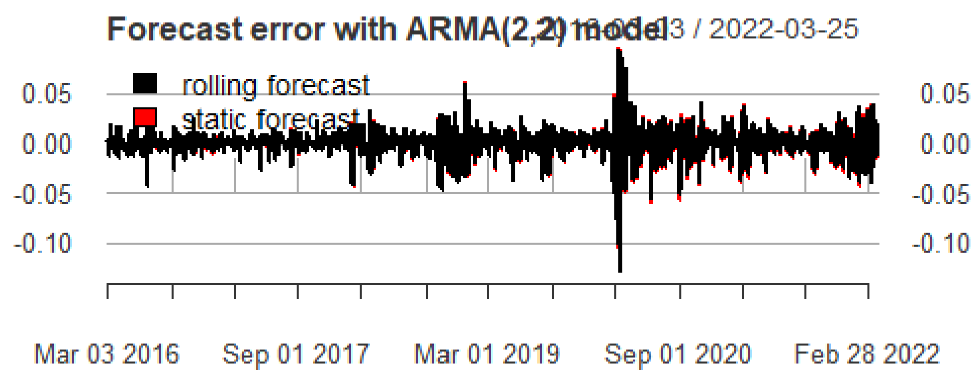
```
plot of forecast error returns:
```



it's possible to obtain better results by refitting the models, doing that the error should tends to become better as the time advances.

```
I did a static fit and forecast and a rolling fit and forecast.

Then I plotted the forecasts and errors of both:
```

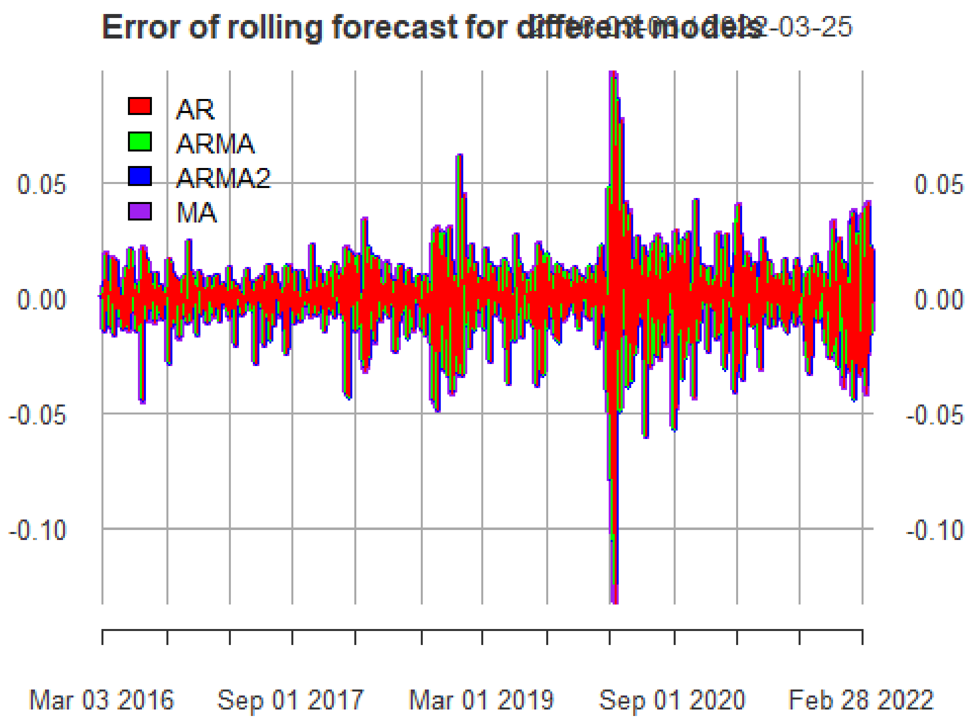




We can observe the effect of the rolling-window process in keeping track with the time series is (with surprise) very smooth.

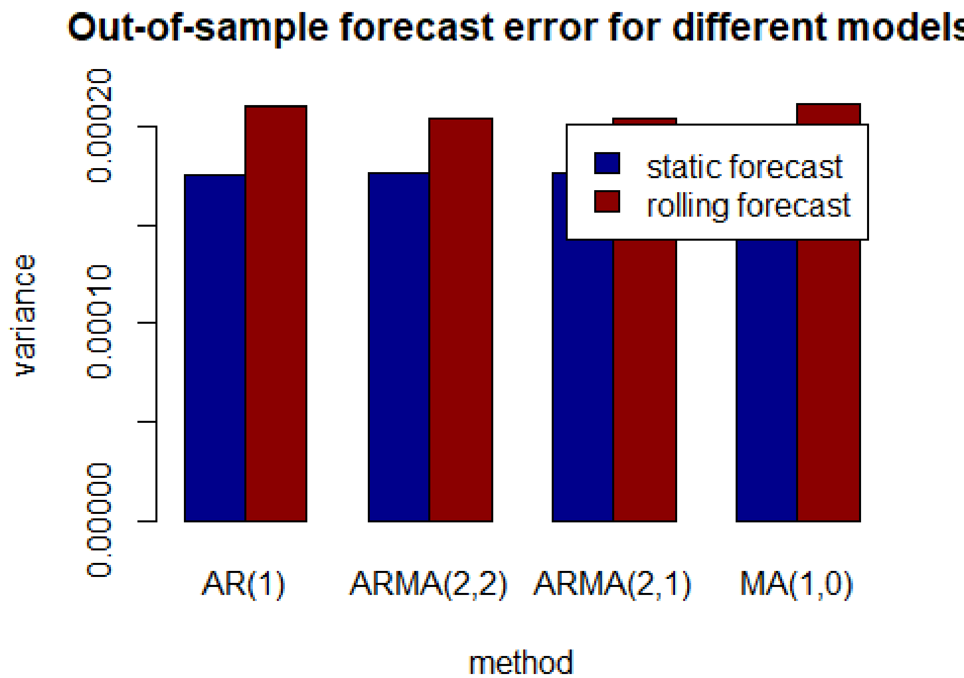
```
Now I did a rolling forecast with AR(1), ARMA(2,2), ARMA(2,1) and MA(1).

Calculate then the rolling forecast errors variance and print them

##                in-sample out-of-sample
## AR(1)     0.0001989092  0.0002099782
## ARMA(2,2) 0.0001984664  0.0002041272
## ARMA(2,1) 0.0001984664  0.0002041272
## MA(1,0)   0.0001988444  0.0002111230
```

```
Plot of error returns:
```



all the models should track better the time series. However, we do not observe any significant difference among the models and the methods

```
compare the static vs rolling basis errors:
```




```
it confirms the previous results.
```

Mean model fit

```
Another way to find the best model:

library(SBAGM) I choose the one with lower AIC
```

```
##           q=0       q=1       q=2       q=3
## p=0 -29078.46 -29122.69 -29120.73 -29118.80
## p=1 -29122.56 -29120.72 -29118.73 -29116.76
## p=2 -29120.73 -29118.73 -29116.73 -29114.76
## p=3 -29118.74 -29116.73 -29114.73 -29115.88
```

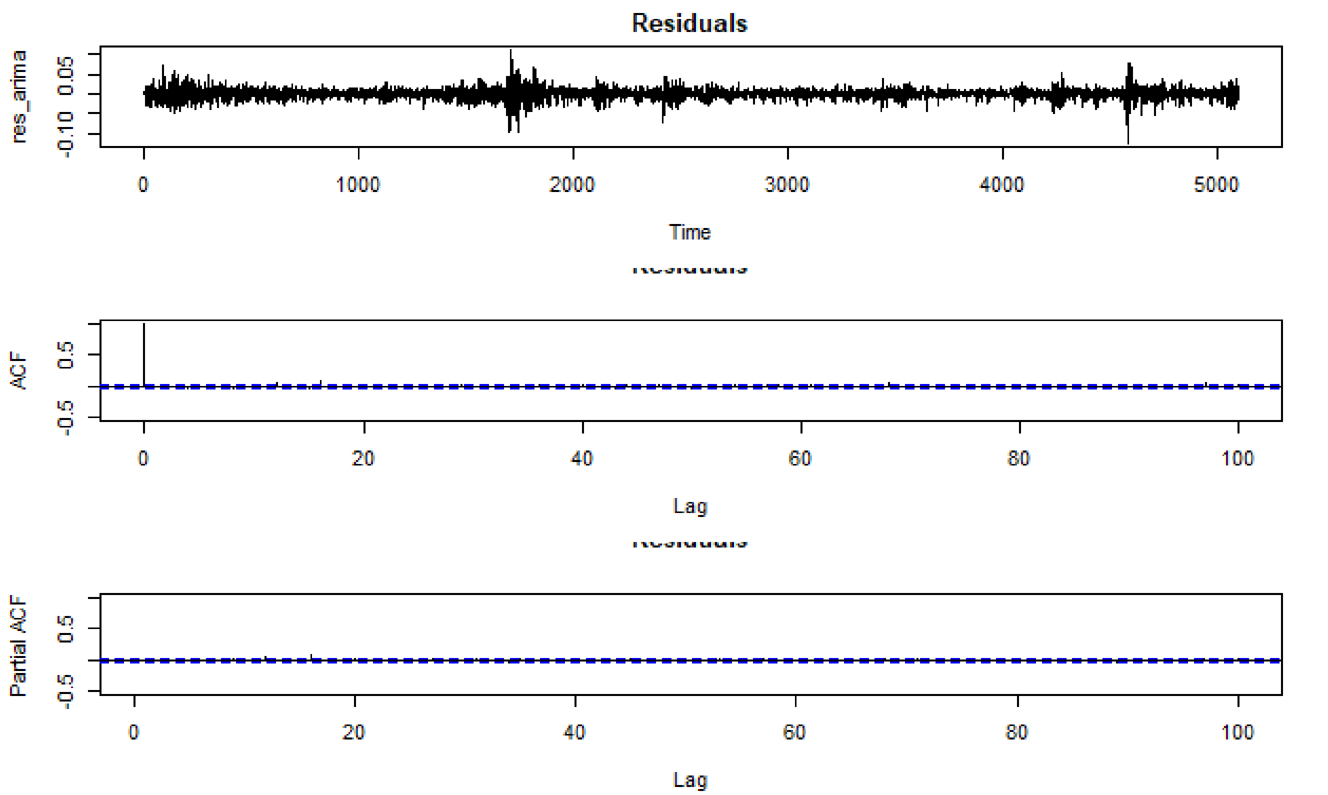
```
library(forecast)
auto.arima(y = diff(log(nasdaq_adj))[-1])
```

```
## Series: diff(log(nasdaq_adj))[-1]
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##           ma1   mean
##       -0.0952  4e-04
## s.e.   0.0139  2e-04
##
## sigma^2 = 0.000192:  log likelihood = 14564.34
## AIC=-29122.69   AICc=-29122.68   BIC=-29103.08
```

As the previous approach, AR(1) and MA(1) give us good fits, but the auto.arima and SBAGM functions have selected MA(1).

residuals analysis

```
Firstly I plot arima’s residuals, acf and pacf:
```



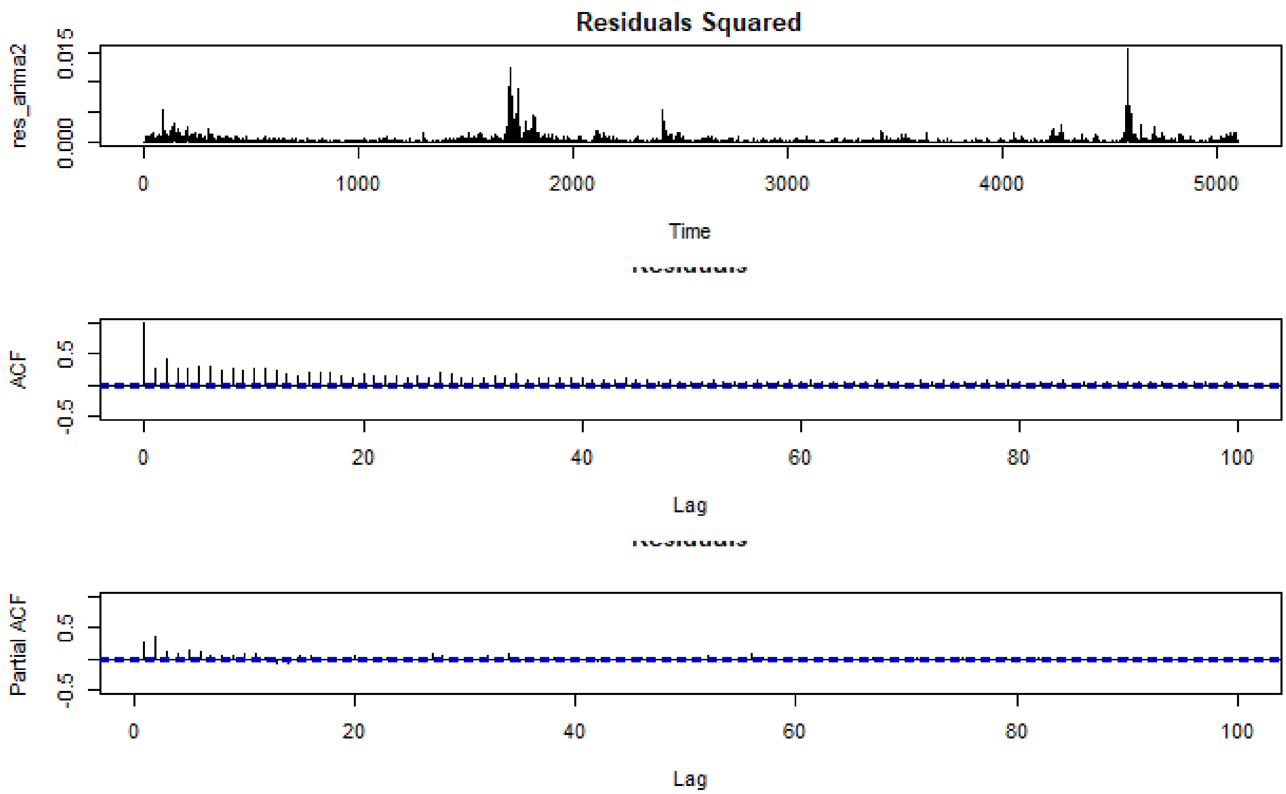
The residual plot, ACF and PACF do not have any significant lag, indicating ARIMA(1,1,0) is a good model to represent the series.

```
Secondly I did a Box test for lags 1,5 and 10:

Box.test(res_arima, lag = c(2), type = c( "Ljung-Box"), fitdf = 1)

##
##  Box-Ljung test
##
## data:  res_arima
## X-squared = 0.051853, df = 1, p-value = 0.8199

Box.test(res_arima, lag = c(5), type = c( "Ljung-Box"), fitdf = 1)

##
##  Box-Ljung test
##
## data:  res_arima
## X-squared = 3.8253, df = 4, p-value = 0.4302
```

```
Box.test(res_arima, lag = c(10), type = c( "Ljung-Box"), fitdf = 1)

##
##  Box-Ljung test
##
## data:  res_arima
## X-squared = 22.842, df = 9, p-value = 0.006561
```

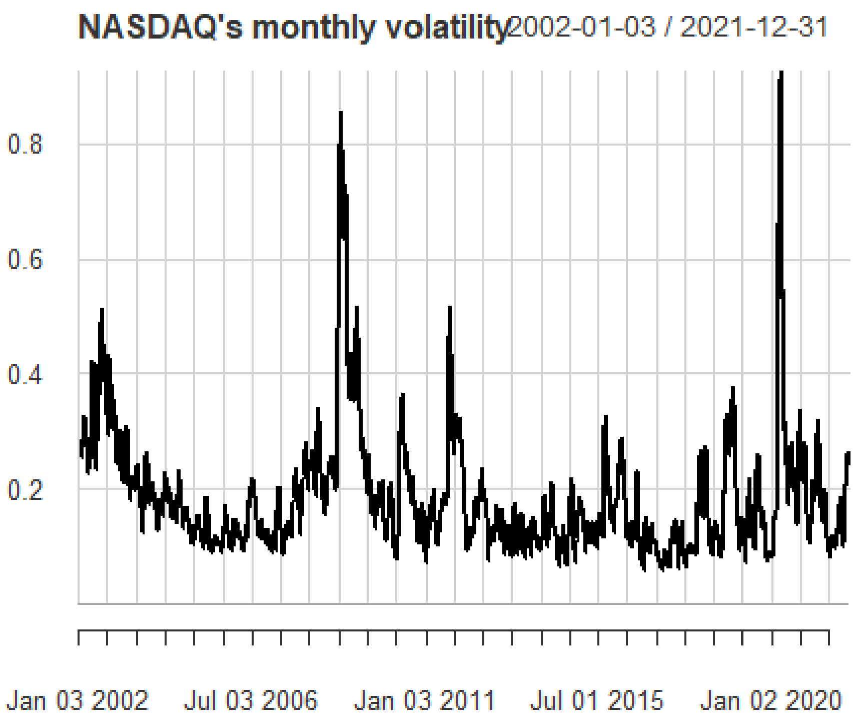
Furthermore, the Box test for the second, fifth and tenth(slightly) lag confirm the hypothesis of independently distributed data.

`Then I plot arima's squared residuals, acf and pacf:`



Squared residuals plot shows cluster of volatility PACF and ACF cuts off after lag 10 and 20 even though some remaining lags are significant Thus, there is presence of ARCH effect.

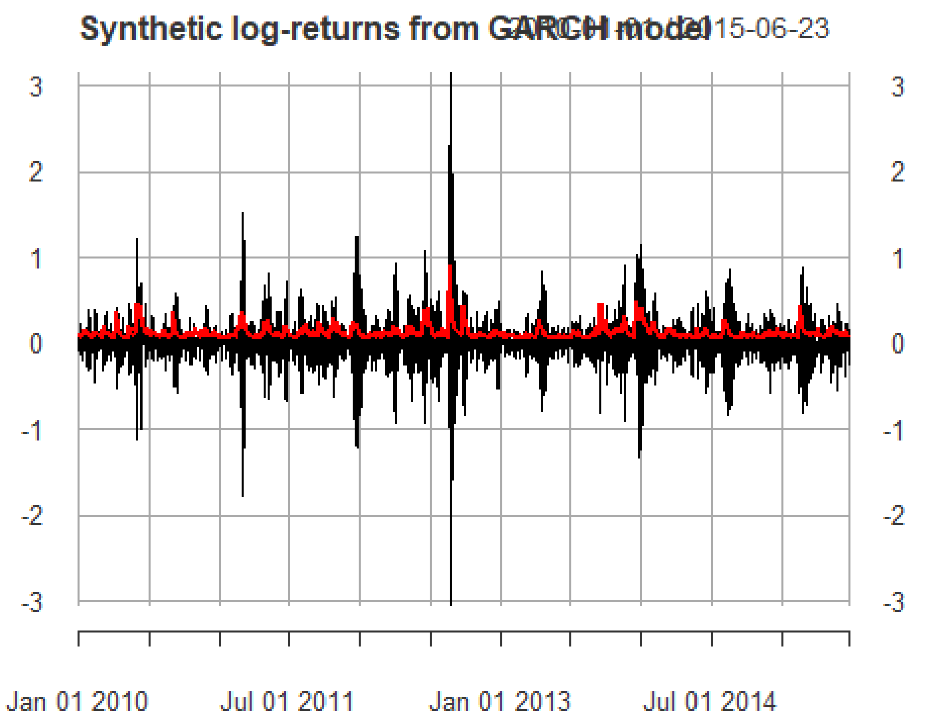
```
Plot of NASDAQ's monthly volatility
```



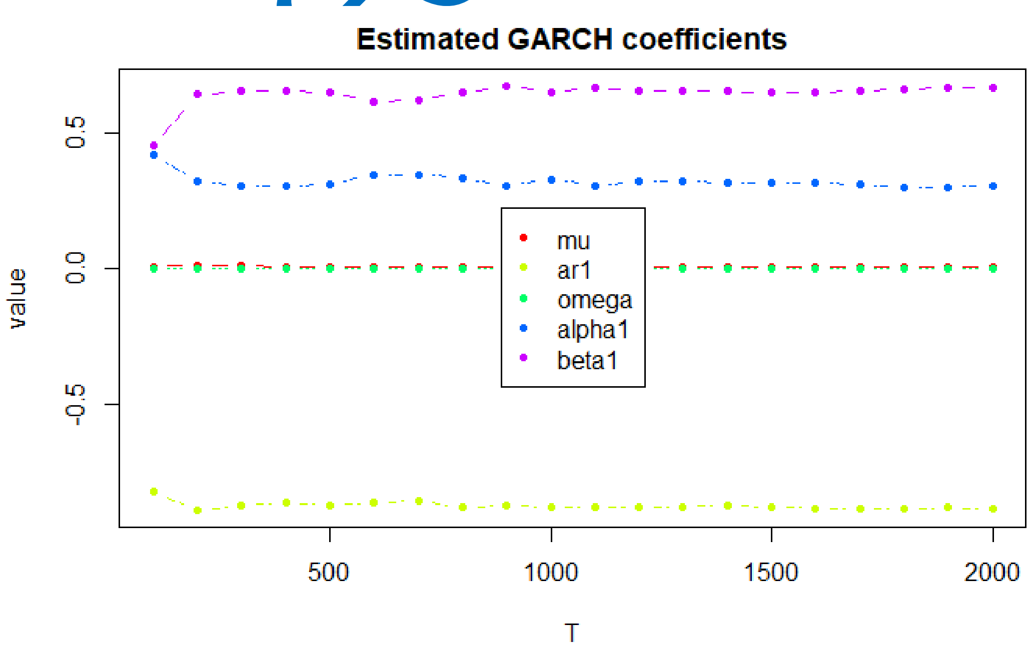
We may need a stochastic model for conditional volatility. In the next session I will try to fit the best model to deal with the conditional volatility.

Theoretical background variance models

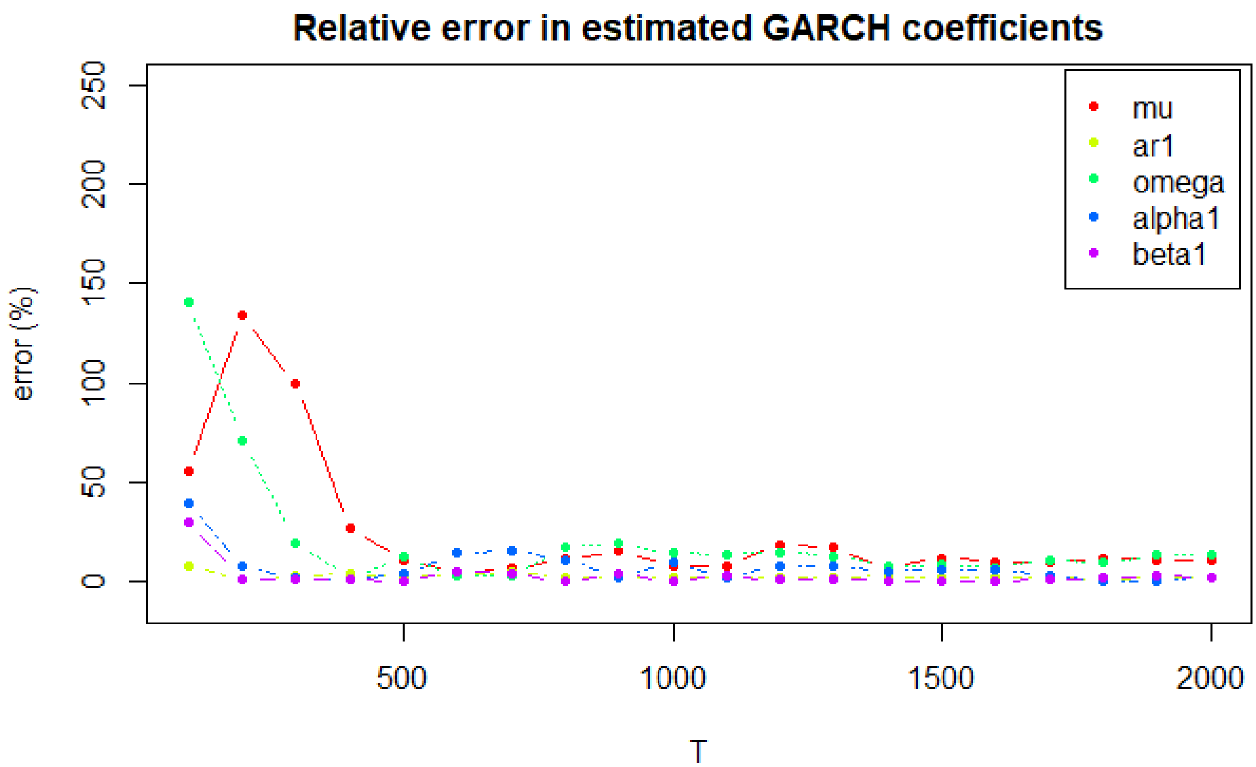
First i specified a GARCH(1,1) model with given coefficients and parameters and I simulated a path and plotted it with synthetic log returns




```
now specify a GARCH model and plot the estimate GARCH coefficients:
```



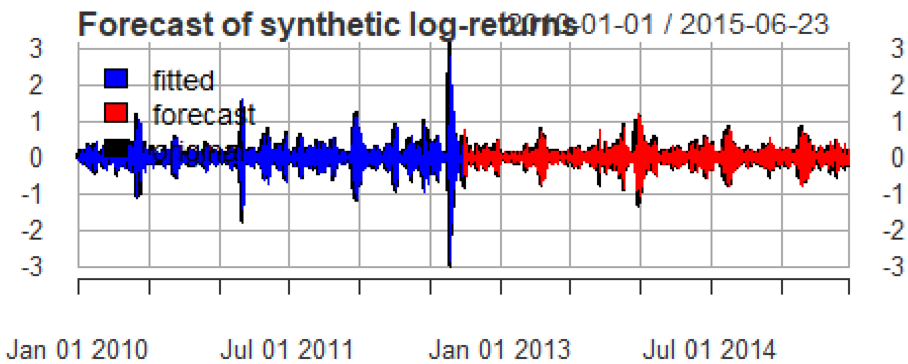
```
Then plot the relative error in estimated GARCH coefficients:
```

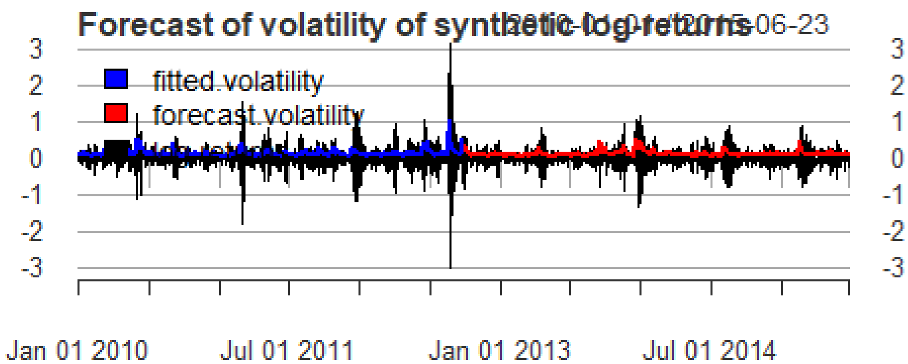


the coefficients seem to be well estimated after T samples

Once the parameters of the GARCH model have been estimated, one can use the model to forecast the values ahead.

```
For the forecasting I firstly estimate model excluding the out-of-sample, then
forecast log-returns along the whole out-of-sample, after that plot the log
returns and the volatility of log returns:
```



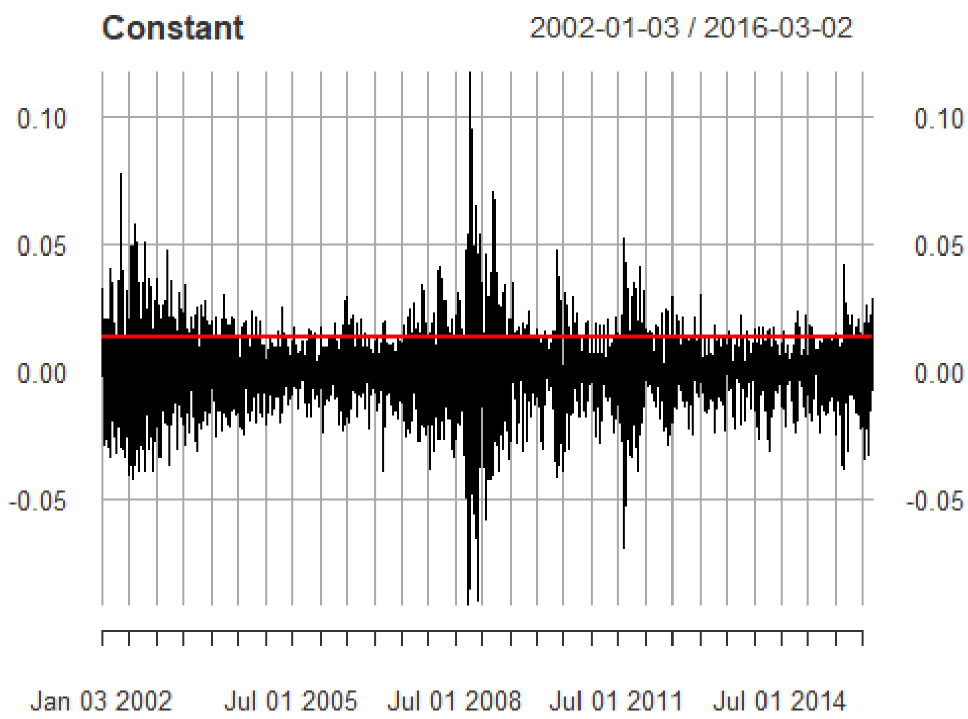


I estimated the model excluding the out-of-sample, then forecasted log-returns along the whole out-of-sample.

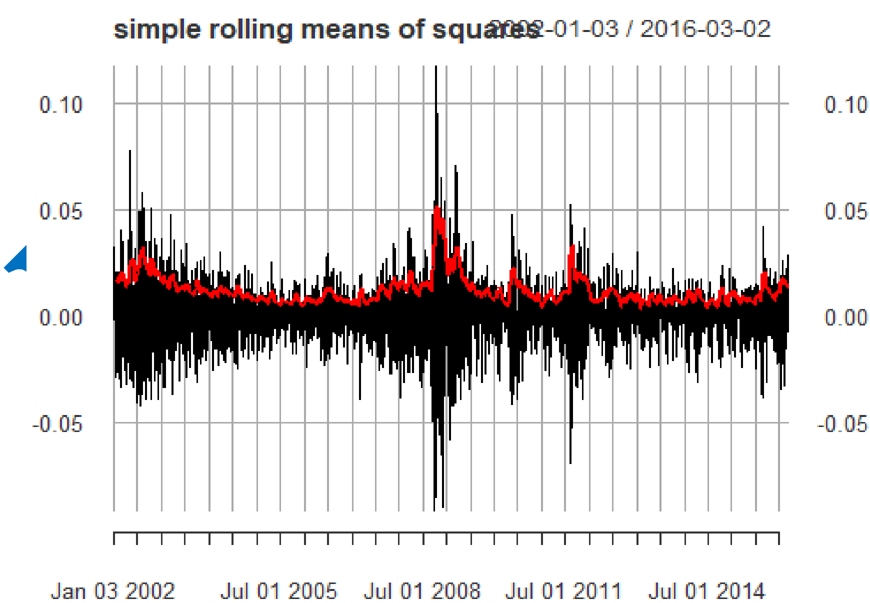
Different methods for conditional volatility

modelling the Nasdaq's volatility different methods:

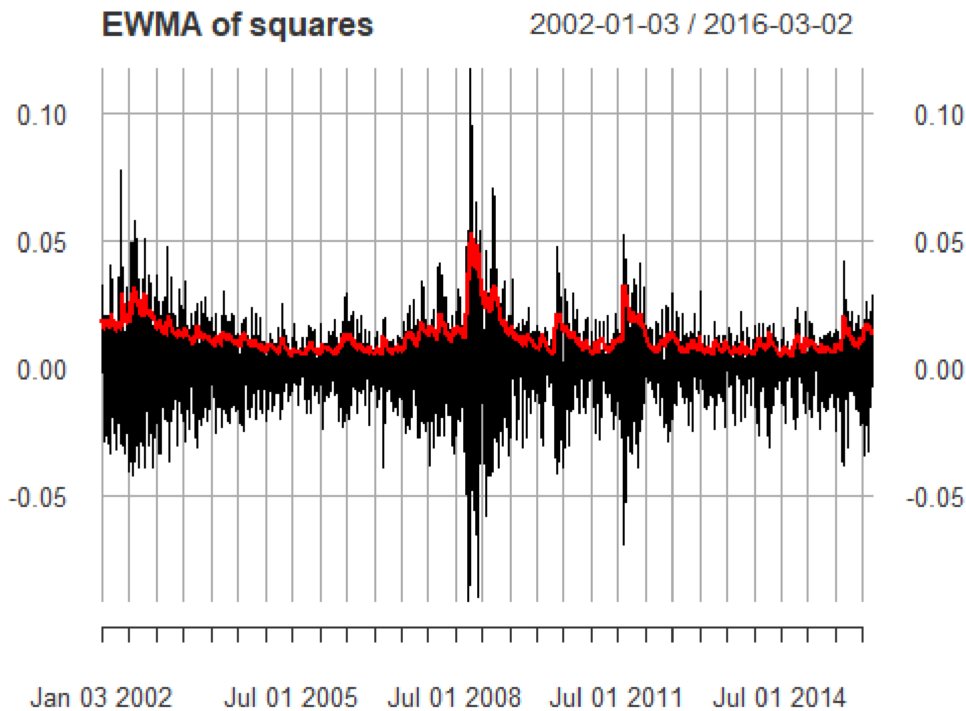
```
Costant variance:
```



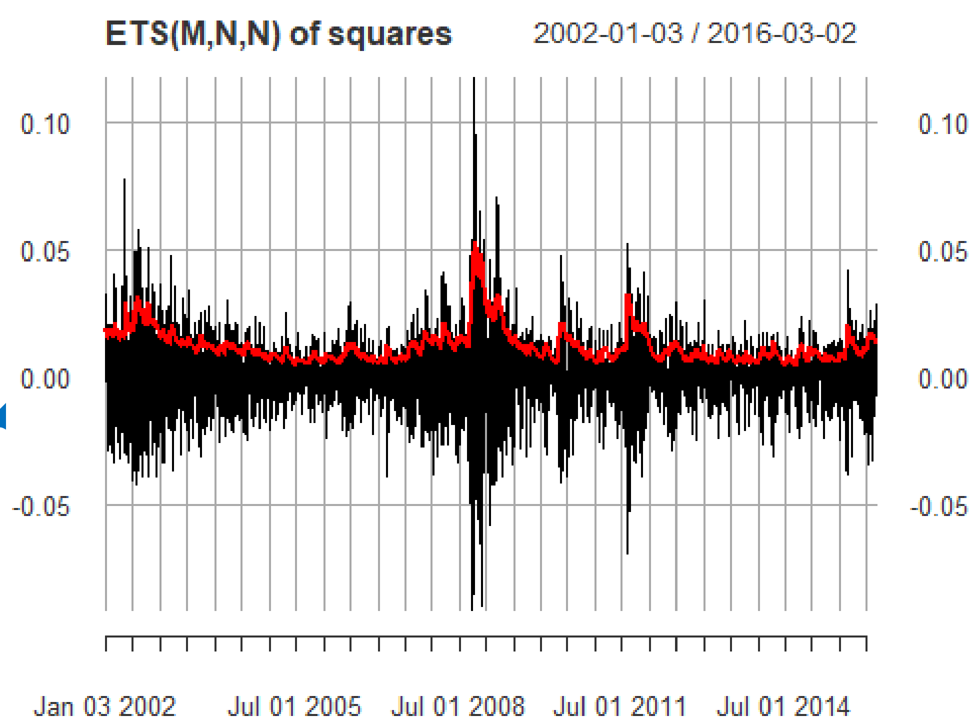

```
Moving Average:
```



`exponentially weighted moving average:`



Multiplicative ETS:



ETS (Error, Trend, Seasonal)method is an approach method for forecasting time series univariate. ETS model focuses on trend and seasonal components. The flexibility of the ETS model lies in its ability to trend and seasonal components of different traits. In this paper it's used only for illustration purposes, it's not further covered by this work

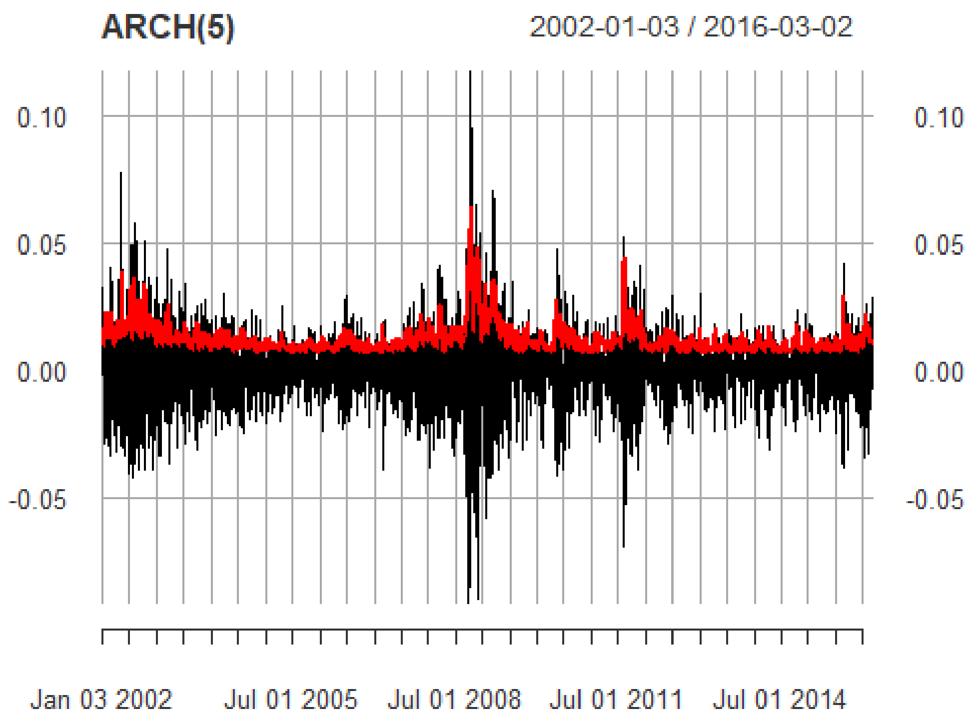
```
I used the “appgarch” algorithm to find the best GARCH(,) + distribution
combination
```

```
appgarch_1 <- appgarch(data = return, methods = c('sGARCH', 'eGARCH', 'gjrGARCH'),
distributions =  c("norm", "std","sstd", "snorm",'ged'),
         aorder = c(0,1), gorder = c(1, 1), stepahead = 2 )

## $rmse_mean
##                 norm        std       sstd      snorm        ged
## sGARCH   0.01280471 0.01266372 0.01264385 0.01263505 0.01267901
## eGARCH   0.01314183 0.01291799 0.01301078 0.01306634 0.01291860
## gjrGARCH 0.01310694 0.01291799 0.01297889 0.01301380 0.01291667
##
## $mae_mean
##                 norm        std       sstd      snorm        ged
## sGARCH   0.01018926 0.01018908 0.01006424 0.01003220 0.01020496
## eGARCH   0.01023014 0.01023245 0.01015853 0.01012516 0.01024431
## gjrGARCH 0.01023609 0.01024236 0.01015405 0.01012270 0.01025683
##
## $forecast_mean
##       sGARCH-norm  sGARCH-std  sGARCH-sstd sGARCH-snorm  sGARCH-ged  eGARCH-norm
## T+1 0.0013941347 0.001629151 0.0016204842 0.0016250429 0.001609034 0.0008588628
## T+2 0.0008460608 0.001080712 0.0008223621 0.0007628482 0.001092364 0.0003925498
##        eGARCH-std  eGARCH-sstd eGARCH-snorm   eGARCH-ged gjrGARCH-norm
## T+1 0.0012210235 0.0010505833 0.0009542401 0.0012235817  0.0009160329
## T+2 0.0007593347 0.0004410439 0.0002779707 0.0007856046  0.0004616043
##     gjrGARCH-std gjrGARCH-sstd gjrGARCH-snorm gjrGARCH-ged
## T+1 0.0012239911  0.0011005645   0.0010370364 0.0012305476
## T+2 0.0007821032  0.0004820636   0.0003558419 0.0008176138
##
## $forecast_sigma
##     sGARCH-norm sGARCH-std sGARCH-sstd sGARCH-snorm sGARCH-ged eGARCH-norm
## T+1  0.01988816 0.02061097  0.02062876   0.02008425 0.02021335  0.01776043
## T+2  0.01978605 0.02058334  0.02058690   0.01998632 0.02014814  0.01757892
##     eGARCH-std eGARCH-sstd eGARCH-snorm eGARCH-ged gjrGARCH-norm gjrGARCH-std
## T+1 0.01822851  0.01812509   0.01787575 0.01803635    0.01691592   0.01691819
## T+2 0.01800036  0.01793108   0.01771687 0.01779125    0.01681810   0.01683897
##     gjrGARCH-sstd gjrGARCH-snorm gjrGARCH-ged
## T+1    0.01696756     0.01703881   0.01690764
## T+2    0.01686481     0.01692407   0.01680765
```

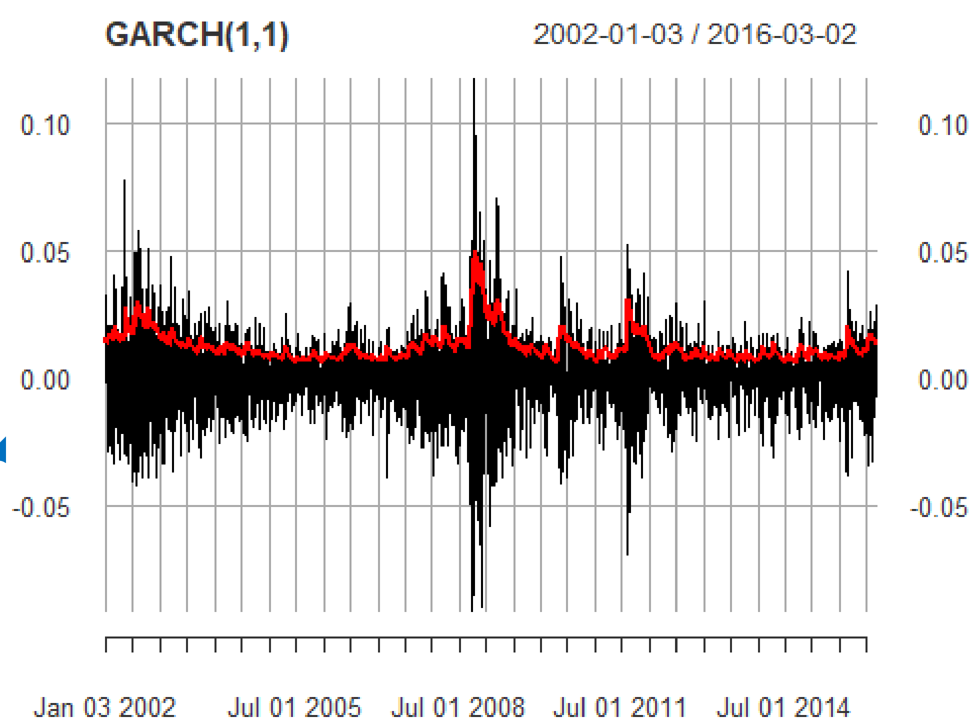
```
appgarch_2 <- appgarch(data = return, methods = c('sGARCH', 'eGARCH', 'gjrGARCH'),
distributions =  c("norm", "std","sstd", "snorm",'ged'),
         aorder = c(0,1), gorder = c(2, 1), stepahead = 2 )


## $rmse_mean
##                 norm        std       sstd      snorm        ged
## sGARCH   0.01282226 0.01267170 0.01264584 0.01264653 0.01269115
## eGARCH   0.01303188 0.01279397 0.01300566 0.01306762 0.01279891
## gjrGARCH 0.01307939 0.01289029 0.01293950 0.01298031 0.01289348
##
## $mae_mean
##                 norm        std       sstd      snorm        ged
## sGARCH   0.01019558 0.01018952 0.01005931 0.01003401 0.01020772
## eGARCH   0.01016173 0.01015717 0.01016290 0.01012935 0.01016948
## gjrGARCH 0.01022309 0.01022777 0.01013247 0.01010509 0.01024497
##
## $forecast_mean
##      sGARCH-norm  sGARCH-std  sGARCH-sstd sGARCH-snorm  sGARCH-ged  eGARCH-norm
## T+1 0.0013671500 0.001615860 0.0016156509 0.0016067254 0.001589563 0.0010176413
## T+2 0.0008317025 0.001068294 0.0008076689 0.0007481441 0.001078409 0.0004145016
##       eGARCH-std  eGARCH-sstd eGARCH-snorm   eGARCH-ged gjrGARCH-norm
## T+1 0.0014018813 0.0010598725 0.0009531664 0.0013975377  0.0009569060
## T+2 0.0007896317 0.0004590677 0.0002852579 0.0008098946  0.0004764823
##     gjrGARCH-std gjrGARCH-sstd gjrGARCH-snorm gjrGARCH-ged
## T+1  0.001265070  0.0011582949   0.0010861489 0.0012650873
## T+2  0.000794015  0.0004966341   0.0003697235 0.0008284343
##
## $forecast_sigma
##     sGARCH-norm sGARCH-std sGARCH-sstd sGARCH-snorm sGARCH-ged eGARCH-norm
## T+1  0.01999185 0.02077562  0.02087408   0.02025395 0.02032970  0.01849086
## T+2  0.01944968 0.02021708  0.02026550   0.01965940 0.01978522  0.01761937
##     eGARCH-std eGARCH-sstd eGARCH-snorm eGARCH-ged gjrGARCH-norm gjrGARCH-std
## T+1 0.01919160  0.01898025   0.01857086 0.01897107    0.01705878   0.01722670
## T+2 0.01801074  0.01791341   0.01771680 0.01788651    0.01685871   0.01708461
##     gjrGARCH-sstd gjrGARCH-snorm gjrGARCH-ged
## T+1    0.01721300     0.01712078   0.01716232
## T+2    0.01705167     0.01691100   0.01697966
```

the appgarch() function suggests eGARCH(1,1) model with skew normal distribution (snorm)

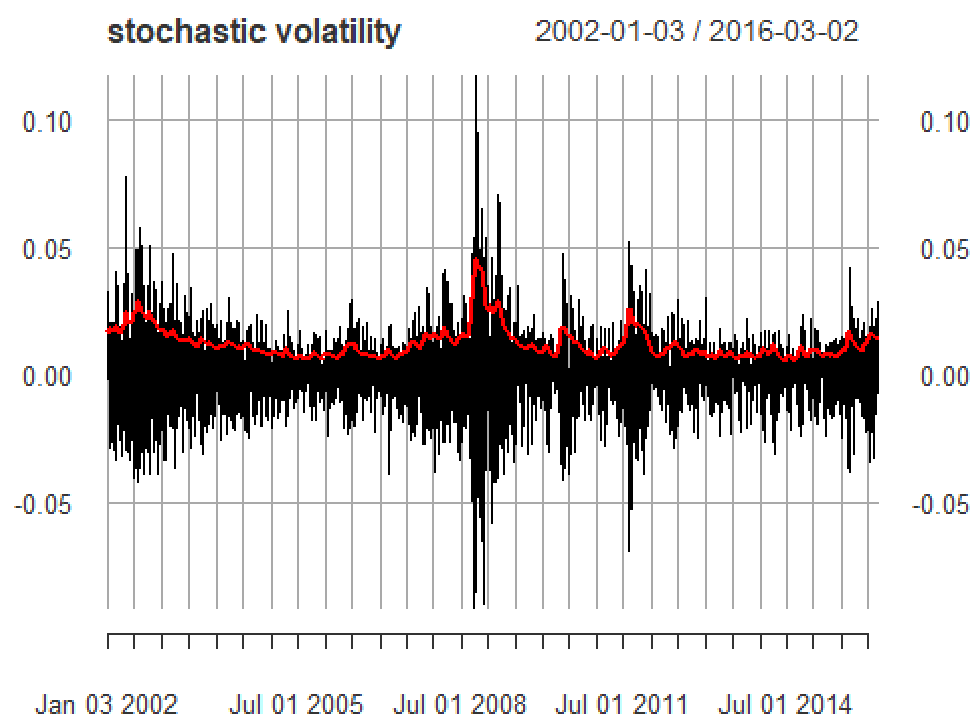
```
ARCH(5)
```




```
GARCH(1,1
```



```
Stochastic Volatility
```

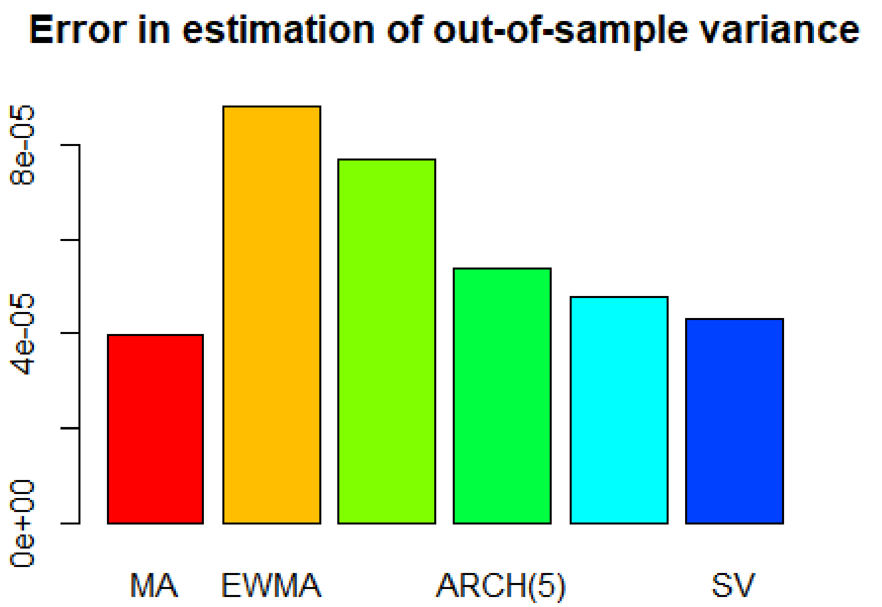


stochastic volatility is not covered by this work, however it has been included for informational purposes

```
I compared the error in the estimation of the variance by each method for the out-of-sample period

print(error_all)
```

```
##           MA         EWMA    ETS(M,N,N)      ARCH(5)   GARCH(1,1)           SV
## 3.960304e-05 8.798757e-05 7.679852e-05 5.383680e-05 4.777268e-05 4.319431e-05
```

```
Plot of errors in estimation of out of sample variance:
```



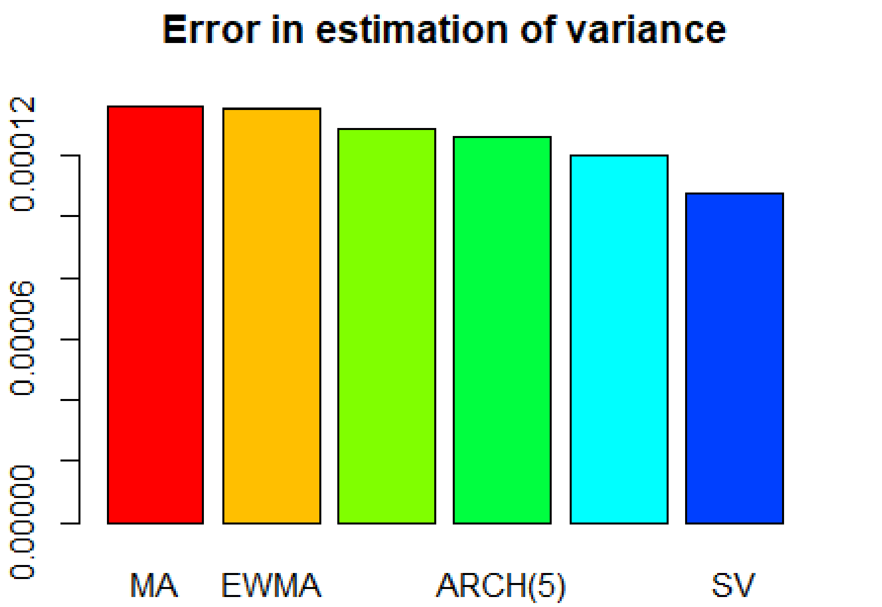
as i expected the SV seems to be the best in describing the conditional variance.

Rolling-Window comparison volatility models

Then, i used a rolling-window comparison of six methods: MA, EWMA, ETS(MNN), ARCH(5), GARCH(1,1), and SV

```
print(error_all)

##           MA         EWMA    ETS(M,N,N)      ARCH(5)   GARCH(1,1)           SV
## 0.0001357935 0.0001352626 0.0001288716 0.0001259760 0.0001201972 0.0001075741
```

```
Plot of errors in estimation of out of sample variance:
```



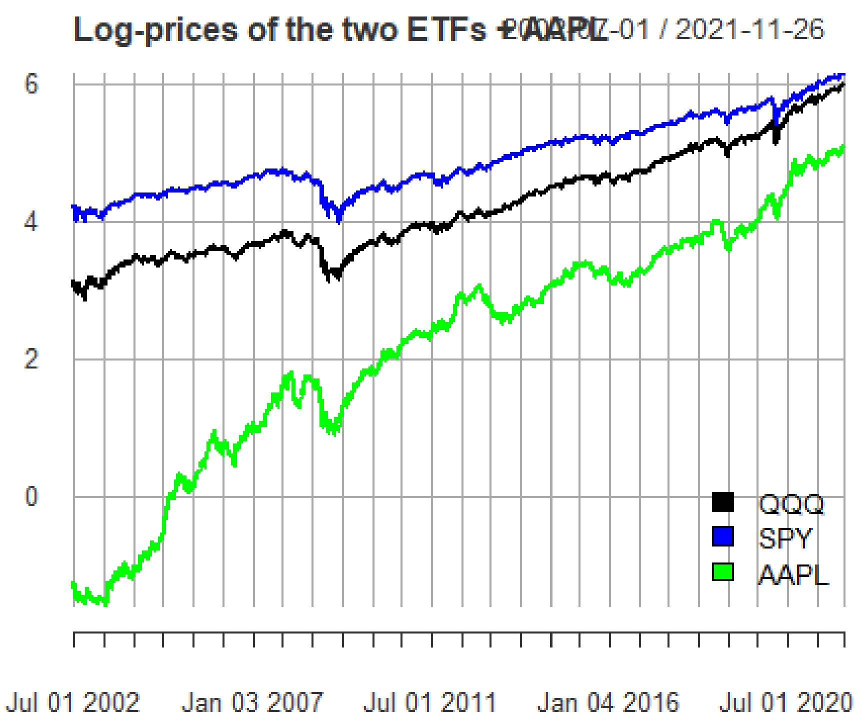
the SV keeps the first position in being the best method for describing conditional variance. in this paper i will continue with GARCH modelling which gives a good solution in dealing with conditional variance

Multivariate GARCH model

#Multivariate GARCH (just a look at the correlations)

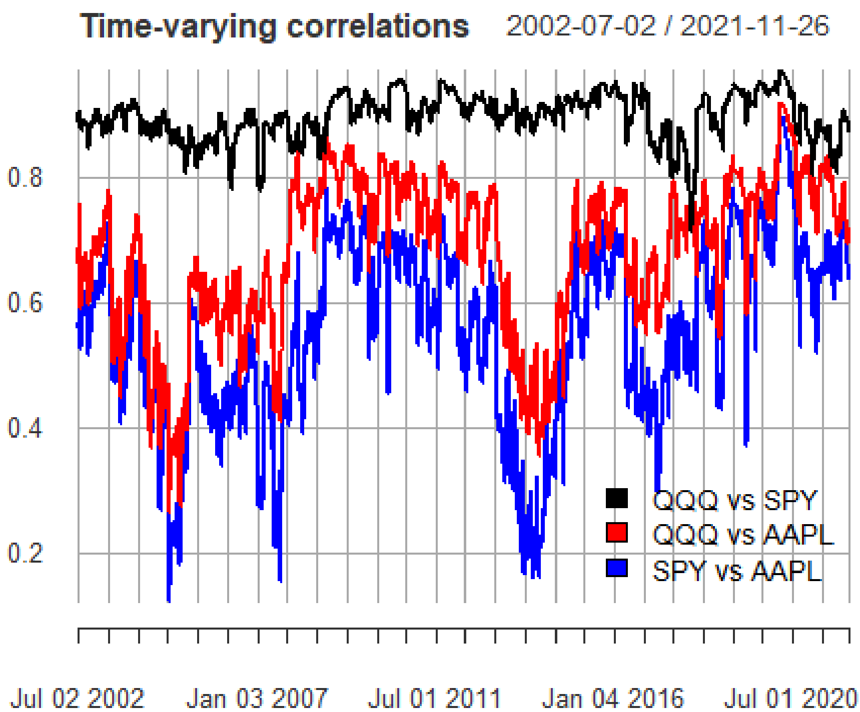
I loaded some multivariate ETF data [QQQ, SPY and AAPL stock] from Yahoo Finance.

```
stock_namelist <- c("QQQ", "SPY", "AAPL")



plot the four series of log-prices:
```




```
First i specify iid model for the univariate time series, then I specify the DCC
model and after that I fit the model

I extract the time-varying covariance and correlation matrix:


dim(dcc_cor)

## [1]    3    3 4887
```

```
plot of the Time-varying correlations:
```



We see the correlation between the two ETFs is extremely high and quite stable. The correlation between AAPL and the ETFs is cyclical, shifting between 0.8 and 0.4

Model specification and fit

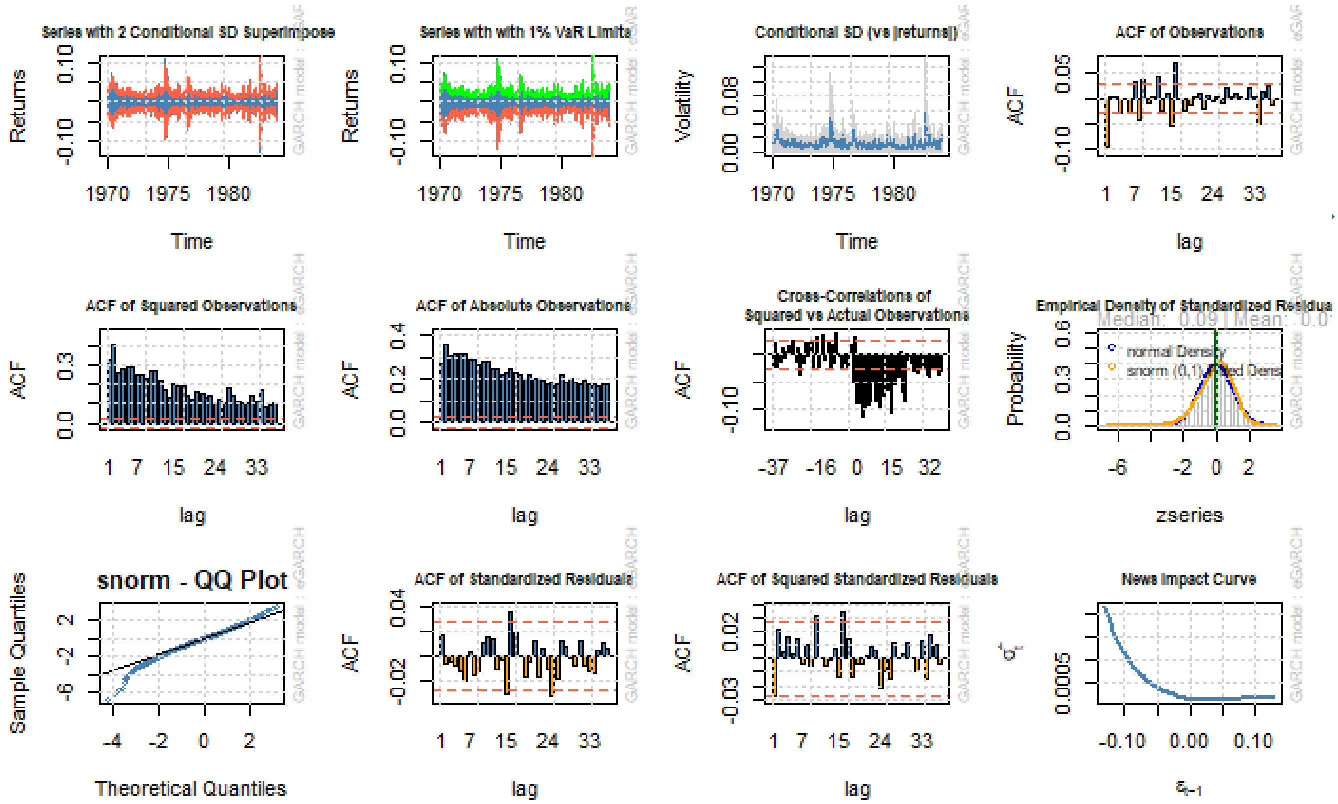
Now it's time for specify the model and to make the fit

ARIMA(0,1,1)-EGARCH(1,1) with skew normal distribution

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : eGARCH(1,1)
## Mean Model   : ARFIMA(0,0,1)
## Distribution : snorm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000198    0.000136    1.4631  0.14344
## ma1    -0.055044    0.015953   -3.4505  0.00056
## omega  -0.231854    0.002219 -104.5015  0.00000
## alpha1 -0.122486    0.005683  -21.5521  0.00000
## beta1   0.973492    0.000364 2673.7185  0.00000
## gamma1  0.141123    0.001885   74.8504  0.00000
## skew    0.800409    0.014777   54.1656  0.00000
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000198    0.000175    1.1303 0.258351
## ma1    -0.055044    0.015321   -3.5928 0.000327
## omega  -0.231854    0.007207  -32.1695 0.000000
## alpha1 -0.122486    0.008573  -14.2871 0.000000
## beta1   0.973492    0.000591 1648.2570 0.000000
## gamma1  0.141123    0.010403   13.5652 0.000000
## skew    0.800409    0.018352   43.6151 0.000000
##
## LogLikelihood : 15729.09
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.1740
## Bayes        -6.1650
## Shibata      -6.1740
## Hannan-Quinn -6.1709
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.619  0.2032
```

```
## Lag[2*(p+q)+(p+q)-1][2]     1.712  0.3290
## Lag[4*(p+q)+(p+q)-1][5]     2.079  0.6885
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      3.720 0.05377
## Lag[2*(p+q)+(p+q)-1][5]     6.238 0.07900
## Lag[4*(p+q)+(p+q)-1][9]     7.485 0.16182
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1154 0.500 2.000  0.7341
## ARCH Lag[5]    1.1551 1.440 1.667  0.6875
## ARCH Lag[7]    1.9140 2.315 1.543  0.7355
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  5.4873
## Individual Statistics:
## mu     0.16639
## ma1    0.07691
## omega  0.36506
## alpha1 0.18691
## beta1  0.30513
## gamma1 0.22965
## skew   2.57418
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.69 1.9 2.35
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value     prob sig
## Sign Bias            1.903 0.057156   *
## Negative Sign Bias   1.801 0.071717   *
## Positive Sign Bias   1.833 0.066867   *
## Joint Effect        13.215 0.004194 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     82.91    5.834e-10
## 2    30     96.34    3.735e-09
## 3    40    108.28    1.951e-08
```

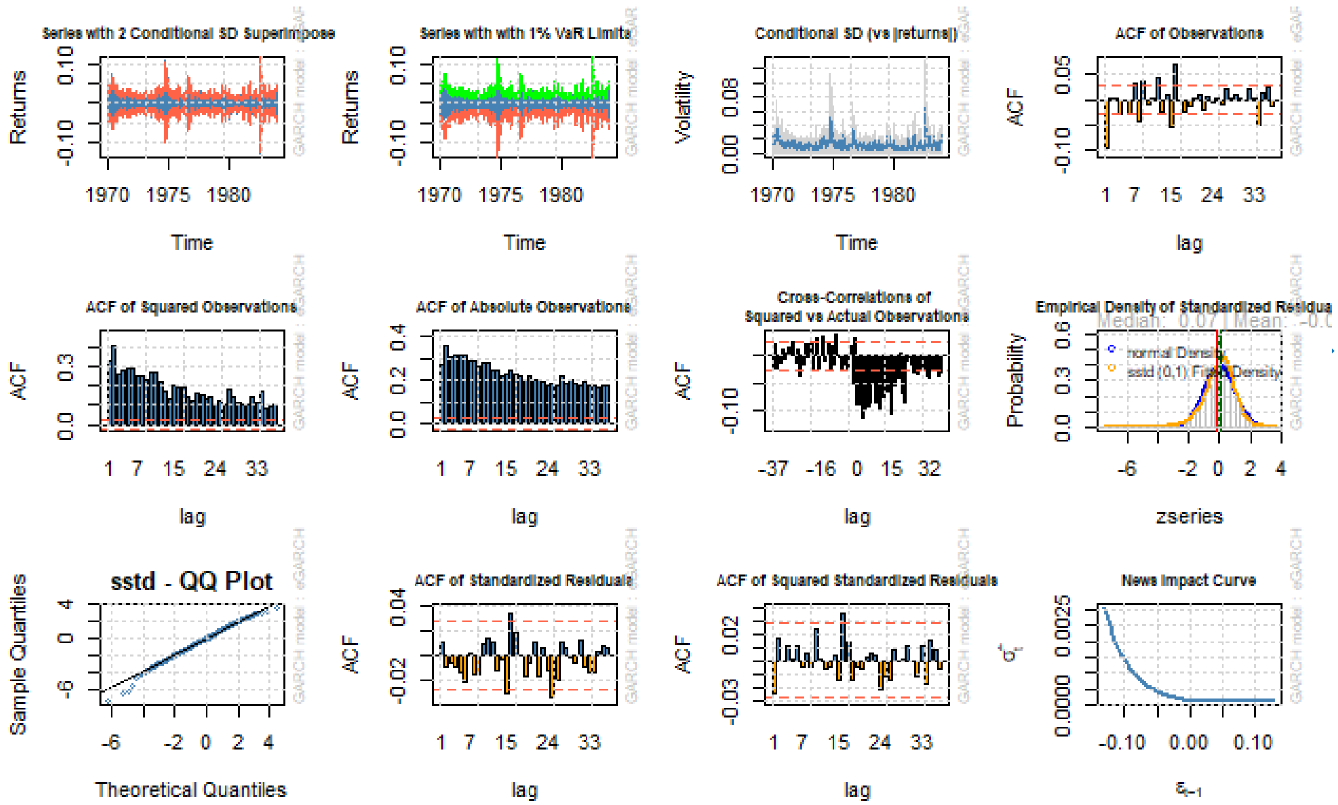
```
## 4     50     126.49     8.973e-09
##
```




```
ARIMA(0,1,1)-EGARCH(1,1) with skew Student-t distribution

##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model   : eGARCH(1,1)
## Mean Model    : ARFIMA(0,0,1)
## Distribution : sstd
##
## Optimal Parameters
## ------------------------------------
##          Estimate  Std. Error    t value Pr(>|t|)
## mu       0.000392    0.000122     3.2166 0.001297
## ma1     -0.048334    0.014064    -3.4367 0.000589
## omega   -0.210633    0.002761   -76.2810 0.000000
## alpha1  -0.140512    0.007253   -19.3722 0.000000
## beta1    0.976552    0.000025 39355.7694 0.000000
```

```
## gamma1  0.144602    0.007111    20.3358 0.000000
## skew    0.822723    0.016302    50.4686 0.000000
## shape   8.595820    0.995034     8.6387 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000392    0.000121     3.2376 0.001205
## ma1    -0.048334    0.011972    -4.0373 0.000054
## omega  -0.210633    0.002855   -73.7875 0.000000
## alpha1 -0.140512    0.008355   -16.8187 0.000000
## beta1   0.976552    0.000026 37642.8096 0.000000
## gamma1  0.144602    0.008143    17.7579 0.000000
## skew    0.822723    0.016172    50.8728 0.000000
## shape   8.595820    1.058131     8.1236 0.000000
##
## LogLikelihood : 15787.07
##
## Information Criteria
## ------------------------------------
##
## Akaike       -6.1964
## Bayes        -6.1861
## Shibata      -6.1964
## Hannan-Quinn -6.1928
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.5803  0.4462
## Lag[2*(p+q)+(p+q)-1][2]    0.7903  0.8562
## Lag[4*(p+q)+(p+q)-1][5]    1.3396  0.8838
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      3.122 0.07723
## Lag[2*(p+q)+(p+q)-1][5]     4.620 0.18621
## Lag[4*(p+q)+(p+q)-1][9]     5.299 0.38687
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3] 0.0007376 0.500 2.000  0.9783
## ARCH Lag[5] 0.5412769 1.440 1.667  0.8714
## ARCH Lag[7] 0.9966121 2.315 1.543  0.9141
##
## Nyblom stability test
## ------------------------------------
```

```
## Joint Statistic:  6.8131
## Individual Statistics:
## mu     0.1984
## ma1    0.1556
## omega  0.8033
## alpha1 0.9552
## beta1  0.6915
## gamma1 0.3379
## skew   2.3596
## shape  1.1097
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.89 2.11 2.59
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value     prob sig
## Sign Bias            1.364 0.172554
## Negative Sign Bias   2.289 0.022120  **
## Positive Sign Bias   2.248 0.024591  **
## Joint Effect        13.476 0.003713 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     29.53      0.05816
## 2    30     36.92      0.14835
## 3    40     49.53      0.12029
## 4    50     55.59      0.24061
##
##
## Elapsed time : 1.306241
```



Graphs: The sstd gives a better distribution based on QQ plot, both models are good in describing the news impact curve (the asymmetrical behavior of the negative news). There's absence of autocorrelation in the Squared Standardized residuals' ACF.

the arch LM test (in both models) shows that the series of residuals exhibits no conditional heteroscedasticity the Ljung-Box Test on Standardized (and Squared) Residuals shows that there is No serial correlation in the models, the eGARCH with sstd distribution shows better pvalues.

the Adjusted Pearson Goodness-of-Fit Test: it tests the null hypothesis that the data comes from a specified distribution: the model with skew Student-t distribution dosen't refuse the null hypothesis while the skew normal does.

In addition to that, the eGARCHwith sstd dist has the higher the value of the log-likelihood and Lower AIC score.

Markov-Switching model

The appmsgarch function (package SBAGM) computes the root mean square error (RMSE) and mean absolute error (MAE) of the different possible combinations of methods and distributions of the MS-GARCH model (link to theory, and R package)

```
package SBAGM – appmsgarch function on log returns


## $rmse_mat
##                snorm-snorm   snorm-sstd    snorm-ged    sstd-sstd     sstd-ged
## sGARCH-sGARCH 0.0002599888 0.0002903842 0.0002992829 0.0003166518 0.0003171690
## sGARCH-eGARCH 0.0001704455 0.0001771039 0.0001915731 0.0001843464 0.0001945981
## sGARCH-tGARCH 0.0002030667 0.0001958892 0.0001973595 0.0001910793 0.0001995379
## eGARCH-eGARCH 0.0002421234 0.0002430714 0.0002510836 0.0002359714 0.0002643738
## eGARCH-tGARCH 0.0002439953 0.0002659066 0.0002436313 0.0002371769 0.0002617593
## tGARCH-tGARCH 0.0001984178 0.0001973887 0.0002076179 0.0002091543 0.0002074019
##                    ged-ged
## sGARCH-sGARCH 0.0003186718
## sGARCH-eGARCH 0.0001907644
## sGARCH-tGARCH 0.0001949679
## eGARCH-eGARCH 0.0002517277
## eGARCH-tGARCH 0.0002416700
## tGARCH-tGARCH 0.0002241362
##
## $mae_mat
##                snorm-snorm   snorm-sstd    snorm-ged    sstd-sstd     sstd-ged
## sGARCH-sGARCH 0.0002177921 0.0002511812 0.0002586679 0.0002780012 0.0002780529
## sGARCH-eGARCH 0.0001581886 0.0001546021 0.0001522445 0.0001551695 0.0001523699
## sGARCH-tGARCH 0.0001535519 0.0001533443 0.0001535570 0.0001527730 0.0001531841
## eGARCH-eGARCH 0.0001870319 0.0001894876 0.0002009011 0.0001793010 0.0002159552
## eGARCH-tGARCH 0.0001838432 0.0002184637 0.0001900387 0.0001814588 0.0002130655
## tGARCH-tGARCH 0.0001534996 0.0001534850 0.0001471818 0.0001528541 0.0001530061
##                    ged-ged
## sGARCH-sGARCH 0.0002799721
## sGARCH-eGARCH 0.0001524465
## sGARCH-tGARCH 0.0001517972
## eGARCH-eGARCH 0.0002003714
## eGARCH-tGARCH 0.0001875279
## tGARCH-tGARCH 0.0001654591
```

the appmsgarch function suggests a eGARCH-eGARCH with sstd-ged distribution.

```
So now I specify a msgarch model with eGARCH – eGARCH and sstd-ged distribution

Summary:

## Specification type: Markov-switching
## Specification name: eGARCH_sstd eGARCH_ged
## Number of parameters in each variance model: 4 4
## Number of parameters in each distribution: 2 1
## ------------------------------------------
## Fixed parameters:
## None
## ------------------------------------------
## Across regime constrained parameters:
## None
## ------------------------------------------
```

Maximum Likelihood estimation:

```
## Specification type: Markov-switching
## Specification name: eGARCH_sstd eGARCH_ged
## Number of parameters in each variance model: 4 4
## Number of parameters in each distribution: 2 1
## ------------------------------------------
## Fixed parameters:
## None
## ------------------------------------------
## Across regime constrained parameters:
## None
## ------------------------------------------
## Fitted parameters:
##          Estimate Std. Error  t value  Pr(>|t|)
## alpha0_1  -0.6853     0.1143  -5.9959 1.012e-09
## alpha1_1   0.0967     0.0195   4.9479 3.751e-07
## alpha2_1  -0.2353     0.0200 -11.7659    <1e-16
## beta_1     0.9258     0.0122  76.1360    <1e-16
## nu_1       8.6408     1.4233   6.0710 6.355e-10
## xi_1       0.7938     0.0207  38.3908    <1e-16
## alpha0_2  -0.2636     0.0553  -4.7672 9.339e-07
## alpha1_2   0.1318     0.0226   5.8297 2.777e-09
## alpha2_2  -0.1275     0.0177  -7.2153 2.690e-13
## beta_2     0.9676     0.0068 141.5611    <1e-16
## nu_2       1.7214     0.0898  19.1716    <1e-16
## P_1_1      0.9978     0.0016 620.7749    <1e-16
## P_2_1      0.0035     0.0011   3.2943 4.933e-04
## ------------------------------------------
## Transition matrix:
##       t+1|k=1 t+1|k=2
## t|k=1  0.9978  0.0022
## t|k=2  0.0035  0.9965
## ------------------------------------------
```

```
## Stable probabilities:
## State 1 State 2
##  0.6115  0.3885
## -----------------------------------------
## LL: 15804.1234
## AIC: -31582.2467
## BIC: -31497.2836
## -----------------------------------------
```

the parameters indicate that the evolution of the volatility process is quite heterogeneous across the 2 regimes.

alpha2_1 = -0.2323, alpha2_2 = -0.1280 indicates different reactions to past returns.

looking at State 1 0.6613, State 2 0.3387 i understand that markov chain evolve persistently over time

persistence of volatility process: the first regime reports alpha1_1 + alpha2_1 + beta_1 = 0.7908 the second regime reports alpha1_2 + alpha2_2 + beta_2 = 0.975

the first regime has lower unconditional volatility, weaker volatility reaction to past negative returns and lower persistence of volatility process. the second instead is characterized by higher unconditional volatility, stronger volatility reaction to past negative returns and higher persistence of volatility process.

```
Unconditional volatility:

## [1]    NA   0.2958819
```

Plot of estimated smoothed probability superimposed on the Nasdaq returns

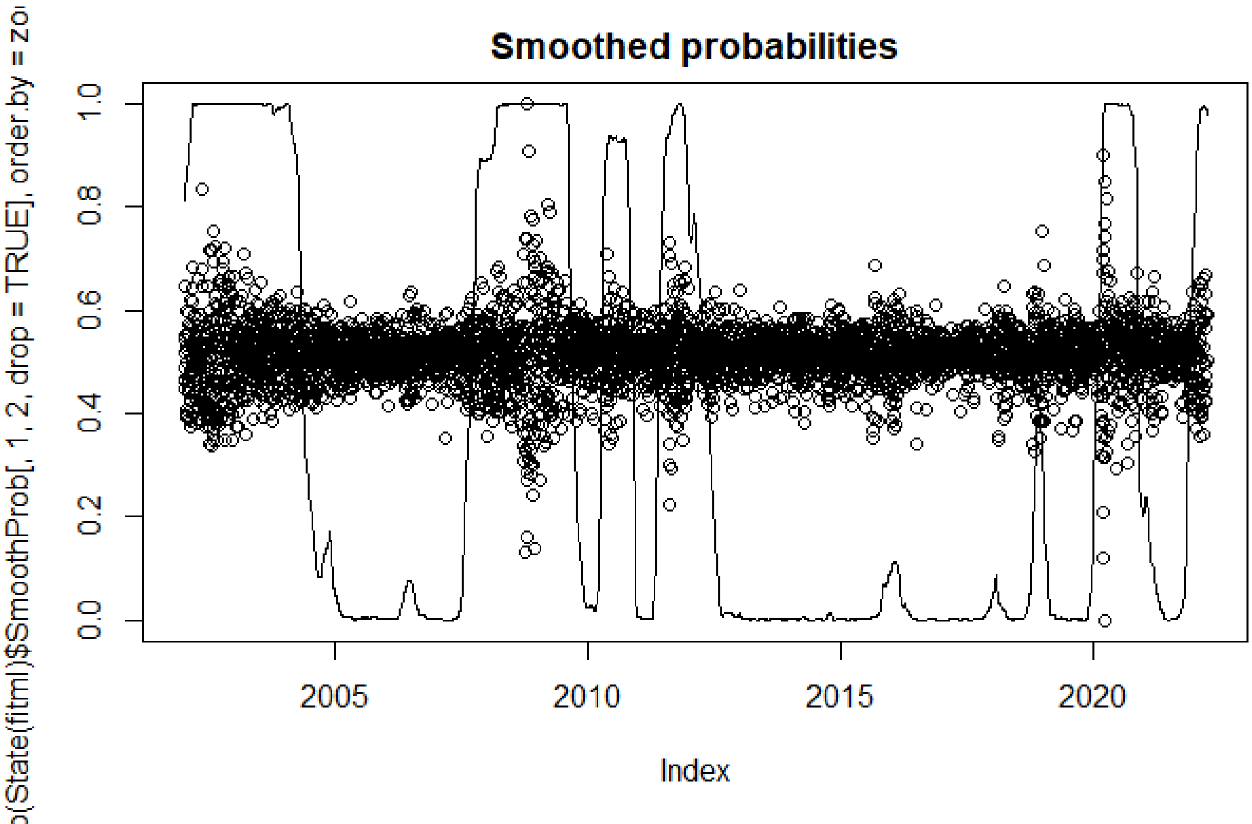


image and code from:Markov-Switching GARCH Models in R:The MSGARCH Package Journal of Statistical Software

State() function reports a list of 4 elements: FiltProb, PredProb and SmoothProb are three array of T x 1 x K dimension containing the filtered, predicted and smoothed probabilities at each time t. The last element, Viterbi is a matrix of dim T x 1 representing decoded states according to the Viterbi algo. [Viterbi is not a topic of this work]

when the smoothed probabilities are near one the filtered volatility of the process (graph below) increases. I further note that the Markov chain evolves persistently over time. [State 1 0.6613, State 2 0.3387]

#Bayesian approach ML estimation can be difficult for MSGARCH-type models. MCMC procedures can be used to explore the joint posterior distribution of the models' parameters, avoiding convergence to local maxima which are possible in ML estimation.

The Bayesian approach also can obtain at low cost, by simulating from the joint posterior distribution, the exact distributions of nonlinear function of the model parameters. In addition, parameter uncertainty can be integrated in the forecasts through the predictive distribution.

The posterior distribution and Markov-switching models often exhibits non elliptical shapes which lead to non-reliable estimation of the uncertainty of model parameters. This invalidates the use of the Gaussian asymptotic distribution for inferential purposes in the finite samples.

MCMC estimation Markov Chain Monte Carlo / Bayesian estimation:

```
fit.mcmc <- FitMCMC(mrs_spec, data = return)
summary(fit.mcmc)
```

```
## Specification type: Markov-switching
## Specification name: eGARCH_sstd eGARCH_ged
## Number of parameters in each variance model: 4 4
## Number of parameters in each distribution: 2 1
## ------------------------------------------
## Fixed parameters:
## None
## ------------------------------------------
## Across regime constrained parameters:
## None
## ------------------------------------------
## Posterior sample (size: 1000)
##             Mean     SD     SE   TSSE    RNE
## alpha0_1 -0.7306 0.0863 0.0027 0.0197 0.0191
## alpha1_1  0.1081 0.0234 0.0007 0.0013 0.3442
## alpha2_1 -0.2672 0.0200 0.0006 0.0013 0.2532
## beta_1    0.9204 0.0088 0.0003 0.0020 0.0195
## nu_1     11.2814 1.2271 0.0388 0.2405 0.0260
## xi_1      0.7590 0.0285 0.0009 0.0067 0.0180
## alpha0_2 -0.1646 0.0264 0.0008 0.0048 0.0297
## alpha1_2  0.1258 0.0224 0.0007 0.0024 0.0889
## alpha2_2 -0.1020 0.0160 0.0005 0.0016 0.0970
## beta_2    0.9804 0.0034 0.0001 0.0006 0.0311
## nu_2      1.5994 0.0829 0.0026 0.0085 0.0956
## P_1_1     0.9921 0.0011 0.0000 0.0003 0.0150
## P_2_1     0.0142 0.0016 0.0001 0.0002 0.0839
## ------------------------------------------
## Posterior mean transition matrix:
##       t+1|k=1 t+1|k=2
## t|k=1  0.9921  0.0079
## t|k=2  0.0142  0.9858
## ------------------------------------------
## Posterior mean stable probabilities:
## State 1 State 2
##  0.6428  0.3572
## ------------------------------------------
## Acceptance rate MCMC sampler: 27.7%
## nmcmc: 10000
## nburn: 5000
## nthin: 10
## ------------------------------------------
## DIC: -31571.7852
## ------------------------------------------
```

Scatter plot of posterior draws form the marginal distribution of alpha1_1, alpha2_2

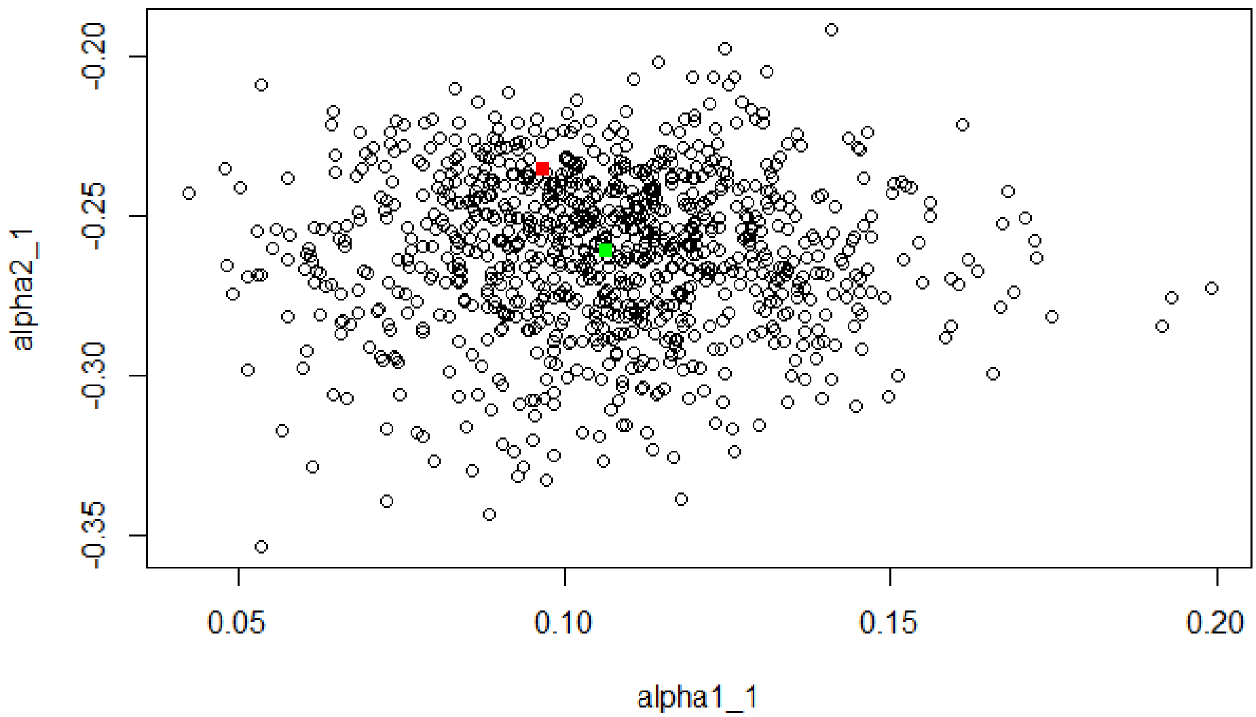


image and code from:Markov-Switching GARCH Models in R:The MSGARCH Package Journal of Statistical Software

i plot 2000 draws of the posterior sample for parameters alpha1_1, alpha1_2. The green square reports the posterior mean, the red square reports the ML estimate.

With the Bayesian estimation we can make distributional statements on any function of model parameters, achieved by simulation. [for each draw in the posterior sample we can compute the unconditional volatility in each regime to get its posterior distribution].

```
dim(tmp)

## [1] 1000    2
```

I made a function that computes the unconditional volatility, extract the posterior distributions of the unconditional annualized volatility in each regime and made a histogram.

```
Histograms of posterior distribution for the unconditional volatility in each regime:
```

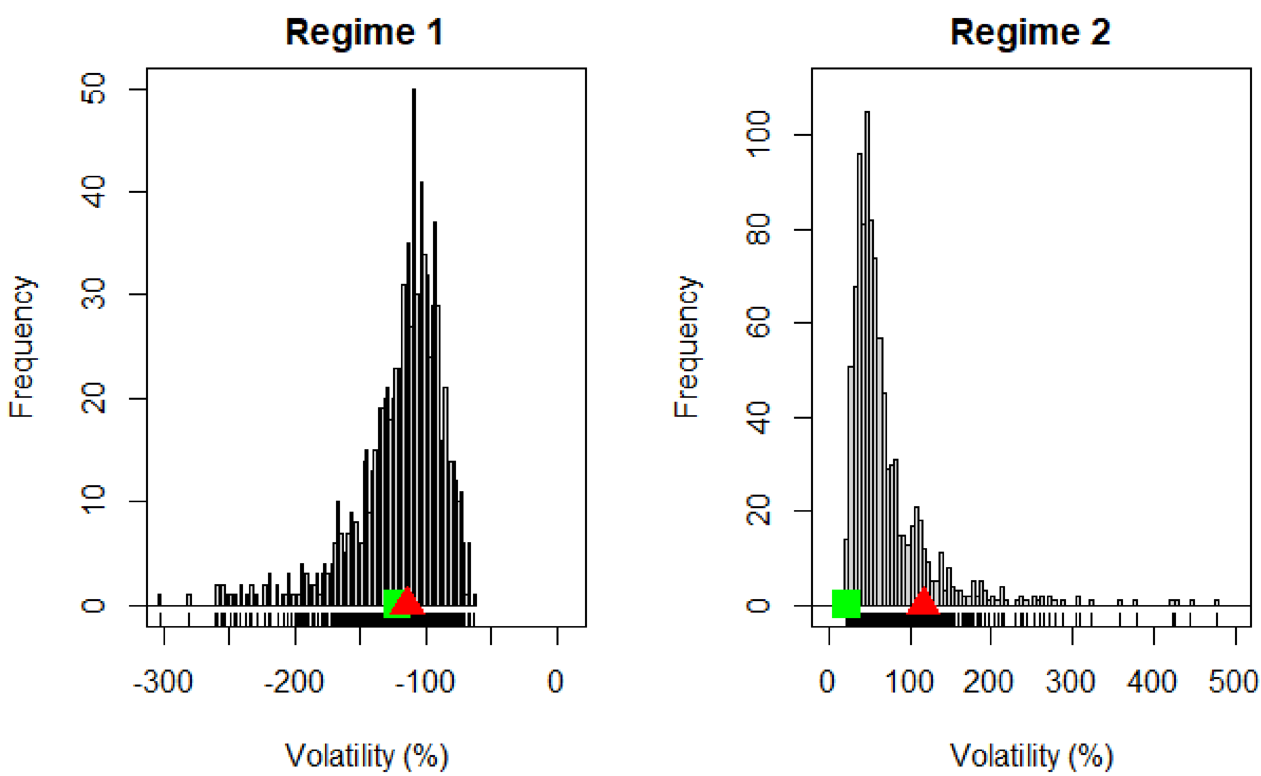


image and code from:Markov-Switching GARCH Models in R:The MSGARCH Package Journal of Statistical Software

based on 2000 draws from the joint posterior sample

```
Quantiles of unconditional volatility:

##          ucvol_1   ucvol_2
## 2.5%  -221.38704  24.69504
## 97.5%  -74.76481 200.36906
```

Now I want to include parameter uncertainty in the one step ahead predictive density of MSGARCH models and evaluate the one-step ahead predictive density in the range of values from −5 to 0

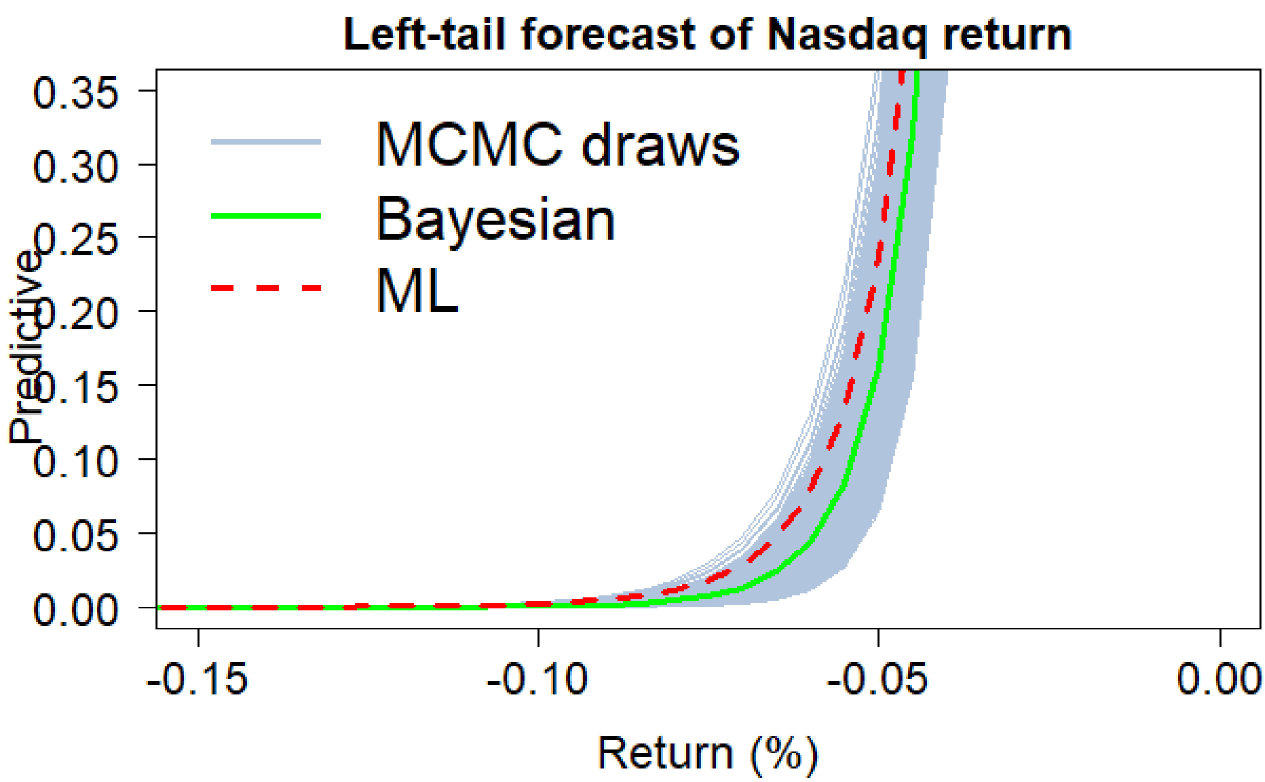


image and code from:Markov-Switching GARCH Models in R:The MSGARCH Package Journal of Statistical Software

Left tail of the one-step ahead predictive distribution for the Nasdaq returns, the green line reports the predictive Bayesian density, the red line reports the ML conditional density, and the blue lines report the conditional densities obtained for each 2000 draws in posterior sample.

The Bayesian predictive density is a particular average of the predictive densities that can be formed with individual poster MCMC draws. It's more conservative that the predictive density with plugged ML estimates and offers additional flexibility by accounting for all likely scenarios within the model structure.

VaR

#Back test analysis of the Markov Switching eGarch-eGarch model against its single-regime counterpart

```
First i create the single-regime model and gather both specification in a list

Then I define the back test settings: 1000 out-of-sample observations and focus on the
one-step ahead VaR forecasts at 5% risk lvl

Forecasts based on rolling windows of 1500 observations and models are re-estimated every
100 observations.
```

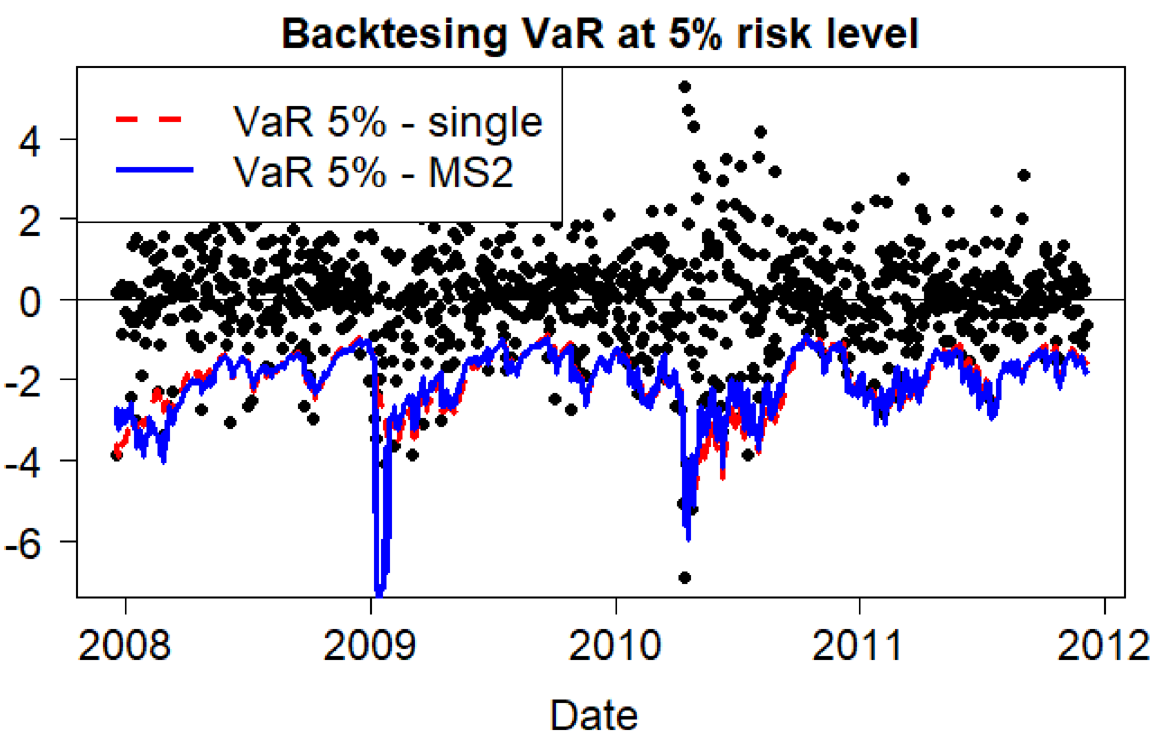


image and code from:Markov-Switching GARCH Models in R:The MSGARCH Package Journal of Statistical Software

1 day ahead VaR forecasts at 5% risk level, the Markov-switching (blue line) and single-regime (red line), with the realized returns.

I notice discrepancy during the stress periods of the stock market, I need to evaluate which VaR forecasts are most accurate in predicting the α-quantile loss (that I expect to have a proportion α of exceedances).

[GAS package]

compute the p values of two back testing hypothesis tests of correct conditional coverage of the VaR (CC) and the dynamic quantile (DQ) test

they determine if the VaR forecasts achieve correct unconditional coverage and if the violations of the VaR are independent over time

```
print(CC.pval)

##    single       MS2
## 0.0848907 0.0891077

print(DQ.pval)

##    single       MS2
## 0.1104897 0.1385840
```

For both tests I notice the better performance of Markov-Switching specification

Conclusion

CONCLUSION (MSGARCH)

MRS-GARCH model enjoy popularity in the financial studies with the presence of structural breaks. In this

study, we sought to identify the optimal model to capture the characteristics of Nasdaq Index over the period of

2002-01-02–2022-03-25. We used the MRS-GARCH family models to investigate the volatility of the series. In this paper we used the log-likelihood function and the Monte Carlo simulation for modelling the GARCH and MRS-GARCH family models.

Then their performance was compared with those of ARMA-GARCH family models under the normal, student-t (skewed) and GED distributions. Overall, our findings demonstrated strong evidence of switching behaviour in the Nasdaq's stock market between two states. We found the MRS-GARCH models to provide a more powerful tool in modelling the series' volatility than the traditional volatility models. The findings indicated that MRS-EGARCH model under Student-t skewed distributions has better results than other MRS-GARCH models.

We also investigated the advantages of Bayesian approach, especially while working with posterior distribution and parameters uncertainty for forecasting through the predictive distribution. Then we analysed the resulting VaR forecasts of a single regime and two-regime model together with out-of-sample (realized) returns. Surprisingly the most accurate VaR forecasts in term of predicting the α-quantile loss (such that we expect to have a proportion α of exceedances) discovered by CC and DQ tests is the single-regime.

"I would like to conclude this paper by outlining some points, which deserve further investigation. First, one could investigate the performances of MRS-GARCH models for forecasting. The estimation of the VaR depends on the reasonable assumption of innovations, and is, therefore, beyond the scope of this paper. Second, the univariate MRS-GARCH models could be extended to multivariate ones, which could model the covariance between two different asset returns in order to investigate hedging and asset allocations".

Conclusion written with the help of [Mehdi Zolfaghari, Bahram Sahabi paper].

Hybrid ARIMA-GARCH in Trading strategies

For the last part of my thesis, I applied the theory of ARIMA-(MS)GARCH models and used for building a trading strategy.

Even though I tried to build my own strategy (from quantstart) I found it very difficult, so I chose to illustrate the strategy from others work.

Fist I am going to compare the ARIMA's performance with the combination of ARIMA and GARCH family models to forecast log returns (SP500).

Source: Article, *Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk*

By using a rolling window approach, i compared ARIMA with the hybrid models (arima-sGarch, arima-eGarch)

To compare the precision of these models (in forecasting), i compared their equity lines, their forecasting error metrics (MAE, MAPE, RMSE, MAPE), and their performance metrics (annualized return compounded, annualized standard deviation, maximum drawdown, information ratio, and adjusted information ratio).

The aim of this work is to show that the hybrid models outperform ARIMA and the benchmark (Buy&Hold strategy on S&P500) over the long-term period.

The results are not sensitive to varying window sizes, the type of distribution, and the type of the GARCH model.

◊ Firstly, i conducted a rolling forecast based on the ARIMA model (window size s = 1000), while the most common approach was based on simple division of in-sample and one out-of-sample sets.
The optimized combination of p and q which has the lowest AIC is used to predict the return for the next day.

◊ Secondly, i describe and the implementation of ARIMA(p,1,q)-SGARCH(1,1) models with generalized error distribution (GED) and window size =1000, in which optimized ARIMA(p,1,q) is taken from the first step

◊ Thirdly, i evaluate the performance of sGarch with different window sizes as well as various distributions to check the sensitivity of the results.
eGARCH was also applied in the sensitivity analysis in order to check the robustness of the initial assumptions. (checking various types of GARCH models and various assumptions concerning the error distributions).

• forecasts' precision is verified in the 2-step procedure which combines the evaluation of econometric model forecasts with standard error metrics and the evaluation of investment signals constructed on these forecasts with the help of performance metrics calculated based on final equity lines.

Methodology and input Parameter

• rolling forecast based on an ARIMA model with window size s= 1000 (optimized combination of p, q with the lowest AIC is used to predict return for the next step)

• implementation of dynamic ARIMA(p,1,q)-sGARCH(1,1) models with GED distribution and window size = 1000 and where optimized ARIMA(p,1,q) is given by the first step

• evaluate the results based on error metrics, performance metrics, and equity curve

•• then hybrid models with different input parameters are built, different window size (500, 1500) and different (skewed) distributions: SNORM, SSTD, SGED

••• replace sGARCH with eGARCH, conduct forecasting ARIMA on different window sizes
Does the hybrid model outperform ARIMA in different input variables?

Sets of input parameters (ARIMA/hybrid ARIMA-xGARCH).

Parameters	Values
Sample sizes	s ∈ {500, **1000**, 1500} (days)
Distribution	**Generalized Error Distribution (GED)** Skewed Normal Distribution (SNORM) Skewed Generalized Error Distribution (SGED) Skewed Student t Distribution (SSTD)
xGARCH MODEL	x ∈ {**SGARCH**, eGARCH} (x represents the type of tested GARCH model. In other words, x is either **symmetric** (s)GARCH or exponential (e)GARCH.

Note: The letters in bold represent the parameters in the main test.

Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk implementation of dynamic **ARIMA(p,1,q)**-sGARCH(1,1):

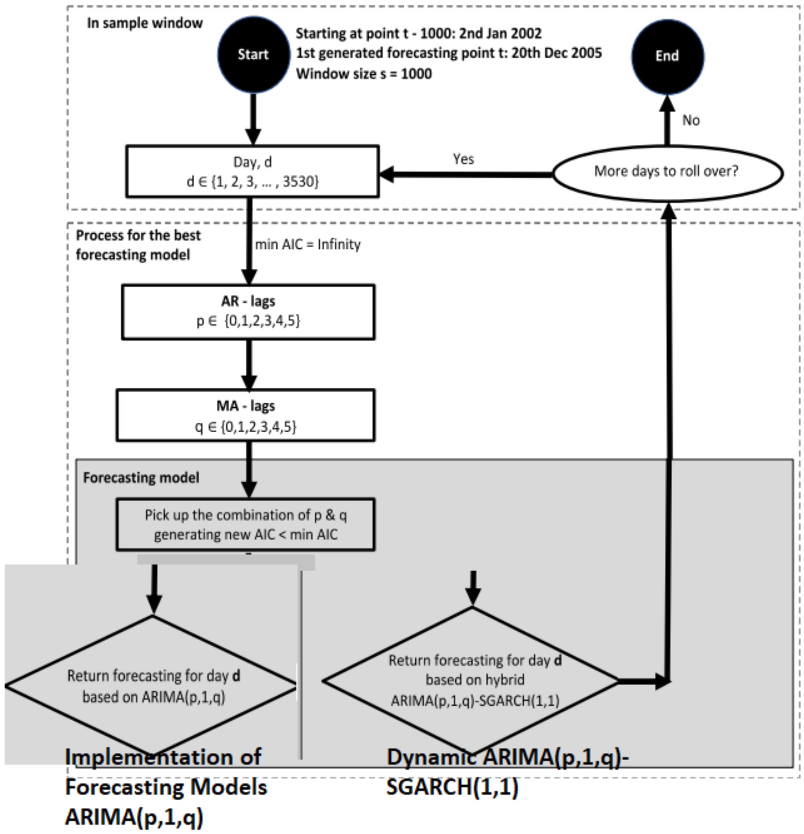


Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

Data

The performance of ARIMA, hybrid model ARIMA(p,1,q)-SGARCH(1,1) with GED distribution and Buy&Hold strategy (window size = 1000)

Forecasting performance of ARIMA(p,1,q) and ARIMA(p,1,q)-SGARCH(1,1).

	Error Metrics				Performance Statistics				
METHOD	MAE	MSE	RMSE	MAPE	ARC	ASD	MD	IR	IR*
BUY&HOLD S&P500					6.931%	**18.826%**	56.775%	0.368	0.045
ARIMA 1000	12.122	310.372	17.617	0.00775	8.084%	18.878%	50.007%	0.428	0.069
SGARCH.GED 1000	**11.831**	**303.044**	**17.408**	**0.00754**	**14.026%**	18.893%	**25.885%**	**0.742**	**0.402**

Note: In order to simplify the structure of the table, SGARCH.GED 1000 is understood as ARIMA(p,1,q)-SGARCH(1,1) with GED distribution and window size equal to 1000 days. MAE: mean absolute error; MSE: mean squared error; RMSE: root mean squared error; MAPE: mean absolute percentage error; ARC: annualized return compounded; ASD: annualized standard deviation; MD: maximum drawdown; IR = ARC/ASD: information ratio; IR* = (ARC^2 * sign(ARC))/(ASD * MD): adjusted information ratio. Figures in bold indicate the best results.

Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

SGARCH.GED is more accurate than ARIMA in predicting returns and has the lowest values of MAE, MSE, RMSE, and MAPE.

Looking at performance statistics, the hybrid model generates the highest IR among the 3 methods.
The difference between IR and IR* is that we additionally take into account MD as a measure of risk beside ASD.

cumulative returns of the strategies are plotted to visualize the performance of ARIMA(p,1,q), hybrid model with GED distribution and benchmark.

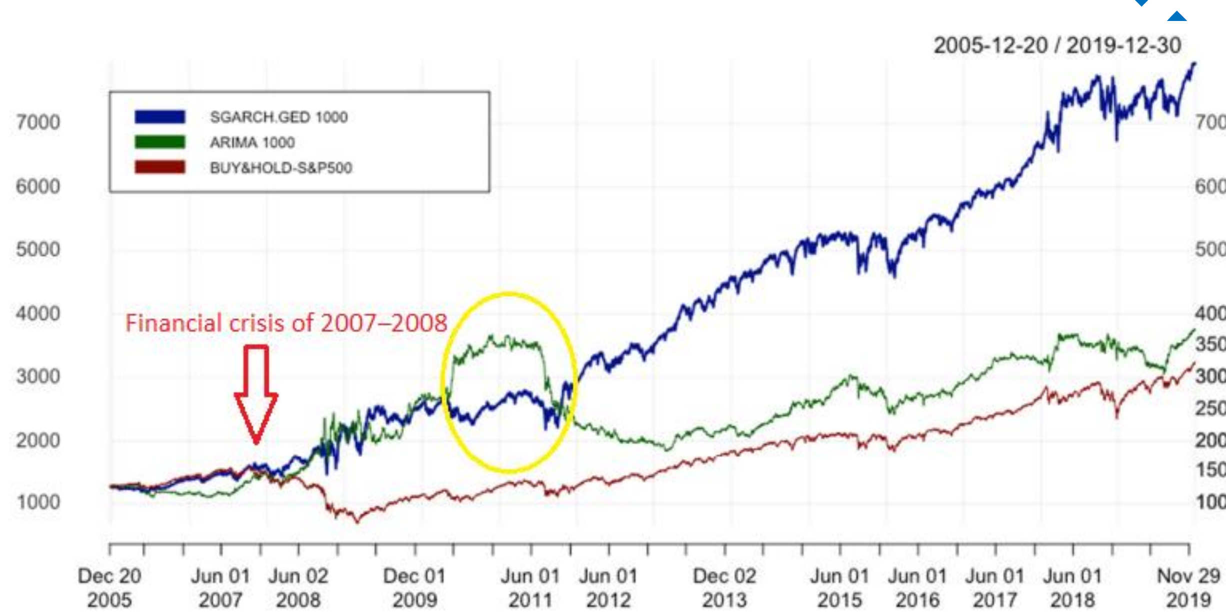


Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

Before the financial crisis the buy and hold strategy remained above the ARIMA-GARCH, then in a short period time of 2 years ARIMA outperforms ARIMA-GARCH, then the hybrid model did proof to be the best model, it captures well all the movements of time series and is much better when compared with the benchmark

Robustness Test:

verify whether the results we obtained above are robust to varying family of GARCH, various distributions, and different window lengths.

◊ first robustness test → substitute sGARCH with eGARCH (keeping GED distribution and same window size).

◊ second → change GED to SNROM, SSTD, and SGED (other conditions remain unchanged).

◊ last → replace window size of 1000 to 500 and 1500 (remaining conditions are kept the same).

Forecasting performance of ARIMA(p,1,q) & ARIMA(p,1,q)-EGARCH(1,1).

	Error Metrics				Performance Statistics				
METHOD	**MAE**	**MSE**	**RMSE**	**MAPE**	**ARC**	**ASD**	**MD**	**IR**	**IR***
BUY & HOLD S&P500					6.931%	**18.826%**	56.775%	0.368	0.045
ARIMA 1000	12.122	310.372	17.617	0.00775	8.084%	18.878%	50.007%	0.428	0.069
SGARCH.GED 1000	11.831	303.044	17.408	0.00754	**14.026%**	18.893%	**25.885%**	**0.742**	**0.402**
EGARCH.GED 1000	**11.828**	**301.745**	**17.371**	**0.00753**	11.010%	18.901%	29.150%	0.582	0.220

Note: In order to simplify the structure of the table, EGARCH.GED 1000 is understood as ARIMA(p,1,q)-EGARCH(1,1) with GED distribution and window size equal to 1000 days. MAE: mean absolute error; MSE: mean squared error; RMSE: root mean squared error; MAPE: mean absolute percentage error; ARC: annualized return compounded; ASD: annualized standard deviation; MD: maximum drawdown; IR = ARC/ASD: information ratio; IR* = (ARC^2 * sign(ARC))/(ASD * MD): adjusted information ratio. The figures in bold indicate the best results.

Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

The combination of ARIMA(p,1,q) and EGARCH(1,1) outperforms ARIMA in a similar way as the combination of ARIMA(p,1,q) and SGARCH(1,1), even though error metrics of EGARCH.GED.1000 have the lowest values, it cannot beat the SGARCH.GED 1000 in terms of performance statistics.

EGARCH is introduced as more advanced than SGARCH since it takes the magnitude of volatility into consideration. However, the result based on IR*, which is selected as the most important performance statistic to evaluate the model, does not support this theory. *The best model is not necessarily the same when the selection is based on the best error metrics or the best performance statistics*.

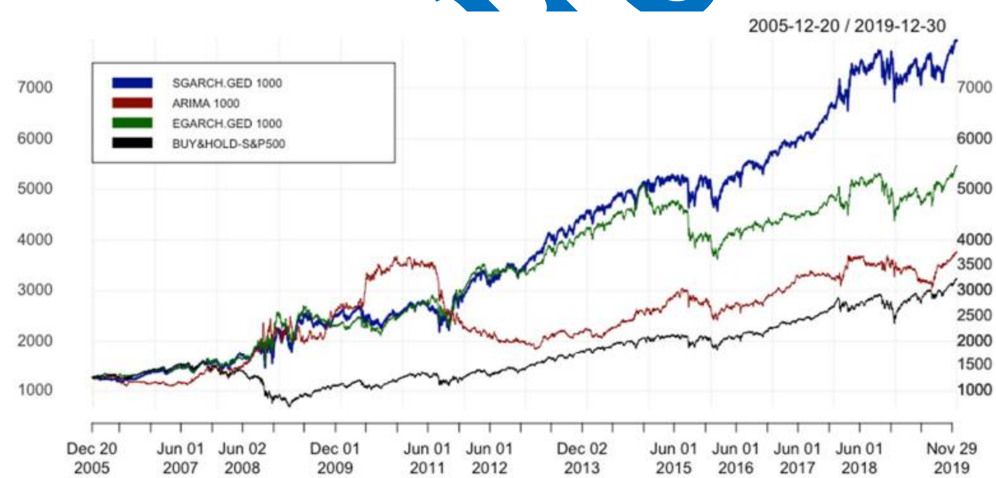


Figure 8. Equity curves of ARIMA(p,1,q) & ARIMA(p,1,q)-EGARCH(1,1). In order to simplify the structure of the legend, SGARCH.GED 1000 is understood as ARIMA(p,1,q)-SGARCH(1,1) with GED distribution and window size equal to 1000 days; EGARCH.GED 1000 is understood as ARIMA(p,1,q)-EGARCH(1,1) with GED distribution and window size equal to 1000; ARIMA 1000 is ARIMA(p,1,q) with window size s = 1000; BUY&HOLD-S&P500 is the benchmark strategy.

Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

Varying Window Sizes:

Performance of ARIMA(p,1,q) and hybrid models in different window sizes.

	Error Metrics				Performance Statistics				
METHOD	**MAE**	**MSE**	**RMSE**	**MAPE**	**ARC**	**ASD**	**MD**	**IR**	**IR***
BUY & HOLD S&P500					6.931%	**18.826%**	56.775%	0.368	0.045
ARIMA 500	12.216	318.342	17.842	0.00777	−0.573%	18.830%	46.471%	−0.030	0.000
SGARCH.GED 500	11.91	307.812	17.545	0.00758	5.912%	18.871%	35.666%	0.313	0.052
ARIMA 1000	12.122	310.372	17.617	0.00775	8.084%	18.878%	50.007%	0.428	0.069
SGARCH.GED 1000	11.831	**303.044**	**17.408**	**0.00753**	**14.026%**	18.893%	**25.885%**	**0.742**	**0.402**
ARIMA 1500	12.069	308.983	17.578	0.00771	5.005%	18.852%	50.733%	0.265	0.026
SGARCH.GED 1500	**11.825**	303.298	17.415	0.00753	12.186%	18.896%	25.885%	0.645	0.304

Note: In order to simplify the structure of the table, SGARCH.GED 500 is understood as ARIMA(p,1,q)-SGARCH(1,1) with GED distribution and window size s = 500 days, similar for s = 1000 and = 1500 days. MAE: mean absolute error; MSE: mean squared error; RMSE: root mean squared error; MAPE: mean absolute percentage error; ARC: annualized return compounded; ASD: annualized standard deviation; MD: maximum drawdown; IR = ARC/ASD: information ratio; IR* = (ARC^2 * sign(ARC))/(ASD * MD): adjusted information ratio. The figures in bold indicate the best results.

Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

With the window size switched from 500 to 1500, the hybrid models are superior to ARIMA (as I want to demonstrate),

Figure 9. Equity curves of ARIMA(p,1,q) and hybrid models with window sizes = 500 and 1000. In order to simplify the structure of the legend, SGARCH.GED 500/SGARCH.GED 1000 is understood as ARIMA(p,1,q)-SGARCH(1,1) with GED distribution and window size(s) equal to 500/1000 days.

Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

ARIMA 500 and SGARCH.GED 500 underperform benchmark at the end. The Buy&Hold strategy is not the worst. In general, hybrid models seem to be sensitive to the values of window size. However, i can still conclude that *hybrid models outperform ARIMA regardless of the values of window size as an input parameter*.

Varying Distributions:

Performance of ARIMA(p,1,q) and hybrid models in different distributions.

	Error Metrics				Performance Statistics				
METHOD	**MAE**	**MSE**	**RMSE**	**MAPE**	**ARC**	**ASD**	**MD**	**IR**	**IR***
BUY & HOLD S&P500					6.931%	**18.826%**	56.775%	0.368	0.045
ARIMA 1000	12.122	310.372	17.617	0.00775	8.084%	18.879%	50.007%	0.428	0.069
SGARCH.GED 1000	**11.831**	**303.044**	**17.408**	**0.00754**	**14.026%**	18.893%	**25.885%**	**0.742**	**0.402**
SGARCH.SNORM 1000	11.880	303.151	17.411	0.00758	8.987%	18.890%	33.079%	0.476	0.129
SGARCH.SSTD 1000	11.928	305.642	17.483	0.00762	8.860%	18.881%	28.373%	0.469	0.147
SGARCH.SGED 1000	11.848	302.362	17.389	0.00755	9.201%	18.859%	37.566%	0.488	0.119

Note: In order to simplify the structure of the table, SGARCH.GED 1000 is understood as ARIMA(p,1,q)-SGARCH(1,1) with GED distribution and window size s = 1000 days, similar for SNORM, SGED, and SSTD. MAE: mean absolute error; MSE: mean squared error; RMSE: root mean squared error; MAPE: mean absolute percentage error; ARC: annualized return compounded; ASD: annualized standard deviation; MD: maximum drawdown; IR = ARC/ASD: information ratio; IR* = (ARC^2 * sign(ARC))/(ASD * MD): adjusted information ratio. Figures in bold indicate the best results.

Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

ARIMA has the worst performance with the highest values of MAE, MSE, RMSE, and MAPE if compared to hybrid models.

Figure 12. Equity curves of all hybrid models with different distributions. In order to simplify the structure of the legend, SGARCH.GED 1000 is understood as ARIMA(p,1,q)-SGARCH(1,1) with GED distribution and window size(s) equal to 1000 days, similar for SNORM, SSTD, and SGED.

Image from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

Cumulative returns of hybrid models with SNORM, SSTD, and SGED distributions show no significant differences at the end, but all of them surpass ARIMA's.

Results

The ARIMA(p,1,q)-SGARCH(1,1) (hybrid model) with window size 1000 can generate a trading strategy that outperforms ARIMA(p,1,q) and Buy&Hold.

The *main test* are robust to varying family of GARCH models, varying window sizes, and varying distributions.

The performance of hybrid models in the *main test* change with varying family of GARCH model, varying window sizes and varying distributions.

Conclusion from: Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index Nguyen Vo and Robert Slepaczuk

Trading using Garch Volatility Forecast (regime -switching)

Regime Switching System Using Volatility Forecast

This topic has been taken from *systematicinvestor in R bloggers*, both code and comments

Build an algorithm to switch between mean-reversion and trend-following strategies based on the market volatility.

◊ Two models are examined: one using the historical volatility and another using the Garch(1,1) Volatility Forecast

• The mean-reversion strategy is modelled with RSI(2) [long with RSI(2), short otherwise]

• The trend-following strategy is modeled with SMA 50/200 crossover (long if SMA 50 > 200, short if reverse)

Systematic Investor Toolbox (SIT) is used

Methodology and data

Compares performance of the Buy and Hold, Mean-Reversion, and Trend-Following strategies using the back testing library in the Systematic Investor Toolbox

Image from: Trading using Garch Volatility Forecast by systematicinvestor in R bloggers

Now create a strategy that switches between mean-reversion and trend-following strategies based on historical market volatility:

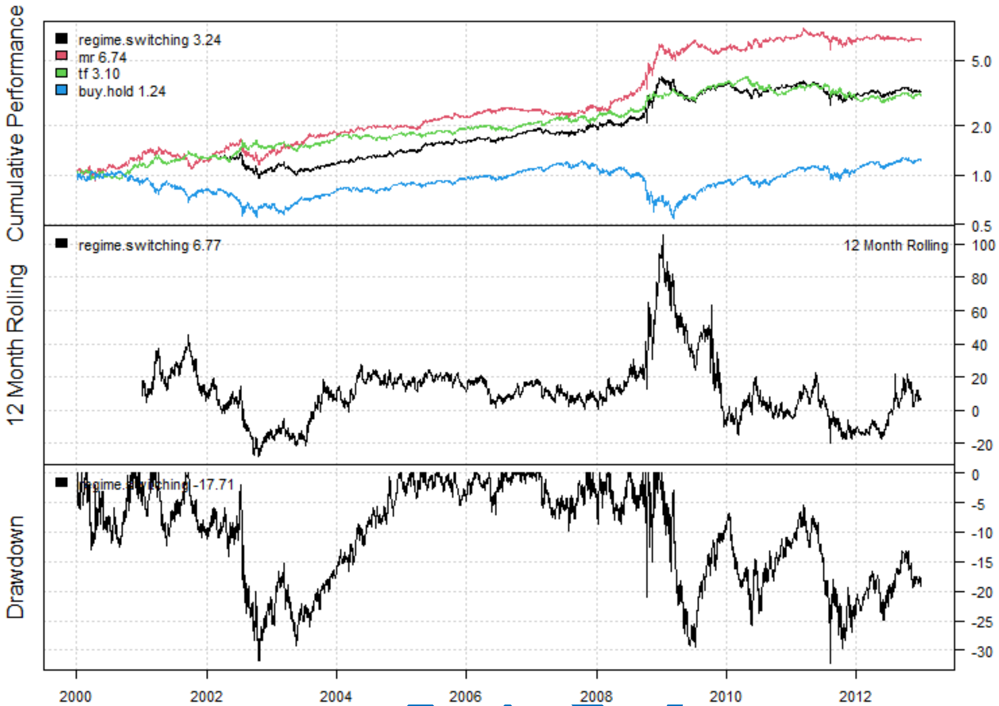


Image from: Trading using Garch Volatility Forecast by systematicinvestor in R bloggers

Next, create a GARCH(1,1) Volatility Forecast (references in the web page).

Used: `packages('tseries,fGarch')`

`garch` function from `tseries` package and `garchFit` function from `fGarch` package.

Garch function from tseries package is faster but doesn't always find solution, while garchFit function from fGarch package is slower but does converge more consistently.

Then create a strategy that switches between mean-reversion and trend-following strategies based on GARCH(1,1) volatility forecast:

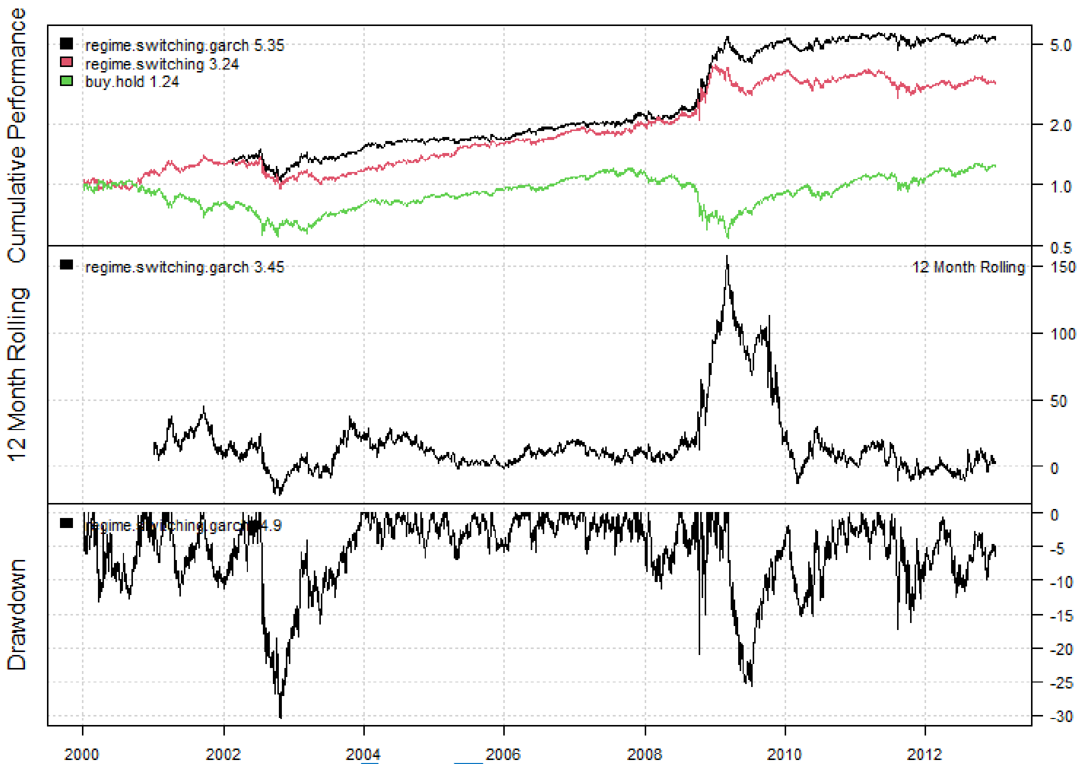


Image from: Trading using Garch Volatility Forecast by systematicinvestor in R bloggers

Results

The switching strategy that uses GARCH(1,1) volatility forecast performs better than the one which uses historical volatility.

Full code: here

Code used here.

Plagiarism checks here.

References

Akaike H (1974). "A New Look at the Statistical Model Identification." IEEE Transactions on Automatic Control, 19(6), 716–723. doi:10.1007/978-1-4612-1694-0_16.

Ardia D (2008). Financial Risk Management with Bayesian Estimation of GARCH Models: Theory and Applications, volume 612 of Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin Heidelberg. doi:10.1007/978-3-540-78657-3.

Ardia D, Bluteau K, Boudt K, Catania L (2018). "Forecasting Risk with Markov-Switching GARCH Models: A Large-Scale Performance Study." International Journal of Forecasting, 34(4), 733–747. doi:10.1016/j.ijforecast.2018.05.004.

Ardia D, Bluteau K, Boudt K, Catania L, Ghalanos A, Peterson B, Trottier DA (2019a). MSGARCH: Markov-Switching GARCH Models in R. R package version 2.31, URL https://CRAN.R-project.org/package=MSGARCH.

Ardia D, Boudt K, Catania L (2019b). "Generalized Autoregressive Score Models in R: The GAS Package." Journal of Statistical Software, 88(6), 1–28. doi:10.18637/jss.v088.i06.

Ardia D, Hoogerheide LF (2010). "Bayesian Estimation of the GARCH(1, 1) Model with Student-t Innovations." The R Journal, 2(2), 41–47. doi:10.32614/rj-2010-014.

34 MSGARCH: Markov-Switching GARCH Models in R

Ardia D, Hoogerheide LF, Van Dijk HK (2009). "Adaptive Mixture of Student-t Distributions as a Flexible Candidate Distribution for Efficient Simulation: The R Package AdMit." Journal of Statistical Software, 29(3), 1–32. doi:10.18637/jss.v029.i03.

Ardia D, Kolly J, Trottier DA (2017). "The Impact of Parameter and Model Uncertainty on Market Risk Predictions from GARCH-Type Models." Journal of Forecasting, 36(7), 808–823. doi:10.1002/for.2472.

Bauwens L, Backer B, Dufays A (2014). "A Bayesian Method of Change-Point Estimation with Recurrent Regimes: Application to GARCH Models." Journal of Empirical Finance, 29, 207–229. doi:10.1016/j.jempfin.2014.06.008.

reference source.

Ringraziamenti (in ITA)

Ringrazio il Prof. Guizzardi, per avermi guidato nella stesura della tesi.

Ringrazio i miei compagni di corso Amadori Nicolò e Zignani Tommaso per le giornate passate in biblioteca.

Ringrazio Benzi Simone per le dritte su opzioni e i confronti sul value investing.

Ringrazio Tedi Chausheva e Camilla Guardigli per i consigli.

Ringrazio specialmente Soldati Federico per l'aiuto -essenziale- soprattutto col codice usato per la stesura di questa tesi.