SAPIENZA
UNIVERSITÀ DI ROMA

Department of Methods and Models for Economics, Territory and Finance

THE USE OF GARCH MODELS IN VAR ESTIMATION

Prof.ssa
Lea Petrella

Studenti
Luca Petrucci
Martina Silvestri
FrancescaTarchini

ANNO ACCADEMICO 2017-2018

ABSTRACT

In this paper we develop the technique to analyze different extensions of ARCH and GARCH's family models in market risk estimation, measured by taking into account the " Value at Risk ". Then, we develop a list of Markov-switching GARCH model which enables us to choose which model is adapt for each index and, specify complex GARCH equations in two distinct Markov-regimes: high and low volatility. Periods of high volatility are persistent over time and similar periods of relative market quiet. This phenomenon is known as "volatility clustering." To give a simple example, we have estimated 1.032 models for each index with the package RUGARCH. With package MSGARCH we have estimated 72 models with Maximum Likehood estimation and 72 with Bayesian estimation. Our empirical study has two major findings. First, our estimation results said that the parsimoniously models are preferred to the other. Second, we have demonstrated the powerful of Markov-switching GARCH models over traditional GARCH family.

INDEX

Introduction

Over recent years the study of the cash flow of the portfolio is exponentially spreading. This kind of analysis is crucial for a better measurement of equity securities' financial risk.

Our work is focused mostly on the evaluation of the stability of the estimates of ARCH (Engle, 1982) and GARCH (Bollerslev, 1986) models parameters, which aim at better describing volatility clustering phenomenon and the related effects, such as kurtosis. The main idea behind those two models is that the volatility depends on previous performances of the series.

The functional analysis of these parameters estimates was carried out considering the historical series for 5 completely differentiated stock price indices:

- S&P500: is the most representative US stock market index. It includes 500 Blue Chips quoted on the major US stock markets;

-Nikkey: is the official stock exchange index of Japan. It is calculated on the basis of two different baskets of securities listed/quoted mainly on the largest Tokyo stock market;

-DAX30: is the principal German stock exchange index (FWB), it summarizes the trend of the 30 main securities;

-CAC40: is the Paris stock exchange benchmark and it summarizes the trend of the 40 main securities;

- FTSE100: is the main British stock exchange benchmark. Clearly, its principal market (LSE) is the most important amongst the Eurpeans. It is based on the first 100 society in terms of capitalization.

When examining the diversified portfolios daily VaR amongst the 5 indices above, by using a number of assumption on the distribution and on the size of the sample (500, 1000, 1500 and 2000 observations), it results that the distributions that take into account also the skewness parameter provide a more accurate measure of the VaR at high and low confidence levels.

Indeed, in the light of asymmetry effects on the accuracy of VaR estimate, the models that don't take into account this component in the unconditional distribution of returns or in the volatility underestimate the real VaR absolute value.

In the first chapter we will carry out the analysis of the data considering the series of prices and returns on the above stock indices and we in particular will focus on the analysis of the simple linear model.

In the second chapter we will review further models, particularly the ARCH and GARCH models, in which persistence is captured taking into account the second moments.

Following reading of the Paper *"The use of GARCH models in VaR estimation" (Timotheos Angelidisa, Alexandros Benosa, Stavros Degiannakisb)*, we chose to estimate the models that are able to capture the financial markets most relevant characteristics, looking at the better one in the light of theirs predictive and statistic accuracy in terms of risk management through 3 backtesting criteria:

- Unconditional coverage test of Kupiek $LRuc$
- Conditional coverage test of Christoffersen $LRcc$
- Dynamic quantile test DQ

Finally, once estimated in the third chapter traditional GARCH models, we will carry out a comparison between the latter and Markov-Switching GARCH models in order to figure out which one between the two is able to better fit the series used.

Markov-Switching models therefore add to the ARCH and GARCH models a latent dynamic (not observed) component, which takes into account the potential scheme (expansion or contraction) that occur in the economic system; a traditional model couldn't indeed capture these shifts.

We will be discussing these type of model in the last chapter of these paper

Chapter 1: Analysis of the variable

1.1 Stochastic processes and data time series

In the context of time series, the data collected at the instant t is closer to the one collected at the preceding instant because of the natural tendency of various phenomena to evolve in a homogeneous way during time. Because of this characteristic we can say that the analyzed time series has "memory of itself".

It is called "time series" one of the multiple finite potential realizations of a stochastic process; a "stochastic process" is an orderly sequence of random variables dependent on time t .

Time series analyzed in the Paper and downloaded by Bloomberg relate to 5 financial indices occurring from 09/07/1987 to 18/10/2002:

- Standard and Poor's (S&P 500), with the "price.SPX" variable related
- NIKKEI 225, with the "price.NKY" variable related
- Deutsche Akitienindex (DAX 30), with the "price.DAX" variable related
- Cotation Assistée en Continu (CAC 40), with the "price.CAC" variable related
- Financial Times Stock Exchange (FTSE 100), with the "price.UKX" variable related

For every index, by simultaneously using the `"na.omit()"` command to remove any NA values, we have defined the prices vector as follow:

```
###   Prezzo   ###
price.SPX <- na.omit(SPX.Index)
price.NKY <- na.omit(NKY.Index)
price.DAX <- na.omit(DAX.Index)
price.CAC <- na.omit(CAC.Index)
price.UKX <- na.omit(UKX.Index)
```

A stochastic process, whose t-th component is indicated by x_t, is characterized by:

- Density function f($x_t; t$)
- Marginal density functions for every process realizations (x_t), and for every couple of elements (x_t, x_{t-1}) and so on;

- Moments such as:

  1. the expected value E (X_t) = μ which is steady for every t: it doesn't change with time;

  2. the variance Var (X_t) = σ^2 = (0) < ∞ steady for every t;

  3. the covariance Cov (x_t, x_{t+h}) = γ(h) at lag h, which depends on laps of time h, not on t;

- Conditional density functions

Nevertheless, there are two problems arising from the inference procedure: on one hand we can't accurately say that observed series characteristics are specific of that single realisation; on the other hand, even by using a single realisation it is necessary that the process is stable in covariance, namely that it is stationary.

We talk about a stationary stochastic process with two different meaning: strong stationariness and weak stationariness. For the purpose of this work we will consider the weak stationarity which concerns a 2 lag size window.

As a general rule, this requirement on stationarity is necessary to carry out the inference analysis.

Prices can be indexed in the moment they're observed (P_t) and they're interpretable as capitalisation regarding the price observed in the previous moment (P_{t-1}). Clearly, prices are closely related to returns. Indeed, prices are defined by daily securities prices as follow:

$$Returns\ r_t = log\ (P_t) - log\ (P_{t-1}) = log\ (P_t\ /P_{t-1}) \quad (1)$$

So, starting from prices time series, we have calculated the returns as logarithmic prime differences of prices themself: in so doing we've stabilized the examined stochastic process cutting out the not stationarity on average, variance, and covariance.

We used the "`diff()`" command on R to calculate every index returns vector

```
###   Log-Returns   ###
returns.SPX <- diff(log(price.SPX), lag=1)
returns.NKY <- diff(log(price.NKY), lag=1)
returns.DAX <- diff(log(price.DAX), lag=1)
returns.CAC <- diff(log(price.CAC), lag=1)
returns.UKX <- diff(log(price.UKX), lag=1)
```

We are hereby studying a stochastic process in every random variable r_t , each one referred to a different moment. The "lag=1" setting represents a one period temporal "switching" e.g. P_{t-1} (see equation 1).

We can graphically compare price series evolution with return series. It appears that the procedure seen before has rendered the stationary process in average (not stationary in variance).

Throughout this work is often used the "`function()`" command which enables to develop complex mathematical operation with the possibility to retrieve it when needed. The used syntax shall be as follow:

```
name <- function(variables) {

  # corpo_della_funzione

}
```

In which "`name`" shall indicate the function name; the n input variables selected shall be entered in field "`variables`"; in curly brackets ther's the function , i.e. the codex that will be repeated each time the function will be recalled.

As an example, we here present the function build to plot the return series. See Appendix A to further insights and examples. The used syntax shall be as follow:

```
###   Funzione per i grafici Rendimenti  ###
grafico.R <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  ggplot() +
    coord_cartesian(xlim = c(150, 3720), ylim = NULL, expand = TRUE)+

    geom_line(mapping = aes(x = seq(1:length(returns.SPX)), y = returns.SPX),
              color = "black", size=1)+

    geom_hline(yintercept = mean(returns.SPX), col = 'green', size = 1)+

    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Fonte: Bloomberg",
         y = paste(asix.x),
         x = "Osservazioni"
    )
}
```

"grafico.R" shall be the name of the function, and the chosen input variables are: "x.ts" for returns time series; "tkr.titolo", "subtitle", "asix.x" are strings of characters which respectively indicate the title, subtitle and name for horizontal axis. The body of the function shall contain "ggplot()" instruction of the namesake library . This instruction makes it possible to create the graphic surface.

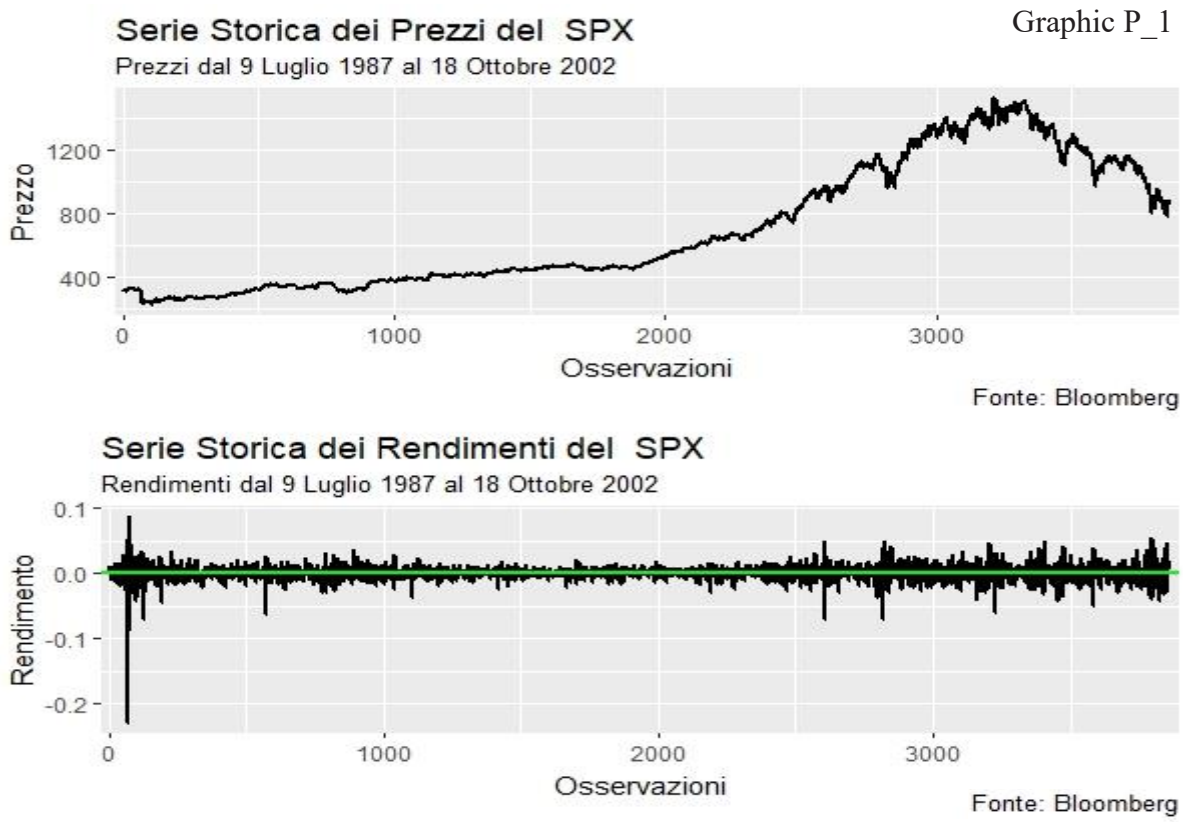
Furthermore, we have the objects: "coord_cartesian" allows us to set the size of the graph, meanwhile "geom_line" and "geom_hline" enable to build respectively returns time series and the green line representing the average of the process along the timeframe concerned. The "labs" object includes all the settings with regard to the title, subtitle and axes.

The use of the function just described is showed in another "TwO_Plot" function. The latter uses "grafico.P" and "grafico.R" to build prices and returns graphs to simultaneously plot them in a single image.

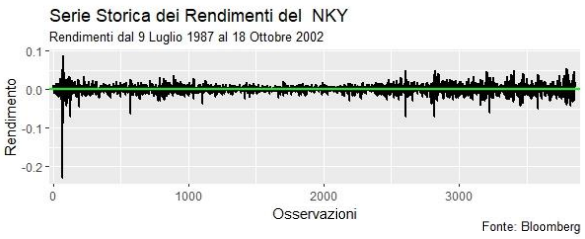
```
###   Funzione per entrambi i grafici: Prezzi & Rendimenti  ###
TwO_Plot <- function(x.price, x.return, tkr) {

  G1 <- grafico.P(x.price,
                  paste('Serie Storica dei Prezzi del ', tkr),
                  'Prezzi dal 9 Luglio 1987 al 18 Ottobre 2002', 'Prezzo')
  G2 <- grafico.R(x.return,
                  paste('Serie Storica dei Rendimenti del ', tkr),
                  'Rendimenti dal 9 Luglio 1987 al 18 Ottobre 2002', 'Rendimento')
  grid.arrange(G1, G2, nrow=2, ncol=1)

}
```

A visible example of those script are P_1 … P_5 graphs.

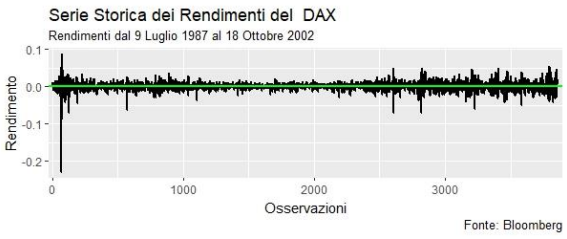
Graphic P_1



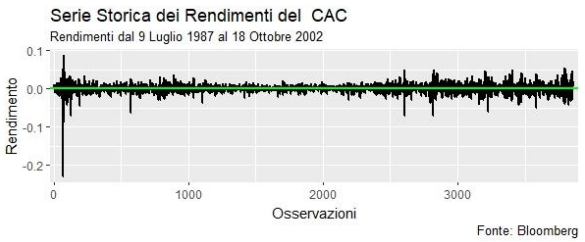
Graphic P_2



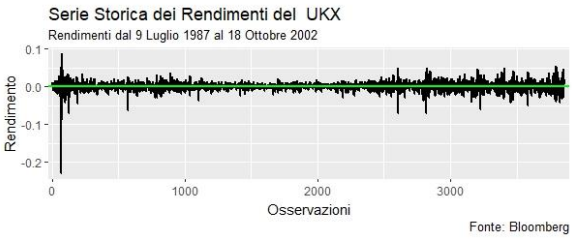
Graphic P_3



Graphic P_4



Graphic P_5



As can be seen from prices time series graphs, the process doesn't fluctuate around a constant value: process conditional average varies during time and autocorrelation function, as we will see, will have a very long memory: clearly the process isn't stationary.

Unlike prices, by analysing indices returns along the period taken into account, graphs show that the series result stationary on average, i.e. the observed values range around a number that is the series average: this evidence shall also be supported by some unit root tests (Dickey-Fuller); furthermore, the series variability isn't steady.

1.2 Dikey-Fuller test for stationarity

To verify that returns series really is stationary, we can use the *Dikey-Fuller test* or "unit root test", which tests the not stationarity in covariance.

```
> Dikey_Fuller(price.SPX, returns.SPX)
           Statistica           P-Value              Stazionaria
Prezzi     "-0.924808638241051" "0.950493638276216" "NO"
Rendimenti "-15.0828688388246"  "0.01"              "SI"

> Dikey_Fuller(price.NKY, returns.NKY)
           Statistica           P-Value              Stazionaria
Prezzi     "-2.75024350431515"  "0.260660305836609" "NO"
Rendimenti "-14.8427933619198"  "0.01"              "SI"

> Dikey_Fuller(price.DAX, returns.DAX)
           Statistica           P-Value              Stazionaria
Prezzi     "-1.1418488627827"   "0.915885051309971" "NO"
Rendimenti "-13.9368625193974"  "0.01"              "SI"

> Dikey_Fuller(price.DAX, returns.DAX)
           Statistica           P-Value              Stazionaria
Prezzi     "-1.1418488627827"   "0.915885051309971" "NO"
Rendimenti "-13.9368625193974"  "0.01"              "SI"

> Dikey_Fuller(price.UKX, returns.UKX)
           Statistica           P-Value              Stazionaria
Prezzi     "-1.2494149212084"   "0.896156538037062" "NO"
Rendimenti "-14.4387802570417"  "0.01"              "SI"
```

From the stationarity test arises that the prices first order differences, that are used to calculate the returns, make the series stationary for each index.

However, as shown before, returns variability (and so the variance) doesn't result constant in time.

Returns volatility is a phenomenon with a long memory; it frequently happens that the stochastic process that rules isn't stationary in covariance because it presents unit roots in its lags this means that every value of the conditional variance is related with its absolute value in the previous moments of observation.

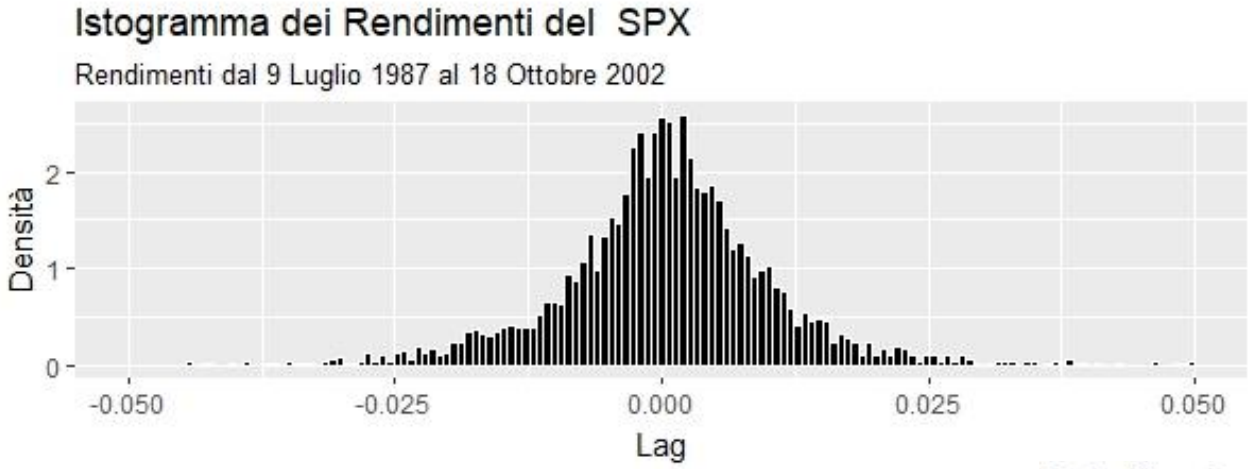
Returns variance changes due to the change in the intensity with which trading activity develops on financial markets: this is explained by the fact that prices have the tendency to change more often in certain days rather than others, resulting in what in the literature is called volatility clustering.

We can then make reference to a graphic representation using returns frequency distribution histogram which gives us a first information on distribution shape and a summary of some data characteristics.

For series volatility we consider returns absolute value. We simultaneously plotted the two graphs, by using the already seen "`TwO_Plot`" function to create a more accurate comparison.

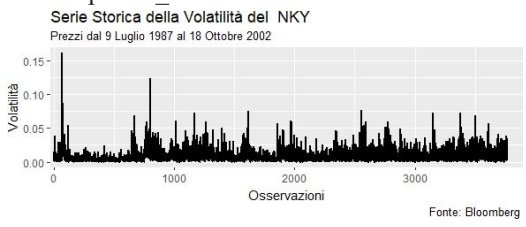
Graphic P_6

Fonte: Bloomberg

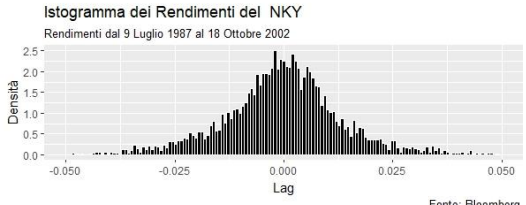


Fonte: Bloomberg

Graphic P_7

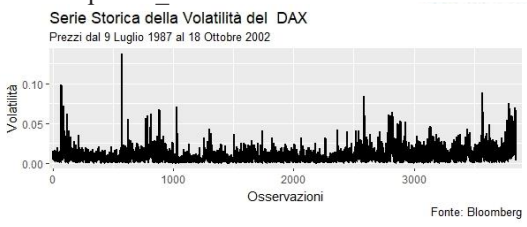


Fonte: Bloomberg

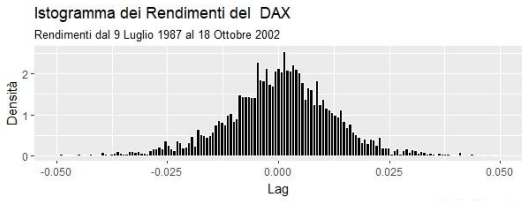


Fonte: Bloomberg

Graphic P_8

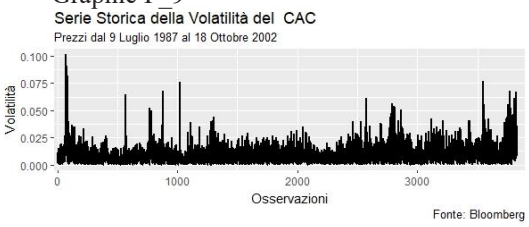


Fonte: Bloomberg

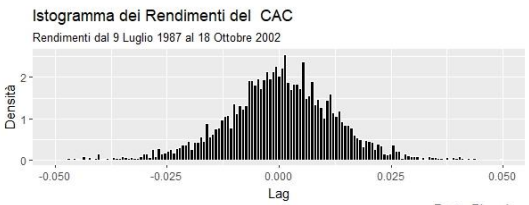


Fonte: Bloomberg

Graphic P_9

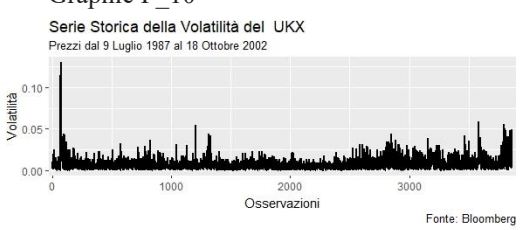


Fonte: Bloomberg

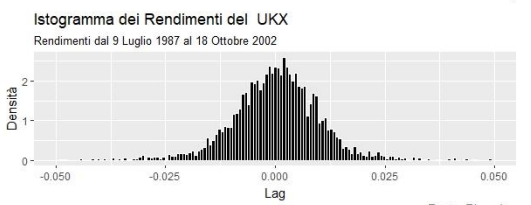


Fonte: Bloomberg

Graphic P_10



Fonte: Bloomberg



Fonte: Bloomberg

To calculate the moments for each one of the examined processes, we have built the "moments" function. This function is a list of values that is triggered with the last command "return(statistiche)".

```
###    Funzione Calcolo Statistiche   ###
momenti <- function(x.ts) {
  media <- mean(x.ts, na.rm = TRUE)
  mediana <- median(x.ts, na.rm = TRUE)
  massimo <- max(x.ts, na.rm = TRUE)
  minimo <- min(x.ts, na.rm = TRUE)
  varianza <- var(x.ts, na.rm = TRUE)
  dev_std <- sqrt(varianza)
  asimmetria <- skewness(x.ts)
  curtosi <- kurtosis(x.ts, na.rm = FALSE, method = c("excess"))
  test.jb <- jarque.bera.test(x.ts)
  statistiche <- list(Media = media,
                      Mediana = mediana,
                      Massimo = massimo,
                      Minimo = minimo,
                      Varianza = varianza,
                      Std_Dev = dev_std,
                      Asimmetria = asimmetria,
                      Curtosi = curtosi,
                      JB.Stat = test.jb$statistic[[1]],
                      JB.Pvalue = test.jb$p.value[[1]])
  return(statistiche)
}
```

The function has been applied to every stock market index that we've considered. Results have been ordered in a matrix named "df0",whose output is as shown below[1]:

Table 1: Descriptive statistics of daily log returns, for the period of 9 July 1987 to 18 October 2002

	stat.SPX	stat.NKY	stat.DAX	stat.CAC	stat.UKX
Media	0.0002738807	-0.0002550642	0.0002099406	0.0001974461	0.0001436932
Mediana	0.0004162731	-0.0001135587	0.0007774265	0.0003469953	0.0004545561
Massimo	0.08708879	0.1243033	0.07552676	0.08225436	0.07596966
Minimo	-0.2289972	-0.1613542	-0.1370612	-0.1013757	-0.130286
Varianza	0.0001299643	0.0002232236	0.0002101827	0.0001896902	0.0001190473
Std.Dev	0.01140019	0.01494067	0.01449768	0.01377281	0.01091088
Asimmetria	-2.296291	-0.05996877	-0.5311798	-0.3233085	-0.8804831
Curtosi	46.73206	7.107121	6.449065	4.384675	11.39993
JB.Stat	354748.5	7944.494	6854.253	3137.771	21444.79
JB.Pvalue	0	0	0	0	0

It can be seen that the data that this work estimates replicate those obtained in the "Table 2" of the Paper.

The table, together with skewness calculus, kurtosis[2] and Jarque-Bera (JB) test, shows the principal descriptive statistics of returns for S&P500, NIKKEI225, DAX30, CAC40 and FTSE100 indices.

As can be seen in the table, we are dealing here with a strong negative asymmetry; moreover kurtosis index value is much higher than 3, to show that even the test of normality leads to largely unsatisfactory results;
In all cases infact, the null hypothesis of normality is rejected at every significance level (as can be seen from the Jarque-Brera test which we will consider afterwards). This fact is symptom of a kurtosis excess and of negative asymmetry in every market.

[1] The following table, as well as the subsequent ones in the paper, were created with "TexStudio". The relevant codes are shown in the final appendix

1.3 Residue Analysis

After making stationary the series, we can pass on a first analysis of our processes. The easiest method is to create a linear model as follow:

$$Y_t = T_t + \varepsilon_t \tag{2}$$

This formulation presupposes that is ε_t IID and that it distributes as a normal N(0,1): hence, errors are homoscedastic (with constant variance) and independent between them (without serial autocorrelation). These are the hypothesis behind the model and they need to be verified with appropriate statistic tests. The blurring of these hypothesis could undermine the validity of the model used and favour the use of other more complex models.

To proceed to residues analysis, it is firstly necessary to calculate them. We have defined a function that allows to extract the residues through "`lm()`" command, Fitting Linear Models.

```
###   Scarti dalla media (Errori)     ###
residui <- function(x.ts) {
  model <- lm(x.ts~c(1:length(x.ts)))
  res <- model$residuals
  return(res)
}
```

It is preferable to work on standardized residues to deal with raw numbers. We use the following function:

$$Z = (Y_t - m)/\,\text{dev.st} \tag{3}$$

For this purpose we wrote a function that standardize data vector in input:

```
###   Scarti dalla media (Errori) Standardizzati   ###
residui.stndz <- function(x.ts) {
  m <- mean(x.ts)
  s <- (var(x.ts)^0.5)
  z <- (x.ts - m)/s
  return(z)
}
```

[2] The kurtosis index measures the "thikness" of the tail of a distribution: a distribution with heavier tails than those of the normal distribution is called Leptokurtic (Kurtosis Normal=3), i.e. if that index is > 3. The leptokurtic curve is very concentrated around its average.

1.4 Autocorrelation functions and the use of the correlogram

Because of observed values inertia or stability, it's possible to have a temporal autocorrelation phenomenon so that every value is influenced by the preceding one and shall significantly determine the following one.

A quite simple way to check if there's autocorrelation in a series is that to build the correlogram with "`acf()`" function. If there's not autocorrelation, the asymptotic distribution of the autocorrelation coefficients estimate is normal and we will have a stripe of confidence such as:

$$\left[{}^{-Z_{1-\alpha/2}}\Big/_{\sqrt{n}} \;\; ; \;\; {}^{Z_{1-\alpha/2}}\Big/_{\sqrt{n}} \right] \tag{4}$$

where external values of this stripe indicate the presence of significant autocorrelation.

The autocorrelation function (ACF) measures the linear dependency of a process "value" with the preceding (al lag h), and gives information about the size and duration of process memory:

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = Corr(X_{t+h}, X_t) \tag{5}$$

where $\gamma(h) = Cov(X_{t+h}, X_t)$ represents the autocovariance function.

As will be seen from the graphs, this function is even and so is only plotted for positive lag. In our case, since we have a stationary process, the ACF approaches zero when h increases (otherwise the process would tend to explode).

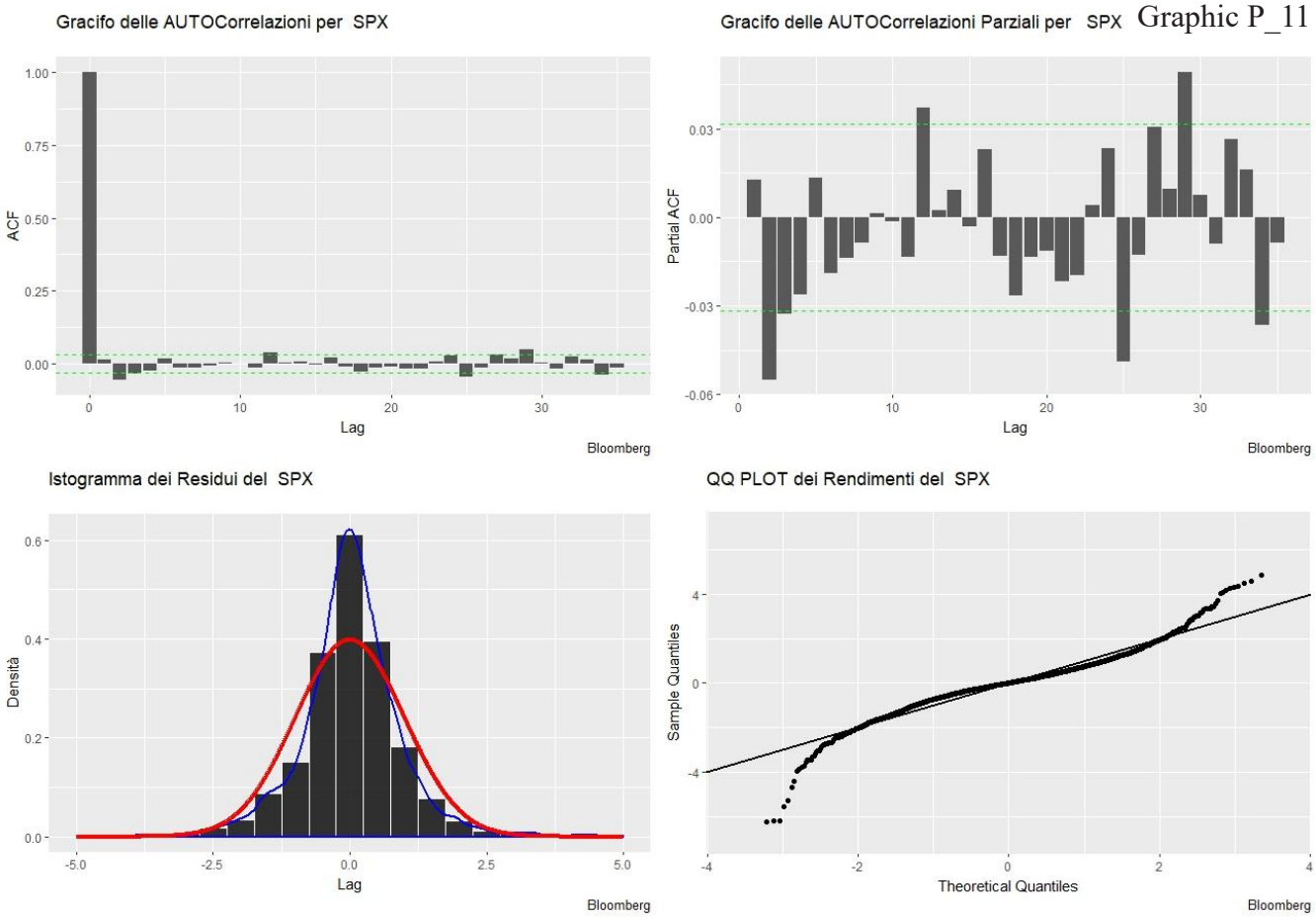
The partial autocorrelation function (PACF), instead, measures the autocorrelation between X_t and X_{t+h} after having eliminated the linear dependence of these two variables from the interim ones.

As an example, we reported the function that we've defined in order to build the residues histogram. (To see the other graphic functions used in P_11-P_15 graphs refer to appendix A.)

The "`grafico_hist_rl`" function is defined in the same way as "`grafico.R`" function. The main difference consists in the objects uploaded on the plot: "`geom_histogram`", "`geom_density`" and "`geom_line`": those instructions allow to build the errors histogram, the blu curve for errors density and lastly the red curve that measures the density of a N(0,1).

```r
###    Funzione per istogramma dei residui ###
grafico_hist_rl <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  #Parametri per la normale
  x <- seq(min(x.ts), max(x.ts), length = length(x.ts))
  y <- dnorm(x, mean = 0, sd = 1)
  #Costruzione grafico
  ggplot() +

    geom_histogram(mapping = aes(x = x.ts, y=..density..),
                   alpha=0.8,
                   binwidth = 0.5,
                   fill='black',
                   colour='white',
                   stat = "bin",
                   position = "stack")+

    geom_density(mapping=aes(x.ts),
                 col = 'blue',
                 size = 1,
                 stat = "density")+

    geom_line(mapping = aes(x=x, y=y),
              col = 'red',
              size = 1.5)+

    scale_x_continuous(limits = c(-5, 5))+

    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Bloomberg",
         y = paste(asix.x),
         x = "Lag")
}
```

A quite easy and intuitive way to verify the normality of errors distribution is to call upon the assistence of a graph with a histogram and with a QQ-plot. After, indeed, we plot the graphs of ACF, PACF, histogram and QQ Plot of standardized residues related to each stock index considered.



Graphic P_11

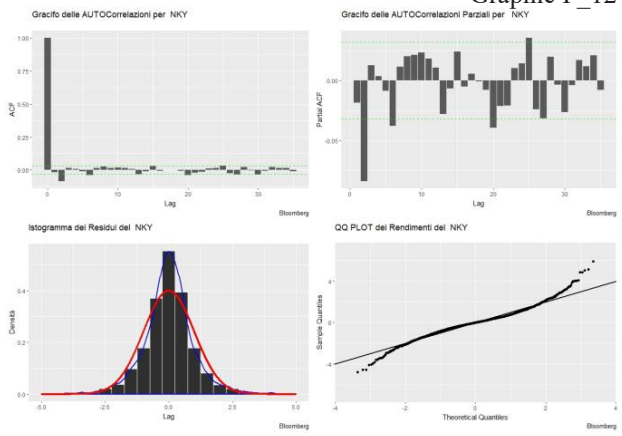
In the correlograms of standardized residues, the green horizontal dotted lines, above and below the zero, indicates the 95% level confidence stripes inside which the correlation are statistically not different from zero.

When the temporal lag changes, autocorrelation coefficients of residues turn out to be partially intern and partially external to the confidence stripe, showing the presence or absence of autocorrelation. In particular:

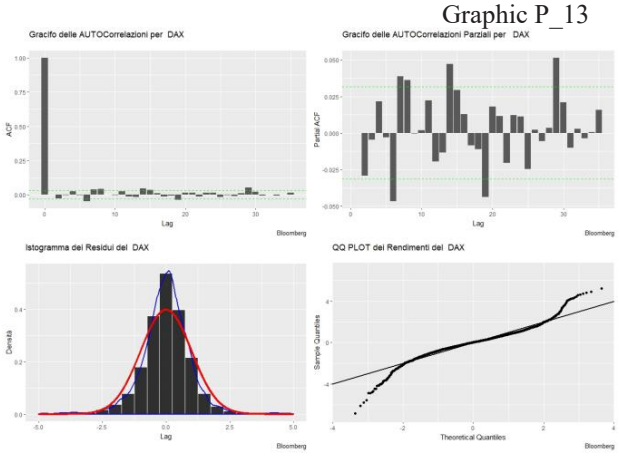
- → the autocorrelation function of the SPK index's series approaches to zero with a monotonic trend until the annulment of itself. This is symptom of a stationary process of the type "white noise", and so the process hasn't memory;
- → taking into account the PACF graph, we see a behavior at intervals, in positive and negative autocorrelation: this is due to the fact that the processes considered net of interim variables;
- → the histogram built on residues density provides us with an insight on a distribution with leptokurtic tails and so it isn't possible to approximate to a normal distribution of residues;
- → the Q-Q plot further highlights a huge difference of empiric distribution's tails compared to Gaussian distribution of residues (on the main diagonal) and the quantiles of errors' theoretical distribution.

Similar observations can be applied for the following graphs with some variations amongst each cases discussed

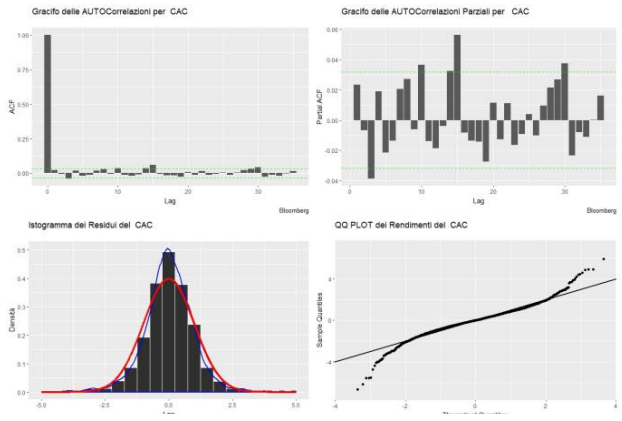
Graphic P_12



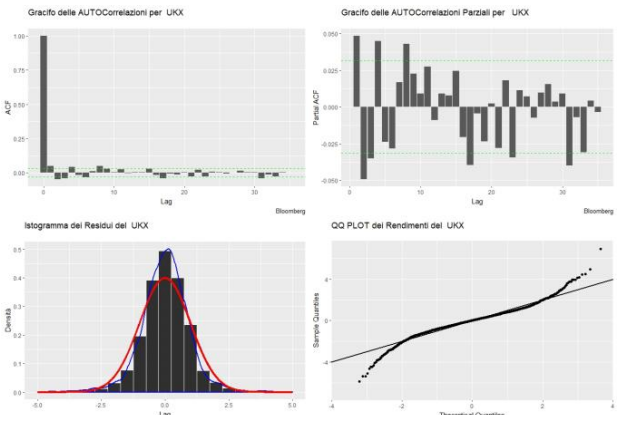
Graphic P_13



Graphic P_14



Graphic P_15



In our case the assumptions aren't met since returns' distribution isn't normal but it is leptokurtic, i.e. it is sharper than the normal and with heavier tails: so, to have a more statistically reliable result it is necessary to conduct normality tests.

1.5: Statistical tests

In our work, we have taken into account:

- Jarque-Bera test to verify the normality assumption;
- Breusch-Pagan test to verify the homoscedasticity assumption;

On a financial level, normality assumption is rare to find, but it is easy to imagine how the normal distribution simplifies calculations and enables more easily to obtain certain results.

Jarque and Bera have built a test based on sample skewness and kurtosis to understand weather or not the distribution is normal. Indeed, a normal distribution is characterized by a skewness statistically not different from zero and from a kurtosis index equal to three.

Nevertheless, in reality the majority of the distributions (for example the returns' distribution) is characterized by different from zero skewness and by greater than three kurtosis.

Jarque-Bera test verify the normality assumption.

Starting from simple regression like:

$$Y = \beta_1 X_{t,1} + \beta_2 X_{t,2} + \ldots + \beta_k X_{t,k} \quad (6)$$

The test is used to verify the following system of assumption on residues:

$H_0: \rho_1 = 0$ the series has a normal marginal distribution

$H_1: \rho_1 > 0$ the series has a marginal distribution different from the normal

If the p-value < α, H0 is refused; if the p-value > α, we do not refuse the normality assumption of residues distribution.

The statistic is calculated as follows

$$JB = \frac{T-1}{6}\left[\widehat{SK}^2 + \frac{1}{4}(k-3)^2\right] \underset{H_0}{\overset{\alpha}{\approx}} \chi_2^2 \quad (7)$$

It is a normality test that simultaneously verifies if skewness and kurtosis are consistent with the values that they should assume under the null hypothesis, i.e. 0 and 3: under the null hypothesis the test is distributed as a chi-square with 2 degrees of freedom, meaning that the JB statistic is asymptotically distributed as a random variable chi-square with two degrees of freedom; it can be used to test the null hypothesis that the sample has been extracted from a data population distributed as a normal random variable.

In R this test is in "`tseries`" set and is activated through the "`jarque.bera.test()`" command that gives back the value of the statistic, the degrees of freedom and the p-value.

The Breusch-Pagan test is used to verify if the variance estimated on regression models' residues depends from one of the independent variables.

Starting from a previous regression (view equation 6), the test is used to verify the following hypothesis system on residues:

$H_0: \rho_1 = 0$ Homoscedasticity

$H_1: \rho_1 > 0$ Heteroscedasticity

The test's null hypothesis is the presence of homoscedasticity, while the alternative hypothesis is the presence of heteroscedasticity.

Therefore, the test verify if the esteemed variance of sample's residues depends on the values of the covariance and it is as follows:

$$H_0: \sigma_1^2 = \cdots = \sigma_n^2$$
$$H_1: \sigma_i^2 = \sigma^2 h(z_i' \theta)$$

In which $h(z_i' \theta)$ is some sort of function of X that explains the presence of heteroscedasticity by adopting a linear specification of the function.
The statistic test is:

$$BP = \frac{\frac{R_i^2}{(k-1)}}{\frac{(1-R_i^2)}{(n-k)}} \approx F_{k-1,n-k} \tag{8}$$

In order to make the test of Breusch-Pagan in R it is necessary to load the `"lmtest"` set where the function `"bptest()"` is.

Concerning the returns' serial autocorrelation, instead, there are several statistic tests. They let you verify the assumption of presence or absence of correlation between standardized residues.
Here we will refer to Box-Pierce, Ljung-Box and Durbin-Watson tests.

The Durbin-Watson test is used to verify the hypothesis of absence of serial autocorrelation between the regression's residues.
Starting from the equation 6, the test is used to verify the residues and the following system of assumptions

$H_0: \rho_1 = 0$ Absence of autocorrelation

$H_1: \rho_1 > 0$ Presence of autocorrelation

The statistic is then calculated:

$$DW = \frac{\sum_n^{i=1}(e_i - e_{i-1})^2}{\sum_n^{i=1} e_i^2} \tag{9}$$

This test has an asymptotic distribution and must be compared to specific Tables of significance. (Both the test and the p-value are calculated with the software Eviews).

In R we use the command `"dwtest()"` that is in the package `"lmtest".`

The Ljung-Box test is usually used to verify whether the correlation between the standardized residues is present or not.
The hypothesis are the following:

$H_0: \rho_1 = \rho_2 = \cdots = \rho_k = 0$ Absence of autocorrelation

H_1: there is at least a $\rho_i \neq 0$, with i = 1, 2, . . . , q,

where $\rho_i = (1,2, \ldots, k)$, represents the autocorrelation's grade between the observations.

The statistic, which is a proper linear combination of residues autocorrelation's coefficients r(t) with k fixed whole number, is calculated as:

$$LB = n(n+2)\sum_{t=1}^{k}\frac{r^2(t)}{n-t} \quad (10)$$

If p-value < α, H0 is refused; on the other hand if p-value > α, the null hypothesis is accepted.

The Box-Pierce is similar to the Ljung-Box:

$$BP = n\sum_{t=1}^{k} r^2(t) \quad (11)$$

These two statistics just use a different weighting system but asymptotically they have the same distribution.

In R we use the command "`Box.test()`" for both the statistics. In order to verify that there is not a serial correlation among our series of standardized residues, we first use the Ljung-Box test and then the Box-Pierce test.

To calculate the return's moments with the same logic shown earlier, we have built a function that gives us a vector, which includes the statistics and the p-value of all the tests we need to make for the errors.

```
###   Test rugli errori   ###
test_errori <- function(x.ts) {
  media <- mean(x.ts, na.rm = TRUE)
  varianza <- var(x.ts, na.rm = TRUE)
  dev_std <- sqrt(varianza)
  asimmetria <- skewness(x.ts)
  curtosi <- kurtosis(x.ts, na.rm = FALSE, method = c("excess"))
  #Per la normalità
  test.jb <- jarque.bera.test(x.ts)
  #per verificare l'omoschedasticità
  X <- 1:length(x.ts)
  Y <- as.numeric(x.ts)
  test.bp <- bptest(Y~X)
  ### Verifica di assenza di autocorrelazione seriale
  # Box-Ljung
  test.BL <- Box.test(x.ts, lag = 7, type = "Ljung-Box")
  # Box-Pierce
  test.BP <- Box.test(x.ts, lag = 7, type = "Box-Pierce")
  # Durbin-Watson
  test.dw <- dwtest(x.ts~1, alternative = 'two.sided')

  statistiche <- list(Media = media,
                      Varianza = varianza,
                      Sigma = dev_std,
                      Asimmetria = asimmetria,
                      Curtosi = curtosi,
                      JB.Stat = test.jb$statistic[[1]],
                      JB.Pvalue = test.jb$p.value[[1]],
                      BP.Stat = test.bp$statistic[[1]],
                      BP.Pvalue = test.bp$p.value[[1]],
                      LjBOX.Pvalue = test.BL$p.value[[1]],
                      PiBOX.Pvalue = test.BP$p.value[[1]],
                      DW.Stat =  test.dw$statistic[[1]],
                      DW.Pvalue = test.dw$p.value[[1]])
  return(statistiche)
}
```

Now we just need to use the new function to define the "`test.err`" vectors.

```
####################################
##    Test Sugli Errori       #######
####################################
test.err.SPX <- test_errori(residual.std.SPX)
test.err.NKY <- test_errori(residual.std.NKY)
test.err.DAX <- test_errori(residual.std.DAX)
test.err.CAC <- test_errori(residual.std.CAC)
test.err.UKX <- test_errori(residual.std.UKX)
```

By using the command "`cbind()`" we put together the five vectors and create e matrix that we call "`Tab.Res`", shown here:

Table 2: Descriptive statistics and p-value of standardized residues for each index

	test.err.SPX	test.err.NKY	test.err.DAX	test.err.CAC	test.err.UKX
Media	6.863798e-18	-4.438582e-18	-1.038849e-17	1.479291e-17	6.216579e-18
Varianza	1	1	1	1	1
Sigma	1	1	1	1	1
Asimmetria	-2.295237	-0.0556457	-0.5212031	-0.3176482	-0.8758036
Curtosi	46.90119	7.13831	6.46925	4.396002	11.43864
JB.Stat	357293.1	8014.026	6889.349	3151.337	21581.95
JB.Pvalue	0	0	0	0	0
BP.Stat	0.1575567	2.986232	30.40373	27.44081	10.00451
BP.Pvalue	0.691416	0.08397535	3.508523e-08	1.619769e-07	0.00156157
LjBOX.Pvalue	0.003150995	8.736402e-06	0.005862285	0.08022054	6.199032e-06
PiBOX.Pvalue	0.003184783	8.908966e-06	0.005949899	0.0807181	6.32517e-06
DW.Stat	1.974808	2.036204	2.000088	1.953423	1.903096
DW.Pvalue	0.4339276	0.2663638	0.9978147	0.1495614	0.002593341

As we can see from Tab 2 the standardized residues of the processes estimated with simple linear model, are distributed with a null average and a constant variance equal to 1.

About the results of the tests, assuming a level of confidence of 95% (α=0,05) we can observe:

- with the JB test, for all five equity indices, value of the p-value is less than the threshold level and therefore the null hypothesis of normality of the residues is rejected;
- with the BP test, only two of the five analyzed indices does not reject the null hypothesis of homoscedasticity (SPX, NKX);
- the three tests used to verify the presence or absence of serial autocorrelation between residues, report discordant results for all 5 indices, with the exception of the index UKX, for which in all three test the null hypothesis of absence of autocorrelation is rejected.

Following the analytical and graphic analysis, we can say that the assumptions underlying the simple linear model no longer apply; therefore, it is necessary to proceed with the analysis using additional models, in particular, the rejection of normality extends to the use of different distributions from the normal.

The presence of autocorrelation implies that the process has short-term persistence, as evidenced in the ACF graphics.
In these cases it is necessary to ensure the test distribution is not affected by short-term memory: one of the most common ways to achieve this goal is to assume that the process memory characteristics can be well approximated by an AR(p) choosing a large value of p.
Any short-term persistence, modeled on the first moments, is completely captured from the auto-regressive part.
However, there are deserving persistence aspects of care that do not concern the first stages of the process but his other distributional characteristics.
In the course of this paper, we will discuss processes called ARCH and GARCH, where persistence is felt through the second moments.
These models have properties that quite well replicate some features commonly encountered in financial time series.

Appendix A

In this appendix is reported the complete script, which will show step by step the procedure used to obtain the results of tables and graphs shown in the chapter just ended.

This part of code is only concerned with the import libraries used; among all the loaded libraries, the most important are "Tseries", "ggplot2" and "gridExtra": the first for the manipulation of the time series and the test to be performed on the same, the next two for the construction of all graphics.

```
######################################################################
######################################################################
####                                                              ####
#####        ANALISI DELLE SERIE STORICHE PROGETTO CON R         #####
####                                                              ####
######################################################################
######################################################################

########################
### Ripulire Ambiente ###
########################
rm(list=ls())
getwd()
setwd('C:/Users/Luca/Documents/Università/Magistrale/Serie Storiche/Gara/Lavoro')

########################
###    Librerie      ###
########################
library(PerformanceAnalytics)
library(tseries)
library(foreign)
library(ggplot2)
ggplot2::ggplot()
library(lmtest)
library(gridExtra)
###   Importo i dati   ###
df1 <- read.csv('DATASET2.csv', header = TRUE, sep = ",", quote = "\"")
attach(df1)
```

We define the vectors of prices and check the presence of any "Not Available" or "NA". Then, with the "diff" command, we calculate the logarithmic first differences for each stock index defining simultaneously the carrier "returns._" , namely the vector of returns.

Right after are defined airlines from absolute returns "returns ._. abs".

```
########################
#  Definisco i vettori  #
########################
###   Prezzo   ###
price.SPX <- na.omit(SPX.Index)
price.NKY <- na.omit(NKY.Index)
price.DAX <- na.omit(DAX.Index)
price.CAC <- na.omit(CAC.Index)
price.UKX <- na.omit(UKX.Index)

###   Log-Returns   ###
returns.SPX <- diff(log(price.SPX), lag=1)
returns.NKY <- diff(log(price.NKY), lag=1)
returns.DAX <- diff(log(price.DAX), lag=1)
returns.CAC <- diff(log(price.CAC), lag=1)
returns.UKX <- diff(log(price.UKX), lag=1)

###   Log-Returns ABS   ###
returns.SPX.abs <- abs(returns.SPX)
returns.NKY.abs <- abs(returns.NKY)
returns.DAX.abs <- abs(returns.DAX)
returns.CAC.abs <- abs(returns.CAC)
returns.UKX.abs <- abs(returns.UKX)
```

To show the dikey-Fuller test on the stationarity of the process we have built the following function which allows to obtain as output the value of the statistic, the p-value and a string with reference to stationarity.

```r
#####################################################
###   Funzione per verificare la stazionarietà    |###
###              Test di Dikey-Fuller               ###
#####################################################
Dikey_Fuller <- function(x.price, x.return) {

  # Creo la matrice e inserisco i nomi colonne
  Dikey.Fuller <- matrix(NA, 2, 3)

  # Calcolo gli oggetti test
  test.p <- adf.test(x.price)
  test.r <- adf.test(x.return)

  # Estraggo la statistica
  stat.p <- test.p$statistic
  stat.r <- test.r$statistic
  stat <- c(stat.p,  stat.r)

  # Estraggo il P-Value
  pval.p <- test.p$p.value
  pval.r <- test.r$p.value
  pval <- c(pval.p,  pval.r)

  # Verifico la stazionarietà con il test Dikey Fuller
  # Prezzo
  staz.p <- if (abs(pval.p) < 0.05) {
    paste('SI')
  }else{
    paste('NO')
  }
  # Rendimento
  staz.r <- if (abs(pval.r) < 0.05) {
    paste('SI')
  }else{
    paste('NO')
  }
  staz <- c(staz.p,  staz.r)

  Dikey.Fuller <- cbind(stat, pval, staz)
  colnames(Dikey.Fuller) <- c('Statistica', 'P-Value', 'Stazionaria')
  rownames(Dikey.Fuller) <- c('Prezzi', 'Rendimenti')
  return(Dikey.Fuller)
}

#Dikey_Fuller
Dikey_Fuller(price.SPX, returns.SPX)
Dikey_Fuller(price.NKY, returns.NKY)
Dikey_Fuller(price.DAX, returns.DAX)
Dikey_Fuller(price.CAC, returns.CAC)
Dikey_Fuller(price.UKX, returns.UKX)
```

Below we show the part of the script where all the functions necessary for the construction of the graphs have been defined. The functions are similar to those already explained above: for this reason, will not be difficult for a careful reader to read following codes.

The following shows the function used for plotting returns and the series of stock prices:

```
###################################
###  Costruzione delle Funzioni  ###
###################################

###   Funzione per i grafici Prezzi  ###
grafico.P <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  ggplot() +
    coord_cartesian(xlim = c(150, 3720), ylim = NULL, expand = TRUE)+

    geom_line(mapping = aes(x = c(1:length(x.ts)), y = x.ts),
              color = "black", size=1) +

    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Fonte: Bloomberg",
         y = paste(asix.x),
         x = "Osservazioni"
    )
}

###   Funzione per i grafici Rendimenti  ###
grafico.R <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  ggplot() +
    coord_cartesian(xlim = c(150, 3720), ylim = NULL, expand = TRUE)+

    geom_line(mapping = aes(x = seq(1:length(x.ts)), y = x.ts),
              color = "black", size=1)+

    geom_hline(yintercept = mean(x.ts), col = 'green', size = 1)+

    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Fonte: Bloomberg",
         y = paste(asix.x),
         x = "Osservazioni"
    )
}
```

Function to build Graphics P_1 ... P_5:

```
###   Funzione per entrambi i grafici: Prezzi & Rendimenti  ###
TwO_Plot <- function(x.price, x.return, tkr) {

  G1 <- grafico.P(x.price,
                  paste('Serie Storica dei Prezzi del ', tkr),
                  'Prezzi dal 9 Luglio 1987 al 18 Ottobre 2002', 'Prezzo')
  G2 <- grafico.R(x.return,
                  paste('Serie Storica dei Rendimenti del ', tkr),
                  'Rendimenti dal 9 Luglio 1987 al 18 Ottobre 2002', 'Rendimento')
  grid.arrange(G1, G2, nrow=2, ncol=1)

}
```

The function which allows to plot the histogram of the returns is shown in the following script, together with the "TwO_Plot.2" function:

```
#####   Funzione per i grafici (istogramma)  ######
grafico_hist <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  ggplot() +
    #coord_cartesian(xlim = c(150, 3720), ylim = NULL, expand = TRUE)+
    geom_histogram(mapping = aes(x = x.ts, y =..ndensity..),
                   fill='black',
                   colour='white',
                   stat = "bin",
                   bins = 150,
                   position = "stack")+
    scale_x_continuous(limits = c(-0.05, 0.05))+
    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Bloomberg",
         y = paste(asix.x),
         x = "Lag")
}

###   Funzione per entrambi i grafici: Prezzi & Rendimenti  ###
TwO_Plot.2 <- function(x.price, x.return, tkr) {

  G1 <- grafico.P(x.price,
                  paste('Serie Storica della Volatilità del ', tkr),
                  'Prezzi dal 9 Luglio 1987 al 18 Ottobre 2002', 'Volatilità')
  G2 <- grafico_hist(x.return,
                     paste('Istogramma dei Rendimenti del ', tkr),
                     'Rendimenti dal 9 Luglio 1987 al 18 Ottobre 2002', 'Densità')
  grid.arrange(G1, G2, nrow=2, ncol=1)

}
```

The following commands provide to produce the graphics shown through above function:

```
### Grafici ###
TwO_Plot(price.SPX, returns.SPX, 'SPX')
TwO_Plot(price.NKY, returns.NKY, 'NKY')
TwO_Plot(price.DAX, returns.DAX, 'DAX')
TwO_Plot(price.CAC, returns.CAC, 'CAC')
TwO_Plot(price.UKX, returns.UKX, 'UKX')

TwO_Plot.2(returns.SPX.abs, returns.SPX, 'SPX')
TwO_Plot.2(returns.NKY.abs, returns.NKY, 'NKY')
TwO_Plot.2(returns.DAX.abs, returns.DAX, 'DAX')
TwO_Plot.2(returns.CAC.abs, returns.CAC, 'CAC')
TwO_Plot.2(returns.UKX.abs, returns.UKX, 'UKX')
```

It is now shown function used to obtain the Table 2 of the paper:

```
###    Funzione Calcolo Statistiche   ###
momenti <- function(x.ts) {
  media <- mean(x.ts, na.rm = TRUE)
  mediana <- median(x.ts, na.rm = TRUE)
  massimo <- max(x.ts, na.rm = TRUE)
  minimo <- min(x.ts, na.rm = TRUE)
  varianza <- var(x.ts, na.rm = TRUE)
  dev_std <- sqrt(varianza)
  asimmetria <- skewness(x.ts)
  curtosi <- kurtosis(x.ts, na.rm = FALSE, method = c("excess"))
  test.jb <- jarque.bera.test(x.ts)
  statistiche <- list(Media = media,
                      Mediana = mediana,
                      Massimo = massimo,
                      Minimo = minimo,
                      Varianza = varianza,
                      Std_Dev = dev_std,
                      Asimmetria = asimmetria,
                      Curtosi = curtosi,
                      JB.Stat = test.jb$statistic[[1]],
                      JB.Pvalue = test.jb$p.value[[1]])
  return(statistiche)
}

stat.SPX <- momenti(returns.SPX)
stat.NKY <- momenti(returns.NKY)
stat.DAX <- momenti(returns.DAX)
stat.CAC <- momenti(returns.CAC)
stat.UKX <- momenti(returns.UKX)


#stat.SPX ####### Controlla se simili a tabella 2 pagina 114
#stat.NKY ################
#stat.DAX ########################
#stat.CAC ##################################
#Stat.UKX ################################################
######### DF0 Tabella 2 ##########################################
df0 <- cbind(stat.SPX, stat.NKY, stat.DAX, stat.CAC, stat.UKX)  #
df0                                                             #
#################################################################
```

The part of the following script shows how the residuals of an OLS model and their standardization are calculated; subsequently the script shows functions relating to the construction of the graphs ACF, PACF, Histogram, QQ Plot and finally, the function built to carry out diagnostics of residues.

```
########################################
###                                  ###
###      Scarti dalla media (Errori)     ###
###                                  ###
########################################

###   Scarti dalla media (Errori)     ###
residui <- function(x.ts) {
  model <- lm(x.ts~c(1:length(x.ts)))
  res <- model$residuals
  return(res)
}

###   Scarti dalla media (Errori) Standardizzati   ###
residui.stndz <- function(x.ts) {
  m <- mean(x.ts)
  s <- (var(x.ts)^0.5)
  z <- (x.ts - m)/s
  return(z)
}

###   Calcolo i vettori degli errori   ###
residual.SPX <- residui(returns.SPX)
residual.NKY <- residui(returns.NKY)
residual.DAX <- residui(returns.DAX)
residual.CAC <- residui(returns.CAC)
residual.UKX <- residui(returns.UKX)

###   Calcolo i vettori degli errori standardizzati   ###
residual.std.SPX <- residui.stndz(residual.SPX)
residual.std.NKY <- residui.stndz(residual.NKY)
residual.std.DAX <- residui.stndz(residual.DAX)
residual.std.CAC <- residui.stndz(residual.CAC)
residual.std.UKX <- residui.stndz(residual.UKX)


###################################
###                             ###
###  Costruzione delle Funzioni  ####
###                             ###
###################################

###   Funzione per i grafici AUTOCorrelazioni per Lag  ###
grafico_correlation <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  corr.x <- acf(x.ts, type = 'correlation', plot = FALSE)
  n.used = length(x.ts)
  clim0 = qnorm((1 + 0.95)/2)/sqrt(n.used)

  ggplot() +

    geom_bar(mapping = aes(x = corr.x$lag, y = corr.x$acf),stat = "identity") +
    geom_hline(yintercept = c(clim0,-clim0), col = 'green', size = 0.7, linetype = "dashed")+

    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Bloomberg",
         y = paste(asix.x),
         x = "Lag"
    )
}

###  Funzione per i grafici AUTOCorrelazioni Parziali per Lag   ###
grafico_correlation_parziali <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  corr.x <- pacf(x.ts, type = 'correlation', plot = FALSE)
  n.used = length(x.ts)
  clim0 = qnorm((1 + 0.95)/2)/sqrt(n.used)
  ggplot() +
    geom_bar(mapping = aes(x = corr.x$lag, y = corr.x$acf),stat = "identity") +
    geom_hline(yintercept = c(clim0,-clim0), col = 'green', size = 0.7, linetype = "dashed")+

    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Bloomberg",
         y = paste(asix.x),
         x = "Lag"
    )
}
```

```r
###???   Funzione per i grafici (AUTOCovarianze per Lag)  ###
grafico_covariance <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  corr.x <- acf(x.ts, type = 'covariance', plot = FALSE)
  ggplot() +
    geom_bar(mapping = aes(x = corr.x$lag, y = corr.x$acf),stat = "identity") +
    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Bloomberg",
         y = paste(asix.x),
         x = "Lag"
    )
}



###    Funzione per istogramma dei residui  ###
grafico_hist_rl <- function(x.ts, tkr.titolo, subtitle, asix.x) {
  #Parametri per la normale
  x <- seq(min(x.ts), max(x.ts), length = length(x.ts))
  y <- dnorm(x, mean = 0, sd = 1)
  #Costruzione grafico
  ggplot() +

    geom_histogram(mapping = aes(x = x.ts, y=..density..),
                   alpha=0.8,
                   binwidth = 0.5,
                   fill='black',
                   colour='white',
                   stat = "bin",
                   position = "stack")+

    geom_density(mapping=aes(x.ts),
                 col = 'blue',
                 size = 1,
                 stat = "density")+

    geom_line(mapping = aes(x=x, y=y),
              col = 'red',
              size = 1.5)+

    scale_x_continuous(limits = c(-5, 5))+

    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Bloomberg",
         y = paste(asix.x),
         x = "Lag")
}

###???   Funzione per i grafici (Q-QPlot)  ###
grafico_qqplot <- function(x.ts, tkr.titolo, subtitle) {
  ggplot() +
    geom_qq(mapping = aes(sample = x.ts),
            geom = "point",
            #position = "identity",
            distribution = stats::qnorm,
            inherit.aes = TRUE)+
    geom_abline(intercept = 0, slope = 1,
                size = 1)+
    labs(title = paste(tkr.titolo),
         subtitle = paste(subtitle),
         caption = "Bloomberg",
         y = 'Sample Quantiles',
         x = "Theoretical Quantiles")+
    scale_y_continuous(limits = c(-7, 7))
}

###   Funzione per 4 grafici  ###
Res_Plot <- function(x.err, tkr) {

  G1 <- grafico_correlation(x.err,
                            paste('Gracifo delle AUTOCorrelazioni per ', tkr),
                            '', 'ACF')
  G2 <- grafico_correlation_parziali(x.err,
                                     paste('Gracifo delle AUTOCorrelazioni Parziali per  ', tkr),
                                     '', 'Partial ACF')
  G3 <- grafico_hist_rl(x.err,
                        paste('Istogramma dei Residui del ', tkr),
                        '', 'Densità')
  G4 <- grafico_qqplot(x.err,
                       paste('QQ PLOT dei Rendimenti del ', tkr),
                      '')

  grid.arrange(G1, G2, G3, G4, nrow=2, ncol=2)

}
```

```
###   Test rugli errori   ###
test_errori <- function(x.ts) {
  media <- mean(x.ts, na.rm = TRUE)
  varianza <- var(x.ts, na.rm = TRUE)
  dev_std <- sqrt(varianza)
  asimmetria <- skewness(x.ts)
  curtosi <- kurtosis(x.ts, na.rm = FALSE, method = c("excess"))
  #Per la normalità
  test.jb <- jarque.bera.test(x.ts)
  #per verificare l'omoschedasticità
  X <- 1:length(x.ts)
  Y <- as.numeric(x.ts)
  test.bp <- bptest(Y~X)
  ### Verifica di assenza di autocorrelazione seriale
  # Box-Ljung
  test.BL <- Box.test(x.ts, lag = 7, type = "Ljung-Box")
  # Box-Pierce
  test.BP <- Box.test(x.ts, lag = 7, type = "Box-Pierce")
  # Durbin-Watson
  test.dw <- dwtest(x.ts~1, alternative = 'two.sided')

  statistiche <- list(Media = media,
                      Varianza = varianza,
                      Sigma = dev_std,
                      Asimmetria = asimmetria,
                      Curtosi = curtosi,
                      JB.Stat = test.jb$statistic[[1]],
                      JB.Pvalue = test.jb$p.value[[1]],
                      BP.Stat = test.bp$statistic[[1]],
                      BP.Pvalue = test.bp$p.value[[1]],
                      LjBOX.Pvalue = test.BL$p.value[[1]],
                      PiBOX.Pvalue = test.BP$p.value[[1]],
                      DW.Stat =  test.dw$statistic[[1]],
                      DW.Pvalue = test.dw$p.value[[1]])
  return(statistiche)
}
```

This last step was designed to build "`Tab.Res`" of this paper.

```
####################################
####                            ####
###      Analisi sui residui     ###
####                            ####
####################################

####################################
##    Test Sugli Errori       #######
####################################
test.err.SPX <- test_errori(residual.std.SPX)
test.err.NKY <- test_errori(residual.std.NKY)
test.err.DAX <- test_errori(residual.std.DAX)
test.err.CAC <- test_errori(residual.std.CAC)
test.err.UKX <- test_errori(residual.std.UKX)

################################
### OUTPUT FINALE DEI RESIDUI ###
################################

Tab.Res <- cbind(test.err.SPX, test.err.NKY, test.err.DAX,
                 test.err.CAC, test.err.UKX)
Tab.Res


####################################
##    Grafici sugli Errori    #######
####################################

#SPX
Res_Plot(residual.std.SPX, 'SPX')

#NKY
Res_Plot(residual.std.NKY, 'NKY')

#DAX
Res_Plot(residual.std.DAX, 'DAX')

#CAC
Res_Plot(residual.std.CAC, 'CAC')

#UKX
Res_Plot(residual.std.UKX, 'UKX')
```

Chapter 2: Estimate of models

2.1 Modeling and theoretical signs on distributions

In this chapter we will consider extensions of the various GARCH and ARCH models with different error distributions for modeling the variance process. The use of these models is justified by the fact that the financial series, being based on observations that have a greater frequency (daily) than the economic data and financial data being distributed in an asymmetrical way and heavy tails, require models that are able to capture the volatility component.

This led modern economic theory to develop new techniques on historical series that marked the birth of the methods of investigation focused primarily on the study of conditional moments after the first.

Before going into the analysis of the various models it seems appropriate to make a brief introductory overview on the major distributions considered. On data were estimated six different theoretical distributions: a canonical one, which is clearly the normal and the other five have heavier tails of the normal and this is justified by the fact that the empirical returns have a high excess parameter of kurtosis, which makes necessary to use leptokurtic distributions.

- → the 'Student's t'
- → The 'GED' (Generalized Error Distribution)
- → the skew-NORM
- → the skew-GED
- → the skew-t

Student's t
The Student's t is a symmetrical distribution and belongs to the family of polynomial distributions; it has the characteristic of a not finite density function over very low degrees of freedom: it depends on a "v" parameter that indicates the degrees of freedom; the stretch of v to the value zero, the distribution has heavier tails and therefore is able to get leptokurtosis phenomena in a more meticulous of a normal distribution.

Generalized Error Distribution (GED)
The GED is a symmetrical unimodal distribution belonging to the exponential family and its density function depends on three parameters:

- μ It is the location parameter for the average value and assumes all R
- σ^2 It is the scale parameter for the variance and assumes value throughout R^+
- α It is the shape parameter (indicating the degrees of freedom) and takes non-negative values

The above-mentioned symmetric distributions can be made asymmetrical by inserting a parameter 'skewness'; the latter measure the degree of asymmetry of a distribution around its mean: if positive indicates the orientation towards the positive values (long tail to the right), while if negative towards negative values (to the left).

In the context before outlined we first analyze the AR model used to model the mean and then all the ARCH and GARCH models that consider heteroskedasticity of returns and that, therefore, for expected volatility, the conditional variance model using history.
Finally, through a combination of them, we estimate them to different confidence levels and using different parameters.

AR

These autoregressive linear models closely resemble a regression model in which the explanatory variables are the past values of the dependent variable.

There are various types of AR which differ from each other for the model order, ie by the number of past values taken into consideration. Starting from the easier autoregression model of order one, in fact, this concept extends to the more general case in which the order of the model is a generic p> 1. From a mathematical point of view, the model is called AR (1) to indicate that at time n the process depends only on the value of the process to a previous step, namely analytically if:

$$Y_t = \phi Y_{t-1} + u_t \tag{12}$$

where the parameter ϕ represents the linear regression coefficient while the u_t is the error term. The elaborate analysis, we have focused in particular to analyze the AR models (1), AR (2) and AR (3), whose mathematical expressions are respectively:

$$Y_{t=} \phi_1 Y_{t-1} + u_t \tag{13}$$

$$Y_{t=} \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + u_t \tag{14}$$

$$Y_{t=} \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3} + u_t \tag{15}$$

ARCH

The family of ARCH (Autoregressive Conditional Heteroskedasticity) was introduced by Engle in 1982: it is autoregressive models with conditional heteroskedasticity, namely the conditional variance changes with the time parameter and it's depend on a constant and the square of shocks at time t-1. Engle suggests to model the conditional variance as a function of q errors in the most recent square.

The ARCH model (q) takes the following form

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 , \tag{16}$$

where rt = ε_t,
conditional error εt | $I_{t-1} \sim N(0, \sigma_t^2)$
and I_{t-1} represents the t-1 set of available information up to time.

- the restrictions $\alpha_0 > 0$ and $\alpha_i \geq 0$, ∀ i> 0, are necessary to ensure the positivity of the conditional variance;
- The constraint α_i <1 instead, it needs to ensure that the conditional variance is over (this hypothesis is necessary in order to avoid an explosive variance, of course, does not comply with the underlying assumptions of normality).

In this model, a shock on current volatility, which occurred "the" prior periods (i≤q) and whose effect is captured by α_i , has no effect on the conditional variance. Therefore it follows that older news have less impact on the current volatility.
Under the assumption of an ARCH model, high yields tend probabilistically (the probability of obtaining a high value is greater than obtain one small) to be followed by high yields, and vice versa: this feature is similar to the volatility clustering observed for the financial returns.
These models, however, have very obvious problems: on the one hand, if the volatility persists empirical, values of q very large are recorded and, consequently, many α_i coefficients;
on the other, ARCH assume that the positive and negative shocks have the same effect on the volatility because this depends on the square of the positive and negative shocks.

These problems are overcome with the class of GARCH models proposed by Bollerslev in 1986.

AP-ARCH

Ding, Granger and Engle (1993) introduce the class of ARCH models to "asymmetric power" (Asymmetric Power ARCH). The model is formulated thus:

$$\sigma_t = \alpha_0 + \alpha(\left| a_{t-1} \right| - \gamma_{t-1})^{\delta} + \beta\, \sigma_{t-1} \qquad (17)$$

with α, β, ɣ and δ parameters that are estimated; in particular:

- ɣ parameter for asymmetry
- δ power parameter.

This is equivalent to say that the conditional standard deviation to the past of high model to the δ-nth power values is given by a constant added to a series of coefficients related to past shock, adjusted for an asymmetry parameter and a series of coefficients associated with the deviation standard of past high values to the $\delta -$ nth power. The model, however, presents some of the parameters positivity constraints:

- constant non-negative and nothing
- parameters α, β and non-negative δ
- -1 <ɣ <1

The main feature of this class of models is to understand many of the traditional GARCH specifications, in particular with $\delta = 2$ and $\gamma = 0$ and we have a GARCH (1,1), while $\delta =$ 2 making the GJR (1,1) model.

GARCH

These models modulate the conditional variance at time t as a linear combination of p lags of squared residuals and q lags of the conditional variance. Compared to the ARCH model, lagged values of the conditional variance are introduced, without having to estimate the high number of parameters in the model. This model is based on the set information It-1, therefore the equation of a generic GARCH (p, q) (defined model "standard_GARCH") specifies the conditional variance in the following manner:

$$\sigma^2 = \alpha_0 + 1 + \mathrm{j} \sum_{i=1}^{p} \alpha\, \varepsilon_{t-1}^2 \sum_{j=1}^{q} \beta\, \sigma_{t-j}^2 \qquad (18)$$

in which the terms **εi** are the ARCH length p component while the terms **σj** are the GARCH component of length q. Let us go to estimate the parameters α_0 , α_i and β_j that represent respectively the intercept, the coefficient of the errors and the variance; these parameters must satisfy the sufficient condition for the non-negativity of the conditional variance, namely:

$$\alpha_0 \geq 0, \alpha_i \geq 0 \text{ per i} = 1, 2, \ldots, \text{p} \quad \text{and} \quad \beta_j \geq 0 \text{ per j} = 1, 2, \ldots, \text{q}.$$

In the family of models Garch the most used and which is best suited to the development of our data is the GARCH (1,1). The GARCH models are symmetrical because it does not take into account the sign of ε_t.

They will therefore prove suitable for the study of financial series, for which the sign is extremely important, in particular, to capture the so-called leverage effect: the latter supports the idea that there is a negative correlation between volatility and innovations.

In order to take account of this phenomenon, from GARCH models they have been developed asymmetric generalized models (I-garch, E-garch, GJR-garch, CS-garch, T-garch, T-arch) that respond different volatility.

I-GARCH

The IGARCH models have the property of being persistent variance, ie the information available at time t determines the estimated conditional variance behavior of each future time horizon. This degree of persistence of the conditional variance can be approximated by statistically integrated variance models, like the model Integrated GARCH (IGARCH).

This model is obtained when in a general model GARCH (p, q), $\sum_{i=1}^{p} \alpha$i $+\sum_{j=1}^{q} \beta$j = 1; in this case the volatility shocks have a lasting effect.

T-GARCH

The TGARCH model (Glosten, Jagannathan and Runkle, 1993; Zakoian, 1994) introduces a different behavior corresponding to crossing a threshold, usually fixed to zero, by delayed innovation.

The conditional variance of the model is expressed by the equation:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \, \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta j \, \sigma_{t-j}^2 + \gamma \delta_{t-1}^{-} \, \varepsilon_{t-1}^2 \qquad (19)$$

where γ is the dummy variable which equals 1 if $\varepsilon_{t-1} \geq 0$; zero otherwise.

E-GARCH

The Exponential GARCH, introduced by Nelson, has the following characteristics:

- impossibility of obtaining a negative variance σ_t^2 (without having to impose any conditions on the parameters);
- presence of asymmetry in the reactions of the volatility to innovations;
- possibility of measuring an asymmetrical effect proportional to the amount of the innovations (errors).

The EGARCH model is specified in logarithmic terms, unlike the TGARCH in which is inserted a dummy variable "γ" multiplied by the errors. With regard of EGARCH (1.1), we have:

$$\log(\sigma_t^2) = \alpha_0 + \sum_{i=1}^{n} \alpha_i \; g(\varepsilon_{t-1}) + \sum_{j=1}^{n} \beta_j \log(\sigma_{t-j}^2) \qquad (20)$$

regardless of the value attributed to the parameters, the exponential transformation ensures the non-negativity of the variance.

The term $\beta_j \log(\sigma_{t-j}^2)$ captures the effect of persistence in volatility and, given that the expression has a term autoregressive, stationarity is ensured by the condition $0 < \beta_j < 1$.

The advantage of a EGARCH lies in the fact that in its equation there is the logarithm and this implies the elimination of positivity constraints that impose other models. So, our choice fell on this model for just consider the whole range of values (positive and negative) taking returns, taking into account the possibility that the positive and negative returns have a different impact on volatility. The model has therefore best theoretical properties compared to other models, with an additional cost in terms of computation time, since, generally, the estimation requires a greater number of iterations for achieving convergence.

GJR-GARCH

The GJR-GARCH model, as well known to the surnames of its creators, Glosten, Jagannathan and Runkle, is an extension of the GARCH model as it adds a term that captures the asymmetric evolution of the conditional variance.

This model assumes, in fact, a specific parametric form to conditional heteroskedasticity and it is also a threshold model. The model has the formula:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \gamma_i I_{t-i\ i} \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \ \sigma_{t-j}^2 \qquad (21)$$

in which ɣi represents the lever end; the parameter I_t the indicator function which takes the value one if $\varepsilon_t \geq 0$, zero otherwise. Due to the presence of the indicator function, the persistence of the model depends in particular by the asymmetry of the conditional distribution. In order to ensure the positivity of the model is necessary that the parameters $\alpha_0 \geq 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$.

CSGARCH

Designed by LEE and ENGLE (1999), it is a model that specifies the conditional variance taking into account a permanent feature q_t , whose task is to analyze the different shifts in volatility that hit the headlines in the medium and long term. It is analytically expressed by the formula:

$$\sigma_t^2 = q_t + \sum_{i=1}^{p} \alpha_i (\varepsilon_{t-i}^2 - q_{t-i}) + \sum_{j=1}^{q} \beta_j (\sigma_{t-j}^2 - q_{t-j}) \qquad (22)$$

in which the parameter q_t is determined by the formula:

$$q_t = \alpha_0 + \rho q_{t-1} + \phi (\varepsilon_{t-1}^2 - \sigma_{t-1}^2) \qquad (23)$$

TARCH

The T-ARCH model introduced by Zakoian is a threshold model in which it is possible to divide the distribution of innovations into dissociated intervals and then consider a linear function for the conditional variance. More precisely, it is a model that proposes independence for the asymmetric effect of the shocks, so the conditional variance is defined as:

$$\sigma_t^2 = \alpha_0 + \ \phi \ \sigma_{t-1}^2 + u_t \qquad (24)$$

It differs from EGARCH model because, while the latter has an exponential type lever effect, the tARCH model has a quadratic type of leverage effects.

In the environment of R, we used mainly the "Rugarch" package and the following libraries to build "Garch models" with different distributions.

```
#######################
###    Librerie     ###
#######################
library(PerformanceAnalytics)
library(tseries)
library(foreign)
library(lmtest)
library(rugarch)
library(fGarch)
```

Rugarch supports a range of univariate distributions that include the Normal, the T-Student's, the ged and their skew variants ("snorm", "SGED", "SSTD"), the "ghyp", the "ghst", the "nig" and "JSU").

Initially we have defined two functions that are: "S_Stima" and "tG_Stima" using the library "rugarch". These functions use the methods "ugarchspec" and "ugarchfit" respectively construct the model and fit the data

The two functions perform the same task, with the only difference that the function "tG_Stima" was built as a particular case of the first with the aim of estimating the tGARCH model.

About the "S_Stima" function, it provides to estimate each model regardless of the choice of parameters and distributions.

Variables' function are:

"x.ts" represents the historical series of yields;

"ar" refers to the type of choice parameterization and requires that it is an integer greater than or equal to zero;

"md" refers to previously analyzed models and present in "rugarch" package accepting a string as an input value;

"p" indicates the parameters of the ARCH and GARCH models used and requires as input a level of magnitude vector 2;

"di" depicts the distributions and is also made up of a string.

```
# Stima generale
s_Stima <- function(x.ts, ar, md, p, di){

  GARCH = ugarchspec(variance.model = list(model = md,
                                           garchOrder = c(p[1], p[2])),
                     mean.model = list(armaOrder = c(ar, 0),
                                       include.mean = TRUE),
                     distribution.model = di)
  fitGarch = ugarchfit(data = x.ts, spec = GARCH)
  return(fitGarch)
}

# Stima tGARCH
tG_Stima <- function(x.ts, ar, p, di){

  tGARCH = ugarchspec(variance.model = list(model = 'fGARCH',
                                            submodel = "TGARCH",
                                            garchOrder = c(p[1], p[2])),
                      mean.model = list (armaOrder=c(ar,0),
                                         include.mean = TRUE),
                      distribution.model = di)
  fitGarch = ugarchfit(data = x.ts, spec = tGARCH)
  return(fitGarch)
}
```

The choice of the distribution has been inserted through the "distribution.model" argument and is located within the "ugarchspec" command (which describes the specifications of a GARCH process).

Among the arguments of "ugarchspec" we find "variance.model" that serves to model the variance: we have chosen to use the models "sGARCH", "eGARCH", "gjrGARCH", "apARCH", "iGARCH", "csGARCH", "fGARCH"; while in the list containing the model specifications on the mean, "mean.model", we used the ARMA models with reference only to the autoregressive part (AR) to model the average.

The procedure followed was iterative and allowed us to construct a cycle of functions through a concatenation for list succession.

First, we gave each list a name through: "list_models", "list_distributions", "lista_AR", "lista_GR";

"ar_Garch" and "par_AR" instead allow us to obtain the model simulation by specifying the parameters, the order of the model and the parameterization in each list, as in the following script.

```
###   Variabili utilizzate nella funzione ARGARCH   ###
lista_modelli <- c('sGARCH', 'eGARCH','gjrGARCH', 'apARCH',
                   'iGARCH', 'csGARCH','tGARCH')

lista_distribuzioni <- c('norm', 'std', 'ged',
                         'snorm', 'sstd', 'sged')

lista_AR <- c('AR_0', 'AR_1', 'AR_2', 'AR_3')

lista_GR <- c('1_1', '1_2', '2_1',
              '0_1', '0_2','2_2')

par_Garch <- list(c(1 , 1), c(1 , 2), c(2 , 1),
                  c(0, 1), c(0 , 2), c(2 , 2))

par_AR <- c(0, 1, 2, 3)
```

Each stock index passes through this process, through which the model is estimated. The process is schematically shown in the figure below.

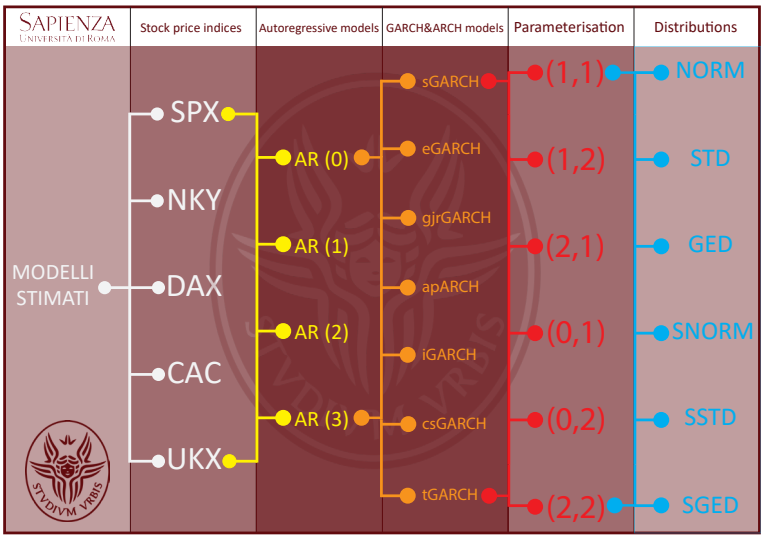


***Figure A**: list " estimated models"*

The goal of this procedure is to obtain a single list named "MODELLI STIMATI" in which we will have the possibility to choose among all the titles and, starting from the title chosen, it will allow us to obtain any type of model.

With reference to the Standard and Poor's, SPX, for example, the model will pass from the "lista_AR", in which the first value AR (0) will be hypothetically selected; with AR (0) we will estimate a GARCH model, which will be selected among all those in the list named "list_models" and the first will be the sGARCH; the latter will be simulated with different combinations of parameters present in the "lista_GR", which gives its name and choosing the combination order through "par_Garch"; finally, the last step will be to simulate the model with a specific distribution among the six present in the "lista_distribuzioni".

At the end of this cycle, the process will resume from the beginning following an identical procedure to the previous one, considering now the parameter 1 for the AR.

By reiterating the process for a number of times equal to the number of stock indices we examine, we managed to get a total of 1008 models estimated for each index.

To achieve this, we have built a function called "ARGARCH", which allows us to estimate any model by creating a list for each title.

```
##########################################
```

```
#######################################
#### Ciclo di costruzione
#######################################
ARGARCH <- function(x.ts, model){
  #Livello 1

  lista_AR_N.GARCH <- list() ## Lista contenente AR
  AN <- 1  # Contatore per gli AR

  for (ar in par_AR){
    # Livello 2

    lista_N.GARCH <- list() ## Lista contenente Modelli
    N <- 1  # Contatore per gli Modelli

    for (md in model) {
      # Livello 3
      PN <- 1
      lista_P.GARCH <- list()

      for (p in par_Garch) {
        # Livello 4B

        n <- 1
        lista_GARCH <- list()

        for (di in lista_distribuzioni){
          # Livello 5
          # if/else per scegliere la funzione da utilizzare
          if (md == 'tGARCH') {

            modello.stimato <- tG_Stima(x.ts, ar, p, di)  # Funzione di stima del modello tGARCH

          }else{

            modello.stimato <- s_Stima(x.ts, ar, md, p, di)  # Funzione di stima modelli in generale

          }
```

```
        lista_GARCH[n] <- list(modello.stimato)
        n <- n + 1

        # Livello 5
      }
      names(lista_GARCH) <- lista_distribuzioni
      lista_P.GARCH[PN] <- list(lista_GARCH)
      PN <- PN + 1

      # Livello 4B
    }
    names(lista_P.GARCH) <- lista_GR
    lista_N.GARCH[N] <- list(lista_P.GARCH)

    N <- N + 1

    # Livello 3
  }

  names(lista_N.GARCH) <- lista_modelli  #Asseglo il nome dell'indice di lista dei Modelli
  lista_AR_N.GARCH[AN] <- list(lista_N.GARCH) # Inserisco la lista con i modelli alla
                                              # posizione AN della lista degli AR
  AN <- AN + 1   # Aggiorna il contatore AR
  # Livello 2
}

names(lista_AR_N.GARCH) <- lista_AR   #Asseglo il nome al posto dell'indice di lista ar
return(lista_AR_N.GARCH)   # Restituisci la lista principale
# Livello 1
}#FIne Funzione
```

Below we show the application of the newly created function and the calculation of all the possible models for each of the equity indices.

```
SPX.models <- ARGARCH(returns.SPX, lista_modelli)
NKY.models <- ARGARCH(returns.NKY, lista_modelli)
DAX.models <- ARGARCH(returns.DAX, lista_modelli)
CAC.models <- ARGARCH(returns.CAC, lista_modelli)
UKX.models <- ARGARCH(returns.UKX, lista_modelli)
```

Finally, all these created lists have been collected and inserted into a single general list which is precisely that of `"estimated models".`

```
MODELLI_STIMATI <- list(
  SPX.models,
  NKY.models,
  DAX.models,
  CAC.models,
  UKX.models)

names(MODELLI_STIMATI) <- c('SPX', 'NKY', 'DAX', 'CAC', 'UKX')
```

To recall the model of interest and view the final output, simply build strings, as shown below:

```
###Á Tabella 3
SPX.norm <- MODELLI_STIMATI$SPX$AR_1$sGARCH$`1_1`$norm
SPX.std <- MODELLI_STIMATI$SPX$AR_1$sGARCH$`1_1`$std
SPX.ged <- MODELLI_STIMATI$SPX$AR_1$sGARCH$`1_1`$ged

NKY.norm <- MODELLI_STIMATI$NKY$AR_1$sGARCH$`1_1`$norm
NKY.std <- MODELLI_STIMATI$NKY$AR_1$sGARCH$`1_1`$std
NKY.ged <- MODELLI_STIMATI$NKY$AR_1$sGARCH$`1_1`$ged

DAX.norm <- MODELLI_STIMATI$DAX$AR_1$sGARCH$`1_1`$norm
DAX.std <- MODELLI_STIMATI$DAX$AR_1$sGARCH$`1_1`$std
DAX.ged <- MODELLI_STIMATI$DAX$AR_1$sGARCH$`1_1`$ged

CAC.norm <- MODELLI_STIMATI$CAC$AR_1$sGARCH$`1_1`$norm
CAC.std <- MODELLI_STIMATI$CAC$AR_1$sGARCH$`1_1`$std
CAC.ged <- MODELLI_STIMATI$CAC$AR_1$sGARCH$`1_1`$ged

UKX.norm <- MODELLI_STIMATI$UKX$AR_1$sGARCH$`1_1`$norm
UKX.std <- MODELLI_STIMATI$UKX$AR_1$sGARCH$`1_1`$std
UKX.ged <- MODELLI_STIMATI$UKX$AR_1$sGARCH$`1_1`$ged
```

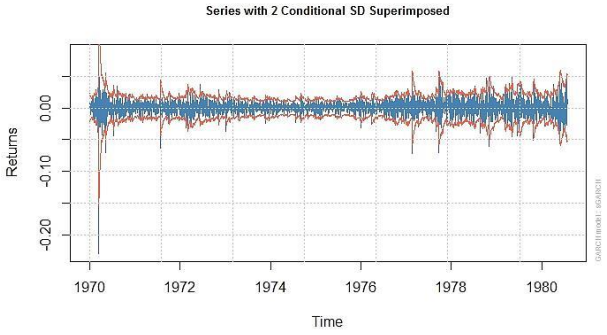
Finally, we have graphically plotted the "MODELLI_STIMATI"function through the previously b uilt strings.
Here,by way of example the SPX index with AR (1) sGARCH (1,1) and with normal distribution; c onsidering that this procedure also applies to all the other titles with their combinations of models, p arameters and distributions.

The command "plot" and the output of R are shown below:

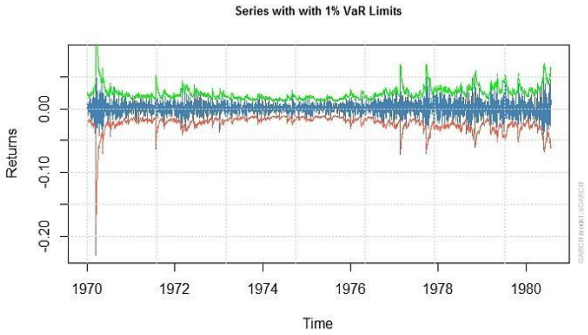
```
plot(MODELLI_STIMATI$SPX$AR_1$sGARCH$`1_1`$norm)
```

```
Make a plot selection (or 0 to exit):

 1:   Series with 2 Conditional SD Superimposed     2:   Series with 1% VaR Limits
 3:   Conditional SD (vs |returns|)                 4:   ACF of Observations
 5:   ACF of Squared Observations                   6:   ACF of Absolute Observations
 7:   Cross Correlation                             8:   Empirical Density of Standardized Residuals
 9:   QQ-Plot of Standardized Residuals            10:   ACF of Standardized Residuals
11:   ACF of Squared Standardized Residuals       12:   News-Impact Curve
```

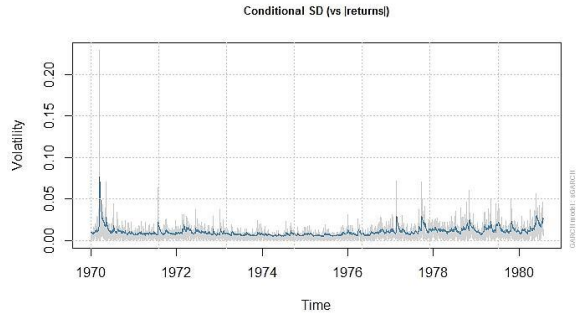
Selection: 1



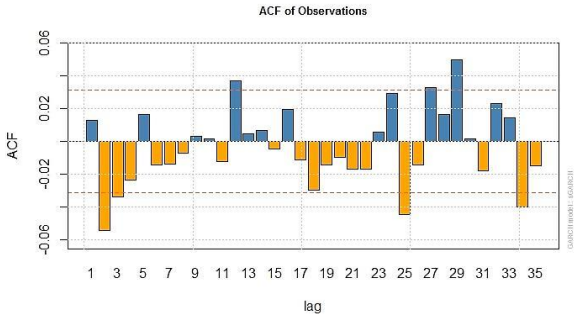
Selection: 2



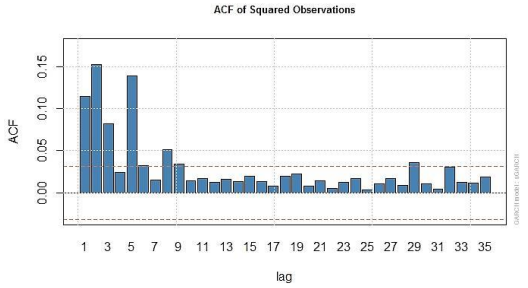
Selection: 3



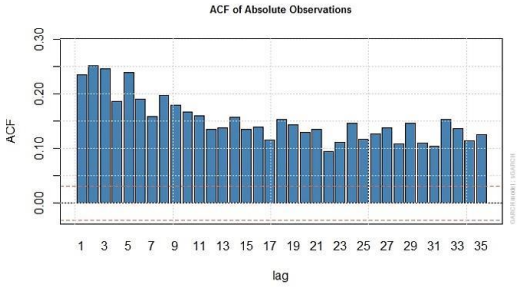
Selection: 4



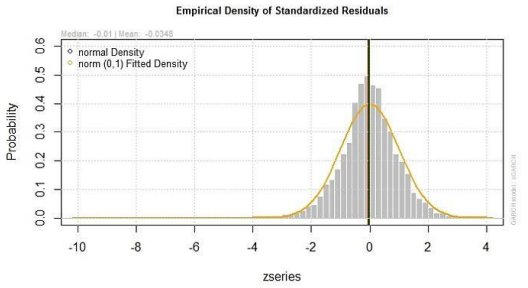
Selection: 5



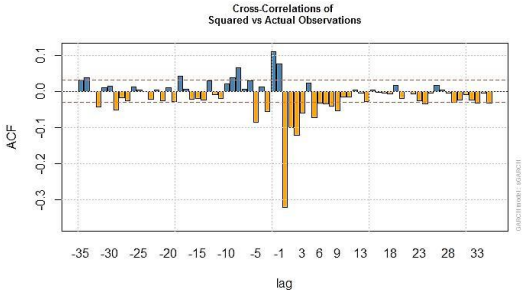
Selection: 6



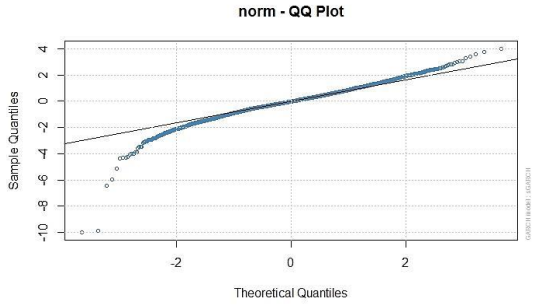
Selection: 7



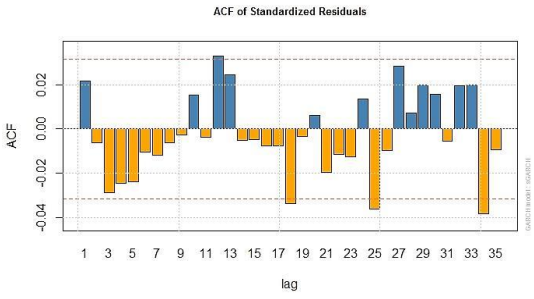
Selection: 8



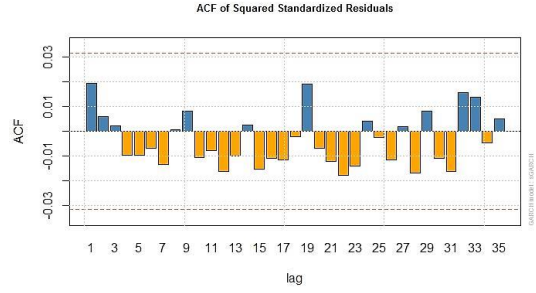
Selection: 9



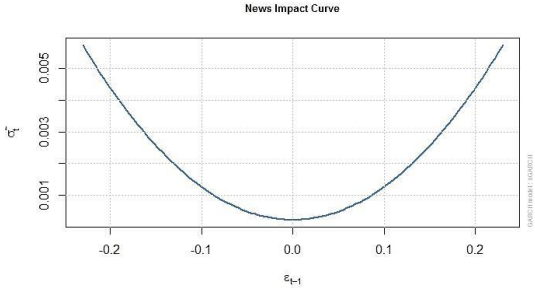
Selection: 10



Selection: 11



Selection: 12



As a last step, we have constructed vector-functions designed to extract the estimate of the various parameters for each index, with its model and distribution. The "R" script is shown below.

```
# Funzione per estrarre subtabella
Vettore <- function(x.ts) {

  v <- round(x.ts@fit$matcoef, 6)
  parametri <- v[,1]
  stand_error <- v[,2]
  lista_v <- seq()
  n <- 1
  i <- 1
  j <- 1

  for (w in 1:length(parametri)) {

    lista_v[n] <- parametri[i]
    n <- n + 1
    i <- i + 1

    lista_v[n] <- stand_error[j]
    n <- n + 1
    j <- j + 1

  }

  lista_v[n] <- likelihood(x.ts)

  return(lista_v)
}
```

By applying this function to each stock vector, we use the `"cbind ()"` command to bundle them together into a single matrix that faithfully reproduces the data in the "Table 3" of the Paper, as the script shows.

```
T.SPX.N <- Vettore(SPX.norm)
T.NKY.N <- Vettore(NKY.norm)
T.DAX.N <- Vettore(DAX.norm)
T.CAC.N <- Vettore(CAC.norm)
T.UKX.N <- Vettore(UKX.norm)

T.N <- cbind(T.SPX.N, T.NKY.N, T.DAX.N, T.CAC.N, T.UKX.N)
rownames(T.N) <- c('mu', '-', 'ar1', '-', 'omega', '-',
                   'alpha1', '-', 'beta1', '-', 'Log LikeHood')

################################################################################
```

Table 3: Estimated Parameters of the AR(1) GARCH (1,1) model for five indices and three different distributions

	T.SPX.N	T.NKY.N	T.DAX.N	T.CAC.N	T.UKX.N
mu	0.000571	0.000561	0.000701	0.000463	0.000406
sd	0.000136	0.000186	0.000181	0.000189	0.000146
ar1	0.028923	0.014978	0.028915	0.045198	0.055030
sd	0.017595	0.017940	0.018081	0.017311	0.017262
omega	0.000002	0.000004	0.000005	0.000006	0.000003
sd	0.000001	0.000002	0.000003	0.000002	0.000001
alpha1	0.104049	0.159091	0.139030	0.105150	0.094280
sd	0.017983	0.015104	0.015511	0.005369	0.012391
beta1	0.889346	0.839905	0.841890	0.865351	0.885826
sd	0.018080	0.014567	0.013139	0.008557	0.013864
Log LikeHood	12485.923050	10921.781688	11427.081531	11400.674735	12509.263159

	T.SPX.T	T.NKY.T	T.DAX.T	T.CAC.T	T.UKX.T
mu	0.000596	0.000283	0.000698	0.000576	0.000419
sd	0.000120	0.000169	0.000161	0.000183	0.000139
ar1	0.011074	-0.007624	0.010082	0.039411	0.049060
sd	0.015767	0.016798	0.015405	0.016680	0.016476
omega	0.000001	0.000002	0.000002	0.000003	0.000002
sd	0.000000	0.000003	0.000004	0.000002	0.000001
alpha1	0.059139	0.099239	0.094135	0.083463	0.078163
sd	0.005022	0.031157	0.053798	0.013138	0.017826
beta1	0.938832	0.899746	0.899609	0.899109	0.906615
sd	0.004586	0.029245	0.054536	0.015829	0.019998
shape	5.581308	6.300967	7.695111	10.594297	10.853064
sd	0.473334	0.706762	0.381954	1.498044	1.322226
Log LikeHood	12667.538313	11086.388004	11575.128408	11452.427882	12609.792388

	T.SPX.G	T.NKY.G	T.DAX.G	T.CAC.G	T.UKX.G
mu	0.000543	0.000314	0.000736	.000507	0.000441
sd	0.000125	0.000179	0.000164	0.000185	0.000142
ar1	-0.000899	-0.008343	0.009663	0.035590	0.044711
sd	0.017106	0.017157	0.016894	0.016743	0.016805
omega	0.000001	0.000002	0.000003	0.000004	0.000002
sd	0.000000	0.000002	0.000002	0.000002	0.000005
alpha1	0.067446	0.111708	0.110197	0.092225	0.084106
sd	0.006768	0.021557	0.015941	0.010594	0.057692
beta1	0.929859	0.886720	0.879404	0.885141	0.899376
sd	0.006789	0.020890	0.016845	0.012930	0.066140
shape	1.247997	1.314053	1.409014	1.556897	1.524015
sd	0.034744	0.037415	0.037996	0.048033	0.064860
Log LikeHood	12643.745372	11052.363415	11524.915805	11435.766468	12569.143407

2.2 tARCH model

As regards the tARCH model, we decided to take care of this model at a later date due to a series of technical requirements and practice. In particular, the TArch has requested a fixed parameter, being effectively a GARCH with parameter (1, 0). In view of the previous code and the `"ARGARCH ()"` function, this model would have weighed down the code and made irregular the flow of the **figure A**. For this reason and with the same logic previously seen, we build a stream that will allow us to obtain a list called `"MODELLI_STIMATI_tARCH".` The following figure shows the combinations to be obtained with the code that will be defined below.

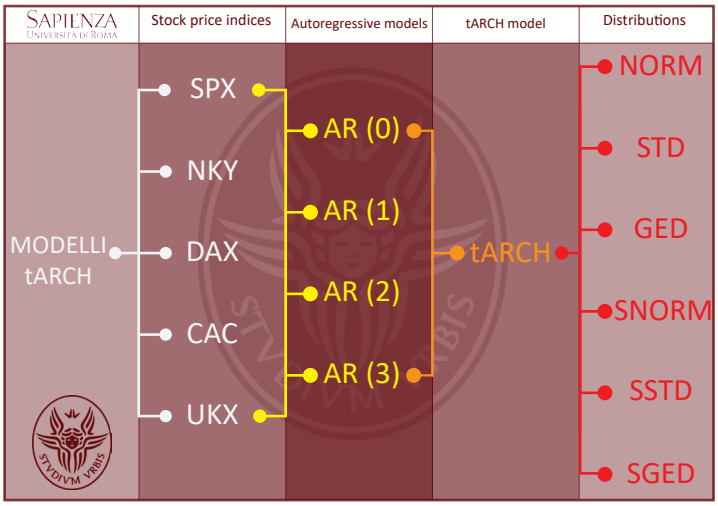


Figure B: *list "estimated models tARCH"*

It is noteworthy that will be used only three lists previously defined for iterations: `"lista_distribuzioni", "lista_AR"` and `"par_AR"`. This is because won't be necessary to choose either the model or the configuration of the latter, as they are both fixed on a standard GARCH (1, 0).

The first step is to build a prediction function that deals with estimation of the model based on the combinations of AR parameters (q), with q = {0, 1, 2, 3}, with the TArch (1, 0) and the six conditioning distributions already mentioned previously.

As can be seen the function takes as input three parameters which are respectively:

`"X.ts"` the vector of the stock's returns;

`"ar"` the parameter that will be attributed to the AR (q model), or ar = q;

`"di"` indicating the distribution to which condition the model residuals. The body of the `"tA_Stima"` function consists of an `"if"` cycle followed by an `"else if"` and a `"final else"`. The three conditions, which are activated respectively in cases where ar = 1, ar = 0 and all other cases, are able to handle some of the computer errors that did not allow the convergence. In any case, the three conditions, with some exceptions, fulfill the same purpose: estimate the model. The estimation is done through the commands previously `"ugarchspec"` and `"ugarchfit"`. The only difference is the `"out.sample"` setting since the last command, which allows to obtain a number of estimated coefficients equal to the order of q in AR (q).

```
# Stima tARCH
tA_Stima <- function(x.ts, ar, di){

  if (ar == 1) {
    tARCH = ugarchspec(variance.model = list(model = 'sGARCH',
                                             garchOrder = c(1, 0)),
                       mean.model = list (armaOrder=c(ar,0),
                                          include.mean = TRUE),
                       distribution.model = di)
    fitGarch = ugarchfit(data = x.ts, spec = tARCH)

    return(fitGarch)

  }else if( ar == 0) {
    if (di == 'snorm') {
      mess <- paste('Processo: AR(0)tARCH(1,0)sNORM ---> Non Converge')
      return(mess)
    }else {
      tARCH = ugarchspec(variance.model = list(model = 'sGARCH',
                                               garchOrder = c(1, 0)),
                         mean.model = list (armaOrder=c(ar,0),
                                            include.mean = TRUE),
                         distribution.model = di)
      fitGarch = ugarchfit(data = x.ts, spec = tARCH, out.sample = ar)
      return(fitGarch)
    }
  }else {
    tARCH = ugarchspec(variance.model = list(model = 'sGARCH',
                                             garchOrder = c(1, 0)),
                       mean.model = list (armaOrder=c(ar,0),
                                          include.mean = TRUE),
                       distribution.model = di)
    fitGarch = ugarchfit(data = x.ts, spec = tARCH, out.sample = ar)
    return(fitGarch)
  }
}
```

At this point we build "TArch" function. It allows us to build a list for each stock price indices in which will be content the regression output for all models estimated according to the scheme shown in **Figure B**. The function requires as only input the historical series of returns.

```
TARCH <- function(x.ts) {
  AN <- 1
  lista_AR_tA_ARCH <- list()

  for (ar in par_AR){

    n <- 1
    lista_DIS <- list()

    for (di in lista_distribuzioni) {

      modello <- tA_Stima(x.ts, ar, di)

      lista_DIS[n] <- list(modello)
      n <- n + 1
    }
    names(lista_DIS) <- lista_distribuzioni
    lista_tA_DIS <- list()
    lista_tA_DIS[1] <-list(lista_DIS)
    names(lista_tA_DIS) <- c('tARCH_1.1')
    lista_AR_tA_ARCH[AN] <- list(lista_tA_DIS)
    AN <- AN + 1
  }
  names(lista_AR_tA_ARCH) <- lista_AR
  return(lista_AR_tA_ARCH)
}
```

The following show the application of the newly created function and the estimate of all possible models based on TArch for each of the equity indices.

```
#Calcolo della stima dei modelli per ogni titolo
tARCH.SPX <- TARCH(returns.SPX)
tARCH.NKY <- TARCH(returns.NKY)
tARCH.DAX <- TARCH(returns.DAX)
tARCH.CAC <- TARCH(returns.CAC)
tARCH.UKX <- TARCH(returns.UKX)
```

As before, we collect all the models in a single list: "MODELLI_STIMATI_tARCH".

```
#Lista Finale
MODELLI_STIMATI_tARCH <- list(
  tARCH.SPX,
  tARCH.NKY,
  tARCH.DAX,
  tARCH.CAC,
  tARCH.UKX)
```

At this point it will be extremely easy to recall, for each index, the model of interest and view the output. Just following the path shown by 'helper' of "R" We can build a strig such as those shown below, to obtain the output.

```
# Modelle AR(0)tARCH(1,0) condizionato alla normale sui rendimenti del SPX
MODELLI_STIMATI_tARCH$SPX$AR_0$tARCH_1.1$norm

# Modelle AR(2)tARCH(1,0) condizionato alla t di Student sui rendimenti del CAC
MODELLI_STIMATI_tARCH$CAC$AR_2$tARCH_1.1$std
```

The resulting output is:

```
> MODELLI_STIMATI_tARCH$SPX$AR_0$tARCH_1.1$norm

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

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

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000431    0.000161    2.674 0.007496
omega   0.000086    0.000003   33.937 0.000000
alpha1  0.313479    0.027948   11.216 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000431    0.000159   2.7120 0.006688
omega   0.000086    0.000006  13.4501 0.000000
alpha1  0.313479    0.073417   4.2699 0.000020

LogLikelihood : 12105.49
```

```
> MODELLI_STIMATI_tARCH$CAC$AR_2$tARCH_1.1$std

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

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(1,0)
Mean Model       : ARFIMA(2,0,0)
Distribution     : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000475    0.000197   2.4162 0.015683
ar1     0.056723    0.018165   3.1226 0.001793
ar2    -0.021272    0.016276  -1.3069 0.191233
omega   0.000148    0.000007  21.9505 0.000000
alpha1  0.214973    0.031654   6.7914 0.000000
shape   5.214317    0.426113  12.2369 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000475    0.000199   2.3807 0.017279
ar1     0.056723    0.017990   3.1531 0.001616
ar2    -0.021272    0.017717  -1.2007 0.229883
omega   0.000148    0.000011  13.6298 0.000000
alpha1  0.214973    0.042121   5.1037 0.000000
shape   5.214317    0.647769   8.0497 0.000000

LogLikelihood : 11248.65
```

The construction of this "loop" , has been produced in order to optimize the speed of the resolution process as a whole.

Chapter 3: Forecast and calculating the VaR

3.1 Value at Risk

The Value at Risk is a risk measurement applied to financial investments and represents the maximum loss that an exhibition or an exhibition's portfolio can suffer, given a certain level of confidence and within a predetermined time horizon.
The confidence levels taken into consideration in the elaborate are two: 95% and 99%, with respective α values of 0.05 and 0.01.
The VaR is a technique commonly used by investment banks to measure the market risk of financial assets held in the portfolio; it analyzes the correlation between the different financial instruments and the likelihood that you achieve certain performance scenarios.
It thus depends on:

- the length of the forecast;
- the unit of measurement: the VaR is expressed as an absolute value in the base currency chosen by the investor;

The VaR is a widely shared standard, also used for quantifying capital requirements of banking institutions (Basel II).
Its spread in time of application lies mainly in the ease of understanding since it corresponds to the loss, compared to exposures in individual positions or wallets, which is expected to be exceeded in the next period with a probability fixed dell'α% said probability coverage (or insolvency): therefore corresponds all'α-quantile of a specific distribution of profit and losses.

VaR is now a standard tool in risk management. A Value-at-Risk model esteems the maximum loss that can be expected, at a particular significance level, over a given trading horizon, with a specified probability.
If Rt denotes return of a portfolio at time t, and α denotes a (pre-determined) significance level, then the respective VaR (VaR_t) is implicitly defined by the following expression:

$$p_r = [Y_t < -VaR_t | F_{t-1}] = \alpha \tag{25}$$

where:

- F_{t-1} is the information set available at time t-1;
- VaR_t is the αth conditional quantile of Y_t. (In other words, VaR_t is the one-step ahead forecast of the αth quantile of Y_t based on the information available up to period t-1.

From Equation (23) it is clear that finding a VaR is equivalent to finding the conditional quantile of VaR_t.
The main goal is to test the null hypothesis that VaR_t correctly approximates the conditional quantile for a specified level α. In this section, we review some of the existing backtests, which are based on an orthogonality condition between a binary variable and some instruments.

3.2 Backtesting procedure

The spread of VaR models, as fundamental tools for measuring market risk, has made it increasingly important to the development of quality assessment techniques of these models.

We now present some backtests usually mentioned in the literature to identify misspecified VaR models.

In this paper, we adapt the idea of Christoffersen et al. (2001) to investigate the accuracy of a given VaR model. In particular, instead of using the conditional volatility as a regressor, we simply use the VaR measure of interest (VaR_t).
We embed this measure in a general class of models for stock returns in which the specification that delivered VaR_t is nested as a special case. In this way, we can provide a test of the VaR model through a conventional hypothesis test.
We first define a violation sequence by the following indicator function or hit sequence:

$$H_t = \begin{cases} 1 : if\ y_t < -VaR_t \\ 0 : if\ y_t < -VaR_t \end{cases}$$

By definition, the probability of violating the VaR should always be:

$$p_r(H_t = 1 \mid F_{t-1}) = \alpha \tag{26}$$

The unconditional coverage (UC) test of Kupiec (1995)

Kupiec proposes a nonparametric test based on the proportion of exceptions. Assume a sample size of n observations and a number of violations of n_1 .

The objective of the test is to know whether $\pi = \frac{n_1}{n}$ is statistically equal to α:

H_0: $\pi = \mathrm{E}(H_t) = \alpha$

The probability of observing n_1 violations over a sample size of n is driven by a Binomial distribution and the null hypothesis H_0: $\pi = \mathrm{E}(H_t) = \alpha$ can be verified through a LR test.
This test rejects the null hypothesis of an accurate VaR if the actual fraction of VaR violations in a sample is statistically different than α. However, Kupiec finds that, the power of his test is generally low in finite samples, and the test becomes more powerful only when the number of observations is very large.
Kupiec's likelihood ratio (LR) UC test examines the hypothesis that the true violation rate is equal to α, as required for an accurate VaR forecasting method. The LR test statistic is:

$$LR_{uc}(\alpha) = 2\ln[\pi^{n_1}(1-\pi)^{n-n_1}] - 2\ln[\alpha^{n}\ (1-\alpha)^{n-n_1}] \tag{27}$$

where n_1 is the number of violations in n observations and π is the observed sample violation rate. This assesses whether the binary violation indicator series $H_t(Y_t < -VaR_t)$. t=1,…..,n, could have an incidence rate equal to α, or not. Under the null hypothesis, which also assumes the binary series is i.i.d. Bernoulli, LRuc tends towards a $\chi^2(1)$ distribution as n gets large.

The conditional coverage (CC) test of Christoffersen (1998)

With this test, the validity of VaR forecasts is tested through parameter restrictions on the transition probability matrix associated with a two-states Markov chain model (violation/no violation). To be more precise, two assumptions are derived from the martingale difference hypothesis, namely the unconditional coverage and the independence hypotheses.

Christoffersen (1998) develops a conditional coverage (CC) joint test, incorporating the UC test, that the binary violations are independent and occur with nominal rate α; the joint LR test is:

$$LR_{cc}(\alpha) = 2\ln\left[(1-\pi_{01})^{n_{oo}}\pi_{01}^{n_{01}}(1-\pi_{11})^{n_{10}}\pi_{11}^{n_{11}}\right] - 2\ln[\alpha^{n_1}(1-\alpha)^{n-n_1}] \tag{28}$$

where n_{ij} is the number of occurrences of $H_t = j$; $H_{t-1} = i$, for i, j = 0, 1, where $H_t = H(Y_t < -VaR_t)$. Under the alternative, the violations follow a two-state Markov chain process,
where $\pi_{ij} = mi_j / \sum_j mi_j$ with i, j = 0, 1, are the observed Markov transition rates, which violates the independence assumption of the null.
Under the null, LRcc tends towards a $\chi^2(2)$ distribution as n gets large. The result LRcc = LRUC + LRind, where LRind is the independence test of Christoffersen (1998), follows from the definitions of these test statistics.

The joint test, also known as the conditional coverage test, includes the unconditional coverage and independence properties. An interesting feature of this test is that a rejection of the conditional coverage may suggest the need for improvements on the VaR model, in order to eliminate the clustering behavior. On the other hand, the proposed test has a restrictive feature, since it only takes into account the autocorrelation of order 1 in the hit sequence.

The dynamic quantile (DQ) test of Engle and Manganelli (2004)

Engle and Manganelli (2004) develop the DQ test, another joint test for correct coverage and independence, but one that can employ more than just the binary violation series:

$$DQ = \frac{Hit_t' X_t [X_t' X_t]^{-1} X_t' Hit_t}{\alpha(1-\alpha)} \tag{29}$$

where the vector of instruments $X_t$might include lags of Hit_t, VaR_t and its lags. They test the null hypothesis that Hit_t and X_t are orthogonal. Under their null hypothesis, the proposed test statistic follows a $\chi^2(q)$ distribution as n gets large in which q = rank(X_t).
More formally, let us denote by $Hit_t = H_t(\alpha) - \alpha$ the demeaned process of violation, that takes the value:

- $1 - \alpha$ every times Y_t is less than the *ex-ante* VaR;
- $-\alpha$ otherwise.

From the definition of the VaR, the conditional expectation of $Hit_t(\alpha)$, given the information known at $t - 1$ must be zero. In particular, under the CC assumption, the variable $Hit_t(\alpha)$ must be uncorrelated with its own lagged values and with any other lagged variable (including past returns, past VaR, etc.), and its expected value must be equal to zero.

We have create a function to calculate the Dynamic Quantile test of Engle and Manganelli (2004), called "`Matrix_DQ`". The variables of input are:

- "`Y`" is a numeric vector of the actual realized returns, that we compare with a vector of the VaR;
- "`VAR`" is a numeric vector of the VaR;
- "`Alpha`" is the quantile used for the VaR;
- "`nHIT`" are the lags used in the Dynamic Quantile test of Engle and Manganelli (2004);
- … other variable are setting for matrix construction.

First one we calculate the vector of HIT and define the matrix of DQ.

```
Matrix_DQ <- function(Y, VAR, alpha, intercept=T, nVAR=0, nHIT=c(1,2,3,4), nY=F){
  N <- length(Y)
  n <- N-max(nVAR, nHIT, nY)
  HIT <- (Y< -VAR) - alpha
  HIT_A <- tail(HIT, N - length(nHIT))
  X <- matrix(ifelse(intercept,1,0),ncol=1,nrow=n)
```

Second step are the construction of the matrix with three different cycle. The first one creates the vector of VaR and put it in a matrix "X"; the second adds the vectors of hits to the same matrix, and the last one adds the vector of actual to matrix.

```
  if(any(is.numeric(nVAR))){
    xVAR=matrix(NA, n, length(nVAR))
    for(i in 1:length(nVAR)){
      xVAR[,i]=VAR[(N-nVAR[i]-n+1):(N-nVAR[i])]
    }
    X <- cbind(X, xVAR)}

  if(any(is.numeric(nHIT))){
    xHIT=matrix(NA, n, length(nHIT))
    for(i in 1:length(nHIT)){
      xHIT[,i]=HIT[(N-nHIT[i]-n+1):(N-nHIT[i])]
    }
    X <- cbind(X, xHIT)}

  if(any(is.numeric(nY))){
    xY=matrix(NA, n, length(nY))
    for(i in 1:length(nY)){
      xY[,i]=Y[(N-nY[i]-n+1):(N-nY[i])]
    }
    X <- cbind(X, xY)}
```

Finally we calculate the statistic and p-value of the test.

```
   #Numertore
   x1 <- tcrossprod(solve(crossprod(X)), X)
   x2 <- X %*% x1
   x3 <- t(HIT_A) %*% x2
   x4 <- x3 %*% HIT_A

   #Denominatore
   d1 <- alpha*(1-alpha)

   #Calcolo Statistica e P-Value
   statistica = (x4)/(d1)
   p_value <- 1 - pchisq(statistica, df = rk(X))
```

At the end we built a list with the both values.

```
  #Lista
  R <- list(statistica, p_value)
  names(R)<- list('Statistica', 'P-Value')
  return(R)
}
```

In this chapter we show the method to creating rolling density forecast from ARMA-GARCH models with the option: refitting every n-1 periods. We have used the same variables used for the "s_Stima" and "ARGARCH" . In this case we define the function "s_Roll", where it is setting "forcast" .

```
s_Roll <- function(x.ts, ar, md, p, di, forcast){

  GARCH = ugarchspec(variance.model = list(model = md,
                                           garchOrder = c(p[1], p[2])),
                     mean.model = list(armaOrder = c(ar, 0),
                                       include.mean = TRUE),
                     distribution.model = di)

  RollGarch = ugarchroll(GARCH, x.ts, n.ahead = 1,
                         forecast.length = forcast,
                         refit.every = forcast-1,
                         refit.window = "moving",
                         solver = 'hybrid',
                         calculate.VaR = TRUE,
                         VaR.alpha = c(0.01, 0.05))

  #MATRICE VaR
  return(RollGarch)
}
```

We use the method "ugarchroll" to calculate the density of model.

Next we define the function "ARGARCH_Roll" that work in the same way of "ARGARCH", but it uses the function "s_Roll"

```
###   Funzione ARGARCH   ###
ARGARCH_Roll <- function(x.ts, forcast){
  NNN <- 1
  AN <- 1
  lista_AR_N.GARCH <- list()
  for (ar in par_AR) {
    N <- 1
    lista_N.GARCH <- list()
    for (md in lista_modelli) {

      PN <- 1
      lista_P.GARCH <- list()
      for (p in par_Garch) {

        n <- 1
        lista_GARCH <- list()

        for (di in lista_distribuzioni){
          if( md == 'tGARCH'){

            rollGarch <- tG_Roll(x.ts, ar, p, di, forcast)

          }else{

            rollGarch <- s_Roll(x.ts, ar, md, p, di, forcast)
          }

          lista_GARCH[n] <- list(rollGarch)
          n <- n + 1

          LISTA_MODELLI_ROLL[NNN] <- list(model)
          NNN <- NNN + 1

          #NOMI MATRICE
          Names_Models[TOT] <- paste('AR_', ar, md, p[1], p[2], di, sep = '.')

          TOT <- 1 + TOT
        }


         names(lista_GARCH) <- lista_distribuzioni
         lista_P.GARCH[PN] <- list(lista_GARCH)
         PN <- PN + 1
       }
       names(lista_P.GARCH) <- lista_GR
       lista_N.GARCH[N] <- list(lista_P.GARCH)
       N <- N + 1
     }
     names(lista_N.GARCH) <- lista_modelli
     lista_AR_N.GARCH[AR] <- list(lista_N.GARCH)
     AN <- AN + 1
   }

   names(lista_AR_N.GARCH) <- lista_AR
   names(LISTA_MODELLI_ROLL) <- Names_Models

   GENERAL <- list(lista_AR_N.GARCH, LISTA_MODELLI_ROLL)

   return(GENERAL)
}
```

The result of ours work is a one list that contains all object "rollGarch" for each parameter, model and distribution.

Next step is to select the good model for each confidence level. We take into account only the models that respects ,at the same time, the three condition: Unconditional Coverage, Conditional Coverage and Dynamic Quantile Test.

This models are stored in a specific list. To make this we have create the function "ARGARCH_Roll_test"; the inputs are only the returns of share index and the length of forecast.

```
###   Funzione ARGARCH   ###
ARGARCH_Roll_test <- function(x.ts, forcast){
  #LISTE 99%
  Lista_ACC.99 <- list()
  ACC.99 <- 1
  Lista_RIF.99 <- list()
  RIF.99 <- 1

  Lista_ACC.NAMES.99 <- as.character()
  ACC.n.99 <- 1
  Lista_RIF.NAMES.99 <- as.character()
  RIF.n.99 <- 1

  #LISTE 95%
  Lista_ACC.95 <- list()
  ACC.95 <- 1
  Lista_RIF.95 <- list()
  RIF.95 <- 1

  Lista_ACC.NAMES.95 <- as.character()
  ACC.n.95 <- 1
  Lista_RIF.NAMES.95 <- as.character()
  RIF.n.95 <- 1

  for (ar in par_AR) {
    for (md in lista_modelli) {
      for (p in par_Garch) {
        for (di in lista_distribuzioni){

          rollGarch <- s_Roll(x.ts, ar, md, p, di, forcast)

          #NOMI MODELLO
          Names_Models <- paste(forcast, 'AR_', ar, md, p[1], p[2], di, sep = '.')

          # Residui
          actual <- rollGarch@forecast$VaR$realized

          #Matrice VaR 99%
          mVaR.99 <- rollGarch@forecast$VaR$`alpha(1%)`

          #Verifica del modello con i test di Kupiec e Cristopher
          test.99 <- try(VaRTest(alpha = 0.01, actual, VaR = mVaR.99, conf.level = 0.99),
                         silent = TRUE)

          ### Ciclo per il test 99%
          if (length(test.99) == 1) {
            msg <- paste(' ERROR SOLVER ---> ', ' -> 99% <- ', Names_Models)
            print(msg)
          }else {

            ### Decisione sui modelli, VAR 99%  ###
            UC.99.p <- test.99$uc.LRp
            CC.99.p <- test.99$cc.LRp

            ### DQ TEST ###
            dq.test.99 <- Matrix_DQ(actual, mVaR.99, 0.99)

            if (is.na(UC.99.p)) {
              msg <- paste(' ERROR ---> ', ' -> 99% <- ', Names_Models)
              print(msg)
            }else {
              if (UC.99.p > 0.01) {
                if (CC.99.p > 0.01) {
                  if (dq.test.99$`P-Value` > 0.01) {
                    Lista_ACC.99[ACC.99] <- list(rollGarch)
                    ACC.99 <- ACC.99 + 1
                    #nomi
                    Lista_ACC.NAMES.99[ACC.n.99] <- Names_Models
                    ACC.n.99 <- ACC.n.99 + 1
                    msg <- paste(' ACCETTATO ---> ', ' -> 99% <- ', Names_Models)
                    print(msg)
                  }else{
                    Lista_RIF.99[RIF.99] <- list(rollGarch)
                    RIF.99 <- RIF.99 + 1
                    #nomi
                    Lista_RIF.NAMES.99[RIF.n.99] <- Names_Models
                    RIF.n.99 <- RIF.n.99 + 1

                    ###
                    msg <- paste(' RIFIUTATO DQ ---> ', ' -> 99% <- ', Names_Models)
                    print(msg)
                  }
                }else {
                  Lista_RIF.99[RIF.99] <- list(rollGarch)
                  RIF.99 <- RIF.99 + 1
                  #nomi
                  Lista_RIF.NAMES.99[RIF.n.99] <- Names_Models
                  RIF.n.99 <- RIF.n.99 + 1

                   msg <- paste(' RIFIUTATO CC ---> ', ' -> 99% <- ', Names_Models)
                   print(msg)
                }
              }else {
                Lista_RIF.99[RIF.99] <- list(rollGarch)
                RIF.99 <- RIF.99 + 1
                #nomi
                Lista_RIF.NAMES.99[RIF.n.99] <- Names_Models
                RIF.n.99 <- RIF.n.99 + 1

                ###
                msg <- paste(' RIFIUTATO UC ---> ', ' -> 99% <- ', Names_Models)
                print(msg)
              }
            }
          }
```

```
        #Matrice VaR 99%
        mVaR.95 <- rollGarch@forecast$VaR$`alpha(5%)`

        #Verifica del modello con i test di Kupiec e Cristopher
        test.95 <- try(VaRTest(alpha = 0.05, actual, VaR = mVaR.95, conf.level = 0.95),
                       silent = TRUE)

        ### Ciclo per il test 95%
        if (length(test.95) == 1) {
          msg <- paste(' ERROR SOLVER ---> ', ' -> 95% <- ', Names_Models)
          print(msg)
        }else {

          ### Decisione sui modelli, VAR 95%  ###
          UC.95.p <- test.95$uc.LRp
          CC.95.p <- test.95$cc.LRp

          ### DQ TEST ###
          dq.test.95 <- Matrix_DQ(actual, mVaR.95, 0.95)


         if (is.na(UC.95.p)) {
           msg <- paste(' ERROR ---> ', ' -> 95% <- ', Names_Models)
           print(msg)
         }else {
           if (UC.95.p > 0.05) {
             if (CC.95.p > 0.05) {
               if (dq.test.95$`P-Value` > 0.05) {
                 Lista_ACC.95[ACC.95] <- list(rollGarch)
                 ACC.95 <- ACC.95 + 1
                 #nomi
                 Lista_ACC.NAMES.95[ACC.n.95] <- Names_Models
                 ACC.n.95 <- ACC.n.95 + 1

                 ###
                 msg <- paste(' ACCETTATO ---> ', ' -> 95% <- ', Names_Models)
                 print(msg)
               }else {}
             }else {}
           }else {}
         }
       }
     }
   }
 }
```

```
  names(Lista_ACC.99) <- Lista_ACC.NAMES.99
  names(Lista_RIF.99) <- Lista_RIF.NAMES.99
  names(Lista_ACC.95) <- Lista_ACC.NAMES.95
  names(Lista_RIF.95) <- Lista_RIF.NAMES.95

  AC <- list(Lista_ACC.99, Lista_ACC.95)
  names(AC) <- c('Lista_ACC.99', 'Lista_ACC.95')
  RF <- list(Lista_RIF.99, Lista_RIF.95)
  names(RF) <- c('Lista_RIF.99', 'Lista_RIF.95')

  GENERAL <- list(AC, RF )
  names(GENERAL) <- c('ACC', 'RIF')

  return(GENERAL)
}
```

This output is very relevant because represents the input for the next script.

We have created a big matrix that have 16 columns:

- "TIK" is the name of share index;
- "PROB" is the expected probability of efficient of model;
- "AR" is the parameter for ARIMA models
- "MODEL" is the model selected;
- "ALPHA" is the fists parameter of ARCH-GARCH models;
- "BETA" is the second parameter of ARCH-GARCH models,
- "DIST" is the distribution selected;
- "FORCAST" is the length of forecast;
- "UC.ST" is the statistic of unconditional coverage;
- "UC.Pv" is the p-value of UC;
- "CC.ST" is the statistic of conditional coverage;
- "CC.Pv" is the p-value of UC;
- "DQ.ST" is the statistic of dynamic quantile test;
- "DQ.Pv is the p-value of DQ;
- "BONTA" is the effective probability realized of model;
- "EXCED" is the effective probability realized the model.

The rows of the matrix are represented by accepted models, those which have exceed all three tests (UC, CC and DQ). Across this data frame we can immediately have all information that we need without problems.

Before talking about the function for the matrix, we define a function to calculate the "Bontà" of each model. The functions are two, one for each confidence level: 95% and 99%.

```
Bonta.99 <- function(Roll) {

  X = Roll@forecast$VaR$realized
  vVaR <- Roll@forecast$VaR$`alpha(1%)`
  Tem <- Roll@model$forecast.length

  Ex.n <- length(X[X<vVaR])
  Ex.p <- (Ex.n/Tem)*100

  Bonta <- 100 - Ex.p

  lista <- list(Bonta, Ex.p)
  names(lista) <- c('Bontà', 'Exceedances')

  return(lista)
}

Bonta.95 <- function(Roll) {

  X = Roll@forecast$VaR$realized
  vVaR <- Roll@forecast$VaR$`alpha(5%)`
  Tem <- Roll@model$forecast.length

  Ex.n <- length(X[X<vVaR])
  Ex.p <- (Ex.n/Tem)*100

  Bonta <- 100 - Ex.p

  lista <- list(Bonta, Ex.p)
  names(lista) <- c('Bontà', 'Exceedances')

  return(lista)
}
```

The following is the function to built a matrix (note that we show only the function for confidence interval of 99%, the other is shown in the Appendix B).

```
DATA_FRAME_99 <- function(lista.modelli.99, tk, prob) {

  tkr = tk
  quant = prob
  I <- 1

  for (model in lista.modelli.99) {

    ar <- model@model$spec@model$maxOrder
    vmodel <- model@model$spec@model$modeldesc$vmodel
    p.alpha <-  as.numeric(model@model$spec@model$modelinc['alpha'])
    p.beta <- as.numeric(model@model$spec@model$modelinc['beta'])
    distribution <- model@model$spec@model$modeldesc$distribution
    forcast <- model@model$forecast.length
    #statistche
    actual <- model@forecast$VaR$realized
    mVaR.99 <- model@forecast$VaR$`alpha(1%)`
    test.99 <- VaRTest(alpha = 0.01, actual, VaR = mVaR.99, conf.level = 0.99)
    #uc
    UC.99.st <- test.99$uc.LRstat
    UC.99.p <- test.99$uc.LRp
    #uc
    CC.99.st <- test.99$cc.LRstat
    CC.99.p <- test.99$cc.LRp
    #DQ
    MX <- Matrix_DQ(actual, mVaR.99, 0.99)
    MX.st <- MX$Statistica
    MX.p <- MX$`P-Value`
    #Bontà
    BT <- Bonta.99(model)
    BT.st <- BT$Bontà
    BT.p <- BT$Exceedances
    vettore <- c(
      paste(tkr),
      paste(quant),
      as.numeric(ar),
      paste(vmodel),
      p.alpha,
      p.beta,
      paste(distribution),
      as.numeric(forcast),
      as.numeric(UC.99.st),
      as.numeric(UC.99.p),
      as.numeric(CC.99.st),
      as.numeric(CC.99.p),
      as.numeric(MX.st),
      as.numeric(MX.p),
      as.numeric(BT.st),
      as.numeric(BT.p))


    if (I==1) {
      DF0<- as.data.frame(vettore)
      I <- I + 1
    }else {
      DF0[,I] <- vettore
      I <- I + 1
    }

  }

  DF1 <- t(DF0)
  vname <- names(lista.modelli.99)
  length(vname)
  colnms <- c('TIK','PROB', 'AR', 'MODEL', 'ALPHA',
              'BETA', 'DIST', 'FORCAST', 'UC.ST',
              'UC.Pv', 'CC.ST', 'CC.Pv', 'DQ.ST',
              'DQ.Pv', 'BONTA', 'EXCED')
  rownames(DF1) <- vname
  colnames(DF1) <- colnms

  return(DF1)
}
```

Here we create also the function “MTX” that uses the function for both confident interval.

```
MTX <- function(tk,
                ACC.99.1, ACC.99.2, ACC.99.3, ACC.99.4,
                ACC.95.5, ACC.95.6, ACC.95.7, ACC.95.8) {

  M1 <- DATA_FRAME_99(ACC.99.1, tk, '99%')
  M2 <- DATA_FRAME_99(ACC.99.2, tk, '99%')
  M3 <- DATA_FRAME_99(ACC.99.3, tk, '99%')
  M4 <- DATA_FRAME_99(ACC.99.4, tk, '99%')

  Mtx_Ac_99 <- rbind(M1, M2, M3, M4)

  M5 <- DATA_FRAME_95(ACC.95.5, tk, '95%')
  M6 <- DATA_FRAME_95(ACC.95.6, tk, '95%')
  M7 <- DATA_FRAME_95(ACC.95.7, tk, '95%')
  M8 <- DATA_FRAME_95(ACC.95.8, tk, '95%')

  Mtx_Ac_95 <- rbind(M5, M6, M7, M8)

  MM <- rbind(Mtx_Ac_99, Mtx_Ac_95)

  return(MM)
}
```

Now we show an example of command that uses this function to create the data frame of SPX.

```
#ESEMPIO
SPX_ACC <- MTX('SPX',
               SPX_Ac_Rf_500$ACC$Lista_ACC.99, SPX_Ac_Rf_1000$ACC$Lista_ACC.99,
               SPX_Ac_Rf_1500$ACC$Lista_ACC.99, SPX_Ac_Rf_2000$ACC$Lista_ACC.99,
               SPX_Ac_Rf_500$ACC$Lista_ACC.95, SPX_Ac_Rf_1000$ACC$Lista_ACC.95,
               SPX_Ac_Rf_1500$ACC$Lista_ACC.95, SPX_Ac_Rf_2000$ACC$Lista_ACC.95)
```

The following tables have been built for each index and they show a little group of the best models; we have created a list with reference to the different statistics.

Table 4: Backtesting and "Bontà" for each index and for two confidence levels.

SPX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***			
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(0)apARCH(1.2)sged	0.010559	91.81%	500.AR(0)apARCH(1.2)snorm	0.066395	96.73%	500.AR(1)gjrGARCH(2.1)norm	0.934538	98.7%	500.AR(1)apARCH(1.2)sged	96.4%	3.6%
	2000.AR(1)iGARCH(1.1)_norm	0.010559	91.81%	500.AR(0)apARCH(1.2)snorm	0.066395	96.73%	500.AR(1)gjrGARCH(2.2)norm	0.934586	98.79%	1000.AR(0)apARCH(1.1)sged	96.2%	3.84%
	2000.AR(2)apARCH(2.1)sged	0.010559	91.81%	500.AR(2)eGARCH(1.1)sged	0.066395	96.73%	500.AR(2)gjrGARCH(2.2)norm	0.947873	98.75%	1000.AR(0)apARCH(1.1)ged	95.7%	4.3%
	2000.AR(3)eGARCH(1.2)sged	0.010559	91.81%	500.AR(3)apARCH(1.1)sged	0.066395	96.73%	500.AR(2)gjrGARCH(2.1)norm	0.947918	98.75%	2000.AR(1)apARCH(1.1)sged	95.7%	4.3%
	2000.AR(0)apARCH(1.2)sged	0.051360	82.07%	1000.AR(0)apARCH(1.1)sstd	0.268159	87.45%	500.AR(3)apARCH(1.2)norm	0.214281	99.98%	1000.AR(0)sGARCH(1.2)ged	99.5%	0.5%
99%	2000.AR(1)eGARCH(1.1)ged	0.051360	82.07%	1000.AR(1)apARCH(1.2)ged	0.268159	87.45%	500.AR(2)eGARCH(1.1)snorm	0.214301	99.98%	1000.AR(0)apARCH(1.1)std	99.5%	0.5%
	2000.AR(2)apARCH(2.1)std	0.051360	82.07%	1000.AR(2)eGARCH(1.1)ged	0.268159	87.45%	500.AR(1)apARCH(1.2)norm	0.214306	99.98%	1000.AR(0)apARCH(1.1)ged	99.5%	0.5%
	2000.AR(3)eGARCH(1.2)sstd	0.051360	82.07%	1000.AR(3)apARCH(1.1)sstd	0.268159	87.45%	500.AR(2)apARCH(1.1)norm	0.214331	99.98%	1000.AR(0)apARCH(1.1)sged	99.5%	0.5%

NKY

	UC_test			***CC_test***			***DQ_test***			***BONTA'***			
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	1500.AR(0)iGARCH(0.2)std	0.014094	90.56%	500.AR(0)iGARCH(0.2)std	0.066395	96.73%	500.AR(1)eGARCH(0.2)std	2.490339	86.95%	1000.AR(1)iGARCH(0.1)sstd	96.2%	3.8%
	1500.AR(1)iGARCH(0.1)sstd	0.014094	90.56%	500.AR(1)eGARCH(0.1)sstd	0.066395	96.73%	500.AR(1)eGARCH(0.1)std	2.493393	86.92%	1000.AR(1)iGARCH(0.2)std	96.1%	3.9%
	1000.AR(2)gjrGARCH(1.1)snorm	0.021187	88.43%	500.AR(0)iGARCH(0.2)sstd	0.066395	96.73%	500.AR(0)iGARCH(0.2)std	2.514683	86.68%	1000.AR(0)eGARCH(0.1)sstd	96%	4%
	1000.AR(3)eGARCH(1.1)sged	0.021187	88.43%	500.AR(1)eGARCH(0.2)sstd	0.066395	96.73%	500.AR(0)iGARCH(0.2)std	2.540576	86.39%	1000.AR(0)eGARCH(0.2)sstd	96%	4%
	2000.AR(0)csGARCH(1.1)ged	0.051360	82.07%	1000.AR(0)apARCH(1.1)std	0.268159	87.45%	1500.AR(0)eGARCH(1.2)sstd	0.578767	99.67%	500.AR(0)sGARCH(0.2)sstd	99.8%	0.2%
99%	2000.AR(1)csGARCH(0.1)std	0.051360	82.07%	1000.A(1)eGARCH(1.1)ged	0.268159	87.45%	1500.AR(1)eGARCH(1.2)sstd	0.584921	99.66%	500.AR(0)eGARCH(0.1)std	99.8%	0.2%
	2000.AR(2)gjrGARCH(2.1)sged	0.051360	82.07%	1000.AR(2)apARCH(2.1)sstd	0.268159	87.45%	1500.AR(2)eGARCH(1.2)sstd	0.587116	99.66%	500.AR(0)eGARCH(0.1)sstd	99.8%	0.2%
	2000.AR(3)gjrGARCH(2.1)sged	0.051360	82.07%	1000.AR(3)eGARCH(1.1)std	0.268159	87.45%	500.AR(2)sGARCH(1.1)snorm	0.633443	99.58%	500.AR(0)eGARCH(0.2)std	99.8%	0.2%

DAX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***			
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(2)eGARCH(2.1)snorm	0.042374	83.69%	2000.AR(2)tGARCH(1.1)sged	0.205844	90.22%	500.AR(0)tGARCH(1.1)snorm	1.855546	93.27%	1000.AR(2)apARCH(2.1)norm	95.7%	4.3%
	2000.AR(3)eGARCH(2.1)snorm	0.095648	75.71%	2000.AR(3)tGARCH(1.1)sged	0.205844	90.22%	500.AR(0)tGARCH(1.2)snorm	3.339442	76.52%	2000.AR(3)eGARCH(2.1)snorm	95.15%	4.85%
	2000.AR(2)tGARCH(1.1)sged	0.010560	91.81%	2000.AR(3)eGARCH(2.1)snorm	0.871268	64.68%	500.AR(1)tGARCH(2.2)snorm	3.330566	76.51%	2000.AR(2)eGARCH(2.1)snorm	95.1%	4.9%
	2000.AR(3)tGARCH(1.1)sged	0.010560	91.81%	2000.AR(2)eGARCH(2.1)snorm	0.896883	63.86%	500.AR(1)tGARCH(2.1)snorm	3.343781	76.46%	2000.AR(2)tGARCH(1.1)sged	95.05%	4.95%
	2000.AR(0)apARCH(1.1)ged	0.051360	82.07%	1000.AR(0)sGARCH(1.1)std	0.268159	87.45%	500.AR(0)csGARCH(1.1)norm	0.301040	99.95%	500.AR(0)eGARCH(1.1)std	99.8%	0.2%
99%	2000.AR(1)apARCH(1.1)std	0.051360	82.07%	1000.AR(1)eGARCH(2.1)snorm	0.268159	87.45%	500.AR(0)csGARCH(1.2)norm	0.302044	99.95%	500.AR(0)eGARCH(1.1)ged	99.8%	0.2%
	2000.AR(2)eGARCH(2.1)std	0.051360	82.07%	1000.AR(2)apARCH(1.1)ged	0.268159	87.45%	500.AR(0)gjrGARCH(1.1)norm	0.302927	99.95%	500.AR(0)eGARCH(1.1)std	99.8%	0.2%
	2000.AR(3)eGARCH(2.1)std	0.051360	82.07%	1000.AR(3)eGARCH(2.1)snorm	0.268159	87.45%	500.AR(0)gjrGARCH(1.2)norm	0.303186	99.94%	500.AR(0)eGARCH(1.1)sged	99.8%	0.2%

CAC

	UC_test			***CC_test***			***DQ_test***			***BONTA'***			
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(0)tGARCH(2.2)ged	0.010560	91.81%	1000.AR(0)tGARCH(1.2)sged	0.100184	95.11%	2000.AR(0)apARCH(1.1)snorm	0.777941	99.26%	2000.AR(0)tGARCH(1.2)snorm	95.7%	4.3%
	2000.AR(1)tGARCH(1.1)ged	0.010560	91.81%	1000.AR(1)tGARCH(2.1)sged	0.100184	95.11%	2000.AR(0)gjrGARCH(2.2)snorm	1.363193	96.80%	2000.AR(2)tGARCH(1.1)snorm	95.65%	4.35%
	2000.AR(2)tGARCH(1.2)norm	0.010560	91.81%	1000.AR(2)tGARCH(2.2)snorm	0.100184	95.11%	2000.AR(0)gjrGARCH(2.1)snorm	1.363225	96.80%	2000.AR(0)tGARCH(1.1)snorm	95.65%	4.35%
	2000.AR(3)tGARCH(2.1)ged	0.042374	83.70%	1000.AR(3)tGARCH(2.1)snorm	0.100184	95.11%	2000.AR(0)apARCH(1.2)snorm	1.459876	96.21%	2000.AR(3)tGARCH(2.2)snorm	95.6%	4.4%
	2000.AR(0)apARCH(1.1)sged	0.051360	82.07%	1000.AR(0)eGARCH(1.1)sstd	0.268159	87.45%	500.AR(0)sGARCH(2.2)snorm	0.215691	99.98%	500.AR(0)eGARCH(1.1)ged	99.6%	0.4%
99%	2000.AR(1)tGARCH(2.1)snorm	0.051360	82.07%	1000.AR(1)tGARCH(2.1)sged	0.268159	87.45%	500.AR(0)tGARCH(2.1)snorm	0.216732	99.98%	500.AR(0)eGARCH(1.2)ged	99.6%	0.4%
	2000.AR(2)apARCH(1.2)sged	0.051360	82.07%	1000.AR(2)tGARCH(2.2)snorm	0.268159	87.45%	500.AR(0)eGARCH(0.1)sged	0.289296	99.96%	500.AR(0)eGARCH(2.1)std	99.6%	0.4%
	2000.AR(3)csGARCH(1.1)sged	0.051360	82.07%	1000.AR(3)tGARCH(2.1)snorm	0.268159	87.45%	500.AR(3)sGARCH(2.1)std	0.296455	99.95%	500.AR(0)eGARCH(2.1)ged	99.6%	0.4%

UKX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***			
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(0)tGARCH(2.1)ged	0.010560	91.81%	2000.AR(1)tGARCH(1.1)norm	0.115449	94.39%	2000.AR(0)csGARCH(2.2)norm	3.258402	77.58%	2000.AR(0)csGARCH(1.1)snorm	95.4%	4.6%
	2000.AR(1)apARCH(2.2)snorm	0.010560	91.81%	2000.AR(2)tGARCH(1.1)norm	0.115449	94.39%	2000.AR(0)csGARCH(2.1)norm	3.482836	74.62%	2000.AR(2)tGARCH(2.2)norm	95.35%	4.65%
	2000.AR(1)tGARCH(2.1)ged	0.010560	91.81%	2000.AR(1)apARCH(2.2)snorm	0.265874	87.55%	2000.AR(3)csGARCH(1.1)norm	3.959473	68.22%	2000.AR(0)csGARCH(1.2)snorm	95.35%	4.65%
	2000.AR(2)iGARCH(2.1)norm	0.010560	91.81%	2000.AR(2)apARCH(2.1)snorm	0.349499	83.97%	2000.AR(2)csGARCH(1.1)norm	4.053052	66.95%	2000.AR(3)csGARCH(2.2)snorm	95.3%	4.7%
	2000.AR(0)eGARCH(1.1)sged	0.051360	82.07%	1000.AR(0)apARCH(1.1)sged	0.268159	87.45%	1500.AR(3)csGARCH(2.2)sstd	1.344650	96.91%	2000.AR(0)tGARCH(2.1)sged	99.5%	0.5%
99%	2000.AR(1)apARCH(1.2)sged	0.051360	82.07%	1000.AR(1)tGARCH(1.1)sstd	0.268159	87.45%	1000.AR(0)apARCH(2.2)std	1.436875	96.36%	2000.AR(0)tGARCH(2.2)sged	99.5%	0.5%
	2000.AR(2)apARCH(2.1)sged	0.051360	82.07%	1000.AR(2)eGARCH(1.1)std	0.268159	87.45%	1000.AR(0)apARCH(2.1)std	1.477043	96.10%	2000.AR(2)tGARCH(2.1)std	99.5%	0.5%
	2000.AR(3)iGARCH(1.1)sged	0.051360	82.07%	1000.AR(3)tGARCH(2.1)snorm	0.268159	87.45%	1000.AR(0)apARCH(1.1)norm	1.507042	95.90%	2000.AR(2)tGARCH(2.2)std	99.5%	0.5%

For each index, and for each confidence level has been selected a sample consisting of four models that show, for the three tests, higher a p-value :the models that present a statistic and therefore an equal p-value, are able to capture the volatility in the data and thus according to the Unconditional and Conditional coverage tests, can not be defined a ranking to identify the best performed models according to probability values of coverage metrics for all the six distributed innovations.
The tables show, for the different models, the goodness in probabilistic terms of how effectively the model thick to better fit financial information and the percentage of errors in the forecast.

In the light of the above tables, it may be stated that each model is good, but it is not true that among these models there will be the best.
Therefore, to assess, in general terms which models are more predictive capabilities, we made for each index and confidence level a summary table showing in percentage terms the times when they have certain distributions, forecast and models.

Table 5: Percentage frequency for each forecast lenght, distribution and model

SPX

Forecast	500	1000	1500	2000
99%	27.88	27.84	22.30	21.98
95%	42.47	23.16	13.65	20.72

Distributions	norm	std	ged	snorm	sstd	sged
99%	11.04	18.59	18.54	14.70	18.59	18.54
95%	21.47	3.58	13.75	32.86	5.08	23.26

Models	sGARCH	eGARCH	gjrGARCH	apARCH	iGARCH	csGARCH
99%	14.66	16.62	14.97	15.95	16.00	21.81
95%	10.36	21.66	10.73	21.00	23.54	12.71

NKY

Forecast	500	1000	1500	2000
99%	30.08	28.91	23.22	17.78
95%	51.13	37.27	6.57	5.03

Distributions	norm	std	ged	snorm	sstd	sged
99%	6.66	20.07	18.60	11.33	22.56	20.78
95%	11.09	11.60	16.94	16.53	17.76	26.08

Models	sGARCH	eGARCH	gjrGARCH	apARCH	iGARCH	csGARCH
99%	18.39	11.53	13.16	12.40	19.21	25.30
95%	21.87	13.96	13.96	10.99	25.05	14.17

DAX

Forecast	500	1000	1500	2000
99%	27.52	26.02	22.75	23.72
95%	16.67	16.36	10.49	56.48

Distributions	norm	std	ged	snorm	sstd	sged
99%	8.92	18.37	18.37	17.58	18.37	18.37
95%	7.41	0.00	8.02	32.10	0.00	52.47

Models	sGARCH	eGARCH	gjrGARCH	apARCH	iGARCH	csGARCH
99%	14.75	15.50	15.37	14.80	16.52	23.06
95%	6.17	19.44	9.88	17.59	37.04	9.88

CAC

Forecast	500	1000	1500	2000
99%	25.05	30.61	17.03	27.30
95%	31.10	21.40	10.70	36.80

Distributions	norm	std	ged	snorm	sstd	sged
99%	11.67	17.38	17.69	17.99	16.58	18.69
95%	9.42	6.85	11.84	30.67	13.27	27.96

Models	sGARCH	eGARCH	gjrGARCH	apARCH	iGARCH	csGARCH
99%	17.79	12.47	13.83	13.68	15.78	26.45
95%	4.56	20.68	8.42	9.70	50.07	6.56

UKX

Forecast	500	1000	1500	2000
99%	5.37	33.69	27.12	33.82
95%	20.51	28.37	15.45	35.67

Distributions	norm	std	ged	snorm	sstd	sged
99%	10.74	12.07	18.17	20.89	19.63	18.50
95%	18.26	3.37	9.55	46.91	8.43	13.48

Models	sGARCH	eGARCH	gjrGARCH	apARCH	iGARCH	csGARCH
99%	13.73	11.67	15.45	15.65	17.57	25.93
95%	2.25	2.25	9.27	7.87	50.56	27.81

Based on these percentages, for example, as regards the SPX, the level of forecast to 2000 comes up on average in the "best models" a lesser extent than at 500 and 1000; therefore we can conclude that increasing the length of the forecast predictions are less reliable. Regarding to the type of model, the csGARCH and iGARCH are the best. Finally, considering distributions, those that present the “skeweness” component result as the best.

For the tARCH model, we report the best results in the following table. (For the construction of the relative function it is necessary consult the appendix to the end of this chapter.)

Table 6: Best models for tARCH

SPX 99%			
RANK	MODELS	BONTA'	EXCESS
1	500.AR(3)tARCH(1,0)snorm	99.8%	0.2
2	1000.AR(0)tARCH(1,0)sstd	99.6%	0.4
3	2000.AR(3)tARCH(1,0)sstd	98.2%	1.8
SPX 95%			
RANK	MODELS	BONTA'	EXCESS
1	500.AR(3)tARCH(1,0)snorm	99.8%	0.2
2	1000.AR(0)tARCH(1,0)sstd	99.6%	0.4
3	2000.AR(3)tARCH(1,0)sstd	98.2%	1.8

NKY 99%			
RANK	MODELS	BONTA'	EXCESS
1	2000.AR(0)tARCH(1,0)sstd	99.25%	0.75
2	2000.AR(1)tARCH(1,0)sstd	99.25%	0.75
3	2000.AR(2)tARCH(1,0)sstd	99.2%	0.8
NKY 95%			
RANK	MODELS	BONTA'	EXCESS
1	2000.AR(0)tARCH(1,0)sstd	99.25%	0.75
2	1000.AR(2)tARCH(1,0)sstd	99%	1
3	1000.AR(3)tARCH(1,0)sstd	99%	1

DAX 99%			
RANK	MODELS	BONTA'	EXCESS
1	1000.AR(0)tARCH(1,0)norm	99.8%	0.2
2	1000.AR(1)tARCH(1,0)norm	99.7%	0.3
3	1000.AR(2)tARCH(1,0)norm	99.7%	0.3
DAX 95%			
RANK	MODELS	BONTA'	EXCESS
1	1000.AR(0)tARCH(1,0)norm	99.8%	0.2
2	1000.AR(2)tARCH(1,0)norm	99.7%	0.3
3	1000.AR(1)tARCH(1,0)norm	99.7%	0.3

UKX 99%			
RANK	MODELS	BONTA'	EXCESS
1	1000.AR(0)tARCH(1,0)norm	99.6%	0.4
2	1500.AR(0)tARCH(1,0)norm	99.4%	0.6
UKX 95%			
RANK	MODELS	BONTA'	EXCESS
1	1000.AR(0)tARCH(1,0)norm	99.6%	0.4
2	1500.AR(0)tARCH(1,0)norm	99.4%	0.6

Appendix B

Here is reported the code of function to create the data frame at confidence level of 95%.

```
DATA_FRAME_95 <- function(lista.modelli.95, tk, prob) {

  tkr = tk
  quant = prob
  I <- 1

  for (model in lista.modelli.95) {

    ar <- model@model$spec@model$maxOrder
    vmodel <- model@model$spec@model$modeldesc$vmodel
    p.alpha <-  as.numeric(model@model$spec@model$modelinc['alpha'])
    p.beta <- as.numeric(model@model$spec@model$modelinc['beta'])
    distribution <- model@model$spec@model$modeldesc$distribution
    forcast <- model@model$forecast.length
    #statistche
    actual <- model@forecast$VaR$realized
    mVaR.95 <- model@forecast$VaR$`alpha(5%)`
    test.95 <- VaRTest(alpha = 0.05, actual, VaR = mVaR.95, conf.level = 0.95)
    #uc
    UC.95.st <- test.95$uc.LRstat
    UC.95.p <- test.95$uc.LRp
    #uc
    CC.95.st <- test.95$cc.LRstat
    CC.95.p <- test.95$cc.LRp
    #DQ
    MX <- Matrix_DQ(actual, mVaR.95, 0.95)
    MX.st <- MX$Statistica
    MX.p <- MX$`P-Value`
    #Bontà
    BT <- Bonta.95(model)
    BT.st <- BT$Bontà
    BT.p <- BT$Exceedances

   vettore <- c(
     paste(tkr),
     paste(quant),
     as.numeric(ar),
     paste(vmodel),
     p.alpha,
     p.beta,
     paste(distribution),
     as.numeric(forcast),
     as.numeric(UC.95.st),
     as.numeric(UC.95.p),
     as.numeric(CC.95.st),
     as.numeric(CC.95.p),
     as.numeric(MX.st),
     as.numeric(MX.p),
     as.numeric(BT.st),
     as.numeric(BT.p))


   if (I==1) {
     DF0<- as.data.frame(vettore)
     I <- I + 1
   }else {
     DF0[,I] <- vettore
     I <- I + 1
   }

  }

  DF1 <- t(DF0)
  vname <- names(lista.modelli.95)
  length(vname)
  colnms <- c('TIK','PROB', 'AR', 'MODEL', 'ALPHA',
              'BETA', 'DIST', 'FORCAST', 'UC.ST',
              'UC.Pv', 'CC.ST', 'CC.Pv', 'DQ.ST',
              'DQ.Pv', 'BONTA', 'EXCED')
  rownames(DF1) <- vname
  colnames(DF1) <- colnms

  return(DF1)
}
```

The following are the codes related to the construction of the functions for the tARCH.

We use the method "`ugarchroll`" that calculate the density of model.

```
lista_distribuzioni <- c('norm', 'std', 'ged', 'snorm', 'sstd', 'sged')
lista_AR <- c('AR_0', 'AR_1', 'AR_2', 'AR_3')
par_AR <- c(0, 1, 2, 3)

tA_Roll <- function(x.ts, ar, di, forcast){
  tARCH = ugarchspec(variance.model = list(model = 'sGARCH',
                                           garchOrder = c(1, 0)),
                     mean.model = list (armaOrder=c(ar,0),
                                        include.mean = TRUE),
                     distribution.model = di)
  fitGarch = try(ugarchroll(tARCH, x.ts, n.ahead = 1,
                            forecast.length = forcast,
                            refit.every = forcast-1,
                            refit.window = "moving",
                            solver = 'hybrid',
                            calculate.VaR = TRUE,
                            VaR.alpha = c(0.01, 0.05)), silent = TRUE)
  return(fitGarch)
}
```

Then we have create the function "`ARGARCH_Roll_test`"; the inputs are only the returns of share index and the length of forecast.

```
ARGARCH_Roll_test_tARCH <- function(x.ts, forcast){
  #LISTE 99%
  Lista_ACC.99 <- list()
  ACC.99 <- 1
  Lista_RIF.99 <- list()
  RIF.99 <- 1

  Lista_ACC.NAMES.99 <- as.character()
  ACC.n.99 <- 1
  Lista_RIF.NAMES.99 <- as.character()
  RIF.n.99 <- 1

  #LISTE 95%
  Lista_ACC.95 <- list()
  ACC.95 <- 1
  Lista_RIF.95 <- list()
  RIF.95 <- 1

  Lista_ACC.NAMES.95 <- as.character()
  ACC.n.95 <- 1
  Lista_RIF.NAMES.95 <- as.character()
  RIF.n.95 <- 1


  for (ar in par_AR) {
    for (di in lista_distribuzioni){

      rollGarch <- tA_Roll(x.ts, ar, di, forcast)

      #NOMI MODELLO
      Names_Models <- paste(forcast, 'AR_', ar, 'tARCH', 1, 0, di, sep = '.')

      # Residui
      actual <- rollGarch@forecast$VaR$realized
```

We show now the scripts at the two levels of confidence: 99% and 95% respectively

```
#Matrice VaR 99%
mVaR.99 <- rollGarch@forecast$VaR$`alpha(1%)`

#Verifica del modello con i test di Kupiec e Cristopher
test.99 <- try(VaRTest(alpha = 0.01, actual, VaR = mVaR.99, conf.level = 0.99),
               silent = TRUE)

### Ciclo per il test 99%
if (length(test.99) < 12) {
  msg <- paste(' ERROR SOLVER VarTEST ---> ', ' -> 99% <- ', Names_Models)
  print(msg)
}else {

  ### Decisione sui modelli, VAR 99%  ###
  ### DQ TEST ###
  dq.test.99 <- try(Matrix_DQ(actual, mVaR.99, 0.99),
                    silent = TRUE)

  if (length(dq.test.99) < 2) {
    msg <- paste(' ERROR SOLVER DQ ---> ', ' -> 99% <- ', Names_Models)
    print(msg)
  }else {
    if (dq.test.99$`P-Value` > 0.01) {
      Lista_ACC.99[ACC.99] <- list(rollGarch)
      ACC.99 <- ACC.99 + 1
      #nomi
      Lista_ACC.NAMES.99[ACC.n.99] <- Names_Models
      ACC.n.99 <- ACC.n.99 + 1

      ###
      msg <- paste(' ACCETTATO ---> ', ' -> 99% <- ', Names_Models)
      print(msg)
    }else{
      Lista_RIF.99[RIF.99] <- list(rollGarch)
      RIF.99 <- RIF.99 + 1
      #nomi
      Lista_RIF.NAMES.99[RIF.n.99] <- Names_Models
      RIF.n.99 <- RIF.n.99 + 1

      ###
      msg <- paste(' RIFIUTATO ---> ', ' -> 99% <- ', Names_Models)
      print(msg)
    }
  }
}
      #Matrice VaR 95%
      mVaR.95 <- rollGarch@forecast$VaR$`alpha(5%)`

      #Verifica del modello con i test di Kupiec e Cristopher
      test.95 <- try(VaRTest(alpha = 0.05, actual, VaR = mVaR.95, conf.level = 0.95),
                     silent = TRUE)

      ### Ciclo per il test 95%
      if (length(test.95) < 12) {
        msg <- paste(' ERROR SOLVER VarTEST ---> ', ' -> 95% <- ', Names_Models)
        print(msg)
      }else {

        ### Decisione sui modelli, VAR 95%  ###
        ### DQ TEST ###
        dq.test.95 <- try(Matrix_DQ(actual, mVaR.95, 0.95),
                          silent = TRUE)

        if (length(dq.test.95) < 2) {
          msg <- paste(' ERROR SOLVER DQ ---> ', ' -> 95% <- ', Names_Models)
          print(msg)
        }else {
          if (dq.test.95$`P-Value` > 0.05) {
            Lista_ACC.95[ACC.95] <- list(rollGarch)
            ACC.95 <- ACC.95 + 1
            #nomi
            Lista_ACC.NAMES.95[ACC.n.95] <- Names_Models
            ACC.n.95 <- ACC.n.95 + 1

            ###
            msg <- paste(' ACCETTATO ---> ', ' -> 95% <- ', Names_Models)
            print(msg)
          }else {
            Lista_RIF.95[RIF.95] <- list(rollGarch)
            RIF.95 <- RIF.95 + 1
            #nomi
            Lista_RIF.NAMES.95[RIF.n.95] <- Names_Models
            RIF.n.95 <- RIF.n.95 + 1
            ###
            msg <- paste(' RIFIUTATO ---> ', ' -> 95% <- ', Names_Models)
            print(msg)
          }
        }

      }
    }
  }

    names(Lista_ACC.99) <- Lista_ACC.NAMES.99
    names(Lista_RIF.99) <- Lista_RIF.NAMES.99
    names(Lista_ACC.95) <- Lista_ACC.NAMES.95
    names(Lista_RIF.95) <- Lista_RIF.NAMES.95

    AC <- list(Lista_ACC.99, Lista_ACC.95)
    names(AC) <- c('Lista_ACC.99', 'Lista_ACC.95')
    RF <- list(Lista_RIF.99, Lista_RIF.95)
    names(RF) <- c('Lista_RIF.99', 'Lista_RIF.95')

    GENERAL <- list(AC, RF )
    names(GENERAL) <- c('ACC', 'RIF')

    return(GENERAL)
  }
```

we show an example of a command used to more easily recall the lists related to each index

```
#############################################################################
SPX_Ac_Rf_500.tARCH <- ARGARCH_Roll_test_tARCH(returns.SPX, 500)
SPX_Ac_Rf_1000.tARCH <- ARGARCH_Roll_test_tARCH(returns.SPX, 1000)
SPX_Ac_Rf_1500.tARCH <- ARGARCH_Roll_test_tARCH(returns.SPX, 1500)
SPX_Ac_Rf_2000.tARCH <- ARGARCH_Roll_test_tARCH(returns.SPX, 2000)

SPX_ACC_tARCH_LISTA <- list(SPX_Ac_Rf_500.tARCH, SPX_Ac_Rf_1000.tARCH, SPX_Ac_Rf_1500.tARCH, SPX_Ac_Rf_2000.tARCH)
names(SPX_ACC_tARCH_LISTA) <- c('SPX_Ac_Rf_500.tARCH', 'SPX_Ac_Rf_1000.tARCH', 'SPX_Ac_Rf_1500.tARCH', 'SPX_Ac_Rf_2000.tARCH')
save(SPX_ACC_tARCH_LISTA, file = 'SPX_ACC_tARCH_LISTA.RData')
load('SPX_ACC_tARCH_LISTA.RData')

SPX_ACC_tARCH <- list(SPX_ACC_tARCH_LISTA$SPX_Ac_Rf_500.tARCH$ACC$Lista_ACC.99$`500.AR_.3.tARCH.1.0.snorm`,
                      SPX_Ac_Rf_500$SPX_Ac_Rf_1000.tARCH$ACC$Lista_ACC.99$`1000.AR_.0.tARCH.1.0.sstd`,
                      SPX_Ac_Rf_2000.tARCH$ACC$Lista_ACC.99$`2000.AR_.3.tARCH.1.0.sstd`)

save(SPX_ACC_tARCH, file = 'SPX_ACC_tARCH.RData')
```

Here the function to built a matrix:

```
DATA_FRAME_99_tARCH <- function(lista.modelli.99, tk, prob) {

  tkr = tk
  quant = prob

  I <- 1

  for (model in lista.modelli.99) {


    ar <- model@model$spec@model$maxOrder
    vmodel <- model@model$spec@model$modeldesc$vmodel
    p.alpha <-  as.numeric(model@model$spec@model$modelinc['alpha'])
    p.beta <- as.numeric(model@model$spec@model$modelinc['beta'])
    distribution <- model@model$spec@model$modeldesc$distribution
    forcast <- model@model$forecast.length
    #statistche
    actual <- model@forecast$VaR$realized
    mVaR.99 <- model@forecast$VaR$`alpha(1%)`
    test.99 <- VaRTest(alpha = 0.01, actual, VaR = mVaR.99, conf.level = 0.99)
    #Bontà
    BT <- Bonta.99(model)
    BT.st <- BT$Bontà
    BT.p <- BT$Exceedances

    vettore <- c(
      paste(tkr),
      paste(quant),
      paste(vmodel),
      p.alpha,
      p.beta,
      paste(distribution),
      as.numeric(forcast),
      as.numeric(BT.st),
      as.numeric(BT.p))


    if (I==1) {
      DF0<- as.data.frame(vettore)
      I <- I + 1
    }else {
      DF0[,I] <- vettore
      I <- I + 1
    }

  }
```

```
  }
  DF1 <- t(DF0)
  vname <- names(lista.modelli.99)
  length(vname)
  colnms <- c('TIK','PROB', 'MODEL', 'ALPHA',
              'BETA', 'DIST', 'FORCAST', 'BONTA', 'EXCED')
  rownames(DF1) <- vname
  colnames(DF1) <- colnms

  return(DF1)
}

DATA_FRAME_95_tARCH <- function(lista.modelli.95, tk, prob) {

  tkr = tk
  quant = prob

  I <- 1

  for (model in lista.modelli.95) {


    vmodel <- model@model$spec@model$modeldesc$vmodel
    p.alpha <-  as.numeric(model@model$spec@model$modelinc['alpha'])
    p.beta <- as.numeric(model@model$spec@model$modelinc['beta'])
    distribution <- model@model$spec@model$modeldesc$distribution
    forcast <- model@model$forecast.length
    #statistche
    actual <- model@forecast$VaR$realized
    mVaR.95 <- model@forecast$VaR$`alpha(5%)`
    test.95 <- VaRTest(alpha = 0.05, actual, VaR = mVaR.95, conf.level = 0.95)
    #Bontà
    BT <- Bonta.95(model)
    BT.st <- BT$Bontà
    BT.p <- BT$Exceedances

    vettore <- c(
      paste(tkr),
      paste(quant),
      paste(vmodel),
      p.alpha,
      p.beta,
      paste(distribution),
      as.numeric(forcast),
      as.numeric(BT.st),
      as.numeric(BT.p))



    if (I==1) {
      DF0<- as.data.frame(vettore)
      I <- I + 1
    }else {
      DF0[,I] <- vettore
      I <- I + 1
    }

  }
  DF1 <- t(DF0)
  vname <- names(lista.modelli.95)
  length(vname)
  colnms <- c('TIK','PROB', 'MODEL', 'ALPHA',
              'BETA', 'DIST', 'FORCAST', 'BONTA', 'EXCED')
  rownames(DF1) <- vname
  colnames(DF1) <- colnms

  return(DF1)
}
```

Finally, are shown the scripts related to the construction of tables that show in percentage terms the times in which each good model is chosen.

```
NAMES_CONF_LIV <- c('99%', '95%')
NAMES_FORECAST <- c('500', '1000', '1500', '2000')
NAMES_DISTRIB <- c('norm', 'std', 'ged', 'snorm', 'sstd', 'sged')
NAMES_MODEL <- c('sGARCH', 'eGARCH', 'gjrGARCH', 'apARCH', 'iGARCH', 'csGARCH')
NAMES_F <- c('FORECAST', 'DISTRIBUZIONE', 'MODELLO')

ANALISI <- function(DF99, DF95) {
  T99 <- nrow(DF99)
  T95 <- nrow(DF95)

  #FORECAST
  #99%
  n_99_500 <- length(DF99[DF99 =='500'])
  n_99_1000 <- length(DF99[DF99 =='1000'])
  n_99_1500 <- length(DF99[DF99 =='1500'])
  n_99_2000 <- length(DF99[DF99 =='2000'])

  PERCENT_FORECAST_99 <- c(n_99_500/T99*100, n_99_1000/T99*100,
                           n_99_1500/T99*100, n_99_2000/T99*100)

  #95%
  n_95_500 <- length(DF95[DF95 =='500'])
  n_95_1000 <- length(DF95[DF95 =='1000'])
  n_95_1500 <- length(DF95[DF95 =='1500'])
  n_95_2000 <- length(DF95[DF95 =='2000'])

  PERCENT_FORECAST_95 <- c(n_95_500/T95*100, n_95_1000/T95*100,
                           n_95_1500/T95*100, n_95_2000/T95*100)

  PERCENT_FORECAST <- matrix(NA, 2, 4)
  PERCENT_FORECAST[1,] <- PERCENT_FORECAST_99
  PERCENT_FORECAST[2,] <- PERCENT_FORECAST_95
  rownames(PERCENT_FORECAST) <- NAMES_CONF_LIV
  colnames(PERCENT_FORECAST) <- NAMES_FORECAST
  PERCENT_FORECAST <- round(PERCENT_FORECAST, digits = 2)
```

```
  #DISTRIBUZIONE
  #99%
  n_99_norm <- length(DF99[DF99 =='norm'])
  n_99_std <- length(DF99[DF99 =='std'])
  n_99_ged <- length(DF99[DF99 =='ged'])
  n_99_snorm <- length(DF99[DF99 =='snorm'])
  n_99_sstd <- length(DF99[DF99 == 'sstd'])
  n_99_sged <- length(DF99[DF99 == 'sged'])

  PERCENT_DISTRIB_99 <- c(n_99_norm/T99*100, n_99_std/T99*100,
                          n_99_ged/T99*100, n_99_snorm/T99*100,
                          n_99_sstd/T99*100, n_99_sged/T99*100)

  #95%
  n_95_norm <- length(DF95[DF95 =='norm'])
  n_95_std <- length(DF95[DF95 =='std'])
  n_95_ged <- length(DF95[DF95 =='ged'])
  n_95_snorm <- length(DF95[DF95 =='snorm'])
  n_95_sstd <- length(DF95[DF95 == 'sstd'])
  n_95_sged <- length(DF95[DF95 == 'sged'])

  PERCENT_DISTRIB_95 <- c(n_95_norm/T95*100, n_95_std/T95*100,
                          n_95_ged/T95*100, n_95_snorm/T95*100,
                          n_95_sstd/T95*100, n_95_sged/T95*100)

  PERCENT_DISTRIB <- matrix(NA, 2, 6)
  PERCENT_DISTRIB[1,] <- PERCENT_DISTRIB_99
  PERCENT_DISTRIB[2,] <- PERCENT_DISTRIB_95
  rownames(PERCENT_DISTRIB) <- NAMES_CONF_LIV
  colnames(PERCENT_DISTRIB) <- NAMES_DISTRIB
  PERCENT_DISTRIB <- round(PERCENT_DISTRIB, digits = 2)
```

```
  #MODELLI
  #99%
  n_99_sGARCH <- length(DF99[DF99 =='sGARCH'])
  n_99_eGARCH <- length(DF99[DF99 =='eGARCH'])
  n_99_gjrGARCH <- length(DF99[DF99 =='gjrGARCH'])
  n_99_apARCH <- length(DF99[DF99 =='apARCH'])
  n_99_iGARCH <- length(DF99[DF99 == 'iGARCH'])
  n_99_csGARCH <- length(DF99[DF99 == 'csGARCH'])

  PERCENT_MODEL_99 <- c(n_99_sGARCH/T99*100, n_99_eGARCH/T99*100,
                        n_99_gjrGARCH/T99*100, n_99_apARCH/T99*100,
                        n_99_iGARCH/T99*100, n_99_csGARCH/T99*100)
  #95%
  n_95_sGARCH <- length(DF95[DF95 =='sGARCH'])
  n_95_eGARCH <- length(DF95[DF95 =='eGARCH'])
  n_95_gjrGARCH <- length(DF95[DF95 =='gjrGARCH'])
  n_95_apARCH <- length(DF95[DF95 =='apARCH'])
  n_95_iGARCH <- length(DF95[DF95 == 'iGARCH'])
  n_95_csGARCH <- length(DF95[DF95 == 'csGARCH'])

  PERCENT_MODEL_95 <- c(n_95_sGARCH/T95*100, n_95_eGARCH/T95*100,
                        n_95_gjrGARCH/T95*100, n_95_apARCH/T95*100,
                        n_95_iGARCH/T95*100, n_95_csGARCH/T95*100)

  PERCENT_MODEL <- matrix(NA, 2, 6)
  PERCENT_MODEL[1,] <- PERCENT_MODEL_99
  PERCENT_MODEL[2,] <- PERCENT_MODEL_95
  rownames(PERCENT_MODEL) <- NAMES_CONF_LIV
  colnames(PERCENT_MODEL) <- NAMES_MODEL
  PERCENT_MODEL <- round(PERCENT_MODEL, digits = 2)

  lista <- list(PERCENT_FORECAST, PERCENT_DISTRIB, PERCENT_MODEL)
  names(lista) <- NAMES_F
  return(lista)
}


SPX_ANALISI <- ANALISI(SPX_ACC_99, SPX_ACC_95)
SPX_ANALISI$FORECAST
SPX_ANALISI$DISTRIBUZIONE
SPX_ANALISI$MODELLO

NKY_ANALISI <- ANALISI(NKY_ACC_99, NKY_ACC_95)
DAX_ANALISI <- ANALISI(DAX_ACC_99, DAX_ACC_95)
CAC_ANALISI <- ANALISI(CAC_ACC_99, CAC_ACC_95)
UKX_ANALISI <- ANALISI(UKX_ACC_99, UKX_ACC_95)

FILE_ANALISI <- list(SPX_ANALISI, NKY_ANALISI, DAX_ANALISI, CAC_ANALISI, UKX_ANALISI)
names(FILE_ANALISI) <- c('SPX', 'NKY', 'DAX', 'CAC', 'UKX')
save(FILE_ANALISI, file = 'FILE_ANALISI.RData')


FILE_ANALISI$SPX$FORECAST
FILE_ANALISI$SPX$DISTRIBUZIONE
FILE_ANALISI$SPX$MODELLO
FILE_ANALISI$NKY$FORECAST
FILE_ANALISI$NKY$DISTRIBUZIONE
FILE_ANALISI$NKY$MODELLO

FILE_ANALISI$DAX$FORECAST
FILE_ANALISI$DAX$DISTRIBUZIONE
FILE_ANALISI$DAX$MODELLO
FILE_ANALISI$CAC$FORECAST
FILE_ANALISI$CAC$DISTRIBUZIONE
FILE_ANALISI$CAC$MODELLO

FILE_ANALISI$UKX$FORECAST
FILE_ANALISI$UKX$DISTRIBUZIONE
FILE_ANALISI$UKX$MODELLO
```

Chapter 4: Markov-Switching GARCH Models

4.1 Brief preface to dynamic models

The use of Markov-switching models to capture the volatility dynamics of financial time series has grown considerably during past years, in part because they give rise to a plausible interpretation of nonlinearities the unpredictable behaviour of financial time series: this characteristic has long been a concern for econometricians, making it difficult to find appropriate models with a satisfactory fit.

The Markov-switching regime model is a popular approach, much in behalf of the way it takes the shifts in the time series behaviour into account. A Markov-Switching model is a nonlinear specification in which different states of the world affect the evolution of a time series. The dynamic properties depend on the present regime, with the regimes being realizations of a hidden Markov chain with a finite state space. Markov-Switching models were introduced to the econometric mainstream by Hamilton (1988, 1989) and continue to gain popularity — especially in financial time-series analysis.

The behaviour of the time series is characterized by multiple equations, decided by the different states of the model. What separates the Markov regime switching model from other switching models is that the switching mechanism is controlled by an unobservable variable that follows a hidden Markov chain. By Markov properties, the current value of the variable depends only on its immediate past value. This means that a structure in the series may prevail for a random period of time, before being replaced by another structure when a switching takes place. This way, the Markov regime switching model is able to capture more complex dynamic patterns.

Financial time series occasionally display dramatic breaks in their behaviour, due to e.g. financial crises. Therefore, the idea of the financial market finding itself in different states at different times becomes appealing. Furthermore, it has been found that financial time series exhibit some formalised facts which can advantageously be reproduced by a hidden Markov model.

This has made the Markov regime switching model one of the most popular nonlinear time series models in the literature.

Cai (1994) and Hamilton and Susmel (1994) are the first to apply the seminal idea of regime-switching parameters by Hamilton (1988, 1989 and 1990) into an ARCH specification in order to account for the possible presence of structural breaks. They use an ARCH specification instead of a GARCH to avoid problems of infinite path-dependence.

In particular, Hamilton and Susmel (1994) distinguish a low, moderate and high volatility regime in weekly stock return data, with the high-volatility regime being associated with economic recessions, while Maheu and McCurdy (2000) identify bull and bear markets and also find that volatility is much higher in the bear market.

Even the most basic Markov-switching model with constant regime parameters is capable of describing many of the typical characteristics of financial time series.

To illustrate this, consider the time series $\{y_t\}$ generated by

$$y_t = \eta_t \sigma_{\Delta t} + \mu_{\Delta t} \tag{30}$$

where $\eta_t \overset{iid}{\sim} N(0,1)$; and $\{\Delta_t\}$is a Markov chain with k-dimensional state space. The unconditional distribution of y_t is a k-component mixture of normals with the vector of mixing weights being equal to the stationary distribution of the Markov chain. It is well known that such mixture models can give rise to a skewed unconditional distribution if the regime means are different, and that the distribution is usually leptokurtic relative to the normal. Due to the dependence in the Markov chain, the model also gives rise to conditional heteroskedasticity.

It seems appropriate to clarify following distinction: a Markov-Switching Model is a regime-switching model in which the shifts between regimes evolve according to an unobserved Markov chain; a Regime-Switching Model is a parametric model of a time series in which parameters are allowed to take on different values in each of some fixed number of regimes.

4.2 Standard GARCH Models and Markov-Switching GARCH Models: a comparison

In GARCH-type models, the conditional volatility is driven by shocks in the observed time series. An alternative approach is to assume that volatility is driven by volatility-specific shocks. Recent studies show that volatility predictions by GARCH-type models may fail to capture the true variation in volatility in the case of regime changes in the volatility dynamics. A solution to this problem is to allow the parameters of the GARCH model to vary over time according to a latent discrete Markov process. This approach is called the Markov-switching GARCH (MSGARCH) model, which leads to volatility forecasts that can quickly adapt to variations in the unconditional volatility level.
The property that makes Markov Regime-Switching GARCH (heretofore MRS-GARCH) and GARCH models so different is given by their completely opposite representation of the concept of time-varying volatility. Actually, while GARCH models describe volatility as an ARMA process, thus incorporating innovations directly, the MRS-GARCH models keep the same structure for the volatility, adding the possibility of sudden jumps from the turbulent regime to the tranquil state and vice versa.

Analytically, let Y_t be the variable of interest. We assume that Y_t has zero mean and is not serially correlated, that is, the following moment conditions are assumed:

$$\mathrm{E}[Y_t] = 0 \text{ and } \mathrm{E}[Y_t Y_{t-l}] = 0, \text{ for } l \neq 0 \text{ and } t > 0.$$

The general Markov-switching GARCH specification can then be expressed as:

$$Y_t \mid (s_t = k, I_{t-1}) \sim D(0, \sigma_{k,t}^2, \xi_k) \tag{31}$$

where $\mathrm{D}(0, \sigma_{k,t}^2, \xi_k)$ is a continuous distribution with zero mean, time varying variance $\sigma_{k,t}^2$, and additional shape parameters gathered in the vector ξ_k.
The integer-valued stochastic variable s_t, defined on the discrete space{1,…,K}, characterizes the Markov-switching GARCH model. We define the standardized innovations as $\eta_{k,t} \equiv \frac{Y_t}{\sigma_{k,t}} \overset{iid}{\sim} \mathrm{D}(0,1,\xi_k)$.

4.3 Dynamics and independent states

The dynamic's specification refers to the assumption of a first-order Markov chain which characterizes tis models and the assumption of independent draws from a multinomial distribution.

We assume that s_t evolves according to an unobserved first-order ergodic homogeneous Markov chain with K x K transition probability matrix P:

$$\mathrm{P} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,K} \\ \vdots & \ddots & \vdots \\ p_{K,1} & \cdots & p_{K,K} \end{bmatrix}$$

where $p_{i,j} \equiv P[s_t = \mathrm{j} \mid s_{t-1} = \mathrm{i}]$ is the probability of a transition from state $s_{t-1} = \mathrm{i}$ to state $s_t = \mathrm{j}$. Obviously, the following constraints hold $0 < p_{i,j} < 1$, $\forall$i, j$\in${1,…,K} and $\Sigma_{j=1}^{K} p_{i,j} = 1, \forall \mathrm{i} \in \{1,\ldots,\mathrm{K}\}$. In the MSGARCH model, the conditional variances $\sigma_{k,t}^2$ for k = 1,…, K are assumed to follow K separate GARCH-type processes which evolve in parallel.

The independence's specification refers the Mixture of GARCH. In this case, we assume that s_t is sampled independently over time from a Multinomial distribution with support {1,…,K} and vector of probabilities: $\omega = (\omega_1, \ldots, \omega_k)^{\mathrm{T}}$, that is $P[s_t = \mathrm{k}] = \omega_k$

4.4 Fitting models with ML and MCMC estimations

In this chapter, we show the methodology to construct 72 different models with following parameters:

```
lista_modelli <- c('sGARCH', 'eGARCH','gjrGARCH')
lista_distribuzioni <- c('norm', 'std', 'ged', 'snorm', 'sstd', 'sged')
lista_AR <- c('AR_0', 'AR_1', 'AR_2', 'AR_3')
par_AR <- c(0, 1, 2, 3)
```

As previously stated, we can estimate each model through two methodologies:

- "FitML" is Maximum Likelihood estimation;
- "FitMCMC" is Bayesian estimation.

Below are the two functions that were built specifically to estimate the models with MSGARCH packages.

```
Stima_MS_ML <- function(x.ts, ar, model, dist) {

  x <- auto.arima(x.ts, max.p = ar, max.q = 0)
  x.AR <- x$residuals

  spec_MS  = CreateSpec(variance.spec = list(model= c(model)),
                        distribution.spec = list(distribution = dist),
                        switch.spec = list(do.mix = FALSE, k=2))

  fit.MS = FitML(spec = spec_MS, data = x.AR)

  return(fit.MS)
}
#Esempio
SPX <- Stima_MS_ML(returns.SPX, 3, 'gjrGARCH', 'sstd')
```

```
Stima_MS_MCMC <- function(x.ts, ar, model, dist) {

  x <- auto.arima(x.ts, max.p = ar, max.q = 0)
  x.AR <- x$residuals

  spec_MCMC  = CreateSpec(variance.spec = list(model= c(model)),
                          distribution.spec = list(distribution = dist),
                          switch.spec = list(do.mix = FALSE, k=2))

  fit.MCMC = FitMCMC(spec = spec_MCMC, data = x.AR)

  return(fit.MCMC)
}
#Esempio
SPX <- Stima_MS_MCMC(returns.SPX, 3, 'gjrGARCH', 'sstd')
```

Once created the "`Stima_MS_ML`" function, we estimate models with relative distributions on the errors, using the command "`CreateSpec()`".

Relevant informations of the function "`CreateSpec()`" are summarized with the "summary" method.

Here is shown the output regarding:

- the model, such as whether a Markov-switching or Mixture of GARCH has been selected;
- the GARCH-type specification within each regime;
- the number of variance parameters;
- the number of shape parameters in each regime.

We have selected models considering those present in the MSGARCH package and, based on the Table 5; we have subsequently created a mixture between the eGARCH model and the sged distribution and between gjrGARCH and the snorm. Here we report an example of a two-state MSGARCH model, where each regime is characterized by a different conditional volatility and a different conditional distribution.

```
> SPX <- Stima_MS_ML(returns.SPX, 1, c('eGARCH', 'gjrGARCH'), c('sged','snorm'))
> summary(SPX)
Specification type: Markov-switching
Specification name: eGARCH_sged gjrGARCH_snorm
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  -0.0921     0.0004   -260.1523    <1e-16
alpha1_1   0.1013     0.0002    505.2713    <1e-16
alpha2_1  -0.0719     0.0002   -456.0994    <1e-16
beta_1     0.9908     0.0000  26875.7159    <1e-16
nu_1       1.4894     0.0010   1557.4513    <1e-16
xi_1       0.9926     0.0004   2797.8035    <1e-16
alpha0_2   0.0001     0.0000    349.2748    <1e-16
alpha1_2   0.0004     0.0001      6.9878 1.396e-12
alpha2_2   0.9972     0.0000 528455.5287    <1e-16
beta_2     0.3751     0.0004   1037.4744    <1e-16
xi_2       0.4416     0.0012    368.2875    <1e-16
P_1_1      0.8848     0.0000 207899.2759    <1e-16
P_2_1      0.9992     0.0000 536212.6934    <1e-16
-----------------------------------------
Transition matrix:
      t+1|k=1 t+1|k=2
t|k=1  0.8848  0.1152
t|k=2  0.9992  0.0008
-----------------------------------------
Stable probabilities:
State 1 State 2
 0.8967  0.1033
-----------------------------------------
LL: 12723.7771
AIC: -25421.5542
BIC: -25340.2048
```

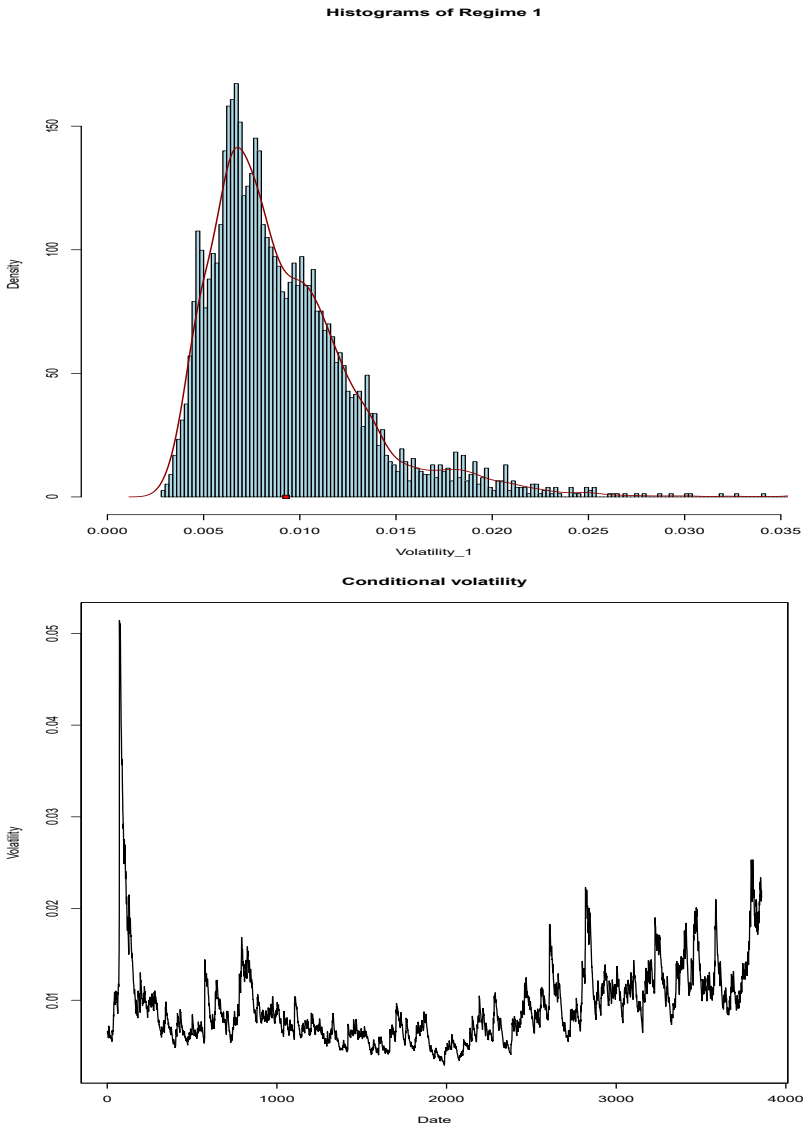
The output of "`summary()`" reports various information regarding model estimation. Estimated parameters are displayed along with significance levels according to their distribution. The summary also returns the unconditional distribution (Stable probabilities) of the Markov chain. Finally, the log-likelihood evaluated at its optimum together with Akaike and Bayesian information criteria are also reported. Note that the ML estimation results are ordered with respect to the unconditional variance in each regime, from lower to higher values, when regimes have the same model specification.

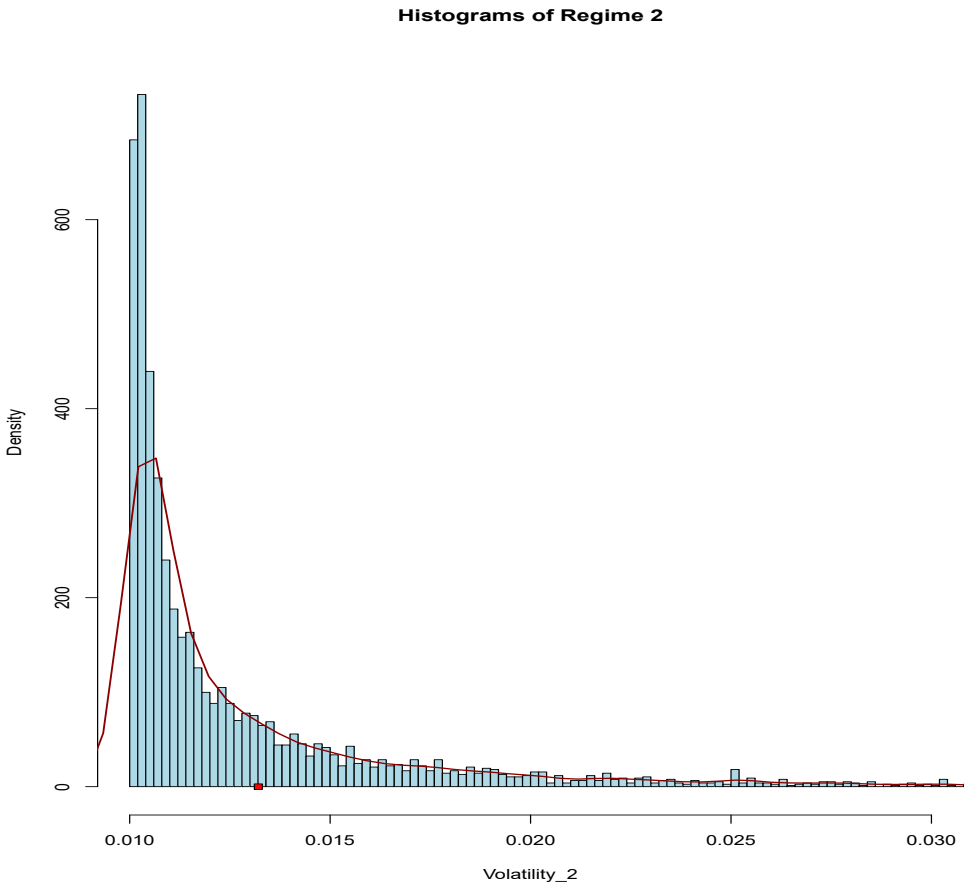
Parameter estimates indicate that the evolution of the volatility process is heterogeneous across the two regimes. Indeed, we first note that the two regimes report different unconditional volatility levels.

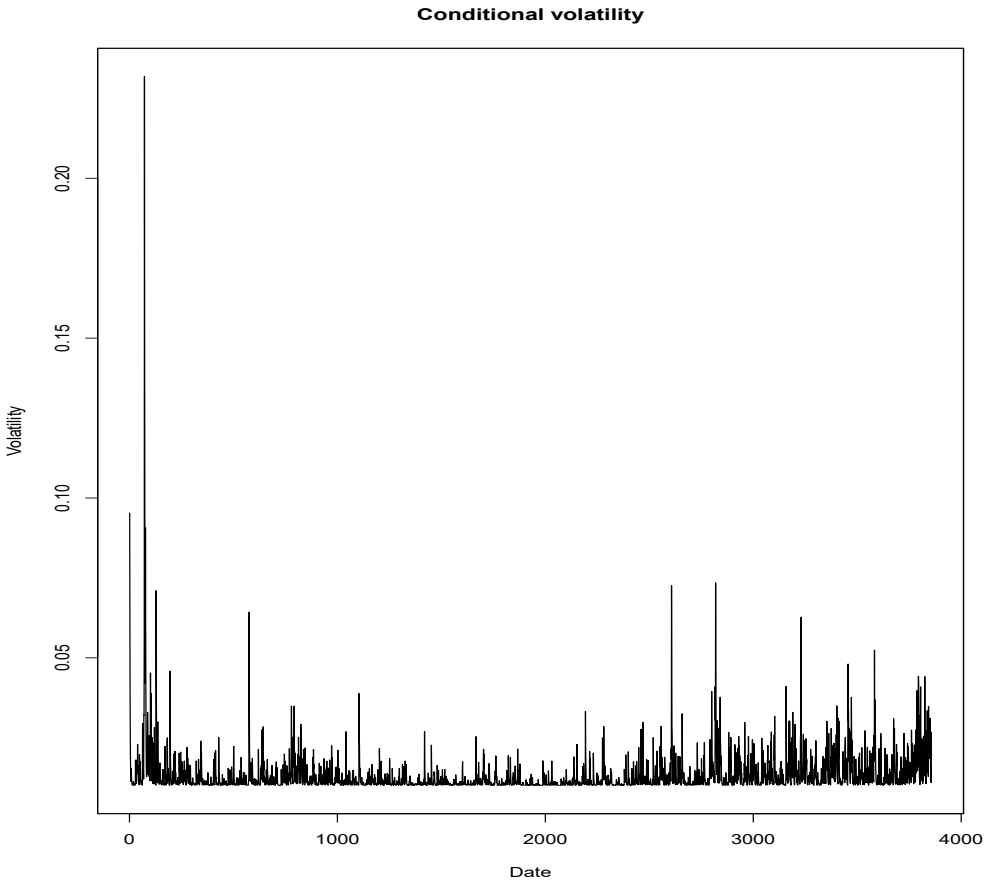
The two volatilities regime (high and low volatility), which graphics are shown below, are extracted from the "`StateFit_ML`" list.

Here we report the script related to the construction of graphics functions:

```
################ Two Regimes
StateFit_ML <- ExtractStateFit(SPX)
names(StateFit_ML)
st1 <- StateFit_ML[[1]]
st2 <- StateFit_ML[[2]]
Volatility_1 <- Volatility(object = st1)
Volatility_2 <- Volatility(object = st2)


#Volatility_1 #low
hist(Volatility_1, nclass = 300, freq = F, col = "light blue",
     main = "Histograms of Regime 1",
     xlim = c(0, 0.034), ylim = c(0, 180))
lines(density(Volatility_1, na.rm=TRUE),col=  "dark red", lwd = 1.5)
points(mean(Volatility_1), 0, pch = 22:25, bg = "red")
plot(Volatility_1)


#Volatililty_2 #high
hist(Volatility_2, nclass = 1000, freq = F, col = "light blue",
     main = "Histograms of Regime 2",
     xlim = c(0.010, 0.030), ylim = c(0, 750))
lines(density(Volatility_2, na.rm=TRUE),col=  "dark red", lwd = 1.5)
points(mean(Volatility_2), 0, pch = 22:25, bg = "red")
plot(Volatility_2)
```




Histograms of Regime 2
Density
0
200
400
600
0.010
0.015
0.020
0.025
0.030
Volatility_2


Conditional volatility
Volatility
0.05
0.10
0.15
0.20
0
1000
2000
3000
4000
Date

Now we build two functions. They will allow us to construct a single list with 72 different models, data from the computations of the previously defined parameters.

Maximum Likelihood method:

```
MS_ML_GARCH <- function(x.ts) {
  n <- 1
  List_Models <- list()
  Names_Models <- list()
  NM <- 1

  for (ar in par_AR) {
    for (md in lista_modelli) {
      for (di in lista_distribuzioni){

        Names_Models[n] <- paste('AR_', ar, md, di, sep = '.')
        fitGarch <- Stima_MS_ML(x.ts, ar, md, di)
        print(paste('Modello Stimato: ', Names_Models[n]))

        List_Models[n] <- list(fitGarch)
        n <- n + 1

      }
    }
  }
  names(List_Models) <- Names_Models
  return(List_Models)
}
#Esempio
SPX_MS_ML <- MS_ML_GARCH(returns.SPX)
SPX_MS_ML$AR_.1.gjrGARCH.sged
```

Bayesian method:

```
MS_MCMC_GARCH <- function(x.ts) {
  n <- 1
  List_Models <- list()
  Names_Models <- list()
  NM <- 1

  for (ar in par_AR) {
    for (md in lista_modelli) {
      for (di in lista_distribuzioni){

        Names_Models[n] <- paste('AR_', ar, md, di, sep = '.')
        fitGarch <- Stima_MS_MCMC(x.ts, ar, md, di)
        print(paste('Modello Stimato: ', Names_Models[n]))

        List_Models[n] <- list(fitGarch)
        n <- n + 1

      }
    }
  }
  names(List_Models) <- Names_Models
  return(List_Models)
}
#Esempio
SPX_MS_MCMC <- MS_MCMC_GARCH(returns.SPX)
```

After obtaining the lists of the 72 models, we have calculated the Value-at-Rick for the two confidence intervals, and considering the four forecast levels (500, 1000, 1500, 2000). The goal is to calculate the goodness of the estimated VaR.

In this way we will have a measure of comparison with the previous standard models. The function for goodness is the same as that seen in the previous chapters.

```
Bonta_MS <- function(vReT, vVaR, frcst) {

  R <- tail(vReT, n = frcst)
  V <- tail(vVaR, n = frcst)

  Ex.n <- length(R[R<V])
  Ex.p <- (Ex.n/frcst)*100

  Bonta <- 100 - Ex.p

  lista <- list(Bonta, Ex.p)
  names(lista) <- c('Bontà', 'Excedances')
  print('Bontà Stimata')
  return(lista)
}
```

We use the "`Risk()`" method to estimate the VaR and by calling the `"Bonta_MS"` function we calculate the desired vector.

```
VaR_MS <- function(x.ts, risk, frcst) {

  VaR_99 <- risk$VaR[,1]
  VaR_95 <- risk$VaR[,2]

  BNT_99 <- Bonta_MS(x.ts, VaR_99, frcst)
  BNT_95 <- Bonta_MS(x.ts, VaR_95, frcst)

  vettore99 <- c(BNT_99$Bontà, BNT_99$Excedances)
  vettore95 <- c(BNT_95$Bontà, BNT_95$Excedances)
  return(c(vettore99, vettore95))
}
```

Another part of the script consist to calculate the vector of DQ-Test for the both confidence levels.

```
DQ_Vet <- function(m, rk, fr) {
  X <- tail(m$data, fr)
  V99 <- tail(rk$VaR[,1], fr)
  V95 <- tail(rk$VaR[,2], fr)

  DQ.T99 <- try(Matrix_DQ(X, V99, 0.99), silent = TRUE)
  DQ.T95 <- try(Matrix_DQ(X, V95, 0.95), silent = TRUE)

  if (typeof(DQ.T99) == "character") {
    DQ.T99 <- c(0, 0)
  }else {
    DQ.T99 <- c(DQ.T99$Statistica, DQ.T99$`P-Value`)
  }

  if (typeof(DQ.T95) == "character") {
    DQ.T95 <- c(0, 0)
  }else {
    DQ.T95 <- c(DQ.T95$Statistica, DQ.T95$`P-Value`)
  }

  return(c(DQ.T99, DQ.T95))
}
```

Last function provides to return the vector that contains the statistic and p-values of UC and CC test. The function is "`Test_Vet()`".

It used three function called "`HitSequence()`", "`Kupiec()`" and "`Christoffersen()`", they are prebuilt in package GAS, written by Leopoldo Catania. We have downloaded one of the files of the package, then we have readjusted the function for our purpose.

Here the function readjusted:

```
HitSequence <- function(returns_X, VaR_X) {
  N = length(returns_X)
  Hit_X = numeric(N)
  Hit_X[which(returns_X <= VaR_X)] = 1L
  return(Hit_X)
}

Kupiec <- function(vY, vVaR, tau, fr) {
  Y <- tail(vY, fr)
  VaR <- tail(vVaR, fr)
  Hit <- HitSequence(Y, VaR)
  N = length(Hit)
  x = sum(Hit)
  rate = x/N
  test = -2 * log(((1 - tau)^(N - x) * tau^x)/((1 - rate)^(N - x) * rate^x))
  if (is.nan(test))
    test = -2 * ((N - x) * log(1 - tau) + x * log(tau) - (N - x) * log(1 - rate) - x * log(rate))
  # threshold = qchisq(alphaTest, df = 1)
  pvalue = 1 - pchisq(test, df = 1)

  LRpof = c(test, pvalue)
  names(LRpof) = c("Test", "Pvalue")
  return(LRpof)
}

Christoffersen <- function(vY, vVaR, tau, fr) {
  Y <- tail(vY, fr)
  VaR <- tail(vVaR, fr)
  Hit <- HitSequence(Y, VaR)
  n00 = n01 = n10 = n11 = 0
  N = length(Hit)
  for (i in 2:N) {
    if (Hit[i] == 0L & Hit[i - 1L] == 0L)
      n00 = n00 + 1
    if (Hit[i] == 0L & Hit[i - 1L] == 1L)
      n01 = n01 + 1
    if (Hit[i] == 1L & Hit[i - 1L] == 0L)
      n10 = n10 + 1
    if (Hit[i] == 1L & Hit[i - 1L] == 1L)
      n11 = n11 + 1
  }
  pi0 = n01/(n00 + n01)
  pi1 = n11/(n10 + n11)
  pi = (n01 + n11)/(n00 + n01 + n10 + n11)
  LRind = -2 * log(((1 - pi)^(n00 + n10) * pi^(n01 + n11))/((1 - pi0)^n00 * pi0^n01 * (1 - pi1)^n10 *
                                                              pi1^n11))
  if (is.nan(LRind))
    LRind = -2 * ((n00 + n10) * log(1 - pi) + (n01 + n11) * log(pi) - n00 * log(1 - pi0) - n01 *
                    log(pi0) - n10 * log(1 - pi1) - n11 * log(pi1))
  LRpof = Kupiec(vY, vVaR, tau, fr)["Test"]
  LRcc = LRpof + LRind
  pvalue = 1 - pchisq(LRcc, df = 2L)
  LRcc = c(LRcc, pvalue)
  names(LRcc) = c("Test", "Pvalue")
  return(LRcc)
}
```

The next is our function that it utilizes this three functions.

```
Test_Vet <- function(m, rk, fr) {
  X <- m$data
  V99 <- rk$VaR[,1]
  V95 <- rk$VaR[,2]
  #UC
  UC.99 <- Kupiec(X, V99, 0.01, fr)
  UC.95 <- Kupiec(X, V95, 0.05, fr)
  #CC
  CC.99 <- Christoffersen(X, V99, 0.01, fr)
  CC.95 <- Christoffersen(X, V95, 0.05, fr)

  return(c(UC.99, UC.95, CC.99, CC.95))
}
```

Now we can show the last function that calculate all data frame for each model. This is the function that allows us to iterate the entire list of models.

This function takes the list as input, chooses the first model and invokes the function for calculating the goodness, unconditional and conditional coverage and dynamic quantile test. After which, it inserts this vector as the first row of the final data frame. The function will repeat this operation until it finds models in the list.

```
DF_BONTA_MS <- function(x.ts, FfG) {
  Nomi_List <- list()
  Nomi_List_FfG <- names(FfG)
  FN <- 1
  I <- 1
  ar <- 0
  for (m in FfG) {
    print(paste('Analisi modello:', Nomi_List_FfG[FN]))
    risk <- Risk(m, alpha = c(0.01, 0.05), do.its = TRUE)
    for (fr in frcst) {
      print(I)
      vB <- VaR_MS(x.ts, risk, fr)
      NOME <- paste(fr, Nomi_List_FfG[FN], sep = '_')
      Nomi_List[I] <- NOME

      vDQ <- DQ_Vet(m, risk, fr)

      vT <- Test_Vet(m, risk, fr)

      V <- c(vT, vDQ, vB)

      if (I == 1) {
        DF0 <- as.data.frame(V)
        I <- I + 1
      }else {
        DF0[,I] <- V
        I <- I + 1
      }
      print(paste('Modello:', NOME, 'inserito nella matrice', sep = ' '))
    }
    FN <- FN + 1
  }

  colnames(DF0) <- Nomi_List

  rownames(DF0) <- c('UC-Stat 99%', 'UC-PValue 99%', 'UC-Stat 95%', 'UC-PValue 95%',
                     'CC-Stat 99%', 'CC-PValue 99%', 'CC-Stat 95%', 'CC-PValue 95%',
                     'DQ-Test 99%', 'P-Value 99%', 'DQ-Test 95%', 'P-Value 95%',
                     'Bontà 99%', 'ECC 99%', 'Bontà 95%', 'ECC 95%')

  return(t(DF0))
}
```

The same operation is repeated to lease the models using the Bayesian method. The last three functions can also be used for models estimated with "fitMCMC".

By executing the functions described so far, we obtain two results:

- the file named "NameIndex_MS_ML" contains the 72 estimated models;
- the file " NameIndex_BT_ML" is a matrix of 288 lines (72x4) and 4 columns.

We have selected the best models choosing among the highest goodness *("Bontà"),* for 99% and 95% of confident intervals.

Below we show the tables of the best models for each share index, and we compare the two methods of analysis: ML and MCMC.

Table 7: Best models with Maximum Likelihood estimation

SPX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	500.AR(1)eGARCH_snorm	0.010125	99,9%	500.AR(0)eGARCH_snorm	0.060211	97.03%	1500.AR(0)gjrGARCH_norm	3.741405	71.16%	2000.AR(2)sGARCH_snorm	95.65%	4.35%
	500.AR(1)eGARCH_sstd	0.010125	99,9%	500.AR(0)eGARCH_ged	0.060211	97.03%	1500.AR(1)gjrGARCH_norm	3.741405	71.16%	2000.AR(3)sGARCH_snorm	95.65%	4.35%
	2000.AR(2)gjrGARCH_ged	0.010493	91.84%	500.AR(1)eGARCH_ged	0.060211	97.03%	1000.AR(3)sGARCH_ged	4.214639	64.77%	500.AR(3)eGARCH_sged	95.6%	4.4%
	2000.AR(0)eGARCH_ged	0.010493	91.84%	500.AR(1)eGARCH_sstd	0.060211	97.03%	1000.AR(0)gjrGARCH_sstd	4.226057	64.61%	2000.AR(3)eGARCH_sged	95.55%	4.45%
99%	500.AR(1)eGARCH_sstd	0.943116	33.15%	500.AR(0)eGARCH_snorm	0.101216	95.07%	500.AR(0)eGARCH_norm	0.215045	99.98%	1000.AR(3)sGARCH_sged	99.5%	0.5%
	500.AR(1)eGARCH_ged	0.943116	33.15%	500.AR(3)eGARCH_snorm	0.101216	95.07%	500.AR(1)eGARCH_norm	0.215045	99.98%	1000.AR(2)sGARCH_sged	99.5%	0.5%
	2000.AR(2)gjrGARCH_ged	0.049692	82.36%	500.AR(3)gjrGARCH_snorm	0.101216	95.07%	500.AR(2)eGARCH_norm	0.215342	99.98%	1000.AR(0)sGARCH_sstd	99.5%	0.5%
	500.AR(0)eGARCH_ged	0.943116	33.15%	500.AR(3)sGARCH_snorm	0.101216	95.06%	500.AR(0)gjrGARCH_norm	0.216125	99.98%	1000.AR(3)sGARCH_sstd	99.5%	0.5%

NKY

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	1000.AR(0)sGARCH_snorm	0.010125	99,9%	1000.AR(2)sGARCH_sged	0.085109	95.83%	500.AR(2)gjrGARCH_std	0.786821	99.24%	1000.AR(0)eGARCH_snorm	95.4%	4.6%
	1000.AR(1)sGARCH_snorm	0.010125	99,9%	1000.AR(3)sGARCH_sged	0.085109	95.83%	500.AR(3)gjrGARCH_std	0.786821	99.24%	1000.AR(1)eGARCH_snorm	95.4%	4.6%
	1000.AR(2)eGARCH_sstd	0.010125	99,9%	1000.AR(2)sGARCH_snorm	0.473742	78.91%	500.AR(2)gjrGARCH_sstd	1.393899	96.62%	2000.AR(2)eGARCH_snorm	95.35%	4.65%
	1000.AR(3)eGARCH_sstd	0.010125	99,9%	1000.AR(3)sGARCH_snorm	0.473742	78.91%	500.AR(3)gjrGARCH_sstd	1.393899	96.62%	2000.AR(3)eGARCH_snorm	95.35%	4.65%
99%	1000.AR(0)sGARCH_snorm	0.830571	36.21%	500.AR(2)gjrGARCH_snorm	0.101216	95.07%	500.AR(2)sGARCH_snorm	0.264954	99.97%	500.AR(3)gjrGARCH_std	99.8%	0.2%
	1000.AR(1)sGARCH_snorm	0.830571	36.21%	500.AR(3)gjrGARCH_snorm	0.101216	95.07%	500.AR(3)sGARCH_snorm	0.264954	99.96%	500.AR(2)gjrGARCH_std	99.8%	0.2%
	1000.AR(2)eGARCH_sstd	1.886232	16.96%	500.AR(0)eGARCH_snorm	0.101216	95.06%	500.AR(0)sGARCH_snorm	0.279679	99.96%	500.AR(2)gjrGARCH_sstd	99.8%	0.2%
	1000.AR(3)eGARCH_sstd	1.886232	16.96%	500.AR(1)eGARCH_norm	0.101216	95.06%	500.AR(1)sGARCH_snorm	0.279678	99.96%	500.AR(3)gjrGARCH_sstd	99.8%	0.2%

DAX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(0)sGARCH_snorm	0.041842	83.79%	2000.AR(0)sGARCH_snorm	0.050880	97.49%	500.AR(0)sGARCH_norm	1.055220	98.34%	2000.AR(0)eGARCH_ged	94.85%	5.15%
	2000.AR(1)sGARCH_snorm	0.041842	83.79%	2000.AR(1)sGARCH_snorm	0.050880	97.49%	500.AR(1)sGARCH_norm	1.055220	98.34%	2000.AR(1)eGARCH_ged	94.85%	5.15%
	2000.AR(2)sGARCH_snorm	0.041842	83.79%	2000.AR(2)sGARCH_snorm	0.050880	97.49%	500.AR(2)sGARCH_norm	1.055220	98.34%	2000.AR(2)eGARCH_ged	94.85%	5.15%
	2000.AR(3)sGARCH_snorm	0.041842	83.79%	2000.AR(3)sGARCH_snorm	0.050880	97.49%	500.AR(3)sGARCH_norm	1.055220	98.34%	2000.AR(3)eGARCH_ged	94.85%	5.15%
99%	2000.AR(0)gjrGARCH_sstd	0.051360	82.07%	500.AR(0)sGARCH_norm	0.101216	95.07%	500.AR(0)gjrGARCH_norm	1.539248	95.68%	500.AR(0)gjrGARCH_ged	99.8%	0.2%
	2000.AR(1)gjrGARCH_sstd	0.051360	82.07%	500.AR(1)sGARCH_norm	0.101216	95.07%	500.AR(1)gjrGARCH_norm	1.539248	95.68%	500.AR(1)gjrGARCH_ged	99.8%	0.2%
	2000.AR(1)eGARCH_std	0.051360	82.07%	500.AR(2)sGARCH_norm	0.101216	95.06%	500.AR(2)gjrGARCH_norm	1.539248	95.68%	500.AR(2)gjrGARCH_ged	99.8%	0.2%
	2000.AR(2)eGARCH_std	0.051360	82.07%	500.AR(3)sGARCH_norm	0.101216	95.06%	500.AR(3)gjrGARCH_norm	1.539248	95.68%	500.AR(0)gjrGARCH_sstd	99.8%	0.2%

CAC

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(0)sGARCH_snorm	0.010560	91.82%	2000.AR(0)sGARCH_snorm	0.012674	99.37%	2000.AR(0)gjrGARCH_norm	3.103646	79.57%	2000.AR(0)eGARCH_snorm	95.25%	4.75%
	2000.AR(1)sGARCH_snorm	0.010560	91.82%	2000.AR(1)sGARCH_snorm	0.012674	99.37%	2000.AR(1)gjrGARCH_norm	3.103646	79.57%	2000.AR(1)eGARCH_snorm	95.25%	4.75%
	2000.AR(0)eGARCH_sged	0.010559	91.81%	2000.AR(2)sGARCH_snorm	0.012673	99.37%	2000.AR(2)gjrGARCH_norm	3.103646	79.57%	2000.AR(2)eGARCH_snorm	95.25%	4.75%
	2000.AR(1)eGARCH_sged	0.010559	91.81%	2000.AR(3)sGARCH_snorm	0.012673	99.37%	2000.AR(3)gjrGARCH_norm	3.103646	79.57%	2000.AR(3)eGARCH_snorm	95.25%	4.75%
99%	2000.AR(0)gjrGARCH_sged	0.051360	82.07%	500.AR(0)sGARCH_norm	0.281518	86.87%	500.AR(0)sGARCH_norm	1.025144	98.46%	500.AR(0)gjrGARCH_snorm	99.4%	0.6%
	2000.AR(1)gjrGARCH_sged	0.051360	82.07%	500.AR(0)eGARCH_norm	0.281518	86.87%	500.AR(1)sGARCH_norm	1.025144	98.46%	500.AR(1)sGARCH_std	99.4%	0.6%
	2000.AR(2)gjrGARCH_sged	0.051359	82.07%	500.AR(1)sGARCH_norm	0.281517	86.87%	500.AR(2)sGARCH_norm	1.025144	98.46%	500.AR(1)sGARCH_ged	99.4%	0.6%
	2000.AR(3)gjrGARCH_sged	0.051359	82.07%	500.AR(1)eGARCH_norm	0.281517	86.87%	500.AR(3)sGARCH_norm	1.025144	98.46%	500.AR(2)gjrGARCH_ged	99.4%	0.6%

UKX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(1)eGARCH_snorm	0.010560	91.82%	2000.AR(1)eGARCH_snorm	0.012674	99.37%	1500.AR(0)eGARCH_norm	1.748615	94.13%	2000.AR(2)sGARCH_norm	94.8%	5.2%
	2000.AR(2)eGARCH_snorm	0.010560	91.82%	2000.AR(1)gjrGARCH_snorm	0.199477	90.51%	1500.AR(0)eGARCH_std	3.853291	69.65%	2000.AR(2)sGARCH_ged	94.75%	5.25%
	2000.AR(0)eGARCH_snorm	0.042374	83.69%	2000.AR(1)gjrGARCH_sged	0.218629	89.65%	1500.AR(0)eGARCH_ged	3.903682	68.97%	2000.AR(3)sGARCH_ged	94.75%	5.25%
	2000.AR(1)gjrGARCH_snorm	0.042374	83.69%	2000.AR(0)eGARCH_sged	0.323143	85.08%	500.AR(2)sGARCH_norm	4.181635	65.21%	2000.AR(0)gjrGARCH_norm	94.7%	5.3%
99%	500.AR(1)eGARCH_ged	0.010125	99,9%	500.AR(1)eGARCH_ged	0.101216	95.07%	2000.AR(1)sGARCH_norm	1.611140	95.18%	1000.AR(0)eGARCH_sged	99.4%	0.6%
	500.AR(1)eGARCH_snorm	0.010125	99,9%	500.AR(1)eGARCH_snorm	0.101216	95.06%	1500.AR(1)sGARCH_norm	1.611681	95.17%	2000.AR(0)sGARCH_sged	99.3%	0.7%
	500.AR(2)eGARCH_sstd	0.010125	99,9%	500.AR(2)eGARCH_sstd	0.101216	95.06%	1000.AR(0)gjrGARCH_norm	1.680625	94.66%	2000.AR(2)sGARCH_sged	99.3%	0.7%
	500.AR(2)gjrGARCH_sstd	0.010125	99,9%	500.AR(2)gjrGARCH_sstd	0.101216	95.06%	1000.AR(1)sGARCH_norm	1.803905	93.68%	1000.AR(2)eGARCH_sstd	99.3%	0.7%

Table 8: Best models with Bayesian estimation

SPX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	500.AR(1)eGARCH_sstd	2.459194	11.68%	500.AR(0)eGARCH_snorm	0.060211	97.03%	1500.AR(0)gjrGARCH_norm	3.763635	70.86%	2000.AR(2)sGARCH_snorm	95.6%	4.4%
	500.AR(1)eGARCH_ged	0.010125	99.9%	500.AR(0)eGARCH_ged	0.060211	97.03%	1500.AR(1)_gjrGARCH_norm	3.750385	71.04%	2000.AR(3)sGARCH_snorm	95.65%	4.35%
	2000.AR(2)gjrGARCH_ged	0.093853	75.93%	500.AR(1)eGARCH_ged	0.060211	97.03%	1000.AR(3)sGARCH_ged	5.753464	45.14%	500.AR(3)eGARCH_sged	93%	7%
	500.AR(0)eGARCH_ged	0.010125	99.9%	500.AR(1)eGARCH_sstd	3.355909	18.68%	1000.AR(0)gjrGARCH_sstd	4.476733	61.24%	2000.AR(3)eGARCH_sged	95.05%	4.95%
99%	500.AR(1)eGARCH_sstd	0.943116	33.15%	500.AR(0)eGARCH_snorm	0.101216	95.07%	500.AR(0)eGARCH_norm	0.215809	99.98%	1000.AR(3)sGARCH_sged	99.5%	0.5%
	500.AR(1)eGARCH_ged	0.943116	33.15%	500.AR(3)eGARCH_snorm	0.101216	95.07%	500.AR(1)eGARCH_norm	0.215069	99.98%	1000.AR(2)sGARCH_sged	99.5%	0.5%
	2000.AR(2)gjrGARCH_ged	0.010125	99.9%	500.AR(3)gjrGARCH_snorm	0.101216	95.06%	500.AR(2)eGARCH_norm	0.216139	99.98%	1000.AR(0)sGARCH_sstd	99.4%	0.6%
	500.AR(0)eGARCH_ged	0.943116	33.15%	500.AR(3)sGARCH_snorm	0.101216	95.06%	500.AR(0)gjrGARCH_norm	0.215777	99.98%	1000.AR(3)sGARCH_sstd	99.5%	0.5%

NKY

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	1000.AR(0)sGARCH_snorm	0.020921	88.50%	1000.AR(2)sGARCH_sged	0.199764	90.49%	500.AR(2)gjrGARCH_std	0.798973	99.21%	1000.AR(0)eGARCH_snorm	95.4%	4.6%
	1000.AR(1)sGARCH_snorm	0.020921	88.49%	1000.AR(3)sGARCH_sged	0.199764	90.49%	500.AR(3)gjrGARCH_std	0.792737	99.27%	1000.AR(1)eGARCH_snorm	95.4%	4.6%
	1000.AR(2)eGARCH_sstd	0.185988	66.68%	1000.AR(2)sGARCH_snorm	0.473742	78.91%	500.AR(2)gjrGARCH_sstd	3.362658	76.21%	2000.AR(2)eGARCH_snorm	95.4%	4.6%
	1000.AR(3)eGARCH_sstd	1.616237	20.36%	1000.AR(3)sGARCH_snorm	0.473742	78.91%	500.AR(3)gjrGARCH_sstd	1.334089	96.97%	2000.AR(3)eGARCH_snorm	95.4%	4.6%
99%	500.AR(2)gjrGARCH_snorm	0.010125	99.9%	500.AR(2)gjrGARCH_snorm	0.101216	95.07%	500.AR(2)sGARCH_snorm	0.263118	99.97%	500.AR(3)gjrGARCH_std	99.8%	0.2%
	500.AR(3)gjrGARCH_snorm	0.010125	99.9%	500.AR(3)gjrGARCH_snorm	0.101216	95.07%	500.AR(3)sGARCH_snorm	0.272158	99.96%	500.AR(2)gjrGARCH_std	99.8%	0.2%
	1000.AR(0)eGARCH_std	0.104521	74.65%	500.AR(0)eGARCH_snorm	0.101216	95.06%	500.AR(0)sGARCH_snorm	1.293613	97.20%	500.AR(2)gjrGARCH_sstd	99.8%	0.2%
	1000.AR(0)eGARCH_sstd	2.189248	13.90%	500.AR(1)eGARCH_norm	0.101216	95.06%	500.AR(1)sGARCH_snorm	1.287865	97.23%	500.AR(3)gjrGARCH_sstd	99.8%	0.2%

DAX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(0)sGARCH_snorm	0.041842	83.79%	2000.AR(0)sGARCH_snorm	0.050880	97.49%	500.AR(0)sGARCH_norm	1.063943	98.31%	2000.AR(0)eGARCH_ged	94.95%	5.05%
	2000.AR(1)sGARCH_snorm	0.041842	83.79%	2000.AR(1)sGARCH_snorm	0.050880	97.48%	500.AR(1)sGARCH_norm	1.062726	98.31%	2000.AR(1)eGARCH_ged	94.85%	5.15%
	2000.AR(2)sGARCH_snorm	0.041843	83.79%	2000.AR(2)sGARCH_snorm	0.050880	97.49%	500.AR(2)sGARCH_norm	1.052929	98.35%	2000.AR(2)eGARCH_ged	94.95%	5.05%
	2000.AR(3)sGARCH_snorm	0.041842	83.79%	2000.AR(3)sGARCH_snorm	0.050880	97.48%	500.AR(3)sGARCH_norm	1.067239	98.29%	2000.AR(3)eGARCH_ged	94.9%	5.1%
99%	2000.AR(0)gjrGARCH_sstd	0.209041	64.75%	500.AR(0)sGARCH_norm	0.101216	95.07%	500.AR(0)gjrGARCH_norm	1.530475	95.75%	500.AR(0)gjrGARCH_ged	99.8%	0.2%
	2000.AR(1)gjrGARCH_sstd	0.209041	64.75%	500.AR(1)sGARCH_norm	0.101216	95.07%	500.AR(1)gjrGARCH_norm	1.530836	95.74%	500.AR(1)gjrGARCH_ged	99.8%	0.2%
	2000.AR(1)eGARCH_std	0.051360	82.07%	500.AR(2)sGARCH_norm	0.101216	95.06%	500.AR(2)gjrGARCH_norm	1.531219	95.74%	500.AR(2)gjrGARCH_ged	99.8%	0.2%
	2000.AR(2)eGARCH_std	0.051360	82.07%	500.AR(3)sGARCH_norm	0.101216	95.06%	500.AR(3)gjrGARCH_norm	1.531602	95.73%	500.AR(0)gjrGARCH_sstd	99.8%	0.2%

CAC

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(0)sGARCH_snorm	0.010560	91.82%	2000.AR(0)sGARCH_snorm	0.012674	99.37%	2000.AR(0)gjrGARCH_norm	2.969361	81.27%	2000.AR(0)eGARCH_snorm	95.25%	4.75%
	2000.AR(1)sGARCH_snorm	0.010560	91.82%	2000.AR(1)sGARCH_snorm	0.012674	99.37%	2000.AR(1)gjrGARCH_norm	3.058222	80.15%	2000.AR(1)eGARCH_snorm	95.25%	4.75%
	2000.AR(0)eGARCH_sged	0.386352	53.42%	2000.AR(2)sGARCH_snorm	0.012673	99.37%	2000.AR(2)gjrGARCH_norm	3.055319	80.19%	2000.AR(2)eGARCH_snorm	95.25%	4.75%
	2000.AR(1)eGARCH_sged	0.166335	68.34%	2000.AR(3)sGARCH_snorm	0.0126736	99.37%	2000.AR(3)gjrGARCH_norm	3.058185	80.15%	2000.AR(3)eGARCH_snorm	95.25%	4.75%
99%	2000.AR(0)gjrGARCH_sged	0.051360	82.07%	500.AR(0)sGARCH_norm	0.101216	95.07%	500.AR(0)sGARCH_norm	1.011163	98.52%	500.AR(0)gjrGARCH_snorm	99.4%	0.6%
	2000.AR(1)gjrGARCH_sged	0.051359	82.07%	500.AR(0)eGARCH_norm	0.281518	86.87%	500.AR(1)sGARCH_norm	1.024844	98.47%	500.AR(1)sGARCH_std	99.4%	0.6%
	2000.AR(2)gjrGARCH_sged	0.759521	38.35%	500.AR(1)sGARCH_norm	0.101216	95.07%	500.AR(2)sGARCH_norm	1.009595	98.52%	500.AR(1)sGARCH_ged	99.4%	0.6%
	2000.AR(3)gjrGARCH_sged	0.051359	82.07%	500.AR(1)eGARCH_norm	0.281518	86.87%	500.AR(3)sGARCH_norm	1.004846	98.54%	500.AR(2)gjrGARCH_ged	99.4%	0.6%

UKX

	UC_test			***CC_test***			***DQ_test***			***BONTA'***		
Levels	Models	Statistic	P-value	Models	Statistic	P-value	Models	Statistic	P-value	Models	P-value	Excess
95%	2000.AR(1)eGARCH_snorm	0.010560	91.82%	2000.AR(1)eGARCH_snorm	0.012674	99.37%	1500.AR(0)eGARCH_norm	1.709081	94.44%	2000.AR(2)sGARCH_norm	94.8%	5.2%
	2000.AR(2)eGARCH_snorm	0.010560	91.81%	2000.AR(1)gjrGARCH_snorm	0.263544	87.65%	1500.AR(0)eGARCH_std	3.470961	74.78%	2000.AR(2)sGARCH_ged	94.85%	5.15%
	2000.AR(0)eGARCH_snorm	0.042374	83.69%	2000.AR(1)gjrGARCH_sged	0.883464	64.29%	1500.AR(0)eGARCH_ged	3.849287	69.71%	2000.AR(3)sGARCH_ged	94.75%	5.25%
	2000.AR(1)gjrGARCH_snorm	0.170592	67.96%	2000.AR(0)eGARCH_sged	0.208433	90.10%	500.AR(2)sGARCH_norm	4.181086	65.22%	2000.AR(0)gjrGARCH_norm	94.8%	5.2%
99%	500.AR(1)eGARCH_ged	0.010125	99.9%	500.AR(1)eGARCH_ged	0.101216	95.07%	2000.AR(1)sGARCH_norm	6.103839	41.17%	1000.AR(0)eGARCH_sged	98.6%	1.4%
	500.AR(1)eGARCH_snorm	0.010125	99.9%	500.AR(1)eGARCH_snorm	0.101216	95.06%	1500.AR(1)sGARCH_norm	5.721976	45.50%	2000.AR(0)sGARCH_sged	99.35%	0.65%
	500.AR(2)eGARCH_sstd	2.612571	10.60%	500.AR(2)eGARCH_sstd	2.943201	22.96%	1000.AR(0)gjrGARCH_norm	1.733888	94.25%	2000.AR(2)sGARCH_sged	99.3%	0.7%
	500.AR(2)gjrGARCH_sstd	0.216870	64.14%	500.AR(2)gjrGARCH_sstd	0.281518	86.87%	1000.AR(1)sGARCH_norm	12.159239	5.85%	1000.AR(2)eGARCH_sstd	98.3%	1.7%

Table 9: Analysis of a model with the three estimation models

SPX

		UC_test		*CC_test*		*DQ_test*		*BONTA'*	
Levels	***2000AR(2)sGARCH_norm***	Statistic	P-value	Statistic	P-value	Statistic	P-value	P-value	Excess
	MSGARCH_ML	1.087647	29.70%	1.088382	58.03%	14.593220	2.37%	95.65%	4.35%
95%	MSGARCH_MCMC	0.878021	34.87%	0.883464	64.29%	14.239989	2.71%	95.6%	4.4%
	RUGARCH_Garch	0.093853	75.93%	0.481845	78.59%	11.342123	7.84%	94.85%	5.15%
Levels	***1000AR(3)sGARCH_sged***								
	MSGARCH_ML	4.813361	2.82%	4.817377	8.99%	3.257706	77.59%	99.8%	0.2%
99%	MSGARCH_MCMC	4.813361%	2.82%	4.817377	8.99%	3.257608	77.59%	99.8%	0.2%
	RUGARCH_Garch	2.352982	12.50%	2.369079	30.59%	2.123652	90.80%	99.6%	0.4%

NKY

		UC_test		*CC_test*		*DQ_test*		*BONTA'*	
Levels	***1000AR(0)eGARCH_snorm***	Statistic	P-value	Statistic	P-value	Statistic	P-value	P-value	Excess
	MSGARCH_ML	0.345711	55.66%	4.788150	9.13%	10.727594	9.72%	95.4%	4.6%
95%	MSGARCH_MCMC	0.345711	55.66%	4.788150	9.12%	10.722607	9.73%	95.4%	4.6%
	RUGARCH_Garch	0.010125	99.9%	5.271144	7.17%	12.535435	5.10%	95%	5%
Levels	***500AR(3)gjrGARCH_std***								
	MSGARCH_ML	0.943116	33.15%	0.979407	61.28%	0.913136	98.87%	99.8%	0.2%
99%	MSGARCH_MCMC	0.943116	33.15%	0.979407	61.28%	0.909296	98.88%	99.8%	0.2%
	RUGARCH_Garch	2.352982	12.50%	2.369079	30.59%	0.961339	98.70%	99.6%	0.4%

DAX

		UC_test		*CC_test*		*DQ_test*		*BONTA'*	
Levels	***2000AR(0)eGARCH_ged***	Statistic	P-value	Statistic	P-value	Statistic	P-value	P-value	Excess
	MSGARCH_ML	0.093853	75.93%	0.644726	72.44%	10.452522	10.68%	94.85%	5.15%
95%	MSGARCH_MCMC	0.010493	91.84%	0.719140	69.80%	11.062856	8.65%	94.95%	5.05%
	RUGARCH_Garch	3.968443	4.636100	3.975010	13.70%	85.139688	0	94%	6%
Levels	***500AR(0)gjrGARCH_ged***								
	MSGARCH_ML	4.813361	2.82%	4.817377	8.99%	3.257706	77.59%	99.8%	0.2%
99%	MSGARCH_MCMC	4.813361	2.82%	4.8173767	8.99%	3.257608	77.59%	99.8%	0.2%
	RUGARCH_Garch	2.352982	12.50%	2.369079	30.59%	2.123652	90.80%	99.6%	0.4%

CAC

		UC_test		*CC_test*		*DQ_test*		*BONTA'*	
Levels	***2000AR(0)eGARCH_snorm***	Statistic	P-value	Statistic	P-value	Statistic	P-value	P-value	Excess
	MSGARCH_ML	0.267420	60.52%	1.577946	45.43%	8.231765	22.16%	95.25%	4.75%
95%	MSGARCH_MCMC	0.267420	60.51%	1.577946	45.43%	8.155218	22.69%	95.25%	4.75%
	RUGARCH_Garch	1.461575	22.67%	1.955648	37.61%	7.157329	30.65%	94.4%	5.6%
Levels	***500AR(0)gjrGARCH_snorm***								
	MSGARCH_ML	0.943116	33.15%	0.979407	61.28%	1.933342	92.57%	99.4%	0.6%
99%	MSGARCH_MCMC	0.943116	33.15%	0.979407	61.28%	1.918297	92.71%	99.4%	0.6%
	RUGARCH_Garch	0.189880	66.30%	0.335928	84.54%	1.952064	92.41%	98.8%	1.2%

UKX

		UC_test		*CC_test*		*DQ_test*		*BONTA'*	
Levels	***2000AR(2)sGARCH_norm***	Statistic	P-value	Statistic	P-value	Statistic	P-value	P-value	Excess
	MSGARCH_ML	0.259104	61.07%	0.672872	71.43%	10.987235	8.88%	94.8%	5.2%
95%	MSGARCH_MCMC	0.166335	68.34%	0.646099	72.39%	10.769937	9.58%	94.8%	5.2%
	RUGARCH_Garch	Inf	0	Inf	0	62.948023	0	87.85%	12.15%
Levels	***1000AR(0)eGARCH_sged***								
	MSGARCH_ML	1.886232	16.96%	1.958740	37.55%	5.663274	46.20%	99.4%	0.6%
99%	MSGARCH_MCMC	1.437406	23.06%	1.835389	39.94%	18.723136	0.47%	98.6%	1.4%
	RUGARCH_Garch	0.104520	74.65%	0.268159	87.45%	3.743300	71.14%	99.1%	0.9%

We show below the tables that compare nine different statistically significant models which have passed, at the same time, the three test we used and which are compared on the three estimation methodologies (Rugarch, Maximum Likelihood, Bayesian).

SPX

		RUGARCH _Garch		MSGARCH _ML		MSGARCH _MCMC	
Levels	Models	Bontà	Excess	Bontà	Excess	Bontà	Excess
	500.AR(0)sGARCH(1,1)sged	93.6%	6.4%	94.4%	5.6%	94.4%	5.6%
	500.AR(0)gjrGARCH(1,1)sged	94.6%	5.4%	94.8%	5.2%	94.8%	5.2%
	500.AR(1)eGARCH(1,1)norm	94.4%	5.6%	94.8%	5.2%	94.6%	5.4%
95%	500.AR(2)sGARCH(1,1)snorm	94.2%	5.8%	95%	5%	94.8%	5.2%
	500.AR(3)eGARCH(1,1)sged	94.8%	5.2%	95.6%	4.4%	93%	7%
	1000.AR(2)eGARCH(1,1)snorm	94.9%	5.1%	95.1%	4.9%	95.1%	4.9%
	2000.AR(0)gjrGARCH(1,1)norm	94.05%	5.95%	94.95%	5.05%	94.95%	5.05%
	2000.AR(3)sGARCH(1,1)norm	94.15%	5.85%	95.1%	4.9%	95.1%	4.9%
	2000.AR(1)eGARCH(1,1)snorm	95.5%	4.5%	95.5%	4.5%	95.5%	4.5%
	500.AR(0)sGARCH(1,1)snorm	98.8%	1.2%	98.8%	1.2%	98.8%	1.2%
	500.AR(0)eGARCH(1,1)norm	99%	1%	99%	1%	99%	1%
	500.AR(0)eGARCH(1,1)sged	99.4%	0.6%	99.4%	0.6%	99.4%	0.6%
99%	1000.AR(0)eGARCH(1,1)norm	98.7%	1.3%	98.8%	1.2%	98.8%	1.2%
	1000.AR(1)eGARCH(1,1)snorm	98.9%	1.1%	98.9%	1.1%	98.9%	1.1%
	1500.AR(3)eGARCH(1,1)sged	99.07%	0.93%	99.2%	0.8%	99.2%	0.8%
	2000.AR(0)sGARCH(1,1)sged	99.15%	0.85%	99.15%	0.85%	99.15%	0.85%
	2000.AR(2)eGARCH(1,1)sstd	99.05%	0.95%	99.2%	0.8%	98.7%	1.3%
	2000.AR(3)sGARCH(1,1)std	98.6%	1.4%	99%	1%	98.95%	1.05%

NKY

		RUGARCH _Garch		MSGARCH _ML		MSGARCH _MCMC	
Levels	Models	Bontà	Excess	Bontà	Excess	Bontà	Excess
	500.AR(0)sGARCH(1,1)sged	93.6%	6.4%	93.8%	6.2%	93.6%	6.4%
	500.AR(2)sGARCH(1,1)norm	93%	7%	93.8%	6.2%	93.4%	6.6%
	500.AR(2)gjrGARCH(1,1)sstd	94%	6%	94.4%	5.6%	94.4%	5.6%
95%	500.AR(3)sGARCH(1,1)sged	93.8%	6.2%	94%	6%	93.8%	6.2%
	1000.AR(3)sGARCH(1,1)sstd	94.1%	5.9%	94.3%	5.7%	94.1%	5.9%
	2000.AR(1)eGARCH(1,1)sstd	94.15%	5.85%	94.85%	5.15%	94.7%	5.3%
	1500.AR(3)sGARCH(1,1)sged	94%	6%	94.4%	5.6%	94.33%	5.67%
	2000.AR(3)eGARCH(1,1)sstd	94.15%	5.85%	94.9%	5.1%	94.45%	5.55%
	1000.AR(3)sGARCH(1,1)std	93.8%	6.2%	93.8%	6.2%	93.8%	6.2%
	500.AR(0)sGARCH(1,1)sstd	99.6%	0.4%	99.6%	0.4%	99.6%	0.4%
	500.AR(0)gjrGARCH(1,1)ged	99.8%	0.2%	99.8%	0.2%	99.8%	0.2%
	500.AR(1)eGARCH(1,1)sstd	99.6%	0.4%	99.6%	0.4%	99.4%	0.6%
99%	1000.AR(0)sGARCH(1,1)std	99.3%	0.7%	99.4%	0.6%	99.4%	0.6%
	1000.AR(0)eGARCH(1,1)ged	99.1%	0.9%	99.2%	0.8%	99.2%	0.8%
	1000.AR(0)gjrGARCH(1,1)ged	99.3%	0.7%	99.4%	0.6%	99.4%	0.6%
	1000.AR(2)eGARCH(1,1)ged	99.1%	0.9%	99.3%	0.7%	99.3%	0.7%
	1500.AR(2)sGARCH(1,1)snorm	98.53%	1.47%	98.8%	1.2%	98.8%	1.2%
	2000.AR(1)eGARCH(1,1)sstd	99.35%	0.65%	99.25%	0.75%	99.45%	0.55%

DAX

		RUGARCH _Garch		MSGARCH _ML		MSGARCH _MCMC	
Levels	Models	Bontà	Excess	Bontà	Excess	Bontà	Excess
	500.AR(0)eGARCH(1,1)sged	93.2%	6.8%	94.2%	5.8%	93.8%	6.2%
	1000.AR(1)eGARCH(1,1)sged	94.1%	5.9%	94.8%	5.2%	93.6%	6.4%
	2000.AR(0)sGARCH(1,1)snorm	94.5%	5.5%	94.9%	5.1%	94.9%	5.1%
95%	2000.AR(3)sGARCH(1,1)snorm	94.55%	5.45%	94.9%	5.1%	94.9%	5.1%
	2000.AR(3)gjrGARCH(1,1)sged	94.3%	5.7%	95.25%	4.75%	95.1%	4.9%
	1000.AR(3)eGARCH(1,1)sged	93.6%	6.4%	94.8%	5.2%	94.6%	5.4%
	500.AR(1)eGARCH(1,1)sged	93.4%	6.6%	94.2%	5.8%	92.2%	7.8%
	1000.AR(0)eGARCH(1,1)snorm	94%	6%	94.4%	5.6%	94.4%	5.6%
	1000.AR(2)eGARCH(1,1)snorm	93.9%	6.1%	94.4%	5.6%	94.4%	5.6%
	500.AR(0)sGARCH(1,1)sstd	99.6%	0.4%	99.6%	0.4%	99.6%	0.4%
	500.AR(0)eGARCH(1,1)snorm	99.6%	0.4%	99.8%	0.2%	99.8%	0.2%
	500.AR(0)gjrGARCH(1,1)sged	99.8%	0.2%	99.8%	0.2%	99.8%	0.2%
99%	500.AR(1)gjrGARCH(1,1)snorm	99.6%	0.4%	99.8%	0.2%	99.8%	0.2%
	1000.AR(1)eGARCH(1,1)std	99.2%	0.8%	99.2%	0.8%	99.2%	0.8%
	1000.AR(2)sGARCH(1,1)sstd	99.4%	0.6%	99.4%	0.6%	99.4%	0.6%
	1500.AR(1)sGARCH(1,1)ged	98.8%	1.2%	99.07%	0.93%	99.07%	0.93%
	1500.AR(3)gjrGARCH(1,1)sstd	98.8%	1.2%	99%	1%	99%	1%
	2000.AR(2)eGARCH(1,1)snorm	98.9%	1.1%	99%	1%	99%	1%

CAC

		RUGARCH _Garch		MSGARCH _ML		MSGARCH _MCMC	
Levels	Models	Bontà	Excess	Bontà	Excess	Bontà	Excess
	500.AR(0)Egarch(1,1)ged	93.2%	6.8%	94.4%	5.6%	94.6%	5.4%
	500.AR(1)eGARCH(1,1)ged	93.6%	6.4%	94.4%	5.6%	94.6%	5.4%
	500.AR(3)gjrGARCH(1,1)snorm	93%	7%	94.2%	5.8%	94.2%	5.8%
95%	500.AR(3)gjrGARCH(1,1)sged	93.4%	6.6%	94.2%	5.8%	94%	6%
	1000.AR(0)eGARCH(1,1)snorm	94%	6%	94.9%	5.1%	94.9%	5.1%
	1000.AR(3)eGARCH(1,1)sstd	93.6%	6.4%	94.6%	5.4%	94.4%	5.6%
	2000.AR(0)gjrGARCH(1,1)snorm	94.6%	5.4%	95.2%	4.8%	95.2%	4.8%
	2000.AR(1)sGARCH(1,1)sged	94.1%	5.9%	95.1%	4.9%	95.1%	4.9%
	2000.AR(3)eGARCH(1,1)snorm	94.7%	5.3%	95.25%	4.75%	95.25%	4.75%
	500.AR(3)gjrGARCH(1,1)sged	99.6%	0.4%	99.6%	0.4%	99.4%	0.6%
	500.AR(1)eGARCH(1,1)ged	99.6%	0.4%	99.6%	0.4%	99.6%	0.4%
	1000.AR(1)eGARCH(1,1)snorm	98.5%	1.5%	99.1%	0.9%	99.1%	0.9%
99%	1000.AR(3)eGARCH(1,1)sged	99%	99.9%	99.2%	0.8%	98.8%	1.2%
	2000.AR(0)gjrGARCH(1,1)snorm	98.55%	1.45%	98.75%	1.25%	98.85%	1.15%
	2000.AR(1)sGARCH(1,1)snorm	98.65%	1.35%	98.8%	1.2%	98.85%	1.15%
	2000.AR(2)eGARCH(1,1)sged	99.2%	0.8%	99.25%	0.75%	98.7%	1.3%
	2000.AR(2)gjrGARCH(1,1)sged	98.9%	1.1%	99.05%	0.95%	98.8%	1.2%
	2000.AR(3)eGARCH(1,1)sged	99.2%	0.8%	99.25%	0.75%	98.95%	1.05%

UKX

		RUGARCH _Garch		MSGARCH _ML		MSGARCH _MCMC	
Levels	Models	Bontà	Excess	Bontà	Excess	Bontà	Excess
	500.AR(1)gjrGARCH(1,1)snorm	93%	7%	93.4%	6.6%	93.6%	6.4%
	500.AR(3)gjrGARCH(1,1)snorm	93%	7%	93.2%	6.8%	93.6%	6.4%
	2000.AR(0)gjrGARCH(1,1)snorm	94.65%	5.35%	95.2%	4.8%	95.3%	4.7%
95%	2000.AR(1)gjrGARCH(1,1)snorm	94.8%	5.2%	95.1%	4.9%	95.3%	4.7%
	2000.AR(2)gjrGARCH(1,1)snorm	94.8%	5.2%	95.05	4.95	95.25	4.75
	2000.AR(3)gjrGARCH(1,1)snorm	94.8%	5.2%	95.15	4.85	95.25	4.75
	2000.AR(1)Egarch(1,1)snorm	94.95%	5.05%	95.1	4.9	95.1	4.9
	2000.AR(2)eGARCH(1,1)snorm	94.95%	5.05%	95.1	4.9	95.15	4.85
	2000.AR(1)eGARCH(1,1)snorm.1	94.95%	5.05%	95.1	4.9	95.1	4.9
	500.AR(2)sGARCH(1,1)snorm	98.4%	1.6%	99%	1%	99%	1%
	1000.AR(1)sGARCH(1,1)snorm	98.5%	1.5%	99.1%	0.9%	99.1%	0.9%
	500.AR(3)sGARCH)(1,1)snorm	98.2%	1.8%	99%	1%	99%	1%
99%	1000.AR(0)gjrGARCH(1,1)sstd	98.8%	1.2%	99.1%	0.9%	99.2%	0.8%
	1000.AR(0)gjrGARCH(1,1)std	98.7%	1.3%	98.9%	1.1%	99.1%	0.9%
	1000.AR(1)eGARCH(1,1)std	98.5%	1.5%	98.8%	1.2%	98.9%	1.1%
	1000.AR(2)gjrGARCH(1,1)snorm	98.7%	1.3%	99%	1%	99%	1%
	1500.AR(0)gjrGARCH(1,1)ged	98.6%	1.4%	98.87%	1.13%	98.87%	1.13%
	2000.AR(1)eGARCH(1,1)sstd	98.9%	1.1%	99.15%	0.85%	98.9%	1.1%

Conclusions

In this paper series of five stock price index were analyzed using different models (with and without dynamic component), error distributions and forecast.

Because of the limits of the linear model that we have analyzed in the first chapter, to build a performing model that is able to capture volatility of time series and to carry out reliable forecasts on the future, we take into account the RUGARCH package.

Thanks to his feasibility, the package allows us to estimate about 1032 models for each index that we have calculated for each forecast length.

Indeed, the size of the rolling sample used in estimation turns out to be rather important: in simpler models and low confidence levels a sample size smaller than 2000 improves probability values.

At a confidence level of 95%, the length of forecast is longer than the other one at 99%, as is evident in the Table 9.

As concern to the distributions, the leptokurtic distributions and especially the "sSdt" and "sGed", are more appropriate than the normal assumption, as they generate more accurate forecasts, while there is no volatility model, which is in absolute superior than the others.

Since volatility is used as a key ingredient for VaR estimates a risk-management, a particularly function-"Bontà"- is adopted to compare the forecasting performances of the competing models.

Finally, after the comparison between general GARCH models and Markov Switching GARCH in term of their ability to predict volatility; the main results from the Tables 7 and 8, using stock market data, point out that MRS-GARCH models significantly outperform the usual GARCH models in forecasting volatility at shorter and long horizons

However, there is no clear answer about which model fares the best under this VaR-based loss function.

In sum, no model seems to outperform all the others in forecasting volatility in according to in-sample evaluation.

Reference

1. Markov Switching GARCH Models in R: The MSGARCH Package.
2. The use of GARCH models in VaR estimation. *T. Angelidis et al. / Statistical Methodology 1 (2004)*
3. Analysis of Financial Time Series, Third Edition RUEY S. TSAY The University of Chicago Booth School of Business Chicago, IL
4. Appunti di Inferenza Statistica Prof.ssa Lea Petrella, Dipartimento MEMOTEF September 11, 2013
5. Forecasting Stock Market Volatility with Regime-Switching Garch Models, Juri Marcucci, Department of Economics, University of Capifornia at San Diego, March 2005
6. Vito Ricci - ANALISI DELLE SERIE STORICHE CON R - R 0.4 del 21/02/05
7. Vito Ricci. Rappresentazione analitica delle distribuzioni statistiche con R (prima parte). *Economia e Commercio*, 1/2:47--60, 2005
8. Fitting Distributions with R" by Vito Ricci Release 0.4-21 February 2005
9. Una (mica tanto) breve introduzione a LATEX 2ε, Tobias Oetiker Hubert Partl, Irene Hyna e Elisabeth Schlegl Versione 3.16, 25 settembre 2000

Sitography

10. http://www.cran.r-project.org
11. https://www.rstudio.com/
12. http://www.github.com
13. http://www.stackoverflow.com
14. https://www.latex-project.org/
15. http://texstudio.sourceforge.net/

APPENDIX LATEX

```latex
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=3.00cm, right=3.00cm, top=2.00cm, bottom=2.00cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
\begin{table}[ht]
\centering
\begin{tabular}{rrrrrr}
\hline
& stat.SPX & stat.NKY & stat.DAX & stat.CAC & stat.UKX \\
\hline
Media &0.0002738807& -0.0002550642& 0.0002099406& 0.0001974461& 0.0001436932 \\
Mediana & 0.0004162731 & -0.0001135587&0.0007774265& 0.0003469953& 0.0004545561 \\
Massimo &   0.08708879 &  0.1243033  &   0.07552676 &  0.08225436 &  0.07596966 \\
Minimo  &   -0.2289972 &  -0.1613542  &  -0.1370612 &  -0.1013757 &  -0.130286 \\
Varianza &  0.0001299643 & 0.0002232236 & 0.0002101827 & 0.0001896902 & 0.0001190473 \\
Std.Dev &   0.01140019 &  0.01494067 &  0.01449768 &  0.01377281 &  0.01091088 \\
Asimmetria & -2.296291  &  -0.05996877 &  -0.5311798 &  -0.3233085 &  -0.8804831 \\
Curtosi &   46.73206   &7.107121   &   6.449065  &   4.384675 &    11.39993 \\
JB.Stat &   354748.5   & 7944.494  &    6854.253  &   3137.771  &   21444.79 \\
JB.Pvalue &  0        &    0        &     0       &     0      &      0 \\
\hline
\end{tabular}
\end{table}
\end{document}
```

```latex
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=3.00cm, right=3.00cm, top=2.00cm, bottom=2.00cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrr}
        \hline
        & test.err.SPX & test.err.NKY & test.err.DAX & test.err.CAC & test.err.UKX \\
        \hline
        Media & 6.863798e-18 &-4.438582e-18 & -1.038849e-17 & 1.479291e-17 & 6.216579e-18 \\
        Varianza & 1 &  1  &  1  &  1  &   1 \\
        Sigma &  1  &   1  & 1  &   1  &  1 \\
        Asimmetria& -2.295237  &  -0.0556457 &   -0.5212031 &   -0.3176482 &  -0.8758036 \\
        Curtosi  &    46.90119  &   7.13831  &     6.46925  &     4.396002 &   11.43864 \\
        JB.Stat  &    357293.1   &  8014.026  &    6889.349  &    3151.337  &   21581.95 \\
        JB.Pvalue &    0       &     0         &    0         &    0         &  0 \\
        BP.Stat   &   0.1575567  &   2.986232   &   30.40373   &   27.44081   &  10.00451 \\
        BP.Pvalue &   0.691416  &   0.08397535   & 3.508523e-08 & 1.619769e-07  & 0.00156157 \\
        LjBOX.Pvalue & 0.003150995 &  8.736402e-06 &  0.005862285 &   0.08022054 &  6.199032e-06 \\
        PiBOX.Pvalue & 0.003184783 &  8.908966e-06 & 0.005949899   & 0.0807181 &   6.32517e-06 \\
        DW.Stat  &    1.974808  &   2.036204  &    2.000088  &    1.953423  &   1.903096 \\
        DW.Pvalue &   0.4339276 &   0.2663638  &   0.9978147  &   0.1495614 &   0.002593341 \\
        \hline
    \end{tabular}
\end{table}
\end{document}
```

```
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=3.00cm, right=3.00cm, top=2.00cm, bottom=2.00cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrr}
        \hline
        & T.SPX.N & T.NKY.N & T.DAX.N & T.CAC.N & T.UKX.N \\
        \hline
         mu & 0.000571 & 0.000561 & 0.000701 & 0.000463 & 0.000406 \\
        sd & 0.000136 & 0.000186 & 0.000181 & 0.000189 & 0.000146 \\
         ar1 & 0.028923 & 0.014978 & 0.028915 & 0.045198 & 0.055030 \\
        sd & 0.017595 & 0.017940 & 0.018081 & 0.017311 & 0.017262 \\
        omega & 0.000002 & 0.000004 & 0.000005 & 0.000006 & 0.000003 \\
        sd & 0.000001 & 0.000002 & 0.000003 & 0.000002 & 0.000001 \\
        alpha1 & 0.104049 & 0.159091 & 0.139030 & 0.105150 & 0.094280 \\
        sd & 0.017983 & 0.015104 & 0.015511 & 0.005369 & 0.012391 \\
        beta1 & 0.889346 & 0.839905 & 0.841890 & 0.865351 & 0.885826 \\
        sd & 0.018080 & 0.014567 & 0.013139 & 0.008557 & 0.013864 \\
        Log LikeHood&12485.923050 & 10921.781688 & 11427.081531 & 11400.674735 & 12509.263159 \\
            \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrr}
        \hline
        & T.SPX.T & T.NKY.T & T.DAX.T & T.CAC.T & T.UKX.T \\
        \hline
    mu&0.000596& 0.000283 & 0.000698& 0.000576 & 0.000419 \\
    sd&0.000120&0.000169 & 0.000161 & 0.000183 &  0.000139 \\
    ar1& 0.011074 &-0.007624 & 0.010082 & 0.039411 & 0.049060 \\
    sd & 0.015767 &0.016798& 0.015405& 0.016680 & 0.016476 \\
    omega & 0.000001 & 0.000002 & 0.000002 & 0.000003 & 0.000002 \\
    sd & 0.000000 & 0.000003 &  0.000004  &  0.000002 &  0.000001 \\
    alpha1& 0.059139 & 0.099239& 0.094135& 0.083463 &0.078163 \\
    sd & 0.005022 & 0.031157& 0.053798 & 0.013138 & 0.017826 \\
    beta1& 0.938832 & 0.899746 & 0.899609 &  0.899109 & 0.906615 \\
    sd & 0.004586 & 0.029245& 0.054536 & 0.015829 & 0.019998 \\
    shape & 5.581308  & 6.300967 & 7.695111 & 10.594297 & 10.853064 \\
    sd & 0.473334 & 0.706762 &  0.381954 & 1.498044 & 1.322226 \\
    Log LikeHood & 12667.538313& 11086.388004& 11575.128408& 11452.427882 & 12609.792388 \\
        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrr}
        \hline
        & T.SPX.G & T.NKY.G & T.DAX.G & T.CAC.G & T.UKX.G \\
        \hline
```

```
        mu & 0.000543 & 0.000314& 0.000736&.000507 & 0.000441 \\
        sd &  0.000125&0.000179 & 0.000164 &0.000185 & 0.000142 \\
        ar1 &  -0.000899 & -0.008343 &  0.009663 &  0.035590 &  0.044711 \\
        sd&  0.017106 &   0.017157 &  0.016894 & 0.016743  & 0.016805 \\
        omega  &   0.000001 & 0.000002 & 0.000003 & 0.000004 & 0.000002 \\
        sd & 0.000000 & 0.000002 & 0.000002 &  0.000002& 0.000005 \\
        alpha1& 0.067446 & 0.111708 & 0.110197 & 0.092225 & 0.084106 \\
        sd & 0.006768 & 0.021557 &  0.015941 &    0.010594  &   0.057692 \\
        beta1 &  0.929859  &   0.886720  &   0.879404   & 0.885141 & 0.899376 \\
        sd & 0.006789 & 0.020890  & 0.016845 &  0.012930 &  0.066140 \\
        shape &  1.247997 & 1.314053 &    1.409014  &   1.556897 & 1.524015 \\
        sd &  0.034744 & 0.037415 &  0.037996 &  0.048033 & 0.064860 \\
        Log LikeHood & 12643.745372 & 11052.363415 &11524.915805& 11435.766468& 12569.143407 \\
        \hline
    \end{tabular}
\end{table}


\end{document}
```

```
\documentclass[12pt,a1paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.5cm, right=0.5cm, top=0.5cm, bottom=0.5cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
  \textit{\textbf{\LARGE{SPX}}} \\
       \begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|}
           \hline
 &\textit{\textit{\textbf{UC\_test}}}&&&\textit{\textit{\textbf{CC\_test}}}&&&\textit{\textit{\textbf{DQ\_test}}}&&&
 \textit{\textit{\textbf{BONTA'}}}&&\textit{\textit{\textbf{}}}    \\

    \hline
    Levels & Models & Statistic & P-value & Models & Statistic & P-value & Models & Statistic & P-value & Models & P-value & Error \\
           \hline\

           \textit{\textit{\textbf{}}}& 2000.AR(0)apARCH(1,2)sged  &0.010559 & 91,81\% & 500.AR(0)apARCH(1,2)snorm & 0.066395 & 96,73\% &
           500.AR(1)gjrGARCH(2,1)norm &  0.934538 & 98,7\% & 500.AR(1)apARCH(1,2)sged  & 96.4\%  & 3.6\% \\
           \textit{\textit{\textbf{95\%}}} & 2000.AR(1)iGARCH(1,1).norm & 0.010559 & 91,81\% & 500.AR(0)apARCH(1,2)snorm & 0.066395 & 96,73\% &
           500.AR(1)gjrGARCH(2,2)norm & 0.934586 & 98,79\% & 1000.AR(0)apARCH(1,1)sged & 96.2\% & 3.84\% \\
           \textit{\textit{\textbf{}}} & 2000.AR(2)apARCH(2,1)sged & 0.010559 & 91,81\% & 500.AR(2)eGARCH(1,1)sged & 0.066395 & 96,73\% &
           500.AR(2)gjrGARCH(2,2)norm & 0.947873 & 98,75\% & 1000.AR(0)apARCH(1,1)ged & 95.7\% & 4.3\% \\
           \textit{\textit{\textbf{}}} & 2000.AR(3)eARCH(1,2)sged & 0.010559 & 91,81\% & 500.AR(3)apARCH(1,1)sged & 0.066395 & 96,73\% &
           500.AR(2)gjrGARCH(2,1)norm & 0.947918 & 98,75\% & 2000.AR(1)apARCH(1,1)sged  & 95.7\% & 4.3\% \\
           \hline \

          \textit{\textit{\textbf{}}} & 2000.AR(0)apARCH(1,2)sged & 0.051360 & 82.07\%&1000.AR(0)apARCH(1,1)sstd& 0.268159 & 87,45\% &
          500.AR(3)apARCH(1,2)norm & 0.214281 & 99.98\% & 1000.AR(0)sGARCH(1,2)ged & 99.5\% & 0.5\%  \\
          \textit{\textit{\textbf{99\%}}} & 2000.AR(1)eGARCH(1,1)ged & 0.051360 & 82.07\% &1000.AR(1)apARCH(1,2)ged& 0.268159 &87,45\% &
          500.AR(2)eGARCH(1,1)snorm & 0.214301 & 99.98\% & 1000.AR(0)apARCH(1,1)std & 99.5\% &  0.5\%  \\
          \textit{\textit{\textbf{}}} & 2000.AR(2)apARCH(2,1)sstd &0.051360 &82.07\% &1000.AR(2)eGARCH(1,1)ged&0.268159 &  87,45\% &
          500.AR(1)apARCH(1,2)norm & 0.214306 & 99.98\% & 1000.AR(0)apARCH(1,1)ged & 99.5\% & 0.5\%  \\
           \textit{\textit{\textbf{}}} &  2000.AR(3)eGARCH(1,2)sstd & 0.051360 &82.07\% & 1000.AR(3)apARCH(1,1)sstd& 0.268159 & 87,45\% &
           500.AR(2)apARCH(1,1)norm & 0.214331 & 99.98\% & 1000.AR(0)apARCH(1,1)sged & 99.5\% & 0.5\% \\

           \hline
       \end{tabular}
```

```
\begin{table}[ht]

    \textit{\textbf{\LARGE{NKY}}} \\

    \begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|}

        \hline
        &\textit{\textit{\textbf{UC\_test}}}&&&\textit{\textit{\textbf{CC\_test}}}&&&\textit{\textit{\textbf{DQ\_test}}}&&&
        \textit{\textit{\textbf{BONTA'}}}&&\textit{\textit{\textbf{}}}\\
        \hline
        Levels & Models & Statistic & P-value & Models & Statistic & P-value & Models & Statistic & P-value & Models & P-value & Error \\
        \hline\

        \textit{\textit{\textbf{}}}&1500.AR(0)iGARCH(0,2)std&    0.014094 &    90,56\%& 500.AR(0)iGARCH(0,2)std &  0.066395&    96,73\%&
        500.AR(1)eGARCH(0,2)sstd&  2.490339&   86,95\%& 1000.AR(1)iGARCH(0,1)sstd&  96.2\% &        3.8\%\\
        \textit{\textit{\textbf{95\%}}}&1500.AR(1)iGARCH(0,1)sstd& 0.014094&     90,56\%& 500.AR(1)eGARCH(0,1)sstd& 0.066395&    96,73\%&
        500.AR(1)eGARCH(0,1)sstd&  2.493393&  86,92\%& 1000.AR(1)iGARCH(0,2)std&  96.1\%&          3.9\%\\
        \textit{\textit{\textbf{}}}&1000.AR(2)gjrGARCH(1,1)snorm&    0.021187&   88,43\%& 500.AR(0)iGARCH(0,2)sstd& 0.066395&    96,73\%&
        500.AR(0)iGARCH(0,2)sstd&  2.514683&  86,68\%& 1000.AR(0)eGARCH(0,1)sstd & 96\%&           4\%\\
        \textit{\textit{\textbf{}}}&1000.AR(3)eGARCH(1,1)sged &0.021187&   88,43\%&  500.AR(1)eGARCH(0,2)sstd& 0.066395&    96,73\%&
        500.AR(0)iGARCH(0,2)std&  2.540576&  86,39\%& 1000.AR(0)eGARCH(0,2)sstd&  96\% &          4\%\\
        \hline


        \textit{\textit{\textbf{}}}&2000.AR(0)csGARCH(1,1)ged & 0.051360& 82.07\%& 1000.AR(0)apARCH(1,1)std& 0.268159& 87.45\%&
        1500.AR(0)eGARCH(1,2)sstd &  0.578767&  99.67\%& 500.AR(0)sGARCH(0,2)sstd&  99.8\%&  0.2\% \\
        \textit{\textit{\textbf{99\%}}}&2000.AR(1)csGARCH(0,1)std& 0.051360& 82.07\% & 1000.A(1)eGARCH(1,1)ged& 0.268159& 87.45\%&
        1500.AR(1)eGARCH(1,2)sstd &  0.584921 & 99.66\%& 500.AR(0)eGARCH(0,1)std& 99.8\%&  0.2\% \\
        \textit{\textit{\textbf{}}}&2000.AR(2)gjrGARCH(2,1)sged& 0.051360& 82.07\%&  1000.AR(2)apARCH(2,1)sstd& 0.268159& 87.45\%&
        1500.AR(2)eGARCH(1,2)sstd&   0.587116&  99.66\%& 500.AR(0)eGARCH(0,1)sstd& 99.8\%&  0.2\%\\
        \textit{\textit{\textbf{}}} &2000.AR(3)gjrGARCH(2,1)sged &0.051360& 82.07\% & 1000.AR(3)eGARCH(1,1)std& 0.268159& 87.45\%&
        500.AR(2)sGARCH(1,1)snorm&  0.633443&  99.58\%&  500.AR(0)eGARCH(0,2)std& 99.8\%&  0.2\% \\

        \hline
    \end{tabular}

\end{table}
\begin{table}[ht]

    \textit{\textbf{\LARGE{DAX}}} \\

    \begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|}


        \hline
        &\textit{\textit{\textbf{UC\_test}}}&&&\textit{\textit{\textbf{CC\_test}}}&&&\textit{\textit{\textbf{DQ\_test}}}&&&
        \textit{\textit{\textbf{BONTA'}}}&&\textit{\textit{\textbf{}}}\\
        \hline
        Levels & Models & Statistic & P-value & Models & Statistic & P-value & Models & Statistic & P-value & Models & P-value & Error \\
        \hline\

        \textit{\textit{\textbf{}}}&2000.AR(2)eGARCH(2,1)snorm&    0.042374&    83,69\%& 2000.AR(2)iGARCH(1,1)sged&   0.205844&   90,22\%&
        500.AR(0)iGARCH(1,1)snorm&  1.853546&   93,27\%& 1000.AR(2)apARCH(2,1)norm&   95.7\%&  4.3\%\\
        \textit{\textit{\textbf{95\%}}}&2000.AR(3)eGARCH(2,1)snorm&     0.095648 &   75,71\% &2000.AR(3)iGARCH(1,1)sged&   0.205844&   90,22\%&
        500.AR(0)iGARCH(1,2)snorm&  3.339442&   76,52\%& 2000.AR(3)eGARCH(2,1)snorm&   95.15\%& 4.85\%\\
        \textit{\textit{\textbf{}}}&2000.AR(2)iGARCH(1,1)sged&   0.010560&   91,81\%& 2000.AR(3)eGARCH(2,1)snorm&  0.871268&  64,68\%&
        500.AR(1)iGARCH(2,2)snorm&  3.339566&   76,51\%&  2000.AR(2)eGARCH(2,1)snorm&   95.1\%&  4.9\%\\
        \textit{\textit{\textbf{}}}&2000.AR(3)iGARCH(1,1)sged&    0.010560&    91,81\%& 2000.AR(2)eGARCH(2,1)snorm&  0.896883&  63,86\%&
        500.AR(1)iGARCH(2,1)snorm&  3.343781&  76,46\% & 2000.AR(2)iGARCH(1,1)sged&    95.05\%&  4.95\%\\
        \hline

        \textit{\textit{\textbf{}}}& 2000.AR(0)apARCH(1,1)ged& 0.051360& 82.07\%& 1000.AR(0)sGARCH(1,1)std& 0.268159& 87.45\%&
        500.AR(0)csGARCH(1,1)norm& 0.301040&  99.95\%& 500.AR(0)eGARCH(1,1)std&  99.8\%& 0.2\% \\
        \textit{\textit{\textbf{99\%}}}&2000.AR(1)apARCH(1,1)sstd& 0.051360& 82.07\% &  1000.AR(1)eGARCH(2,1)snorm& 0.268159& 87.45\%&
        500.AR(0)csGARCH(1,2)norm& 0.302044&  99.95\%& 500.AR(0)eGARCH(1,1)ged& 99.8\%& 0.2\%\\
        \textit{\textit{\textbf{}}}&2000.AR(2)eGARCH(2,1)sstd& 0.051360& 82.07\%&   1000.AR(2)apARCH(1,1)ged& 0.268159& 87.45\%&
        500.AR(0)gjrGARCH(1,1)norm&  0.302927&  99.95\%& 500.AR(0)eGARCH(1,1)sstd& 99.8\% &  0.2\%\\
        \textit{\textit{\textbf{}}}&2000.AR(3)eGARCH(2,1)sstd& 0.051360& 82.07\%&  1000.AR(3)eGARCH(2,1)snorm& 0.268159& 87.45\%&
        500.AR(0)gjrGARCH(1,2)norm& 0.303186&  99.94\%& 500.AR(0)eGARCH(1,1)sged& 99.8\%& 0.2\%\\

        \hline
    \end{tabular}

\end{table}
\begin{table}[ht]

    \textit{\textbf{\LARGE{CAC}}} \\

    \begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|}
        \hline
        &\textit{\textit{\textbf{UC\_test}}}&&&\textit{\textit{\textbf{CC\_test}}}&&&\textit{\textit{\textbf{DQ\_test}}}&&&
        \textit{\textit{\textbf{BONTA'}}}&&\textit{\textit{\textbf{}}}\\
        \hline
        Levels & Models & Statistic & P-value & Models & Statistic & P-value & Models & Statistic & P-value & Models & P-value & Error \\
        \hline\
```

```
        \textit{\textit{\textbf{}}}&2000.AR(0)iGARCH(2,2)ged&   0.010560&   91,81\%&   1000.AR(0)iGARCH(1,2)sged &  0.100184&   95,11\%&
        2000.AR(0)apARCH(1,1)snorm& 0.777941&  99,26\%&  2000.AR(0)iGARCH(1,2)snorm&  95.7\%&  4.3\%\\
        \textit{\textit{\textbf{95\%}}}&2000.AR(1)iGARCH(1,1)ged& 0.010560&   91,81\%&   1000.AR(1)iGARCH(2,1)sged&  0.100184&   95,11\%&
        2000.AR(0)gjrGARCH(2,2)snorm& 1.363193&  96,80\%&   2000.AR(2)iGARCH(1,1)snorm&  95.65\%&  4.35\%\\
        \textit{\textit{\textbf{}}}&2000.AR(2)iGARCH(1,2)norm& 0.010560&   91,81\%&  1000.AR(2)iGARCH(2,2)snorm&  0.100184&   95,11\%&
        2000.AR(0)gjrGARCH(2,1)snorm&  1.363225&   96,80\%&   2000.AR(0)iGARCH(1,1)snorm&  95.65\%&  4.35\%\\
        \textit{\textit{\textbf{}}}&2000.AR(3)iGARCH(2,1)ged&   0.042374&  83,70\%&  1000.AR(3)iGARCH(2,1)snorm&  0.100184&   95,11\%&
        2000.AR(0)apARCH(1,2)snorm& 1.459876&  96,21\%& 2000.AR(3)iGARCH(2,2)snorm &  95.6\%&  4.4\%\\

        \hline \

        \textit{\textit{\textbf{}}}& 2000.AR(0)apARCH(1,1)sged& 0.051360& 82.07\% &1000.AR(0)eGARCH(1,1)sstd& 0.268159& 87.45\%&
        500.AR(0)sGARCH(2,2)snorm& 0.215691&  99.98\%& 500.AR(0)eGARCH(1,1)ged&  99.6\%&  0.4\%\\
        \textit{\textit{\textbf{99\%}}}&2000.AR(1)iGARCH(2,1)snorm& 0.051360& 82.07\% &  1000.AR(1)iGARCH(2,1)sged& 0.268159& 87.45\%&
        500.AR(0)sGARCH(2,1)snorm& 0.216732&  99.98\%& 500.AR(0)eGARCH(1,2)ged&  99.6\%&  0.4\%\\
        \textit{\textit{\textbf{}}}&2000.AR(2)apARCH(1,2)sged& 0.051360& 82.07\%& 1000.AR(2)iGARCH(2,2)snorm& 0.268159& 87.45\%&
        500.AR(0)eGARCH(0,1)sged&   0.289296&   99.96\%& 500.AR(0)eGARCH(2,1)std&  99.6\%&  0.4\% \\
        \textit{\textit{\textbf{}}}&2000.AR(3)csGARCH(1,1)sged& 0.051360& 82.07\%&  1000.AR(3)iGARCH(2,1)snorm& 0.268159&
        87.45\%&500.AR(3)sGARCH(2,1)std&      0.296455&  99.95\%& 500.AR(0)eGARCH(2,1)ged&  99.6\%&  0.4\%\\

        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]

    \textit{\textbf{\LARGE{UKX}}} \\

    \begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|}
        \hline
        &\textit{\textit{\textbf{UC\_test}}}&&&\textit{\textit{\textbf{CC\_test}}}&&&\textit{\textit{\textbf{DQ\_test}}}&&&
        \textit{\textit{\textbf{BONTA'}}}&&\textit{\textit{\textbf{}}}\\
        \hline
        Levels & Models & Statistic & P-value & Models & Statistic & P-value & Models & Statistic & P-value & Models & P-value & Error \\
        \hline\

        \textit{\textit{\textbf{}}}&2000.AR(0)iGARCH(2,1)ged&  0.010560&   91,81\%&   2000.AR(1)iGARCH(1,1)norm&  0.115449&      94,39\%&
        2000.AR(0)csGARCH(2,2)norm& 3.258402&  77,58\%&  2000.AR(0)csGARCH(1,1)snorm& 95.4\%&   4.6\%\\
        \textit{\textit{\textbf{95\%}}}&2000.AR(1)apARCH(2,2)snorm&  0.010560 &  91,81\% &  2000.AR(2)iGARCH(1,1)norm&   0.115449 &  94,39\
        2000.AR(0)csGARCH(2,1)norm& 3.482836&  74,62\%&  2000.AR(2)iGARCH(2,2)norm& 95.35\%&     4.65\%\\
        \textit{\textit{\textbf{}}}&2000.AR(1)iGARCH(2,1)ged&  0.010560&   91,81\%&   2000.AR(1)apARCH(2,2)snorm&  0.265874&    87,55\%&
        2000.AR(3)csGARCH(1,1)norm& 3.959473&  68,22\% & 2000.AR(0)csGARCH(1,2)snorm& 95.35\%&      4.65\%\\
        \textit{\textit{\textbf{}}}&2000.AR(2)iGARCH(2,1)norm&   0.010560 &   91,81\%&   2000.AR(2)apARCH(2,1)snorm&   0.349499&    83,97\%&
        2000.AR(2)csGARCH(1,1)norm& 4.053052&   66,95\%& 2000.AR(3)csGARCH(2,2)snorm & 95.3\%&      4.7\%\\

        \hline \

        \textit{\textit{\textbf{}}}&2000.AR(0)eGARCH(1,1)sged& 0.051360& 82.07\%& 1000.AR(0)apARCH(1,1)sged& 0.268159& 87.45\%&
        1500.AR(3)csGARCH(2,2)sstd& 1.344650&  96.91\%& 2000.AR(0)iGARCH(2,1)sged&  99.5\% & 0.5\%\\
        \textit{\textit{\textbf{99\%}}}&2000.AR(1)apARCH(1,2)sged& 0.051360& 82.07\%& 1000.AR(1)iGARCH(1,1)sstd& 0.268159& 87.45\%&
        1000.AR(0)apARCH(2,2)std&    1.436875&  96.36\%& 2000.AR(0)iGARCH(2,2)sged& 99.5\% & 0.5\%\\
        \textit{\textit{\textbf{}}}&2000.AR(2)apARCH(2,1)sged& 0.051360& 82.07\%& 1000.AR(2)csGARCH(1,1)ged& 0.268159& 87.45\%&
        1000.AR(0)apARCH(1,1)std&    1.477043&  96.10\%& 2000.AR(2)iGARCH(2,1)sstd& 99.5\%&  0.5\%\\
        \textit{\textit{\textbf{}}}&2000.AR(3)iGARCH(1,1)sged&   0.051360& 82.07\%& 1000.AR(3)iGARCH(2,1)snorm& 0.268159& 87.45\%&
        1000.AR(0)apARCH(1,1)norm&  1.507042&  95.90\%& 2000.AR(2)iGARCH(2,2)sstd& 99.5\%&  0.5\% \\

        \hline
    \end{tabular}
\end{table}
\end{document}
```

```
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.50cm, right=0.50cm, top=0.50cm, bottom=0.00cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrr}
       \hline
       \textit{\textbf{{\large SPX 99\% }}} \\
         RANK &MODELLO & BONTA' & EXCED   & \\
            \hline
             1 & 500.AR(3)tARCH(1,0)snorm & 99.8\% & 0.2 \\
             2 & 1000.AR(0)tARCH(1,0)sstd & 99.6\% & 0.4  \\
             3 & 2000.AR(3)tARCH(1,0)sstd & 98.2\% & 1.8  \\
            \hline
        \textit{\textbf{{\large SPX 95\% }}} \\
        RANK &MODELLO & BONTA' & EXCED   & \\
        \hline
        1 & 500.AR(3)tARCH(1,0)snorm & 99.8\% & 0.2 \\
        2 & 1000.AR(0)tARCH(1,0)sstd & 99.6\% & 0.4 \\
        3 & 2000.AR(3)tARCH(1,0)sstd & 98.2\% & 1.8 \\
        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
```

```
    \begin{tabular}{rrrrrr}
        \hline
        \textit{\textbf{{\large NKY 99\% }}} \\
        RANK &MODELLO & BONTA' & EXCED   & \\
        \hline
        1 & 2000.AR(0)tARCH(1,0)sstd & 99.25\% & 0.75 \\
        2 & 2000.AR(1)tARCH(1,0)sstd & 99.25\% & 0.75  \\
        3 & 2000.AR(2)tARCH(1,0)sstd & 99.2\% & 0.8 \\
        \hline
        \textit{\textbf{{\large NKY 95\% }}} \\
        RANK &MODELLO & BONTA' & EXCED   & \\
        \hline
        1 & 2000.AR(0)tARCH(1,0)sstd & 99.25\% & 0.75 \\
        2 & 1000.AR(2)tARCH(1,0)sstd & 99\% & 1  \\
        3 & 1000.AR(3)tARCH(1,0)sstd & 99\% & 1 \\
        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrr}
        \hline
        \textit{\textbf{{\large DAX 99\% }}} \\
        RANK &MODELLO & BONTA' & EXCED   & \\
        \hline
        1 & 1000.AR(0)tARCH(1,0)norm & 99.8\% & 0.2 \\
        2 & 1000.AR(1)tARCH(1,0)norm & 99.7\% & 0.3  \\
        3 & 1000.AR(2)tARCH(1,0)norm & 99.7\% & 0.3 \\
        \hline
        \textit{\textbf{{\large DAX 95\% }}} \\
        RANK &MODELLO & BONTA' & EXCED   & \\
        \hline
```

```
        \textit{\textbf{{\large DAX 95\% }}} \\
        RANK &MODELLO & BONTA' & EXCED   & \\
        \hline
        1 & 1000.AR(2)tARCH(1,0)norm & 99.7\% & 0.3 \\
        2 & 1000.AR(0)tARCH(1,0)norm & 99.8\% & 0.2  \\
        3 & 1000.AR(1)tARCH(1,0)norm & 99.7\% & 0.3 \\
        \hline
\\
\\

        \hline
        \textit{\textbf{{\large UKX 99\% }}} \\
        RANK &MODELLO & BONTA' & EXCED   & \\
        \hline
        1 & 1000.AR(0)tARCH(1,0)norm & 99.6\% & 0.4 \\
        2 & 1500.AR(0)tARCH(1,0)norm & 99.4\% & 0.6 \\
        \hline
        \textit{\textbf{{\large UKX 95\% }}} \\
        RANK &MODELLO & BONTA' & EXCED   & \\
        \hline
        1 & 1000.AR(0)tARCH(1,0)norm & 99.6\% & 0.4 \\
        2 & 1500.AR(0)tARCH(1,0)norm & 99.4\% & 0.6  \\
        \hline
    \end{tabular}
\end{table}
\end{document}
```

```latex
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.50cm, right=0.50cm, top=0.50cm, bottom=0.50cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
        \centering
         \textit{\textbf{\large{SPX}}} \\
        \begin{tabular}{rrrrr}
            \hline
        \textbf{\textsl{Forecast}}  & 500 & 1000 & 1500 & 2000 \\
            \hline
            99\% & 27.88 & 27.84 & 22.30 & 21.98 \\
            95\% & 42.47 & 23.16 & 13.65 & 20.72 \\
            \hline
        \end{tabular}
    \end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Distributions}} & norm & std & ged & snorm & sstd & sged \\
        \hline
        99\% & 11.04 & 18.59 & 18.54 & 14.70 & 18.59 & 18.54 \\
        95\% & 21.47 & 3.58 & 13.75 & 32.86 & 5.08 & 23.26 \\
        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Models}}& sGARCH & eGARCH & gjrGARCH & apARCH & iGARCH & csGARCH \\
        \hline
        99\% & 14.66 & 16.62 & 14.97 & 15.95 & 16.00 & 21.81 \\
        95\% & 10.36 & 21.66 & 10.73 & 21.00 & 23.54 & 12.71 \\
        \hline
    \end{tabular}
\end{table}
\end{document}
```

```
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.50cm, right=0.50cm, top=0.50cm, bottom=0.50cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
        \centering
         \textit{\textbf{\large{NKY}}} \\
        \begin{tabular}{rrrrr}
            \hline
        \textbf{\textsl{Forecast}}  & 500 & 1000 & 1500 & 2000 \\
        \hline
        99\% & 30.08 & 28.91 & 23.22 & 17.78 \\
        95\% & 51.13 & 37.27 & 6.57 & 5.03 \\
            \hline
        \end{tabular}
    \end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Distributions}} & norm & std & ged & snorm & sstd & sged \\
    \hline
    99\% & 6.66 & 20.07 & 18.60 & 11.33 & 22.56 & 20.78 \\
    95\% & 11.09 & 11.60 & 16.94 & 16.53 & 17.76 & 26.08 \\
        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Models}}& sGARCH & eGARCH & gjrGARCH & apARCH & iGARCH & csGARCH \\
    \hline
    99\% & 18.39 & 11.53 & 13.16 & 12.40 & 19.21 & 25.30 \\
    95\% & 21.87 & 13.96 & 13.96 & 10.99 & 25.05 & 14.17 \\
        \hline
    \end{tabular}
\end{table}
\end{document}
```

```
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.50cm, right=0.50cm, top=0.50cm, bottom=0.50cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
        \centering
         \textit{\textbf{\large{DAX}}} \\
        \begin{tabular}{rrrrr}
            \hline
        \textbf{\textsl{Forecast}}& 500 & 1000 & 1500 & 2000 \\
        \hline
        99\% & 27.52 & 26.02 & 22.75 & 23.72 \\
        95\% & 16.67 & 16.36 & 10.49 & 56.48 \\
            \hline
        \end{tabular}
    \end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Distributions}} & norm & std & ged & snorm & sstd & sged \\
    \hline
    99\% & 8.92 & 18.37 & 18.37 & 17.58 & 18.37 & 18.37 \\
    95\% & 7.41 & 0.00 & 8.02 & 32.10 & 0.00 & 52.47 \\
        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Models}}& sGARCH & eGARCH & gjrGARCH & apARCH & iGARCH & csGARCH \\
    \hline
    99\% & 14.75 & 15.50 & 15.37 & 14.80 & 16.52 & 23.06 \\
    95\% & 6.17 & 19.44 & 9.88 & 17.59 & 37.04 & 9.88 \\
        \hline
    \end{tabular}
\end{table}
\end{document}
```

```latex
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.50cm, right=0.50cm, top=0.50cm, bottom=0.50cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
        \centering
         \textit{\textbf{\large{CAC}}} \\
        \begin{tabular}{rrrrr}
            \hline
        \textbf{\textsl{Forecast}} & 500 & 1000 & 1500 & 2000 \\
        \hline
        99\% & 25.05 & 30.61 & 17.03 & 27.30 \\
        95\% & 31.10 & 21.40 & 10.70 & 36.80 \\
            \hline
        \end{tabular}
    \end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Distributions}} & norm & std & ged & snorm & sstd & sged \\
    \hline
    99\% & 11.67 & 17.38 & 17.69 & 17.99 & 16.58 & 18.69 \\
    95\% & 9.42 & 6.85 & 11.84 & 30.67 & 13.27 & 27.96 \\
        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Models}}& sGARCH & eGARCH & gjrGARCH & apARCH & iGARCH & csGARCH \\
    \hline
    99\% & 17.79 & 12.47 & 13.83 & 13.68 & 15.78 & 26.45 \\
    95\% & 4.56 & 20.68 & 8.42 & 9.70 & 50.07 & 6.56 \\
        \hline
    \end{tabular}
\end{table}
\end{document}
```

```
\documentclass[10pt,a4paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.50cm, right=0.50cm, top=0.50cm, bottom=0.50cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
        \centering
         \textit{\textbf{\large{UKX}}} \\
        \begin{tabular}{rrrrr}
            \hline
        \textbf{\textsl{Forecast}}  & 500 & 1000 & 1500 & 2000 \\
        \hline
        99\% & 5.37 & 33.69 & 27.12 & 33.82 \\
        95\% & 20.51 & 28.37 & 15.45 & 35.67 \\
            \hline
        \end{tabular}
    \end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Distributions}}  & norm & std & ged & snorm & sstd & sged \\
    \hline
    99\% & 10.74 & 12.07 & 18.17 & 20.89 & 19.63 & 18.50 \\
    95\% & 18.26 & 3.37 & 9.55 & 46.91 & 8.43 & 13.48 \\
        \hline
    \end{tabular}
\end{table}
\begin{table}[ht]
    \centering
    \begin{tabular}{rrrrrrr}
        \hline
    \textbf{\textsl{Models}} & sGARCH & eGARCH & gjrGARCH & apARCH & iGARCH & csGARCH \\
    \hline
    99\% & 13.73 & 11.67 & 15.45 & 15.65 & 17.57 & 25.93 \\
    95\% & 2.25 & 2.25 & 9.27 & 7.87 & 50.56 & 27.81 \\
        \hline
    \end{tabular}
\end{table}
\end{document}
```

```
\documentclass[12pt,a1paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.5cm, right=0.5cm, top=0.5cm, bottom=0.5cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
  \textit{\textbf{\Large{SPX}}} \\
  \\
        \begin{tabular}{c c c c c c c c c c c c c}
            \hline
 &\textit{\textit{\textbf{UC\_test}}}&&&\textit{\textit{\textbf{CC\_test}}}&&&\textit{\textit{\textbf{DQ\_test}}}&&&
 \textit{\textit{\textbf{BONTA'}}}&&\textit{\textit{\textbf{}}}    \\

    \hline
    Levels & Models & Statistic & P-value & Models & Statistic & P-value & Models & Statistic & P-value & Models & P-value & Excess \\
            \hline\

            \textit{\textit{\textbf{}}} & 500.AR(1)eGARCH\_sstd & 2.459194 & 11.68\% & 500.AR(0)eGARCH\_snorm & 0.060211 & 97.03\% &
            1500.AR(0)gjrGARCH\_norm & 3.763635 & 70.86\% & 2000.AR(2)sGARCH\_snorm & 95.6\% & 4.4\% \\
            \textit{\textit{\textbf{95\%}}} & 500.AR(1)eGARCH\_ged & 0.010125 & 99.9\% & 500.AR(0)eGARCH\_ged & 0.060211 & 97.03\% &
            1500.AR(1)\_gjrGARCH\_norm & 3.750385 & 71.04\% & 2000.AR(3)sGARCH\_snorm & 95.65\% & 4.35\% \\
            \textit{\textit{\textbf{}}} & 2000.AR(2)gjrGARCH\_ged & 0.093853 & 75.93\% & 500.AR(1)eGARCH\_ged & 0.060211 & 97.03\% & 1000.AR(3)sGARCH\_ged &
            5.753464 & 45.14\% & 500.AR(3)eGARCH\_sged & 93\% & 7\% \\
            \textit{\textit{\textbf{}}}& 500.AR(0)eGARCH\_ged & 0.010125 & 99.9\% & 500.AR(1)eGARCH\_sstd & 3.355909 & 18.68\% & 1000.AR(0)gjrGARCH\_sstd &
            4.476733 & 61.24\% & 2000.AR(3)eGARCH\_sged & 95.05\% & 4.95\% \\

           \hline
            \textit{\textit{\textbf{}}} & 500.AR(1)eGARCH\_sstd & 0.943116 & 33.15\% & 500.AR(0)eGARCH\_snorm & 0.101216 & 95.07\% & 500.AR(0)eGARCH\_norm
            & 0.215809 & 99.98\% & 1000.AR(3)sGARCH\_sged & 99.5\% & 0.5\% \\
           \textit{\textit{\textbf{99\%}}} & 500.AR(1)eGARCH\_ged & 0.943116 & 33.15\% & 500.AR(3)eGARCH\_snorm & 0.101216 & 95.07\% &
           500.AR(1)eGARCH\_norm & 0.215069 & 99.98\% & 1000.AR(2)sGARCH\_sged & 99.5\% & 0.5\% \\
           \textit{\textit{\textbf{}}} &  2000.AR(2)gjrGARCH\_ged & 0.010125 & 99.9\% & 500.AR(3)gjrGARCH\_snorm & 0.101216 & 95.06\% &
           500.AR(2)eGARCH\_norm & 0.216139 & 99.98\% & 1000.AR(0)sGARCH\_sstd & 99.4\% & 0.6\% \\
           \textit{\textit{\textbf{}}} & 500.AR(0)eGARCH\_ged & 0.943116 & 33.15\% & 500.AR(3)sGARCH\_snorm & 0.101216 & 95.06\% & 500.AR(0)gjrGARCH\_norm
           & 0.215777 & 99.98\% & 1000.AR(3)sGARCH\_sstd & 99.5\% & 0.5\% \\
            \hline
         \\
            \end{tabular}
      ``
```

```
\documentclass[12pt,a1paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.5cm, right=0.5cm, top=0.5cm, bottom=0.5cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
  \textit{\textbf{\Large{SPX}}} \\
  \\
        \begin{tabular}{c c c c c c c c c c c c c}
            \hline
 &\textit{\textit{\textbf{UC\_test}}}&&&\textit{\textit{\textbf{CC\_test}}}&&&\textit{\textit{\textbf{DQ\_test}}}&&&
 \textit{\textit{\textbf{BONTA'}}}&&\textit{\textit{\textbf{}}}    \\

    \hline
    Levels & Models & Statistic & P-value & Models & Statistic & P-value & Models & Statistic & P-value & Models & P-value & Excess \\
            \hline\

            \textit{\textit{\textbf{}}} & 500.AR(1)eGARCH\_snorm & 0.010125 & 99,9\% & 500.AR(0)eGARCH\_snorm & 0.060211 & 97.03\% &
            1500.AR(0)gjrGARCH\_norm & 3.741405 & 71.16\% & 2000.AR(2)sGARCH\_snorm & 95.65\% & 4.35\% \\
            \textit{\textit{\textbf{95\%}}} & 500.AR(1)eGARCH\_sstd & 0.010125 & 99,9\%  & 500.AR(0)eGARCH\_ged & 0.060211 & 97.03\% &
            1500.AR(1)gjrGARCH\_norm & 3.741405 & 71.16\% & 2000.AR(3)sGARCH\_snorm & 95.65\% & 4.35\% \\
            \textit{\textit{\textbf{}}} & 2000.AR(2)gjrGARCH\_ged & 0.010493 & 91.84\% & 500.AR(1)eGARCH\_ged & 0.060211 & 97.03\% & 1000.AR(3)sGARCH\_ged &
            4.214639 & 64.77\% & 500.AR(3)eGARCH\_sged & 95.6\% & 4.4\% \\
            \textit{\textit{\textbf{}}}& 2000.AR(0)eGARCH\_ged & 0.010493 & 91.84\% & 500.AR(1)eGARCH\_sstd & 0.060211 & 97.03\% & 1000.AR(0)gjrGARCH\_sstd
            & 4.226057 & 64.61\% & 2000.AR(3)eGARCH\_sged & 95.55\% & 4.45\% \\

            \hline
             \textit{\textit{\textbf{}}} & 500.AR(1)eGARCH\_sstd & 0.943116 & 33.15\% & 500.AR(0)eGARCH\_snorm & 0.101216 & 95.07\% & 500.AR(0)eGARCH\_norm
             & 0.215045 & 99.98\% & 1000.AR(3)sGARCH\_sged & 99.5\% & 0.5\% \\
            \textit{\textit{\textbf{99\%}}} & 500.AR(1)eGARCH\_ged & 0.943116 & 33.15\% & 500.AR(3)eGARCH\_snorm & 0.101216 & 95.07\% &
            500.AR(1)eGARCH\_norm & 0.215045 & 99.98\% & 1000.AR(2)sGARCH\_sged & 99.5\% & 0.5\% \\
            \textit{\textit{\textbf{}}} & 2000.AR(2)gjrGARCH\_ged & 0.049692 & 82.36\% & 500.AR(3)gjrGARCH\_snorm & 0.101216 & 95.07\% &
            500.AR(2)eGARCH\_norm & 0.215342 & 99.98\% & 1000.AR(0)sGARCH\_sstd & 99.5\% & 0.5\% \\
            \textit{\textit{\textbf{}}} & 500.AR(0)eGARCH\_ged & 0.943116 & 33.15\% & 500.AR(3)sGARCH\_snorm & 0.101216 & 95.06\% & 500.AR(0)gjrGARCH\_norm
            & 0.216125 & 99.98\% & 1000.AR(3)sGARCH\_sstd & 99.5\% & 0.5\% \\
            \hline
         \\
            \end{tabular}
```

```
\documentclass[12pt,a1paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.5cm, right=0.5cm, top=0.5cm, bottom=0.5cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
  \textit{\textbf{\Large{SPX}}} \\
  \\
        \begin{tabular}{c c c c c c c c c c }
            \hline
&&\textit{\textit{\textbf{UC\_test}}}&&\textit{\textit{\textbf{CC\_test}}}&&\textit{\textit{\textbf{DQ\_test}}}&&
\textit{\textit{\textbf{BONTA'}}}&\textit{\textit{\textbf{}}}    \\

\hline\

Levels & \textit{\textit{\textbf{2000AR(2)sGARCH\_norm}}}  & Statistic & P-value & Statistic & P-value & Statistic & P-value & P-value & Excess
            \hline\

            \textit{\textit{\textbf{}}} & MSGARCH\_ML  & 1.087647 & 29.70\% & 1.088382 & 58.03\% & 14.593220 & 2.37\% & 95.65\% & 4.35\% \\
            \textit{\textit{\textbf{95\%}}} & MSGARCH\_MCMC  & 0.878021 & 34.87\% & 0.883464 & 64.29\% & 14.239989 & 2.71\% & 95.6\% & 4.4\% \\
            \textit{\textit{\textbf{}}} & RUGARCH\_Garch  & 0.093853 & 75.93\% & 0.481845 & 78.59\% & 11.342123 & 7.84\% & 94.85\% & 5.15\% \\
            \hline
             Levels & \textit{\textit{\textbf{1000AR(3)sGARCH\_sged}}}&&&&&&\\
             \hline
            \textit{\textit{\textbf{}}} &  MSGARCH\_ML  & 4.813361 & 2.82\% & 4.817377 & 8.99\% & 3.257706 & 77.59\% & 99.8\% & 0.2\% \\
            \textit{\textit{\textbf{99\%}}} & MSGARCH\_MCMC  & 4.813361\% & 2.82\% & 4.817377 & 8.99\% & 3.257608 & 77.59\% & 99.8\% & 0.2\% \\
            \textit{\textit{\textbf{}}} & RUGARCH\_Garch  & 2.352982 & 12.50\% & 2.369079 & 30.59\% & 2.123652 & 90.80\% & 99.6\% & 0.4\% \\


           \textit{\textit{\textbf{}}} & RUGARCH\_Garch  & 2.352982 & 12.50\% & 2.369079 & 30.59\% & 2.123652 & 90.80\% & 99.6\% & 0.4\% \\
           \hline
         \\
           \end{tabular}

           \textit{\textbf{\Large{NKY}}} \\
       \\
       \begin{tabular}{c c c c c c c c c c }
           \hline
           &&\textit{\textit{\textbf{UC\_test}}}&&\textit{\textit{\textbf{CC\_test}}}&&\textit{\textit{\textbf{DQ\_test}}}&&
           \textit{\textit{\textbf{BONTA'}}}&\textit{\textit{\textbf{}}}    \\

           \hline\

           Levels & \textit{\textit{\textbf{1000AR(0)eGARCH\_snorm}}}  & Statistic & P-value & Statistic & P-value & Statistic & P-value & P-value & Excess
           \\
           \hline\

           \textit{\textit{\textbf{}}} & MSGARCH\_ML  & 0.345711 & 55.66\% & 4.788150 & 9.13\% & 10.727594 & 9.72\% & 95.4\% & 4.6\% \\
           \textit{\textit{\textbf{95\%}}} & MSGARCH\_MCMC  & 0.345711 & 55.66\% & 4.788150 & 9.12\% & 10.722607 & 9.73\% & 95.4\% & 4.6\% \\
           \textit{\textit{\textbf{}}} & RUGARCH\_Garch  & 0.010125 & 99.9\% & 5.271144 & 7.17\% & 12.535435 & 5.10\% & 95\% & 5\% \\
           \hline
           Levels & \textit{\textit{\textbf{500AR(3)gjrGARCH\_std}}}&&&&&&\\
           \hline
           \textit{\textit{\textbf{}}} &  MSGARCH\_ML  & 0.943116 & 33.15\% & 0.979407 & 61.28\% & 0.913136 & 98.87\% & 99.8\% & 0.2\% \\
           \textit{\textit{\textbf{99\%}}}& MSGARCH\_MCMC & 0.943116 & 33.15\% & 0.979407 & 61.28\% & 0.909296 & 98.88\% & 99.8\% & 0.2\% \\
           \textit{\textit{\textbf{}}} & RUGARCH\_Garch  & 2.352982 & 12.50\% & 2.369079 & 30.59\% & 0.961339 & 98.70\% & 99.6\% & 0.4\% \\
           \hline
           \\
       \end{tabular}
```

```
\documentclass[12pt,a1paper]{report}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=0.5cm, right=0.5cm, top=0.5cm, bottom=0.5cm]{geometry}
\author{Francesca Tarchini}
\begin{document}
    \begin{table}[ht]
                \textit{\textbf{\Large{SPX}}} \\
                \\
                \begin{tabular}{c c c c c c c c }
                    \hline
                    \textit{\textit{\textbf{}}}&&\textit{\textit{\textbf{RUGARCH\_Garch}}}&&\textit{\textit{\textbf{MSGARCH\_ML}}}&&
                    \textit{\textit{\textbf{MSGARCH\_MCMC}}}&\textit{\textit{\textbf{}}}    \\

                    \hline
                    Levels & Models & Bontà & Excess & Bontà & Excess & Bontà & Excess   \\
                    \hline\
                    \textit{\textit{\textbf{}}} & 500.AR(0)sGARCH(1,1)sged & 93.6\% & 6.4\% & 94.4\% & 5.6\% & 94.4\% & 5.6\% \\
                    \textit{\textit{\textbf{}}} & 500.AR(0)gjrGARCH(1,1)sged & 94.6\% & 5.4\% & 94.8\% & 5.2\% & 94.8\% & 5.2\% \\
                    \textit{\textit{\textbf{}}} &  500.AR(1)eGARCH(1,1)norm & 94.4\% & 5.6\% & 94.8\% & 5.2\% & 94.6\% & 5.4\% \\
                    \textit{\textit{\textbf{95\%}}} & 500.AR(2)sGARCH(1,1)snorm & 94.2\% & 5.8\% & 95\% & 5\% & 94.8\% & 5.2\% \\
                    \textit{\textit{\textbf{}}} & 500.AR(3)eGARCH(1,1)sged & 94.8\% & 5.2\% & 95.6\% & 4.4\% & 93\% & 7\% \\
                    \textit{\textit{\textbf{}}} & 1000.AR(2)eGARCH(1,1)snorm & 94.9\% & 5.1\% & 95.1\% & 4.9\% & 95.1\% & 4.9\% \\
                    \textit{\textit{\textbf{}}} & 2000.AR(0)gjrGARCH(1,1)norm & 94.05\% & 5.95\% & 94.95\% & 5.05\% & 94.95\% & 5.05\% \\
                    \textit{\textit{\textbf{}}} & 2000.AR(3)sGARCH(1,1)norm & 94.15\% & 5.85\% & 95.1\% & 4.9\% & 95.1\% & 4.9\% \\
                    \textit{\textit{\textbf{}}} & 2000.AR(1)eGARCH(1,1)snorm & 95.5\% & 4.5\% & 95.5\% & 4.5\% & 95.5\% & 4.5\% \\

                    \hline
                    \textit{\textit{\textbf{}}} & 500.AR(0)sGARCH(1,1)snorm & 98.8\% & 1.2\% & 98.8\% & 1.2\% & 98.8\% & 1.2\% \\
                    \textit{\textit{\textbf{}}} & 500.AR(0)eGARCH(1,1)norm & 99\% & 1\% & 99\% & 1\% & 99\% & 1\% \\
                    \textit{\textit{\textbf{}}} & 500.AR(0)eGARCH(1,1)sged & 99.4\% & 0.6\% & 99.4\% & 0.6\% & 99.4\% & 0.6\% \\
                    \textit{\textit{\textbf{}}} & 500.AR(0)eGARCH(1,1)sged & 99.4\% & 0.6\% & 99.4\% & 0.6\% & 99.4\% & 0.6\% \\
                    \textit{\textit{\textbf{99\%}}} & 1000.AR(0)eGARCH(1,1)norm & 98.7\% & 1.3\% & 98.8\% & 1.2\% & 98.8\% & 1.2\% \\
                    \textit{\textit{\textbf{}}} & 1000.AR(1)eGARCH(1,1)snorm & 98.9\% & 1.1\% & 98.9\% & 1.1\% & 98.9\% & 1.1\% \\
                    \textit{\textit{\textbf{}}} & 1500.AR(3)eGARCH(1,1)sged & 99.07\% & 0.93\% & 99.2\% & 0.8\% & 99.2\% & 0.8\% \\
                    \textit{\textit{\textbf{}}} & 2000.AR(0)sGARCH(1,1)sged & 99.15\% & 0.85\% & 99.15\% & 0.85\% & 99.15\% & 0.85\% \\
                    \textit{\textit{\textbf{}}} & 2000.AR(2)eGARCH(1,1)sstd & 99.05\% & 0.95\% & 99.2\% & 0.8\% & 98.7\% & 1.3\% \\
                    \textit{\textit{\textbf{}}} & 2000.AR(3)sGARCH(1,1)std & 98.6\% & 1.4\% & 99\% & 1\% & 98.95\% & 1.05\% \\
                    \hline
                    \\
                \end{tabular}

        \textit{\textbf{\Large{NKY}}} \\
        \\
        \begin{tabular}{c c c c c c c c }
            \hline
            \textit{\textit{\textbf{}}}&&\textit{\textit{\textbf{RUGARCH\_Garch}}}&&\textit{\textit{\textbf{MSGARCH\_ML}}}&&
            \textit{\textit{\textbf{MSGARCH\_MCMC}}}&\textit{\textit{\textbf{}}}    \\

            \hline
            Levels & Models & Bontà & Excess & Bontà & Excess & Bontà & Excess   \\
            \hline\
            \textit{\textit{\textbf{}}} & 500.AR(0)sGARCH(1,1)sged & 93.6\% & 6.4\% & 93.8\% & 6.2\% & 93.6\% & 6.4\% \\
            \textit{\textit{\textbf{}}} & 500.AR(2)sGARCH(1,1)norm & 93\% & 7\% & 93.8\% & 6.2\% & 93.4\% & 6.6\% \\
            \textit{\textit{\textbf{}}} &   500.AR(2)gjrGARCH(1,1)sstd & 94\% & 6\% & 94.4\% & 5.6\% & 94.4\% & 5.6\% \\
            \textit{\textit{\textbf{95\%}}} & 500.AR(3)sGARCH(1,1)sged & 93.8\% & 6.2\% & 94\% & 6\% & 93.8\% & 6.2\% \\
            \textit{\textit{\textbf{}}} &   1000.AR(3)sGARCH(1,1)sstd & 94.1\% & 5.9\% & 94.3\% & 5.7\% & 94.1\% & 5.9\% \\
            \textit{\textit{\textbf{}}} & 2000.AR(1)eGARCH(1,1)sstd & 94.15\% & 5.85\% & 94.85\% & 5.15\% & 94.7\% & 5.3\% \\
            \textit{\textit{\textbf{}}} & 1500.AR(3)sGARCH(1,1)sged & 94\% & 6\% & 94.4\% & 5.6\% & 94.33\% & 5.67\% \\
            \textit{\textit{\textbf{}}} & 2000.AR(3)eGARCH(1,1)sstd & 94.15\% & 5.85\% & 94.9\% & 5.1\% & 94.45\% & 5.55\% \\
            \textit{\textit{\textbf{}}} & 1000.AR(3)sGARCH(1,1)std & 93.8\% & 6.2\% & 93.8\% & 6.2\% & 93.8\% & 6.2\% \\

            \hline
            \textit{\textit{\textbf{}}} & 500.AR(0)sGARCH(1,1)sstd & 99.6\% & 0.4\% & 99.6\% & 0.4\% & 99.6\% & 0.4\% \\
            \textit{\textit{\textbf{}}} & 500.AR(0)gjrGARCH(1,1)ged & 99.8\% & 0.2\% & 99.8\% & 0.2\% & 99.8\% & 0.2\% \\
            \textit{\textit{\textbf{}}} & 500.AR(1)eGARCH(1,1)sstd & 99.6\% & 0.4\% & 99.6\% & 0.4\% & 99.4\% & 0.6\% \\
```