



Università
Ca'Foscari
Venezia

Master's Degree
Programme
in
ECONOMICS AND
FINANCE

Final Thesis

Forecast Ability of Econometrics Models:
Univariate and Multivariate Analysis

Supervisor

Ch. Prof. Irene MAMMI

Graduand

Federico FIORESE
Matriculation number
851653

Academic Year

2019 / 2020

CONTENTS

1) Introduction	2
2) Data & Empirical Analysis	6
2.1) Data for Univariate Analysis	6
2.2) Data for Multivariate Analysis	10
3) Univariate Analysis	12
3.1) ARIMA processes	12
3.2) ARCH/GARCH processes	26
4) Multivariate Analysis	37
5) Results & Conclusions	49
5.1) Univariate Analysis	49
5.2) Multivariate Analysis	51
6) Bibliography	54
7) Appendix	56

INTRODUCTION

In the last decades of the twentieth century, the analysis of time series has been the subject of particular interest by statisticians and econometrics scholars.

The first fundamental contribution to this analysis is due to G.E.P. Box and G.M. Jenkins (1970), who focused their study on developing techniques for the analysis of the dynamic structure of a time series.

In fact, during their research they introduced the formulation of stochastic and dynamic processes that represent the behavior of the individual series, such as the simple Autoregressive process (AR), the simple Moving Average process (MA) and the mixed Autoregressive Moving Average process (ARMA), that includes both autoregressive and moving average components.

These techniques are based on the study of the correlation and dependences of the current value of the series with the past values; once we have captured the effect that an observation has on the following one, the models allow us to generate statistical forecasts on the values that the series will have in the future.

As theorized by E.F. Fama and K.R. French in 1988, the time series is composed of a stationary component and a random walk component. The first part can be decomposed in trend and seasonal components and due to the regularity of movements it is easiest part to be modeled and predicted. The second part is stochastic and not directly observable; for this reason is the hardest component that researchers have to estimate in the process.

The analysis of these components of time series is conducted through statistical tools and models that study the dynamic structure and evolution of the random variables over time, analyzing the series as a linear Difference Equation in which the dependent variable is the current value of y and the impulses variables are its past values (J.D. Hamilton, 1994).

In fact, these models allow us to analyze how the current value of the series is related to the evolution of its past values and to identify possible repetitive patterns that we can exploit to generate forecasts.

The analysis of these patterns and the study of the dynamic path of the time series are what makes the development of forecasts possible.

Of fundamental importance for the development of processes that capture the dynamic structure of the time series was the study of C.W.J. Granger in 1981, who, starting from the research of G.E.P. Box and G.M. Jenkins, studied the formulation of integrated models, such as the Autoregressive Integrated Moving Average process (ARIMA), and the relationships between different stochastic variables.

The first econometric models were formulated under the basic assumption that the conditional variance of the data is constant over time.

In 1982 R.F. Engle developed a new process that took into account the fact that the conditional variance of a variable could depend on its past values, introducing the Autoregressive Conditional Heteroscedastic process (ARCH).

The concept of non-constant variance had already been introduced in 1963 by B. Mandelbrot, who studied the behavior and statistical distribution of the prices of a time series, noting how the large price movements that used to happen over time were not isolated, but they were followed by other equally large movements. This phenomenon took the name of "Persistence of Moves"; nowadays in finance it is called "Volatility Clustering".

This phenomenon, a symptom of heteroskedasticity, can be captured and described by ARCH processes and GARCH processes.

The GARCH models, introduced by T. Bollerslev in 1986, are a generalization of the ARCH process, but allow longer memory and a more flexible lag structure. In the formulation of GARCH process the conditional variance is affected by the past error terms and by its lagged values.

In the analysis of time series it is essential not only to study how the value of a stochastic variable is related to its past values, but also how it is related to the values of other variables.

Fundamental contribution to the analysis of the relationships between different time series is due to C.W.J. Granger. In fact, in the 1981, he studied the fact that the combination of a vector of integrated time series could produce processes that are stationary. This concept is the basis of the study of cointegration relationships among time series.

In the following years many researchers developed this concept, such as J.H. Stock and M.W. Watson in 1986, R.F. Engle and C.W.J. Granger himself in 1987, and R.F. Engle and B.S. Yoo in 1987.

In particular, R.F. Engle and C.W.J. Granger contributed to this analysis by developing a representation theorem, based on the concept of integration and on the formulation of ARMA processes.

We will discuss about this concept in the "Multivariate Analysis" chapter.

This thesis is based on these studies and research, asking questions about the actual possibility of developing satisfactory forecasts on the future price values of financial securities, in this case equity securities, through the application of these econometric models.

In order to produce forecast about the future values of a variable, we have to capture the past information within the time series. The basic assumption is that the values are serially correlated, which means that the past values affect the present and the future ones.

Under this assumption we can formulate models that describe the dynamic structure of the stochastic processes, with the aim of predicting their future evolution.

The purpose of this thesis is to investigate and measure the accuracy of the forecasts that can be produced by the econometric models developed over the years.

We want to question the effective applicability of these processes and the study of relationships between multiple variables in generating short/medium-term forecasts that are accurate enough to be used in a trading strategy, by carrying out an analysis on real data, such as daily stock price.

Through this study we test the hypothesis that it is possible to predict, with a certain level of precision, the future value of the equity securities of big firms, such as JPM, INTC, WMT, ADBE and ADSK, analyzing the dynamic path of the stock price.

We take in consideration univariate models, such as ARMA processes and ARCH/GARCH processes, in order to study the evolution of the time series as a function of its past values, and multivariate models, such as the cointegration study and the estimation of Vector Autoregressive models.

The first part of the study is based on the hypothesis that the forecast obtained through the application of univariate models is more accurate if it is implemented on equities characterized by lower volatility levels compared to securities with higher volatility. Subsequently, the data has been divided into sub-samples to highlight whether there is a substantial difference in the accuracy of prediction if it is made using the whole sample of data or through the use of sub-samples.

The second part of the thesis explores the possibility that two equities, both components of the same stock index and operating in the same industry, are cointegrated, which means that they share a common stochastic trend and for this reason they have a long-term equilibrium relationship. So we focus on the study of this equilibrium and on the analysis of the short-term relationship between the two time series, estimating the parameters of the error correction model (R.F. Engle, C.W.J. Granger, 1986).

The formulation of this model is based on the fact that the residuals of the cointegration relationship are described by a stationary process, therefore mean-reverting.

The equity securities that we analyze are ADBE stock and ADSK stock, both operating in the technology industry.

In order to deeply analyze the relationship between the two stocks, we develop a Vector Autoregressive model (VAR), which allows us to understand the dynamic evolution of the two series starting from their joint study. It is precisely the joint analysis of the past values of both time series that allows us to understand how the dynamic structure of one variable can influence the evolution of the other one.

In this case, the future value of the stochastic variable is a function not only of its lagged values, but also of the past and current values of the other variable.

We consider the individual time series as components of a multivariate process and analyze the two stochastic series jointly.

At this point we decide to investigate the hypothesis that a forecast generated by the formulation of a VAR model, considered as an extension of the ARMA model, is more accurate than a forecast obtained through the formulation and estimation of univariate models; in fact, in this case, we use also the additional information given by the related series to produce forecasts for each variable.

The purpose of this part of analysis is to test how much the forecast that we can obtain from the application of a multivariate process is accurate when the stochastic variables that we are studying are the prices of equity securities.

The analysis of the relationship between the two stocks could lead to the develop trading strategy if the prediction is accurate enough.

It is noteworthy that the existence of a cointegration relationship between two financial securities and the following formulation of the error correction model is the basis for the application of a statistical arbitrage strategy known as "pairs trading". In fact, this strategy is based on the

exploitation of the deviations of the residuals from their expected value in order to capitalize on the subsequent reabsorption of the short-term disequilibrium.

The computational analysis is done using the software R.

The codes are listed in the Appendix. In the univariate analysis part, only the codes regarding the JPM stock are shown, as they are the same as those regarding the INTC stock and WMT stock.

DATA & EMPIRICAL ANALYSIS

Data for Univariate Analysis

For the univariate analysis I examined the historical series of three equities characterized by different levels of volatility, measured by conditional standard deviation. After obtaining the adjusted close prices we apply the logarithmic transformation, in order to make all the data homogeneous.

The equity securities examined are:

JPMorgan Chase & Co. (JPM), listed on New York Stock Exchange (NYSE)

Intel corporation (INTC), listed on Nasdaq-100

Walmart Inc. (WMT), listed on New York Stock Exchange (NYSE)

The data sample consists of daily Adjusted Close Price, starting from 2nd January 2009 to 31th December 2019 for a total of 2768 observations.

In order to estimate, in primis, the econometric models in question and subsequently evaluate their forecasting capabilities, the period under consideration has been divided into two samples: the first part, from 2nd January 2009 to 31th December 2018, used for the estimation, the second, corresponding to the whole year 2019, as a test to compare the real data and the data generated by the forecast method to evaluate its actual accuracy.

Furthermore, as regards the estimation of the ARMA process, the whole sample has been subsequently divided into 5 sub-samples in order to investigate the hypothesis that using a more restricted period for estimating the model causes better fit to the data, therefore a more accurate forecast.

In this case, a Rolling Window procedure has been used for this sub-samples division, in which a period of 6 years is used for the formulation of the model that best fits the data, and a period of 1 year for the comparison between real values and forecast values.

The sub-samples are composed as follows:

2nd January 2009 – 31th December 2014 for the estimation, whole 2015 for the test;

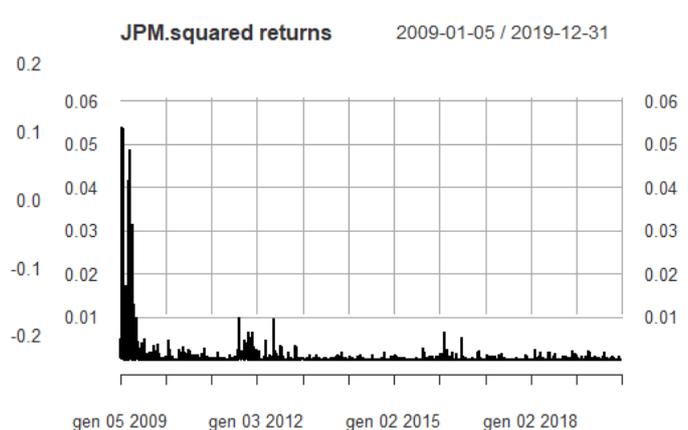
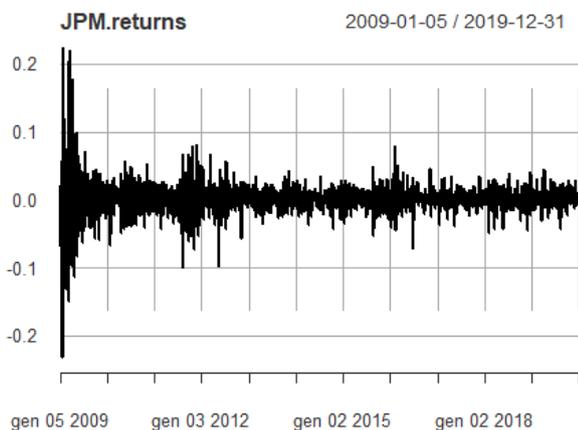
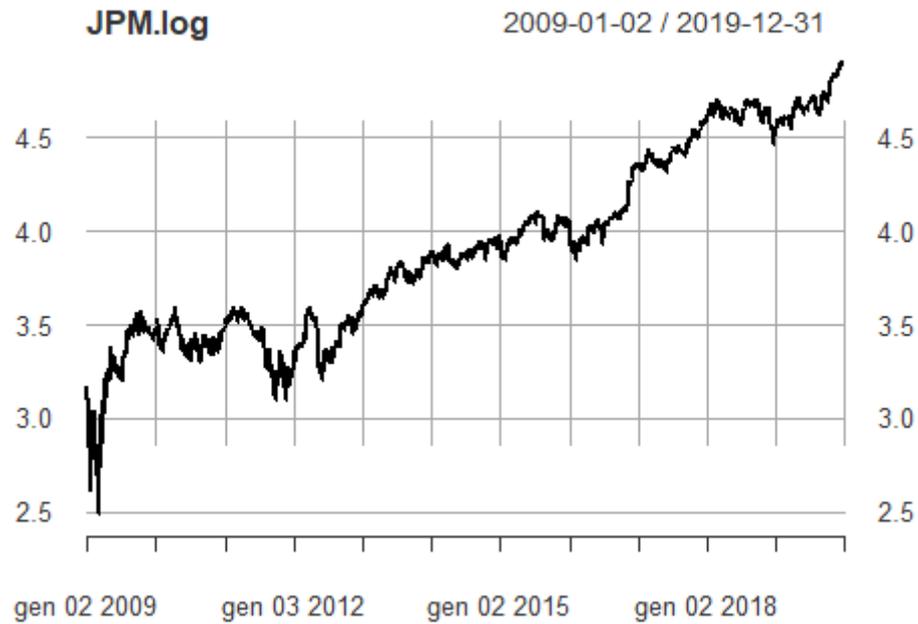
2nd January 2010 – 31th December 2015 for the estimation, whole 2016 for the test;

2nd January 2011 – 31th December 2016 for the estimation, whole 2017 for the test;

2nd January 2012 – 31th December 2017 for the estimation, whole 2018 for the test;

2nd January 2013 – 31th December 2018 for the estimation, whole 2019 for the test.

Once we compose the samples for the estimation of the models and for the forecast evaluation, we begin analyzing the plots of the stock prices, of the logarithmic returns and squared returns.

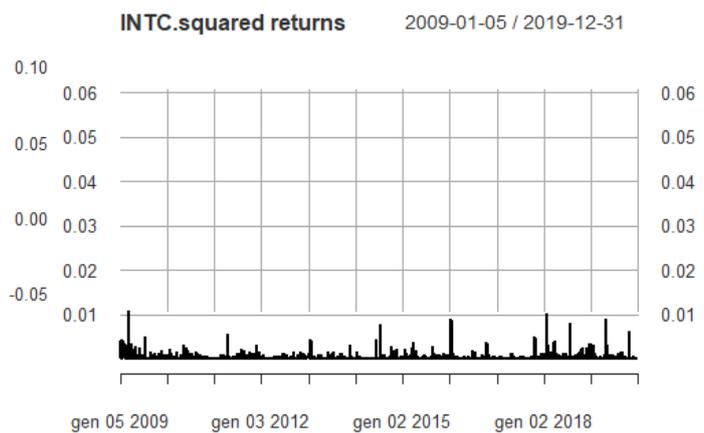
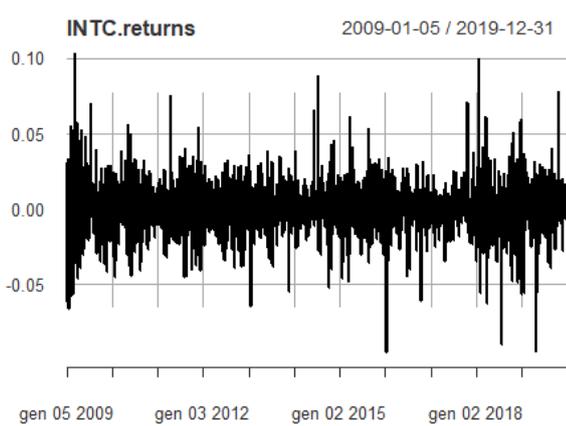
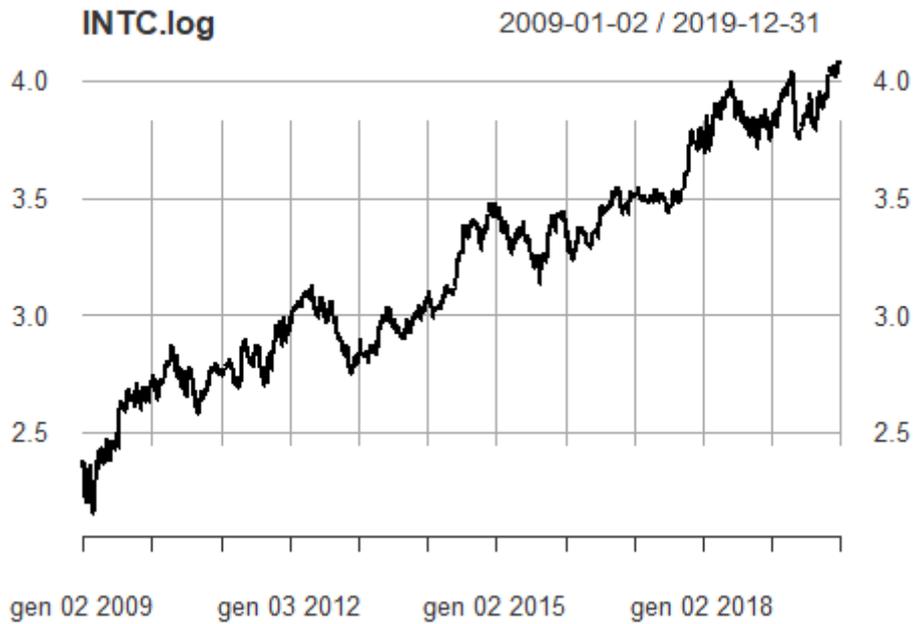


From the plot of the daily logarithmic Close Price of JPM we can see that, after the period of crisis that hit the financial market in 2008 and a consequent lateral movement of the value of the security from 2009 to 2012, it then had a growing trend until the end of 2019.

But more important for the purposes of the study in question is the analysis of the plot of daily logarithmic returns and squared returns: from these plots it appears that the returns are not constant around the average value, but there are several moments in which the deviation is high and moments in which it is more reduced. From the graphical analysis of the returns, in addition to the presence of volatility clusters, we can notice that there is also an asymmetry between negative and positive deviations; the negative deviations are larger than the positive ones.

By the expression volatility clustering (Mandelbrot, 1963) we mean the trend of large changes in price to be followed by large changes and of small changes to be followed by small changes; This phenomenon is a symptom of the presence of autocorrelation in returns and of Heteroskedasticity.

The absence of constancy and homogeneity in returns can also be seen from the squared returns plot, in which periods of higher volatility can be identified.



From the plot of the daily logarithmic close price we can observe a growth trend in the value of the INTC share for the whole period of time examined.

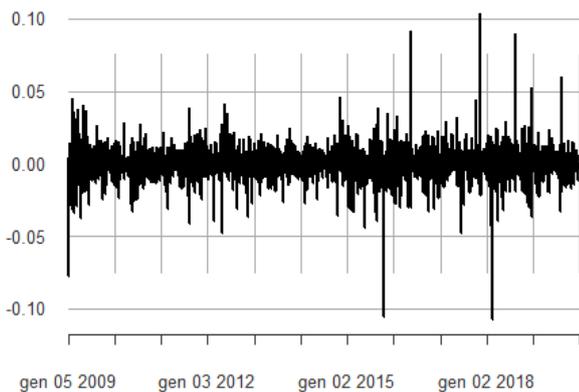
Using the plot of the returns we can observe a certain symmetry between positive and negative deviations from the average value, with peaks in both cases, although more contained than those observable in the plot of JPM returns.

As can be seen from the third graph, in fact, there are moments in which the returns clearly differ from the average, highlighting the presence of periods in which the volatility is higher.

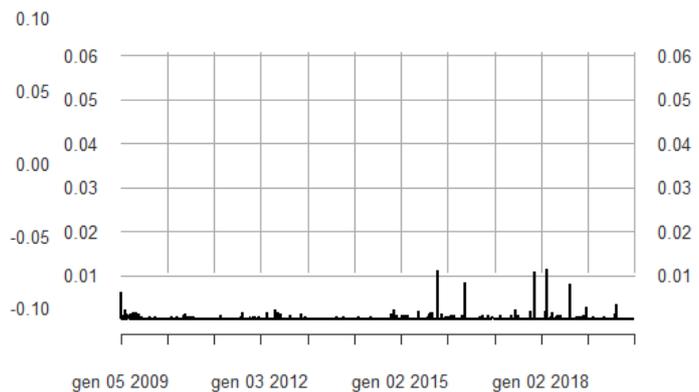
WMT.log 2009-01-02 / 2019-12-31



WMT.returns 2009-01-05 / 2019-12-31



WMT.squared returns 2009-01-05 / 2019-12-31



Also in this case, if we analyze the plot of the WMT share, we notice an initial period of laterality until mid-2011 followed by a growth trend of the stock value.

In the first period, returns are fairly constant around the expected value, then we can see both negative and positive peaks. Looking at the graphs of returns and squared returns and comparing them with the price plot we can see that the volatility peaks occur in periods of sharp fall or sharp rise in the value of the security, such as in the period from 2015 to 2018.

This is consistent with expectations that a period of high price movement is accompanied by an increase in investor frenzy, therefore by an increase in the sales and purchases of the security and an increase in volatility, and vice versa.

Once we have analyzed the three titles graphically we can compute and study the basic statistics of the daily returns, such as the mean, the variance and the standard deviation, the skewness and kurtosis values.

	JPM.returns	INTC.returns	WMT.returns
nobs	2767.00	2767.00	2767.00
mean	0.000630	0.000619	0.000364
variance	0.000455	0.000265	0.000125
sd	0.021335	0.016293	0.011188
skewness	0.606468	0.003510	-0.201477
excess of kurtosis	22.353340	4.045254	13.638424

We can notice that JPM and INTC stocks generate similar daily returns but INTC has a lower volatility, measured by variance and standard deviation; Instead WMT is the security with the lowest risk and return values.

It is also important to observe the values of skewness and kurtosis, measured as an excess from the typical value of the normal distribution. These values in fact provide us with information on the distribution of data useful for the formulation of the econometric processes in question.

A nonzero value of skewness means that the returns of the security are non-symmetrical: if the value is positive, the security has a higher concentration of negative returns, if the value is negative, the security has a higher concentration of positive returns.

The kurtosis value measures the fatness of the tails of the distribution: the higher the value, the fatter the tails and the more likely extreme events are.

If the data has a leptokurtic distribution, volatility can be modeled appropriately by ARCH/GARCH processes.

We can observe that the JPM stock, besides being the one with the most volatility, has a high positive skewness value and an even higher kurtosis value; this means that returns are negatively asymmetric, as we have seen in the first plots, and the data focuses heavily on the tails of the distribution. This indicates that the stock is very risky.

Instead, the INTC security has an almost zero skewness value, so the data are symmetrical; the kurtosis value however indicates that the data does not follow a normal distribution, but has fat tails.

The WMT stock has a negative skewness value; this means that positive deviations from the mean are higher than negative deviations. Also in this case the kurtosis value is quite high.

Data for Multivariate Analysis

As for the part of the analysis previously carried out, also in this case the data taken into consideration for the joint study of the variables are the daily Adjusted Close Price of two shares, both listed on Nasdaq-100:

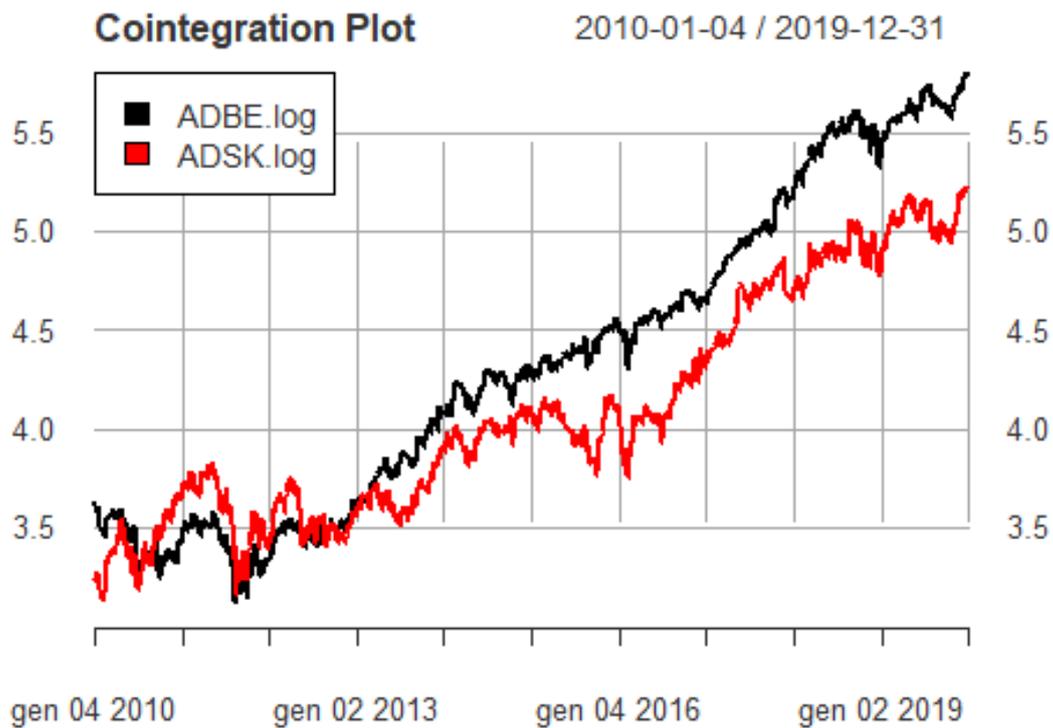
Adobe Inc. (ADBE)

Autodesk Inc. (ADSK)

The data has been transformed through logarithm.

The period considered for the cointegration analysis starts from January 2010 to December 2019; for the estimation of the VAR (Vector Autoregressive model) and the evaluation of the forecast generated by the process we have divided the sample into two parts: the first one starts from 2nd

January 2010 until 31st December 2018 for estimating the model, the second one from 2nd January 2019 until 31st December 2019 to test the forecasting ability by measuring its accuracy. As a first step for the study of the cointegration of two time series we can analyze the plot of the daily logarithmic Close price of the two stocks.



During the period considered we can see that both stocks, after an initial period of lateral movement, show an increasing trend.

Furthermore, we can see that the stocks show similar upward and downward movements. In fact, it seems that they share a common stochastic trend (J.H. Stock & M.W. Watson, 1988), which is the basis for the cointegration relationship.

The hypothesis of cointegration between the two securities will then be investigated through cointegration tests, such as the Phillips-Ouliaris test and the Johansen test, in order to analyze the Long-run Equilibrium relationship of the two time series.

In this thesis we decide to test the existence of cointegration between these two stocks, due to the fact that they belong to the same industry, so we can expect them to share the same systemic risk. This means that a Long-run Equilibrium relationship between the prices of the stochastic variables is more likely to exist.

UNIVARIATE ANALYSIS

ARIMA processes

In order to analyze the composition and the evolution of the time series, we must formulate econometric processes that study the dynamic structure of the data.

These models capture the dynamic evolution of the stochastic variable and they can be used to make probabilistic predictions on the future values of the series.

The theorization and formulation of the first models dates back to the studies of the statisticians G. Box and G.M. Jenkins in 1970, and from that moment on various processes, more or less elaborate, were developed with the aim to be able to model the characteristics of a stochastic variable as accurately as possible.

The simplest processes are:

Autoregressive process (AR)

Moving Average process (MA)

Mixed Autoregressive Moving Average process (ARMA)

In the moment in which we model the current value of a stochastic variable as a function of its lagged values, we are formulating an Autoregressive process.

The simplest form of an Autoregressive process is the AR (1) process; this model represents the value of the time series as a function of its immediately preceding value.

$$y_t = m + \alpha y_{t-1} + \varepsilon_t$$

In this formula m is a constant, the coefficient α measures the weight of the lagged value on the current one, and ε_t is the innovation term. The coefficients m and α are estimated in the process, however the error term ε_t is not directly observable. The error term is considered a White Noise process, which means that it has zero mean, finite variance and it is not serially correlated.

For the purpose of analyzing the time series, it is important to observe whether the coefficient $|\alpha|$ is <1 . In fact, this indicates whether the series is weakly stationary or non-stationary.

By the term weakly stationary we mean a stochastic series in which the dynamic structure and the moments are not influenced by time; this series is characterized by constant mean, finite variance and covariance that are not dependent on time.

The AR (1) process can be expanded by adding lagged values in the function.

Another model that can describe the structure of the time series is the Moving Average process. In this case the current value of the variable is a function of the current and lagged values of the White Noise processes. The simplest formulation is the MA(1):

$$y_t = m + \varepsilon_t + \alpha \varepsilon_{t-1}$$

Where m is a constant and α is the coefficient of the previous value of the disturbance term.

The MA process is stationary by definition, due to the fact that it is a linear combination of White Noise innovation terms.

Also the MA(1) can be extended by adding lagged values of error terms.

The Autoregressive process and the Moving Average process are linked by a structural relationship; in fact, for example, an AR (1) process, if $|\alpha| < 1$, can be rewritten as an infinite MA process.

This feature makes possible to combine the two processes, if they are stationary and invertible. From this we obtain the mixed Autoregressive Moving Average process (ARMA), which combines the specific characteristics of the two previous models. The representation of an ARMA (1,1) process is:

$$y_t = \alpha y_{t-1} + \varepsilon_t - \beta \varepsilon_{t-1}$$

At this point we can formulate the ARMA model that best fits the data.

Consistent with the Box-Jenkins 3-steps approach, the choice of the model is based on the study of the time series and its structural characteristics.

Subsequently, we perform a comparison of the Information Criteria in order to identify the model that best describes the data; the analysis takes into account both the accuracy of the model in adapting to the data, measured as loglikelihood, and the number of parameters used, according to the principle of parsimony.

In the case that, by testing the stationarity of the process, the regressive part of the process shows a unitary root, which is a symptom of non-stationarity, we can expand the ARMA process into an ARIMA process (Autoregressive Integrated Moving Average) that can deal with non-stationary component.

In order to test whether the process specification is correct with respect to the data, it is necessary to perform a diagnostic test on the model residuals. The residuals must be stationary and not serially correlated.

So we begin testing for the stationarity of each sub-samples using the Phillips-Perron test: we can observe that the time series are non-stationary but they become stationary if we compute their first difference. So the series are integrated of order one. This is a common situation when dealing with stocks, in fact the time series of their prices is usually non-stationary, while the returns are stationary and mean-reverting.

In order to estimate the best process for each sub-sample under examination, we use the *"auto.arima"* function, which automatically compares the Information Criteria of multiple models and produces the estimate of the parameters of the model that has the lowest AIC value.

JPM

For the first sub-sample, that starts in January 2009 and finishes in December 2014, the process that better fits the data is an ARIMA(3,1,1) process:

```

Series: JPM.log.20092015
ARIMA(3,1,1) with drift

Coefficients:
      ar1      ar2      ar3      ma1  drift
 0.6081  0.0899 -0.0543 -0.7301 6e-04
s.e.  0.1379  0.0347  0.0272  0.1363 5e-04

sigma^2 estimated as 0.000678:  log likelihood=3366.39
AIC=-6720.78  AICC=-6720.72  BIC=-6688.86

```

We can notice that the coefficients of the first autoregressive component and of the moving average component are high, while the coefficients of the second and third autoregressive components are low; the value of the constant is very low. This means that the current value of the series is affected by its own previous values, especially by the value of the immediately preceding period, and by the value of the lagged disturbance term.

The diagnostic tests, such as the Phillips-Perron test and Box-Ljung test, show that the residuals are stationary and serially uncorrelated.

The process that better fits the data of the second sub-samples is:

```

Series: JPM.log.20102016
ARIMA(1,1,1)

Coefficients:
      ar1      ma1
-0.8215  0.7654
s.e.  0.1905  0.2160

sigma^2 estimated as 0.0003002:  log likelihood=3979.71
AIC=-7953.43  AICC=-7953.41  BIC=-7937.47

```

In this case the data are represented by an ARIMA(1,1,1) process. The coefficient of the components are very high, and the coefficient of the autoregressive part is negative so the previous value of the variable negatively affects the current value.

The diagnostic tests done on the residuals of the model show no signs of non-stationarity and serial correlation. The model is correctly formulated.

Also the data of the third sub-sample follows a ARIMA(1,1,1) process:

```

Series: JPM.log.20112017
ARIMA(1,1,1)

Coefficients:
      ar1      ma1
-0.8954  0.8548
s.e.  0.0624  0.0724

sigma^2 estimated as 0.0002785:  log likelihood=4036.26
AIC=-8066.51  AICC=-8066.5  BIC=-8050.55

```

As in the previous sub-sample, also in this case the coefficients of the process are high and the coefficient of the autoregressive part is negative.

In the case of the fourth sub-sample it seems that the best model is an ARIMA(0,1,0) process:

```
Series: JPM.log.20122018
ARIMA(0,1,0) with drift

Coefficients:
    drift
    9e-04
s.e.    4e-04

sigma^2 estimated as 0.0001878:  log likelihood=4330.13
AIC=-8656.26  AICC=-8656.26  BIC=-8645.63
```

So the current value of the series is not affected by its own lagged values or by the lagged error terms.

Also in the case of the last sub-sample the series follows an ARIMA(0,1,0) process:

```
Series: JPM.log.20132019
ARIMA(0,1,0) with drift

Coefficients:
    drift
    6e-04
s.e.    3e-04

sigma^2 estimated as 0.0001674:  log likelihood=4419.99
AIC=-8835.98  AICC=-8835.97  BIC=-8825.34
```

The diagnostic tests done on the residuals of each process shows that the models are correctly formulated. The residuals are stationary and serially uncorrelated.

INTC

We apply the same procedure also for the INTC stock.

The process that better describes the data of the first sub-sample is an ARIMA(3,1,0) process:

```
Series: INTC.log.20092015
ARIMA(3,1,0) with drift

Coefficients:
    ar1    ar2    ar3  drift
-0.0484  0.0160  0.0217  7e-04
s.e.    0.0257  0.0258  0.0259  4e-04

sigma^2 estimated as 0.0002696:  log likelihood=4061.68
AIC=-8113.35  AICC=-8113.31  BIC=-8086.76
```

In this case the coefficients of the autoregressive components are low, so we can say that the lagged values of the variable have a low impact on the current value. In the data there are no evidence of a moving average component.

The data of the second and third sub-samples follow an ARIMA(0,1,0) process:

```
Series: INTC.log.20102016
ARIMA(0,1,0)
```

```
sigma^2 estimated as 0.0002205: log likelihood=4211.4
AIC=-8420.8 AICc=-8420.79 BIC=-8415.48
```

```
Series: INTC.log.20112017
ARIMA(0,1,0)
```

```
sigma^2 estimated as 0.0002128: log likelihood=4238.29
AIC=-8474.58 AICc=-8474.58 BIC=-8469.26
```

The lagged values of the stochastic variable do not affect its current value.

The data of the fourth sub-sample follows an ARIMA(1,1,0) process:

```
Series: INTC.log.20122018
ARIMA(1,1,0) with drift
```

```
Coefficients:
      ar1  drift
    -0.0140 5e-04
s.e.   0.0258 3e-04
```

```
sigma^2 estimated as 0.000183: log likelihood=4350.05
AIC=-8694.1 AICc=-8694.08 BIC=-8678.14
```

In this case the autoregressive component is present but, due to the fact that the coefficient is low, the effect on the current value of the variable is small.

At last, the fifth sub-sample can be described by an ARIMA(2,1,2) process:

```
Series: INTC.log.20132019
ARIMA(2,1,2) with drift
```

```
Coefficients:
      ar1      ar2      ma1      ma2  drift
    -0.2323 -0.9111  0.1919  0.9136 6e-04
s.e.   0.0619  0.0395  0.0610  0.0383 4e-04
```

```
sigma^2 estimated as 0.0002262: log likelihood=4194.53
AIC=-8377.06 AICc=-8377.01 BIC=-8345.15
```

The coefficients of the autoregressive components are negative and high, especially the coefficient of y_{t-2} ; this means that the past values have a significative influence on the current value of the stochastic series. Also the coefficients of the moving average part are high. So we can say that the lagged values of the variable and the lagged values of the error term highly affect y_t .

The diagnostic tests show that all the models are correctly formulated.

WMT

The model that better fits the data of the first sub-sample is an ARIMA (1,1,2):

```

Series: WMT.log.20092015
ARIMA(1,1,2) with drift

Coefficients:
      ar1      ma1      ma2  drift
      0.93  -0.9868  0.0515  4e-04
s.e.   NaN      NaN  0.0256  2e-04

sigma^2 estimated as 0.0001029:  log likelihood=4788.8
AIC=-9567.6  AICC=-9567.56  BIC=-9541

```

The coefficients of y_{t-1} and ε_{t-1} are very high, so the lagged values of the variable and of the error term have a large impact on the present value of y . Instead, the coefficient of the second component of the moving average is small.

The data of the other sub-samples are described by ARIMA(0,1,0) processes:

```

Series: WMT.log.20102016
ARIMA(0,1,0)

sigma^2 estimated as 9.954e-05:  log likelihood=4811.57
AIC=-9621.14  AICC=-9621.13  BIC=-9615.82

```

```

Series: WMT.log.20112017
ARIMA(0,1,0)

sigma^2 estimated as 0.0001108:  log likelihood=4730.63
AIC=-9459.25  AICC=-9459.25  BIC=-9453.93

```

```

Series: WMT.log.20122018
ARIMA(0,1,0) with drift

Coefficients:
      drift
      4e-04
s.e.   3e-04

sigma^2 estimated as 0.0001139:  log likelihood=4707.03
AIC=-9410.06  AICC=-9410.05  BIC=-9399.43

```

```

Series: WMT.log.20132019
ARIMA(0,1,0)

sigma^2 estimated as 0.0001344:  log likelihood=4584.77
AIC=-9167.53  AICC=-9167.53  BIC=-9162.21

```

This means that the current values of the stochastic variable are not affected by its own lagged values and by the lagged values of the disturbance term.

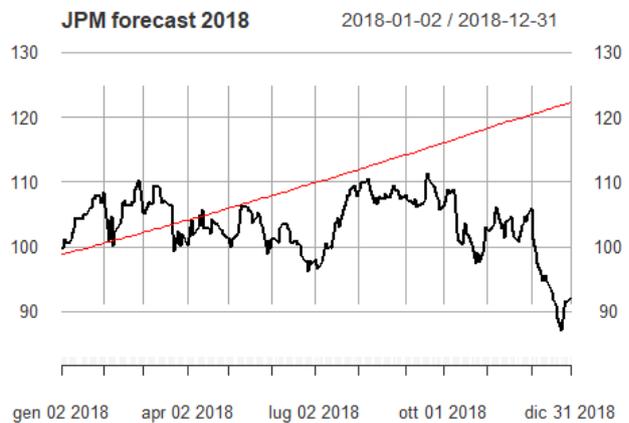
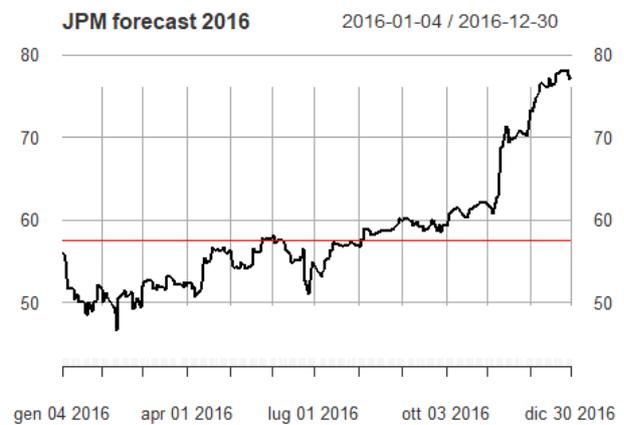
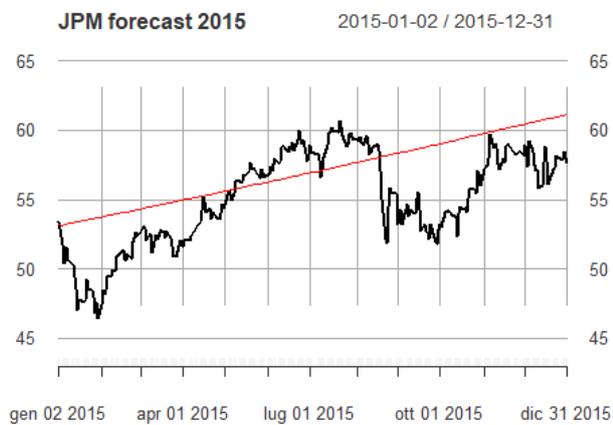
The residuals of the models are stationary and uncorrelated.

The purpose of this chapter is to elaborate forecasts using the processes previously formulated and test their accuracy with real data.

The forecast is based on the information about the structure of the time series that we obtain analyzing its dynamic evolution over the period.

In the next part we expose the graphical representations of the forecasted values, in which we can observe the comparison and the difference between the real data and the prevision of the future evolution of the stock prices.

JPM





As we can see from these plots there are moments in which the process used to forecast is able to capture the evolution of the price over time, for example some growth trends or some lateral movements; but there are also cases in which the forecast produced is not very accurate if compared to the real growth of the stock price.

However it is important to notice that the forecast seems to be more accurate in the first half of the time period, as we can reasonably expect.

Now we can analyze the accuracy of the forecast methods computing the measures of accuracy, such as the Mean Absolute Error, the Mean Squared Error, the Root-Mean-Square Error and the Mean Absolute Percentage Error.

These measures compute the deviation between the real data and the data obtained through the application of the ARIMA processes.

Lower is the values of the measures, higher is the accuracy of the prevision.

	MAE	MSE	RMSE	MAPE
JPM. arima(3,1,1).2015	0.05059132	0.00361274	0.06010607	0.01271047
JPM. arima(1,1,1).2016	0.08776988	0.01401355	0.11837885	0.02138207
JPM. arima(1,1,1).2017	0.08318217	0.01184045	0.10881381	0.01852691
JPM. arima(0,1,0).2018	0.07896711	0.01073035	0.10358740	0.01711784
JPM. arima(0,1,0).2019	0.09041531	0.01156120	0.10752301	0.01904120

We can notice that the value of the forecast accuracy is similar in each period, with the exception of the first period in which the forecast is more accurate.

Then we decide to use a wider period for the estimation of the model to test the hypothesis that if we use smaller samples we do not run into structural breaks and the accuracy of the processes is higher. In fact we use the whole sample, from January 2009 to December 2018, to generate the process that describe the structure of the data.

```
Series: JPM.log.20092019
ARIMA(3,1,1) with drift
```

Coefficients:

```

      ar1      ar2      ar3      ma1  drift
0.6187  0.0748 -0.0469 -0.7239 6e-04
s.e.  0.1193  0.0267  0.0212  0.1182 3e-04
```

```

sigma^2 estimated as 0.0004807:  log likelihood=6041.58
AIC=-12071.16  AICC=-12071.13  BIC=-12036.18
```

In this case the model is an ARIMA(3,1,1) process.

The coefficients of the components are very similar to the ones estimated in the first sub-sample.

Subsequently we use this process to generate the forecasted values. We use the whole 2019 as sample to test the difference between the real data and the forecasted ones.



We can see that the process is able to capture the growth trend of the stock price, but the spread between the real values and the forecast is large.

As we can notice from the table below, the forecast obtained by the use of the whole sample is the less accurate.

	MAE	MSE	RMSE	MAPE
JPM. arima(3,1,1).2015	0.05059132	0.00361274	0.06010607	0.01271047
JPM. arima(1,1,1).2016	0.08776988	0.01401355	0.11837885	0.02138207
JPM. arima(1,1,1).2017	0.08318217	0.01184045	0.10881381	0.01852691
JPM. arima(0,1,0).2018	0.07896711	0.01073035	0.10358740	0.01711784
JPM. arima(0,1,0).2019	0.09041531	0.01156120	0.10752301	0.01904120
JPM. arima(3,1,1).wholesample	0.10090492	0.01401146	0.11837000	0.02125800

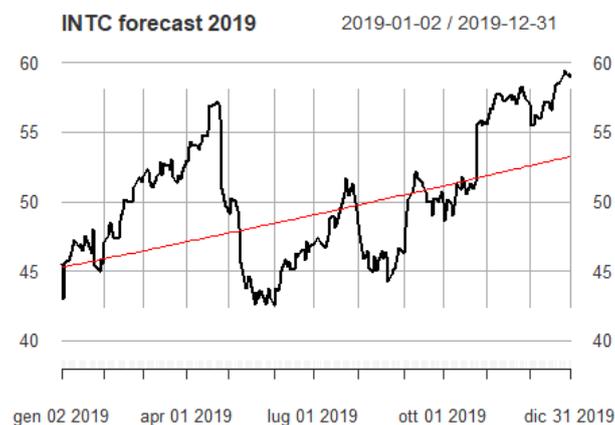
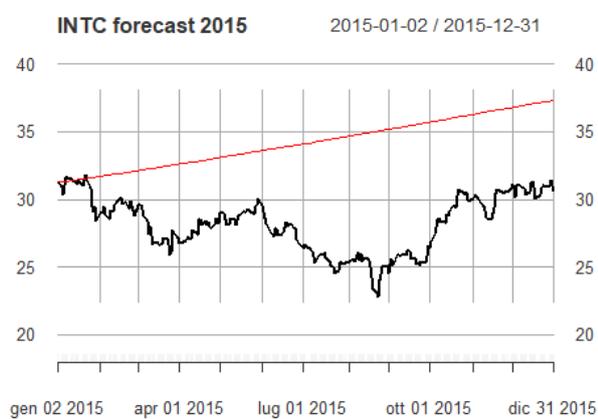
INTC

If we analyze the plots below, we can observe that the forecast method is not very accurate in the first period; in fact it predicts a growth trend for the prices, while the real data exhibits more or less a lateral movement.

Also the forecasted values obtained in the second period are not close to the real values.

Observing the third and fourth plots we can see that in the first part of the period the forecast is quite accurate, while, in the last part, it is not able to capture the evolution of the stock price.

The last period's forecast seems to be the most accurate; in fact the forecasted values are close to the real values of the variable.



So we use the accuracy measures to compare the precision of the forecast; the results are consistent with what we have observed from the plots.

The forecast of the values of the last period seems to be the most accurate.

	MAE	MSE	RMSE	MAPE
INTC.arima(3,1,0).2015	0.19677725	0.047121935	0.21707587	0.05955682
INTC.arima(0,1,0).2016	0.07786017	0.007708528	0.08779822	0.02305465
INTC.arima(0,1,0).2017	0.06749627	0.012451951	0.11158831	0.01835969
INTC.arima(1,1,0).2018	0.07126042	0.006574566	0.08108370	0.01848651
INTC.arima(2,1,2).2019	0.06789653	0.006398816	0.07999260	0.01729144

Then we estimate the ARMA process for the whole sample. The structure of the data are described with an ARIMA(3,1,3) process.

The coefficients of both the autoregressive part and the moving average part are high, so we can say that the lagged values have a quite large impact on the current value of the time series.

```
Series: INTC.log.20092019
ARIMA(3,1,3) with drift

Coefficients:
      ar1      ar2      ar3      ma1      ma2      ma3      drift
    -0.6851  0.5711  0.717  0.6390 -0.6095 -0.6950  6e-04
s.e.   1.1332  0.1987  0.658  1.2151  0.1843  0.7738  3e-04

sigma^2 estimated as 0.0002618:  log likelihood=6806.77
AIC=-13597.55  AICC=-13597.49  BIC=-13550.91
```

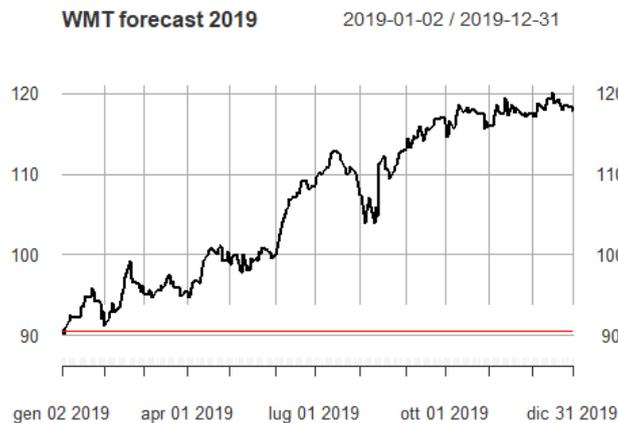
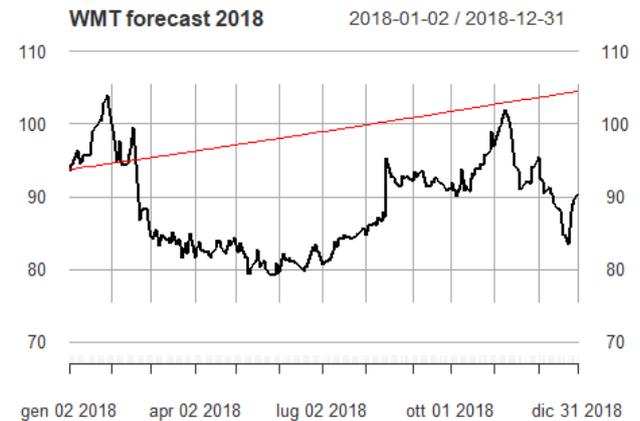
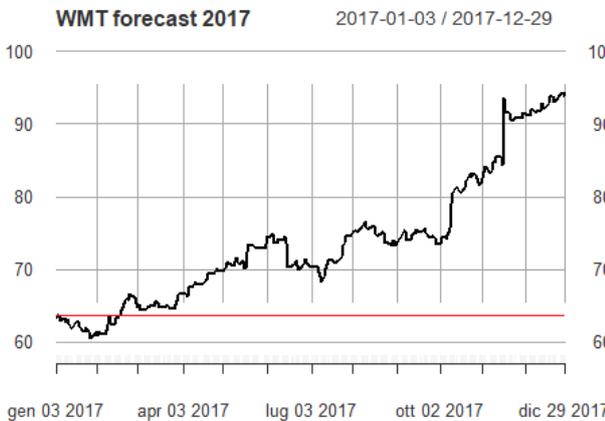
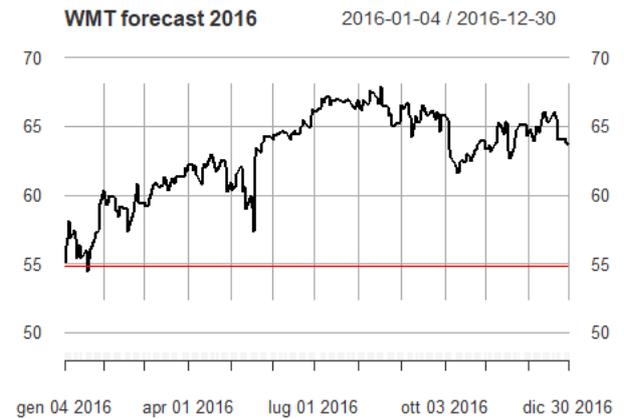
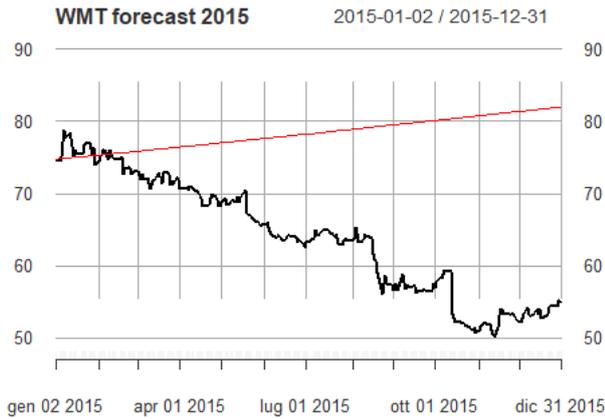
From the plot below we can see that the forecast of the last period is quite similar to the one previously obtained using the sub-sample from January 2013 to December 2018.



The last two measures of accuracy are very similar; there is not a big difference in using the whole sample or the sub-sample.

	MAE	MSE	RMSE	MAPE
INTC.arima(3,1,0).2015	0.19677725	0.047121935	0.21707587	0.05955682
INTC.arima(0,1,0).2016	0.07786017	0.007708528	0.08779822	0.02305465
INTC.arima(0,1,0).2017	0.06749627	0.012451951	0.11158831	0.01835969
INTC.arima(1,1,0).2018	0.07126042	0.006574566	0.08108370	0.01848651
INTC.arima(2,1,2).2019	0.06789653	0.006398816	0.07999260	0.01729144
INTC.arima(3,1,3).wholesample	0.06858748	0.006650867	0.08155285	0.01743698

WMT



In the case of the WMT stock, we can observe that the forecasts are not very accurate; the ARMA processes used to describe the dynamic evolution of the time series are not able to capture the structure of the data.

In fact the values of the accuracy measures are quite high.

This means that the forecast is not precise and the spread between the real values of the stochastic variable and the values obtained by the forecast procedure is large.

	MAE	MSE	RMSE	MAPE
WMT. arima(1,1,2).2015	0.2156437	0.06903769	0.2627503	0.05316138
WMT. arima(0,1,0).2016	0.1393309	0.02183796	0.1477767	0.03348919
WMT. arima(0,1,0).2017	0.1460049	0.03352772	0.1831058	0.03334731
WMT. arima(0,1,0).2018	0.1208463	0.01779006	0.1333794	0.02714120
WMT. arima(0,1,0).2019	0.1583538	0.03289450	0.1813684	0.03360604

We proceed with the estimation of the ARMA process for the whole sample.

The model that better fit the data is an ARIMA(1,1,2) process.

The coefficients of the “ar1” and the “ma1” components are very high, which means that the current value of the stock price is affected by the values of the price in $t - 1$.

The coefficient of the “ma2” component is quite small.

```
Series: WMT.log.20092019
ARIMA(1,1,2) with drift

Coefficients:
      ar1      ma1      ma2  drift
      0.9177 -0.9659  0.0404 3e-04
s.e.      NaN      NaN  0.0119 2e-04

sigma^2 estimated as 0.0001294:  log likelihood=7691.28
AIC=-15372.57  AICC=-15372.54  BIC=-15343.41
```



The forecasted values seem to be a little bit more close to the real values, but also in this case the spread is too much large to consider the forecast accurate and reliable.

As we can observe in the table below, the forecast of the last period is a little bit more accurate if

it is done using the whole sample to estimate the process, but the values of the accuracy measures are still very high.

	MAE	MSE	RMSE	MAPE
WMT. arima(1,1,2).2015	0.2156437	0.06903769	0.2627503	0.05316138
WMT. arima(0,1,0).2016	0.1393309	0.02183796	0.1477767	0.03348919
WMT. arima(0,1,0).2017	0.1460049	0.03352772	0.1831058	0.03334731
WMT. arima(0,1,0).2018	0.1208463	0.01779006	0.1333794	0.02714120
WMT. arima(0,1,0).2019	0.1583538	0.03289450	0.1813684	0.03360604
WMT. arima(1,1,2).wholesample	0.1215040	0.01936108	0.1391441	0.02578675

ARCH/GARCH processes

Usually, when dealing with time series analysis, the researchers make the assumption that the expected value of the error terms is constant over the whole period.

This assumption is useful to produce simpler process since it is a simplification of the reality; but as every simplification it is not very reliable.

For this reason it is better to allow the presence of Heteroskedasticity in the data; this means that the volatility of the stock returns could be higher in some periods than in others.

The change of volatility value over time reflects the fact that some periods could be riskier than others.

When we estimate a process and use it to forecast future values of a stochastic variable, we have to take in consideration the possibility of heteroskedasticity and analyze the error terms.

The econometric tools and processes in this case are the ARCH models (R.F. Engle, 1982) and GARCH models (T. Bollerslev, 1986).

These processes are useful to model the volatility of the data and to describe accurately the structure of the time series; in fact they represent the evolution of the conditional deviation of the returns over time. It is possible to forecast the future values of the volatility of the error terms.

The characteristics for which we can empirically observe heteroskedasticity in the data are:

Volatility Clusters;

Leptokurtosis;

Asymmetry ("Leverage Effect");

The ARCH and GARCH family models allow us to capture these characteristics and use them to describe accurately the evolution of the stock price and of the error terms over time.

There are numerous types of ARCH/GARCH processes;

The first formulation of the ARCH model goes back to 1982, when the statistician R.F. Engle developed the Autoregressive Conditional Heteroskedasticity model. This model allows the conditional volatility to change over time as a function of lagged values of error terms, but keeping the unconditional volatility constant.

The formulation of the ARCH model is:

$$\sigma^2 = a_0 + \sum_{i=1}^q a_i \varepsilon_{t-i}^2$$

The conditional variance is a linear combination of past values of squared error terms; the coefficient a_0 is the mean value and the coefficients a_i are the weight of how much the lagged white noise processes affect the current volatility.

Then, in 1986, the economist T. Bollerslev formulated the GARCH model, that can be considered as a generalization of the ARCH process. As opposite to its predecessor it allows longer memory and presents a more flexible lag structure.

The formulation is:

$$\sigma^2 = \omega + \sum_{i=1}^q a_i \varepsilon_{t-i}^2 + \sum_{j=1}^p b_j \sigma_{t-j}^2$$

In this case the conditional variance is a function of past squared error terms and of past values of conditional variance itself.

The coefficients ω , a_i and b_j are constant and estimated by Maximum Likelihood; $|a_1 + b_j|$ must be <1 , but the closer value is to 1 the more accurate is the estimation process.

To ensure that the variance is non-negative the parameters in the equation must be positive.

The ARCH components ε_{t-i}^2 in the function are the lagged squared residuals and refer to the number of autoregressive lags considered in the process.

A drawback of the standard ARCH and GARCH processes is that they cannot capture the presence of asymmetry in the data; in fact the different signs of the shocks and the different effect that they produce on the current volatility is not taken into consideration.

For this reason B. Nelson, in 1991, formulated a model that takes into account if the shocks in the process are positive or negative. This model is the Exponential GARCH (EGARCH):

$$\log \sigma_t^2 = \omega + \beta \log \sigma_{t-1}^2 + \gamma \frac{\varepsilon_{t-1}}{\sigma_{t-1}} + \alpha \frac{|\varepsilon_{t-1}|}{\sigma_{t-1}}$$

Where ω , α and β are constant parameters; the presence of the parameter γ allow the model to capture the asymmetry in the shocks, if the parameter is nonzero.

If γ is negative, then the negative shocks will have a higher influence on the conditional variance than the positive ones.

The non-negativity of the variance is ensured by the logarithmic transformation.

Now we can estimate the processes that better describe the data of the stocks in order to evaluate if the forecasts produced by the models are more accurate when they are formulated on less volatile stocks.

For each of the three stocks that we are analyzing, we formulate some standard GARCH models, EGARCH models and APARCH models.

The Asymmetric Power ARCH model (Z. Ding et al., 1993) is able to describe the characteristics of the volatility structure, such as the asymmetry, the leptokurtosis and the volatility clusters.

In some cases, we apply also some changes in the ARMA model underlying the Volatility processes.

We make the assumption that the error terms follow a Student t-distribution, to better describe the leptokurtosis of the data.

We recall that the period for the estimation of the model goes from January 2009 to December 2018 in order to forecast the values of the entire 2019.

Applying the processes it is possible to forecast both the future values of the stock returns and the future values of the volatility of the returns.

The forecasted values are compared with the real data and the measures of accuracy are computed.

As "real data" for the volatility of the stock we compute the daily deviation of the returns from the average value.

We test the stationarity of the residuals of each model through the Phillips-Perron test, in order to see if the processes are correctly specified.

JPM

The first model we test is a standard GARCH(1,2):

```

GARCH Model      : sGARCH(1,2)
Mean Model       : ARFIMA(3,0,1)
Distribution      : std

Optimal Parameters
-----
      Estimate  Std. Error  t value  Pr(>|t|)
mu      0.000893   0.000200   4.4692  0.000008
ar1     0.883058   0.014894  59.2902  0.000000
ar2     0.049683   0.026959   1.8429  0.065339
ar3    -0.012591   0.019782  -0.6365  0.524449
ma1    -0.936540   0.013810 -67.8144  0.000000
omega   0.000005   0.000003   1.8314  0.067044
alpha1  0.103078   0.020573   5.0103  0.000001
beta1   0.724787   0.170762   4.2444  0.000022
beta2   0.160881   0.156460   1.0283  0.303831
shape   5.232265    0.530070   9.8709  0.000000

```

From this table we can observe that the coefficient of the first ARMA components are high and statistically significant, while the parameters of the AR(2) and AR(3) parts are not significant. The parameters “alpha1”, which is the coefficient of the lagged squared residuals, and “beta1”, the parameter of the GARCH part, are significant, while the “beta2” coefficient is not significant.

The sum of the “alpha1”, “beta1” and “beta2” is close to 1 (=0.988746).

```

Sign Bias Test
-----
      t-value      prob sig
Sign Bias      0.06279  0.9499368
Negative Sign Bias  3.04631  0.0023408 ***
Positive Sign Bias  1.75468  0.0794364 *
Joint Effect    19.12529  0.0002576 ***

```

We can use this table to evaluate if the process is able to describe the asymmetry of the data; It seems that the sGARCH model is not the best process to fit the structure of the volatility of the JPM stock.

Then we test a sGARCH(1,3) process:

```
GARCH Model      : sGARCH(1,3)
Mean Model       : ARFIMA(3,0,1)
Distribution      : std
```

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.000891	0.000200	4.44494	0.000009
ar1	0.878603	0.015206	57.77954	0.000000
ar2	0.051065	0.027181	1.87871	0.060284
ar3	-0.013693	0.019846	-0.68997	0.490214
ma1	-0.933026	0.013072	-71.37860	0.000000
omega	0.000006	0.000003	1.91490	0.055506
alpha1	0.118303	0.021463	5.51198	0.000000
beta1	0.686728	0.434854	1.57922	0.114286
beta2	0.000000	0.777939	0.00000	1.000000
beta3	0.182092	0.375247	0.48526	0.627492
shape	5.266147	0.539896	9.75400	0.000000

Also in this case the parameters of the lagged conditional variance are not significant in the process.

Sign Bias Test

	t-value	prob	sig
Sign Bias	0.01477	0.9882140	
Negative Sign Bias	2.72523	0.0064702	***
Positive Sign Bias	1.88496	0.0595501	*
Joint Effect	17.48809	0.0005608	***

As before, the standard GARCH model is not able to capture the asymmetry of the data.

So we decide to implement some processes that are able to model the asymmetry of the innovation terms, such as the EGARCH and the APARCH.

```
GARCH Model      : eGARCH(1,5)
Mean Model       : ARFIMA(0,0,0)
Distribution      : std
```

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.00054	0.000219	2.4668	0.013633
omega	-0.14806	0.041778	-3.5441	0.000394
alpha1	-0.14431	0.017893	-8.0647	0.000000
beta1	0.67033	0.000945	709.4795	0.000000
beta2	0.19236	0.112259	1.7136	0.086611
beta3	0.13238	0.092456	1.4318	0.152194
beta4	-0.36638	0.016390	-22.3532	0.000000
beta5	0.35363	0.016797	21.0531	0.000000
gamma1	0.23624	0.026801	8.8145	0.000000
shape	5.46797	0.559804	9.7677	0.000000

In this case we apply an EGARCH(1,5) process. Analyzing the values we can say that the “gamma1” coefficient, that capture the asymmetric part of the data, is positive and statistically significant. The “alpha1”, “beta1”, “beta4” and “beta5” coefficients are significant.

```

GARCH Model      : fGARCH(1,3)
fGARCH Sub-Model : APARCH
Mean Model       : ARFIMA(3,0,1)
Distribution      : std

```

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.000550	0.000076	7.2139e+00	0.000000
ar1	0.947999	0.010644	8.9064e+01	0.000000
ar2	0.054331	0.007057	7.6992e+00	0.000000
ar3	-0.008378	0.002286	-3.6652e+00	0.000247
ma1	-0.998518	0.000028	-3.5077e+04	0.000000
omega	0.000099	0.000080	1.2351e+00	0.216802
alpha1	0.116530	0.022616	5.1526e+00	0.000000
beta1	0.517439	0.250623	2.0646e+00	0.038959
beta2	0.221572	0.268435	8.2542e-01	0.409132
beta3	0.132510	0.120861	1.0964e+00	0.272910
eta11	0.570518	0.142135	4.0139e+00	0.000060
lambda	1.380045	0.181880	7.5877e+00	0.000000
shape	5.483246	0.575293	9.5312e+00	0.000000

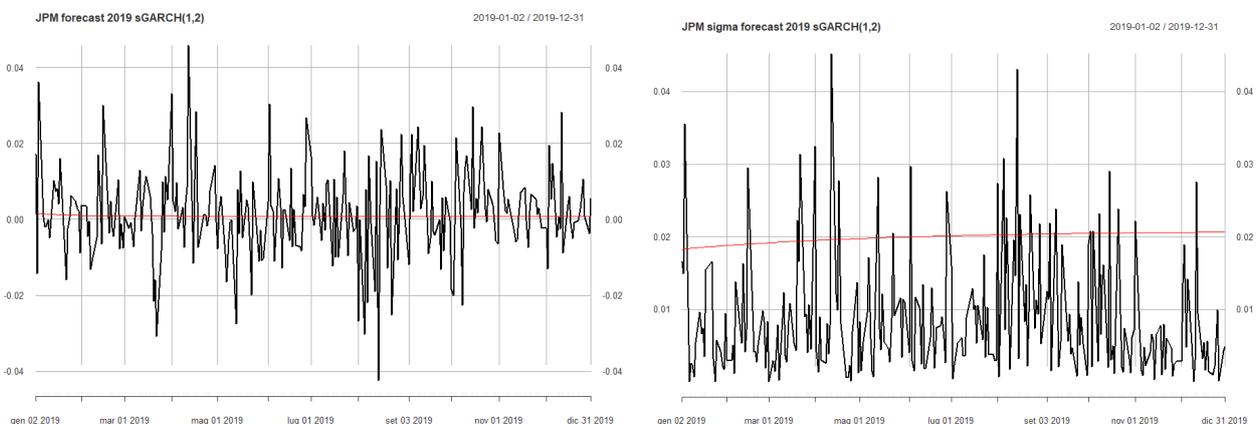
In this table we can see that the specific parameters of the process, “eta11”, which is the parameter that measures the asymmetric structure of the data, and “lambda”, which is the power term, are significative.

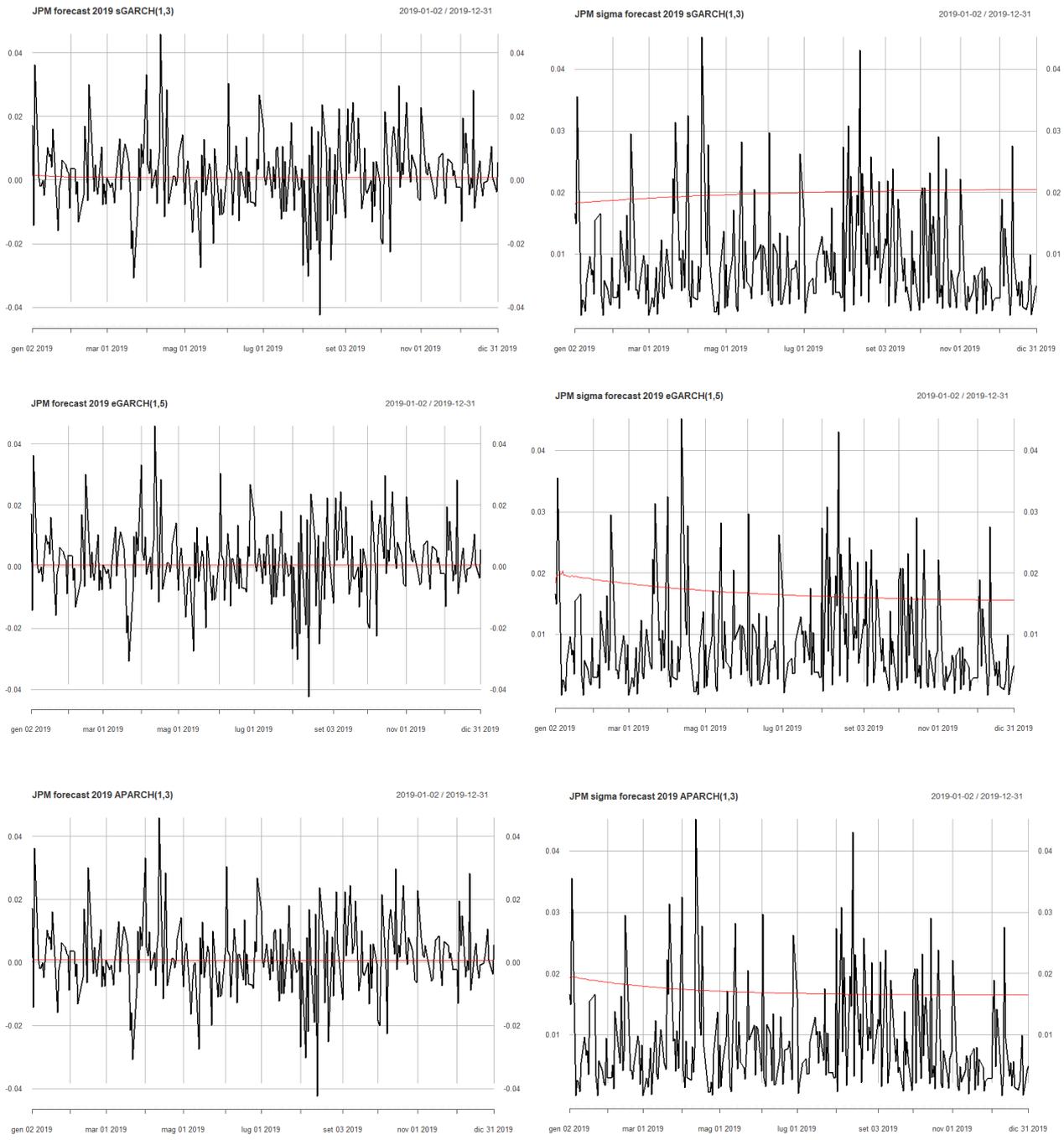
At this point, to observe which process better fits the stock returns we compare the Information Criteria:

	Akaike	Bayes	shibata	Hannan-Quinn
JPM. sGARCH(1,2)	-5.485263	-5.462082	-5.485294	-5.476850
JPM. sGARCH(1,3)	-5.485409	-5.459909	-5.485447	-5.476154
JPM. eGARCH(1,5)	-5.500402	-5.477221	-5.500433	-5.491989
JPM. APARCH(1,3)	-5.501197	-5.471062	-5.501250	-5.490260

The models that better perform are the EGARCH and the APARCH, that are the processes that are able to describe the asymmetry of the data.

Now we proceed with the forecast procedure and the analysis of the plots:





The plots on the left side show the comparison between the real stock returns over the period and the values obtained by the forecasting procedure; we can see that the forecasted values are close to average value of the stock returns for the fact that the processes capture the stationarity and the mean-reversion of the data. There are no large differences between the forecasts obtained by the different models.

On the right side we can see the comparison between the daily volatility of the data and the forecasted values. In this case the spreads between the real data and the data obtained by the processes are quite large.

There is also a difference between the data produced by the standard GARCH models and the EGARCH and APARCH models; in fact the processes that are able to capture the asymmetry in the data seem to be a little bit more accurate.

		MAE	MSE	RMSE
JPM. sgarch(1,2)	returns	0.008553415	0.0001381061	0.01175186
JPM. sgarch(1,3)	returns	0.008553410	0.0001381163	0.01175229
JPM. egarch(1,5)	returns	0.008570921	0.0001388448	0.01178324
JPM. aparch(1,3)	returns	0.008566564	0.0001386586	0.01177534

		MAE	MSE	RMSE
JPM. sgarch(1,2)	sigma	0.01290850	0.0001947074	0.01395376
JPM. sgarch(1,3)	sigma	0.01280924	0.0001918135	0.01384967
JPM. egarch(1,5)	sigma	0.01066288	0.0001354762	0.01163943
JPM. aparch(1,3)	sigma	0.01083375	0.0001391173	0.01179480

As seen before, the accuracy of the forecast of the returns is quite similar for each process; when we use the processes to forecast the volatility of the JPM stock the best models are the EGARCH and the APARCH.

INTC

In this part we analyze the results of the application of some GARCH models to the INTC stock in order to capture the dynamic structure of the volatility of the time series.

We apply three GARCH models: the standard GARCH process, the Exponential GARCH process and the Asymmetric Power ARCH process; in each case we estimate the models, first, using an ARMA process to describe the conditional mean of the time series, then, without the ARMA model.

```
GARCH Model      : sGARCH(1,1)
Mean Model       : ARFIMA(2,0,2)
Distribution      : std
```

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.000780	0.000252	3.0968	0.001956
ar1	-1.796911	0.005202	-345.4093	0.000000
ar2	-0.965082	0.009152	-105.4540	0.000000
ma1	1.792046	0.006732	266.2049	0.000000
ma2	0.956781	0.003773	253.6099	0.000000
omega	0.000002	0.000001	2.0263	0.042736
alpha1	0.048317	0.006457	7.4833	0.000000
beta1	0.945029	0.007699	122.7537	0.000000
shape	4.990241	0.446879	11.1669	0.000000

```
GARCH Model      : sGARCH(1,1)
Mean Model       : ARFIMA(0,0,0)
Distribution      : std
```

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.000777	0.000253	3.0723	0.002124
omega	0.000002	0.000001	1.9651	0.049402
alpha1	0.048467	0.007287	6.6515	0.000000
beta1	0.944733	0.008435	112.0079	0.000000
shape	4.980541	0.444923	11.1942	0.000000

GARCH Model : eGARCH(1,1)
 Mean Model : ARFIMA(2,0,2)
 Distribution : std

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.000687	0.000196	3.4978	0.000469
ar1	-1.803494	0.004729	-381.4004	0.000000
ar2	-0.968296	0.007254	-133.4781	0.000000
ma1	1.799213	0.005715	314.8208	0.000000
ma2	0.961180	0.002968	323.8989	0.000000
omega	-0.113518	0.008328	-13.6302	0.000000
alpha1	-0.035803	0.012216	-2.9307	0.003381
beta1	0.986568	0.001002	984.8263	0.000000
gamma1	0.128676	0.019091	6.7401	0.000000
shape	5.128307	0.516240	9.9340	0.000000

GARCH Model : eGARCH(1,1)
 Mean Model : ARFIMA(0,0,0)
 Distribution : std

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.000686	0.000249	2.7577	0.005820
omega	-0.115525	0.007058	-16.3668	0.000000
alpha1	-0.037102	0.012268	-3.0244	0.002491
beta1	0.986317	0.000831	1187.2778	0.000000
gamma1	0.128688	0.018234	7.0576	0.000000
shape	5.128511	0.505583	10.1438	0.000000

GARCH Model : fGARCH(1,1)
 fGARCH Sub-Model : APARCH
 Mean Model : ARFIMA(1,0,1)
 Distribution : std

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.000623	0.000239	2.6111	0.009025
ar1	-0.202674	0.016367	-12.3828	0.000000
ma1	0.193054	0.016469	11.7226	0.000000
omega	0.000715	0.000607	1.1776	0.238975
alpha1	0.084172	0.015218	5.5311	0.000000
beta1	0.916864	0.017357	52.8228	0.000000
eta11	0.363014	0.109278	3.3219	0.000894
lambda	0.818512	0.171800	4.7643	0.000002
shape	5.124326	0.485360	10.5578	0.000000

GARCH Model : fGARCH(1,1)
 fGARCH Sub-Model : APARCH
 Mean Model : ARFIMA(0,0,0)
 Distribution : std

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
mu	0.000613	0.000235	2.6061	0.009158
omega	0.000756	0.000629	1.2018	0.229457
alpha1	0.084541	0.015113	5.5940	0.000000
beta1	0.916452	0.017241	53.1547	0.000000
eta11	0.368938	0.109863	3.3582	0.000785
lambda	0.807335	0.167603	4.8170	0.000001
shape	5.122098	0.484964	10.5618	0.000000

All the coefficients of the different processes are statistically significant, which means that the models describe the data in a good way.

The models are able to capture the leptokurtosis of the data, and since the distribution of the returns does not show sign of asymmetry, we expect that the application of sGARCH processes allow us to obtain a good estimation of the future values.

Analyzing the values of the Information Criteria we can observe that there is a small difference in the values obtained from the sGARCH models and the EGARCH and APARCH models.

The latter manage to better fit the data.

	Akaike	Bayes	shibata	Hannan-Quinn
INTC. sGARCH(1,1)	-5.610612	-5.589749	-5.610637	-5.603040
INTC. sGARCH(1,1).ARMA(0,0)	-5.611858	-5.600267	-5.611866	-5.607651
INTC. eGARCH(1,1)	-5.623134	-5.599953	-5.623166	-5.614721
INTC. eGARCH(1,1).ARMA(0,0)	-5.625038	-5.611129	-5.625049	-5.619990
INTC. APARCH(1,1)	-5.623366	-5.602503	-5.623391	-5.615794
INTC. APARCH(1,1).ARMA(0,0)	-5.624793	-5.608566	-5.624809	-5.618904

At this point we can analyze the forecast results obtained from the computation of the measures of accuracy.

With regard to the forecast of the future values of the stock returns, we can see that the results are quite equal. The application of one process rather than another does not affect the prevision accuracy.

Instead, if we analyze the results of the conditional volatility forecast, we can notice that the EGARCH and APARCH processes better perform in predicting the future values of the stock's sigma.

	MAE	MSE	RMSE
INTC.sgarch(1,1) returns	0.01203319	0.0002911671	0.01706362
INTC.sgarch(1,1).arma(0,0) returns	0.01204009	0.0002910818	0.01706112
INTC.egarch(1,1) returns	0.01204213	0.0002914197	0.01707102
INTC.egarch(1,1).arma(0,0) returns	0.01204441	0.0002911419	0.01706288
INTC.aparch(1,1) returns	0.01204746	0.0002911948	0.01706443
INTC.aparch(1,1).arma(0,0) returns	0.01204789	0.0002912023	0.01706465

	MAE	MSE	RMSE
INTC.sgarch(1,1) sigma	0.01262537	0.0002237455	0.01495812
INTC.sgarch(1,1).arma(0,0) sigma	0.01256183	0.0002221265	0.01490391
INTC.egarch(1,1) sigma	0.01004900	0.0001674148	0.01293889
INTC.egarch(1,1).arma(0,0) sigma	0.01005308	0.0001674757	0.01294124
INTC.aparch(1,1) sigma	0.01001764	0.0001667128	0.01291173
INTC.aparch(1,1).arma(0,0) sigma	0.01001251	0.0001666162	0.01290799

The inclusion of ARMA process in the model does not provide any significant improvement in the forecast accuracy.

WMT

Taking into consideration the dynamic structure and statistical distribution of the values of the time series, we estimate different GARCH models, such as sGARCH(1,1) model, sGARCH(1,2), EGARCH(1,1), EGARCH(1,2), EGARCH(1,2) model with a different underlying ARMA process and APARCH(1,1).

In each volatility process, the coefficients of ARCH/GARCH parts are all statistically significant; when we observe the coefficients of the ARMA component in the sGARCH processes, we can see that they are not significant.

GARCH Model	: sGARCH(1,1)	GARCH Model	: sGARCH(1,2)
Mean Model	: ARFIMA(1,0,1)	Mean Model	: ARFIMA(1,0,1)
Distribution	: std	Distribution	: std

Optimal Parameters					Optimal Parameters				
	Estimate	Std. Error	t value	Pr(> t)		Estimate	Std. Error	t value	Pr(> t)
mu	0.000509	0.000163	3.11715	0.001826	mu	0.000506	0.000160	3.1633	0.001560
ar1	0.306704	0.776136	0.39517	0.692719	ar1	0.518527	0.401125	1.2927	0.196121
ma1	-0.343087	0.765239	-0.44834	0.653908	ma1	-0.551578	0.388070	-1.4213	0.155219
omega	0.000001	0.000001	1.90776	0.056422	omega	0.000003	0.000002	1.6559	0.097744
alpha1	0.021192	0.002924	7.24649	0.000000	alpha1	0.042487	0.009549	4.4494	0.000009
beta1	0.966999	0.002378	406.68266	0.000000	beta1	0.202183	0.026025	7.7689	0.000000
shape	4.012874	0.361151	11.11135	0.000000	beta2	0.730396	0.026905	27.1470	0.000000
					shape	4.020042	0.309224	13.0004	0.000000

```

GARCH Model      : eGARCH(1,1)
Mean Model       : ARFIMA(1,0,1)
Distribution      : std

GARCH Model      : eGARCH(1,2)
Mean Model       : ARFIMA(1,0,1)
Distribution      : std

Optimal Parameters
-----
      Estimate Std. Error t value Pr(>|t|) mu      Estimate Std. Error t value Pr(>|t|)
mu      0.000476  0.000155  3.0611 0.002205 ar1     0.000478  0.000144  3.3181 0.000906
ar1     0.504261  0.154043  3.2735 0.001062 ma1     0.441844  0.080525  5.4870 0.000000
ma1     -0.537442  0.150396  -3.5735 0.000352 omega  -0.478158  0.078598  -6.0836 0.000000
omega   -0.249157  0.020598 -12.0964 0.000000 alpha1  -0.371461  0.055467  -6.6970 0.000000
alpha1  -0.034974  0.011798  -2.9644 0.003033 beta1   -0.049222  0.017975  -2.7384 0.006175
beta1   0.972790  0.002260 430.3552 0.000000 beta2   0.416363  0.001848 225.2625 0.000000
beta2   0.092790  0.002260 430.3552 0.000000 beta2   0.543101  0.001263 429.9596 0.000000
gamma1  0.116017  0.051809  2.2393 0.025136 gamma1  0.181164  0.025654  7.0617 0.000000
shape   4.222447  0.247442 17.0644 0.000000 shape   0.181164  0.025654  7.0617 0.000000
shape   4.257156  0.347566 12.2485 0.000000

```

```

GARCH Model      : eGARCH(1,2)
Mean Model       : ARFIMA(0,0,1)
Distribution      : std

GARCH Model      : fGARCH(1,1)
fGARCH Sub-Model : APARCH
Mean Model       : ARFIMA(1,0,1)
Distribution      : std

Optimal Parameters
-----
      Estimate Std. Error t value Pr(>|t|) mu      Estimate Std. Error t value Pr(>|t|)
mu      0.000468  0.000166  2.8276 0.004689 mu      0.000498  0.000156  3.18768 0.001434
ma1     -0.039014  0.019883  -1.9622 0.049745 ar1     0.886961  0.173184  5.12148 0.000000
omega   -0.364336  0.053169  -6.8525 0.000000 ma1     -0.897374  0.157796  -5.68694 0.000000
alpha1  -0.050583  0.018303  -2.7637 0.005715 omega   0.000000  0.000000  0.40048 0.688803
beta1   0.409073  0.001809 226.0837 0.000000 alpha1  0.008108  0.003119  2.59959 0.009333
beta2   0.551160  0.001174 469.3966 0.000000 beta1   0.971672  0.002882 337.09932 0.000000
gamma1  0.179471  0.025631  7.0022 0.000000 eta1    0.472502  0.174560  2.70682 0.006793
shape   4.240995  0.345741 12.2664 0.000000 lambda  2.428429  0.052978 45.83822 0.000000
shape   4.056796  0.361040 11.23641 0.000000

```

Analyzing the table below which reports the values of the Information Criteria, we can see that the process that performs worst is the APARCH model; the models that present the lowest values are the EGARCH ones.

The EGARCH processes are able to capture the leptokurtosis and the negative skewness of the WMT stock.

	Akaike	Bayes	Shibata	Hannan-Quinn
WMT. sGARCH(1,1)	-6.399117	-6.382890	-6.399132	-6.393228
WMT. sGARCH(1,2)	-6.400971	-6.382426	-6.400991	-6.394240
WMT. eGARCH(1,1)	-6.414716	-6.396171	-6.414736	-6.407985
WMT. eGARCH(1,2)	-6.416765	-6.395902	-6.416790	-6.409193
WMT. eGARCH(1,2). ARMA(0,1)	-6.417227	-6.398682	-6.417247	-6.410497
WMT. APARCH(1,1)	-6.400709	-6.379846	-6.400734	-6.393137

Comparing the measures of accuracy of the forecast, we can notice that also in this case there is no difference between the forecast precision when the processes are used to predict the future values of the stock returns.

Instead, when we use the models to forecast the future values of the conditional volatility of the stochastic time series, we notice that the EGARCH processes perform a little better and provide a slightly more accurate prediction, as we could expect from the Information Criteria results.

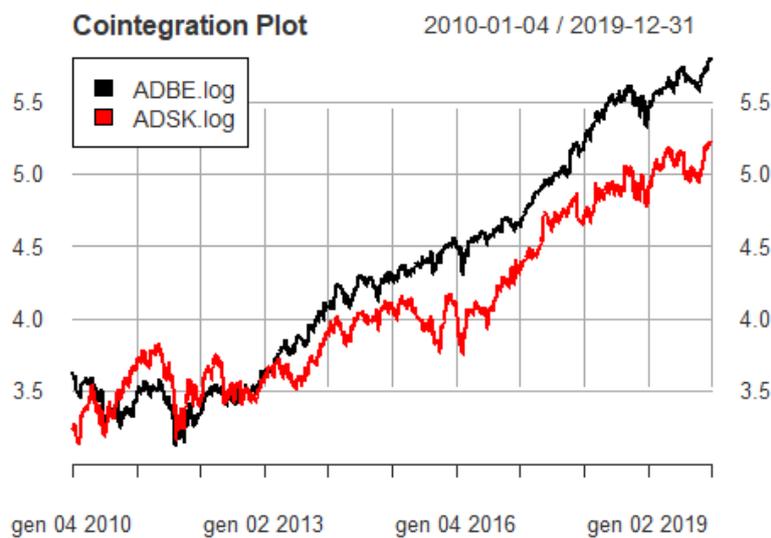
	MAE	MSE	RMSE
WMT.sgarch(1,1) returns	0.006473138	8.124832e-05	0.009013785
WMT.sgarch(1,2) returns	0.006474400	8.126182e-05	0.009014534
WMT.egarch(1,1) returns	0.006477695	8.129476e-05	0.009016361
WMT.egarch(1,2) returns	0.006477227	8.128826e-05	0.009016000
WMT.egarch(1,2).arma(0,1) returns	0.006477439	8.129473e-05	0.009016359
WMT.aparch(1,1) returns	0.006473847	8.126997e-05	0.009014986
	MAE	MSE	RMSE
WMT.sgarch(1,1) sigma	0.007105683	6.959946e-05	0.008342629
WMT.sgarch(1,2) sigma	0.007027529	6.850310e-05	0.008276660
WMT.egarch(1,1) sigma	0.006403920	5.997893e-05	0.007744607
WMT.egarch(1,2) sigma	0.006397638	5.988992e-05	0.007738858
WMT.egarch(1,2).arma(0,1) sigma	0.006405963	5.997706e-05	0.007744486
WMT.aparch(1,1) sigma	0.006503617	6.083423e-05	0.007799630

Through this analysis we can also notice that the GARCH processes that better fit the data and provide a lower IC value, are the ones that provide more accurate forecasts.

MULTIVARIATE ANALYSIS

In this chapter we want to study the relationship between two stochastic variables, their dynamic evolution when jointly analyzed and the effect that the evolution of one variable has on the other.

The basic idea of the analysis is to identify a pair of equity securities whose prices tend to share a common movement. So, at the beginning, we observe the plots of the stock prices analyzing the possible existence of common trends:



From the plot we can see that the prices tend to move together. For this reason we can formulate the hypothesis that the time series are tied by a cointegration relationship.

Cointegration is an extremely important concept when applied to the analysis of multiple time series and economic variables. In fact it can be used to study and jointly describe non-stationary variables through simple estimation methods, such as Least Squares regression and maximum likelihood.

The first step to ensure that the series can be cointegrated is to study their stationarity and their order of integration. It is essential that the time series are integrated of the same order.

In the financial Market it is very common that the prices of equity securities are non-stationary, while their returns are stationary and mean-reverting.

Also in our case, studying the stationarity of the two equities, ADBE and ADSK, through the Phillips-Perron test, we observe that the daily logarithmic close prices are non-stationary, while if the test the first difference of the prices, which is the returns, we see that it is stationary; so both the time series are integrated of order one, by notation $I(1)$.

Once we established that the series are integrated of the same order, we can test if they are cointegrated.

The concept of Cointegration was first formulated in 1981 by C.W.J. Granger, then enhanced by R.F. Engle and C.W.J. Granger in 1987. When a linear combination of two integrated series produces a stochastic variable that is stationary, or $I(0)$, then the two series are cointegrated.

This means that the two variables, even if they are non-stationary, share the same stochastic trend and are tied by a Long-run Equilibrium relationship.

The Long-run equilibrium is an important concept when we study multiple time series. In fact it ensures the fact that even if the prices of two series diverge from the equilibrium in some periods, we know that this deviation will decline over time until it disappears, returning to the state of equilibrium.

This is ensured by the fact that the combination of the series is stationary and mean-reverting.

The first general definition of cointegration given by Engle and Granger is:

DEFINITION: The components of the vector x_t are said to be *co-integrated of order d , b* , denoted $x_t \sim CI(d, b)$, if (i) all components of x_t are $I(d)$; (ii) there exists a vector $\alpha (\neq 0)$ so that $z_t = \alpha' x_t \sim I(d - b)$, $b > 0$. The vector α is called the *co-integrating vector*.

This is a very general definition of when two series are cointegrated. Bringing the concept back to our case we can say:

If y_t, x_t are two time series integrated of same order, where $y_t - \alpha x_t \sim I(0)$, the cointegrating vector is $[1, -\alpha]$.

This means that even if the two series y_t, x_t are non-stationary, there exists a certain value of α such that $z_t = y_t - \alpha x_t$ is stationary.

z_t is the product of the linear combination of the two equities; It is important to highlight the fact that it is a mean-reverting process.

It is called the "disequilibrium term" due to the fact that it captures the spread that elapses between the values of the two time series over time. The expected value of z_t gives us information about the Long-run Equilibrium relationship between y_t and x_t .

The concept of mean-reversion of z_t is essential because it allows us to study the dynamic evolution of the time series through the analysis of the Short-term deviations from the Long-run equilibrium, indicated by the expected value of the disequilibrium term. This concept is formulated in the Error Correction Model, that we will discuss later in the chapter.

There exist various ways to test if two stochastic variables are cointegrated: the most famous ones are the Engle-Granger 2-steps cointegration test, the Johansen cointegration test and the Phillips-Ouliaris test.

Every test is based on the regression of one stochastic variable on the other, and on the stationarity test of the residuals of the regression. If the residuals are stationary, then the series are cointegrated, if the residuals are non-stationary, the combination of the series is called "spurious regression" and there is no cointegration relationship between the variables.

To start the analysis about the existence of a cointegration relationship between the equity securities we apply the Phillips-Ouliaris test: the procedure is based on using the Phillips-Perron stationarity test on the residuals of the regression.

The Null Hypothesis of the test is that the residuals present a Unit root, so they are non-stationary; the results of both regressions are:

Phillips-Ouliaris Cointegration Test

data: ADBE_on_ADSK
Phillips-Ouliaris demeaned = -22.715, Truncation lag parameter = 25, p-value = 0.03325

Phillips-Ouliaris Cointegration Test

data: ADSK_on_ADBE
Phillips-Ouliaris demeaned = -24.593, Truncation lag parameter = 25, p-value = 0.02239

As we can see from the analysis of the p-value, we can reject the Null Hypothesis, so the series are cointegrated.

Once we established that the stock prices share a common stochastic trend and are tied in a Long-run equilibrium relationship, in order to study more deeply the dynamic relationship of the two series we have to analyze the Short-term relationship.

In fact if the combination of the two variables is stationary and mean-reverting, there must exist a process that ensures that the deviations from the expected value of z_t will decline to the equilibrium value.

This process is called Error Correction Model.

It describes the dynamic relationship of y_t and x_t in the short period, taking into consideration the Long-run Equilibrium.

So there exists a relationship between the cointegrated processes and the Error Correction Models, as formulated by R.F. Engle and C.W.J. Granger in 1987. They state that *"for each cointegrated system there exists a ECM representation; if there exists a ECM representation and the series are integrated, then they are cointegrated"*.

The formulation of the Cointegration relationship between the two time series is:

$$\Delta y_t = \beta \Delta x_t + \psi(y_{t-1} - m - \alpha x_{t-1}) + \varepsilon_t$$

Each component of the equation is stationary.

$\beta \Delta x_t$ measures how much a change in the value of x_t affects the movement of y_t , and describes the short-run dynamic evolution of the process.

$\psi(y_{t-1} - m - \alpha x_{t-1})$ is the Error Correction mechanism and describes the Long-run equilibrium relationship.

In order to formulate this equation in our case study we first regress one variable on the other to generate the residuals; Then we test the stationarity of the residuals.

REGRESSION OF ADBE ON ADSK

```
Call:
lm(formula = ADBE.log ~ ADSK.log)

Residuals:
    Min       1Q   Median       3Q      Max
-0.47075 -0.09535  0.00485  0.10037  0.52701

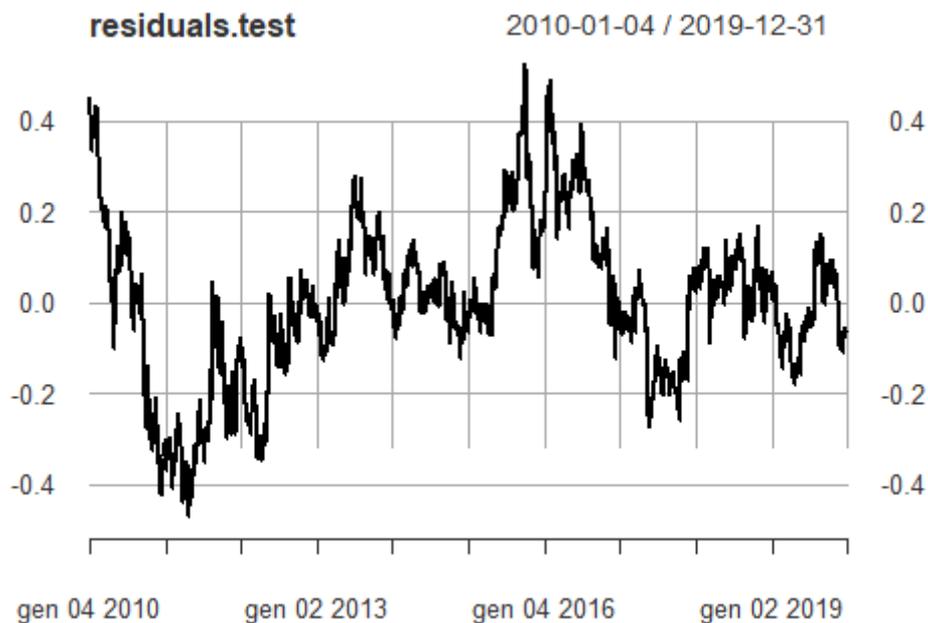
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.179980   0.025564  -46.16  <2e-16 ***
ADSK.log     1.349682   0.006194  217.89  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1759 on 2514 degrees of freedom
Multiple R-squared:  0.9497,    Adjusted R-squared:  0.9497
F-statistic: 4.747e+04 on 1 and 2514 DF,  p-value: < 2.2e-16
```

Phillips-Perron Unit Root Test

```
data: residuals.test
Dickey-Fuller Z(alpha) = -24.889, Truncation lag parameter = 8, p-value = 0.0247
alternative hypothesis: stationary
```

The residuals are stationary; so when plot them to observe the spread between the two time series:



We can see that the spread is mean-reverting around zero.

So we can estimate the ECM:

```

Call:
lm(formula = dADBE.log ~ dADSK.log + z1 - 1)

Residuals:
    Min       1Q   Median       3Q      Max
-0.189084 -0.006374  0.000429  0.007131  0.112706

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
dADSK.log  0.494268    0.013113  37.692 <2e-16 ***
z1         0.002645    0.001610   1.643  0.101
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01417 on 2513 degrees of freedom
Multiple R-squared:  0.3612,    Adjusted R-squared:  0.3607
F-statistic: 710.4 on 2 and 2513 DF,  p-value: < 2.2e-16

```

Analyzing the table we can see that the coefficient of the ADSK variable is statistically significant, which means that in the short period a change in the ADSK price will affect the price of ADBE. The coefficient of the ECT (Error Correction Term) is very low.

The coefficient of the ECT component measures the “speed of adjustment”, that is the measure of how rapidly the spread will go back to the equilibrium value and disappear. In this case the adjustment is not very quick.

REGRESSION OF ADSK ON ADBE

```

Call:
lm(formula = ADSK.log ~ ADBE.log)

Residuals:
    Min       1Q   Median       3Q      Max
-0.38710 -0.06871  0.00300  0.08113  0.31771

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.035885    0.014235  72.77 <2e-16 ***
ADBE.log    0.703653    0.003229 217.89 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.127 on 2514 degrees of freedom
Multiple R-squared:  0.9497,    Adjusted R-squared:  0.9497
F-statistic: 4.747e+04 on 1 and 2514 DF,  p-value: < 2.2e-16

```

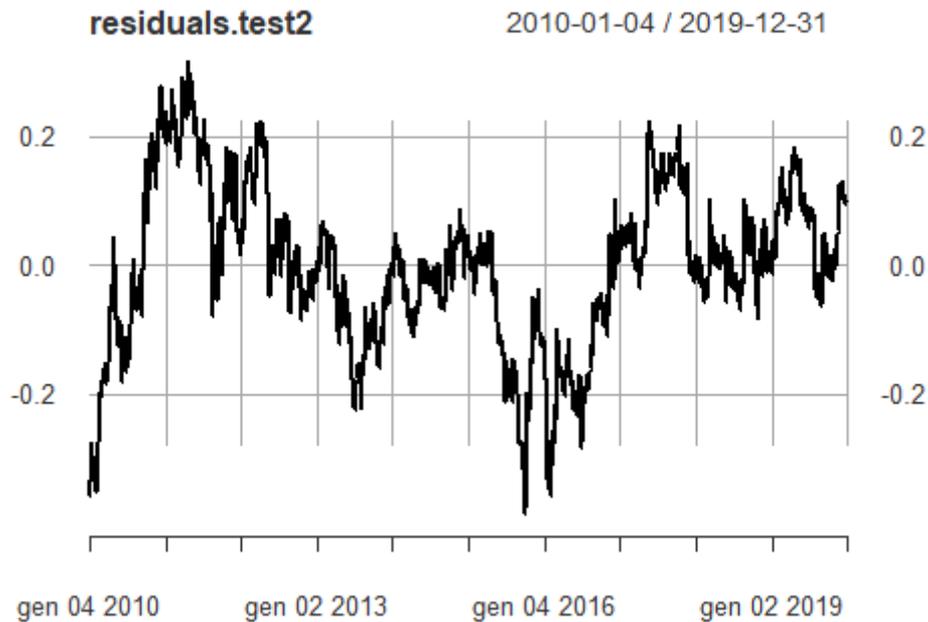
Phillips-Perron Unit Root Test

```

data: residuals.test2
Dickey-Fuller Z(alpha) = -24.41, Truncation lag parameter = 8, p-value = 0.028
alternative hypothesis: stationary

```

Also in this case the error terms are stationary and mean-reverting.



Now we formulate the process:

```

call:
lm(formula = dADSK.log ~ dADBE.log + z2 - 1)

Residuals:
    Min       1Q   Median       3Q      Max
-0.141772 -0.007967 -0.000168  0.008221  0.129050

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
dADBE.log  0.730242    0.019387  37.667 < 2e-16 ***
z2         0.007960    0.002709   2.939  0.00333 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01723 on 2513 degrees of freedom
Multiple R-squared:  0.3627,    Adjusted R-squared:  0.3622
F-statistic:  715 on 2 and 2513 DF,  p-value: < 2.2e-16

```

The coefficient of Δx_t is high and statistically significant; also the coefficient of ECT is significant. In this case the deviations from the long-run equilibrium are reabsorbed more quickly.

At this point of the study we decide to apply the Johansen Cointegration test to examine the cointegration relationship between the series.

The cointegrating vector is estimated by Maximum Likelihood Estimation.

```
#####
# Johansen-Procedure #
#####
```

Test type: trace statistic , without linear trend and constant in cointegration

Eigenvalues (lambda):

```
[1] 6.538301e-03 2.826936e-03 1.040834e-17
```

values of teststatistic and critical values of test:

	test	10pct	5pct	1pct
r <= 1		7.12	7.52	9.24 12.97
r = 0		23.61	17.85	19.96 24.60

Eigenvectors, normalised to first column:
(These are the cointegration relations)

	ADBE.Adjusted.l2	ADSK.Adjusted.l2	constant
ADBE.Adjusted.l2	1.000000	1.000000	1.000000
ADSK.Adjusted.l2	-1.465431	-0.7544289	-0.4932688
constant	1.647212	0.2690554	-2.4477846

From this table we can see that between the two series there exists 1 cointegration vector, since the fact that the Null Hypothesis, which states that $r \leq 1$, is not rejected.

This cointegration vector is composed by the first column of the Eigenvectors matrix: [1,-1.465431, 1.647212].

The fact that there exists one cointegration vector allow us to say that there exists one stationary process that describes the dynamic evolution of the stochastic series and the evolution of the mean-reverting spread.

The cointegration vector is the vector of parameters in z_t .

We can formulate the equations of the equilibrium between the two cointegrated time series:

```
Call:
lm(formula = substitute(form1), data = data.mat)

Coefficients:
          ADBE.Adjusted.d  ADSK.Adjusted.d
ect1          -0.001161          0.006487
ADBE.Adjusted.dl1    -0.045655          0.054687
ADSK.Adjusted.dl1    -0.007239         -0.023772

$beta
          ect1
ADBE.Adjusted.l2  1.000000
ADSK.Adjusted.l2 -1.465431
constant          1.647212
```

This is the formulation of the Vector Error Correction Model (VECM).

It allows to estimate the process taking into consideration both the series symmetrically as a vector.

$$\Delta x_t = \Delta x_{t-1} + \psi(ECT) + \varepsilon_t$$

In this case the values β are the values of the Error Correction mechanism.

The studies about the cointegration between two financial variables have been the basis of the formulation of a Trading strategy called "pairs trading". This strategy, formulated in mid-80s, exploits the deviations from the equilibrium in the short period to simultaneously buy and sell the two stocks. The spread will eventually disappear over time allowing the closure of the trades and the gain. Over the years this strategy became known as "Statistical Arbitrage strategy".

It is essential to recall that, even if we know that the spread between the two stochastic variables is mean-reverting, we do not obtain information about the future evolution of the stock prices. We only know that the two equity securities are tied by a long-run equilibrium relationship and they move together.

The reversion could take a long time to happen, if the residuals present a large amount of autocorrelation and long-term memory.

At this point, we decide to analyze the joint structure of the time series using a VAR process.

This model allows us to study the dynamic structure and evolution of the system of variables using the past values of the series at the same time.

It is based on the ARMA process; in fact it is composed of an autoregressive component and a moving average part.

The purpose of this part of the thesis is to test if the forecast procedure is able to generate accurate and reliable previsions and to test the hypothesis which states that the forecast produced by the application of a VAR model is more accurate than the one obtained estimating an ARMA model.

So we estimate the VAR process:

```
Estimation results for equation ADBE.Adjusted:
=====
ADBE.Adjusted = ADBE.Adjusted.l1 + ADSK.Adjusted.l1 + const

              Estimate Std. Error t value Pr(>|t|)
ADBE.Adjusted.l1  0.9995843  0.0020076 497.906  <2e-16 ***
ADSK.Adjusted.l1  0.0009695  0.0027807   0.349   0.727
const              -0.0012911  0.0034993  -0.369   0.712
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01771 on 2512 degrees of freedom
Multiple R-Squared: 0.9995,    Adjusted R-squared: 0.9995
F-statistic: 2.465e+06 on 2 and 2512 DF,  p-value: < 2.2e-16
```

Estimation results for equation ADSK.Adjusted:

$$\text{ADSK.Adjusted} = \text{ADBE.Adjusted.l1} + \text{ADSK.Adjusted.l1} + \text{const}$$

	Estimate	Std. Error	t value	Pr(> t)
ADBE.Adjusted.l1	0.006982	0.002441	2.860	0.00427 **
ADSK.Adjusted.l1	0.990194	0.003381	292.840	< 2e-16 ***
const	0.010583	0.004255	2.487	0.01294 *

 signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02154 on 2512 degrees of freedom
 Multiple R-squared: 0.9986, Adjusted R-squared: 0.9986
 F-statistic: 8.677e+05 on 2 and 2512 DF, p-value: < 2.2e-16

From the first table we can observe the dynamic relationship between the two time series and the effect that this relationship has on the current value of ADBE. The coefficient of the lagged value of ADBE is significant, as we can expect; the parameter of the lagged value of ADSK is not significant, so we can say that the history of ASDK does not affect the current value of ADBE.

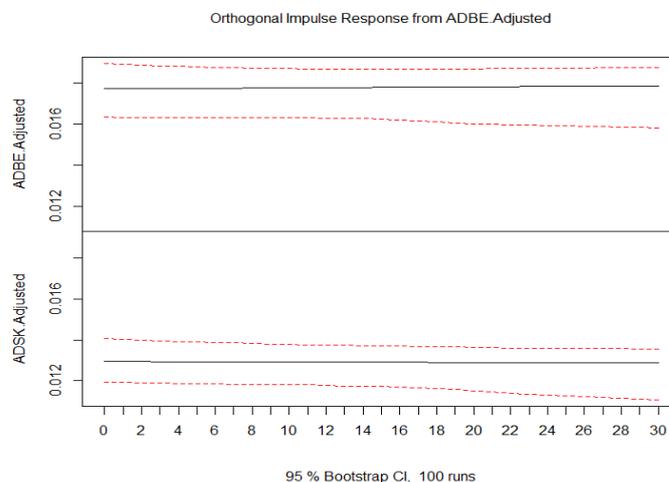
In the case of the second equation, we can see that the coefficients of the lagged values of ADBE and ADSK are significant, which means that the historical values of the two series affect positively the current price of ADSK.

It is considered appropriate to highlight that the coefficient of the lagged value of ADBE is very low, so the effect on ADSK is not so large.

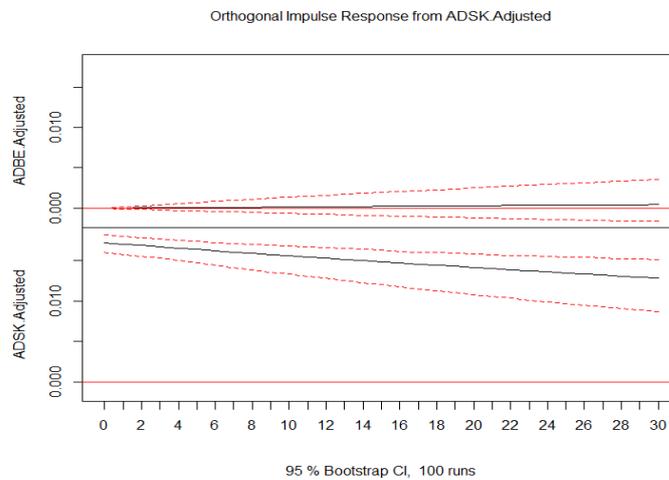
At this point we apply two other procedures in order to analyze deeply the relationship between the series:

- the Impulse Response Function;
- the Granger Causality test.

The IRF describes the dynamic evolution and behavior of one variable of the system when there is a shock in the other variable's error terms. It measures the responsiveness of the stochastic variable to a 1 standard deviation shock in the other variable of VAR model.



From the graph we can see that the ADBE time series is not affected by a shock of the error terms of ADSK.



Also in this case we can see that the effect of the shock happened in ADBE on ADSK is very small.

Now we perform the Granger Causality test. This test is used to determine if the independent variable affects the value of the response variable. The Null Hypothesis states that the coefficients of the lagged values of the impulse variable are equal to zero, which means that it has no effect on the dependent variable.

Granger causality H0: ADSK.Adjusted do not Granger-cause ADBE.Adjusted

```
data: VAR object varmodel1
F-Test = 0.12155, df1 = 1, df2 = 5024, p-value = 0.7274
```

\$Instant

H0: No instantaneous causality between: ADSK.Adjusted and ADBE.Adjusted

```
data: VAR object varmodel1
Chi-squared = 667.64, df = 1, p-value < 2.2e-16
```

Analyzing the p-value we cannot reject the Null Hypothesis of Granger Causality; so the values of ADBE are not affected by the past values of ADSK.

However we reject the hypothesis that there is no instantaneous causality between the two stochastic time series.

Granger causality H0: ADBE.Adjusted do not Granger-cause ADSK.Adjusted

data: VAR object varmodel1
F-Test = 8.18, df1 = 1, df2 = 5024, p-value = 0.004253

\$Instant

H0: No instantaneous causality between: ADBE.Adjusted and ADSK.Adjusted

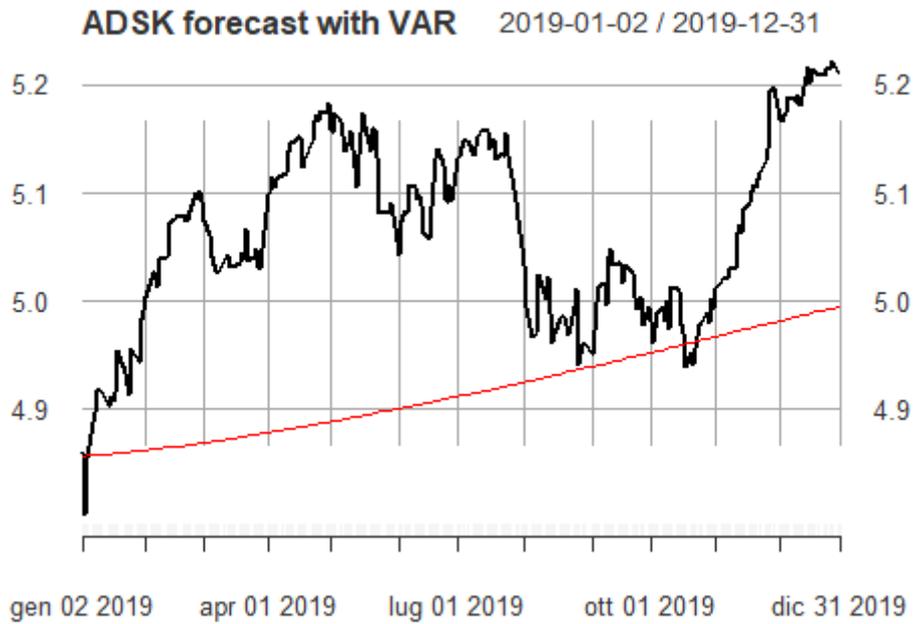
data: VAR object varmodel1
Chi-squared = 667.64, df = 1, p-value < 2.2e-16

In this case we reject the Null Hypothesis in both cases, so the lagged values of ADBE grange-cause the evolution of ADSK.

Once we estimate the VAR process and analyze its dynamic structure, we can use it to forecast the future values of the two stocks.



From this plot we can observe how the VAR process is able to capture the dynamic evolution of the time series; in fact the forecasted values replicate the growth trend of the stock price and spread between real data and the forecasted values is quite small.



In this case the spread between the real values of ADSK and the data obtained by the forecast procedure appears larger, even if the forecasted values replicate the growth trend of the stock price.

	MAE	MSE	RMSE	MAPE
ADBE	0.07787924	0.007954996	0.08919079	0.01380318
ADSK	0.14829433	0.028943939	0.17012918	0.02904907

This table shows the results of the accuracy measures used to compare the forecasted values and the real data; as we could observe from the plots, the forecast of the prices of ADBE is far more accurate than the one of ADSK.

RESULTS & CONCLUSIONS

Univariate Analysis

In this last chapter we present the results of the analysis concerning the accuracy of the forecast methods.

The idea behind this first part of study was to test if the forecasts produced by the application of univariate processes are accurate enough to be used in a short/medium-term strategy. We test also the hypothesis that the forecast of future stock prices is more accurate if done on stochastic variables that exhibits less volatility.

For this reason we choose three different equity securities with different amount of volatility: JPM is the most volatile stock, INTC is the stock that present a medium level of volatility and WMT is the less volatile stock.

So we begin with testing the accuracy of the forecast produced when the dynamic structure of the data is modeled by ARIMA processes:

	MAE	MSE	RMSE	MAPE
JPM. arima(3,1,1). 2015	0.05059132	0.00361274	0.06010607	0.01271047
JPM. arima(1,1,1). 2016	0.08776988	0.01401355	0.11837885	0.02138207
JPM. arima(1,1,1). 2017	0.08318217	0.01184045	0.10881381	0.01852691
JPM. arima(0,1,0). 2018	0.07896711	0.01073035	0.10358740	0.01711784
JPM. arima(0,1,0). 2019	0.09041531	0.01156120	0.10752301	0.01904120
JPM. arima(3,1,1). wholesample	0.10090492	0.01401146	0.11837000	0.02125800

	MAE	MSE	RMSE	MAPE
INTC. arima(3,1,0). 2015	0.19677725	0.047121935	0.21707587	0.05955682
INTC. arima(0,1,0). 2016	0.07786017	0.007708528	0.08779822	0.02305465
INTC. arima(0,1,0). 2017	0.06749627	0.012451951	0.11158831	0.01835969
INTC. arima(1,1,0). 2018	0.07126042	0.006574566	0.08108370	0.01848651
INTC. arima(2,1,2). 2019	0.06789653	0.006398816	0.07999260	0.01729144
INTC. arima(3,1,3). wholesample	0.06858748	0.006650867	0.08155285	0.01743698

	MAE	MSE	RMSE	MAPE
WMT. arima(1,1,2). 2015	0.2156437	0.06903769	0.2627503	0.05316138
WMT. arima(0,1,0). 2016	0.1393309	0.02183796	0.1477767	0.03348919
WMT. arima(0,1,0). 2017	0.1460049	0.03352772	0.1831058	0.03334731
WMT. arima(0,1,0). 2018	0.1208463	0.01779006	0.1333794	0.02714120
WMT. arima(0,1,0). 2019	0.1583538	0.03289450	0.1813684	0.03360604
WMT. arima(1,1,2). wholesample	0.1215040	0.01936108	0.1391441	0.02578675

As we can observe from the tables above, the less accurate forecast is done on the WMT time series, while the measures of accuracy of the forecast produced on JPM and INTC are quite similar; in this case the amount of volatility of the stochastic variable does not affect the precision of the forecast.

Moreover, in this part of the analysis, we decide to test if there is a difference between the accuracy of the prevision if it is obtained modeling the whole sample through ARIMA processes or

dividing the sample in smaller sub-samples. Dividing the whole sample in smaller periods allows us to deal with the possibility that the parameters change over time.

From the results we can say that there is no substantial difference.

At this point we test the hypothesis that the level of volatility affects the accuracy of the forecast. In this case we estimate some GARCH models and use them to produce the prevision.

We forecast both the future returns of the stocks and the future conditional volatility.

	MAE	MSE	RMSE
JPM. sgarch(1,2) returns	0.008553415	0.0001381061	0.01175186
JPM. sgarch(1,3) returns	0.008553410	0.0001381163	0.01175229
JPM. egarch(1,5) returns	0.008570921	0.0001388448	0.01178324
JPM. aparch(1,3) returns	0.008566564	0.0001386586	0.01177534
	MAE	MSE	RMSE
JPM. sgarch(1,2) sigma	0.01290850	0.0001947074	0.01395376
JPM. sgarch(1,3) sigma	0.01280924	0.0001918135	0.01384967
JPM. egarch(1,5) sigma	0.01066288	0.0001354762	0.01163943
JPM. aparch(1,3) sigma	0.01083375	0.0001391173	0.01179480
	MAE	MSE	RMSE
INTC. sgarch(1,1) returns	0.01203319	0.0002911671	0.01706362
INTC. sgarch(1,1).arma(0,0) returns	0.01204009	0.0002910818	0.01706112
INTC. egarch(1,1) returns	0.01204213	0.0002914197	0.01707102
INTC. egarch(1,1).arma(0,0) returns	0.01204441	0.0002911419	0.01706288
INTC. aparch(1,1) returns	0.01204746	0.0002911948	0.01706443
INTC. aparch(1,1).arma(0,0) returns	0.01204789	0.0002912023	0.01706465
	MAE	MSE	RMSE
INTC. sgarch(1,1) sigma	0.01262537	0.0002237455	0.01495812
INTC. sgarch(1,1).arma(0,0) sigma	0.01256183	0.0002221265	0.01490391
INTC. egarch(1,1) sigma	0.01004900	0.0001674148	0.01293889
INTC. egarch(1,1).arma(0,0) sigma	0.01005308	0.0001674757	0.01294124
INTC. aparch(1,1) sigma	0.01001764	0.0001667128	0.01291173
INTC. aparch(1,1).arma(0,0) sigma	0.01001251	0.0001666162	0.01290799
	MAE	MSE	RMSE
WMT. sgarch(1,1) returns	0.006473138	8.124832e-05	0.009013785
WMT. sgarch(1,2) returns	0.006474400	8.126182e-05	0.009014534
WMT. egarch(1,1) returns	0.006477695	8.129476e-05	0.009016361
WMT. egarch(1,2) returns	0.006477227	8.128826e-05	0.009016000
WMT. egarch(1,2).arma(0,1) returns	0.006477439	8.129473e-05	0.009016359
WMT. aparch(1,1) returns	0.006473847	8.126997e-05	0.009014986
	MAE	MSE	RMSE
WMT. sgarch(1,1) sigma	0.007105683	6.959946e-05	0.008342629
WMT. sgarch(1,2) sigma	0.007027529	6.850310e-05	0.008276660
WMT. egarch(1,1) sigma	0.006403920	5.997893e-05	0.007744607
WMT. egarch(1,2) sigma	0.006397638	5.988992e-05	0.007738858
WMT. egarch(1,2).arma(0,1) sigma	0.006405963	5.997706e-05	0.007744486
WMT. aparch(1,1) sigma	0.006503617	6.083423e-05	0.007799630

Observing the results of the accuracy measures, we can notice that the most accurate prevision is done on the values of the WMT stock, which is the less volatile stochastic variable among the three stocks. The accuracy of the forecast done on JPM and INTC is quite similar.

At this point of the results analysis we can say that the hypothesis that the forecast done on less volatile time series is more accurate than the one done on stocks that exhibits higher level of risk is rejected. In fact there is no evidence that the level of volatility affects in a negative way the accuracy of the prevision.

Analyzing the results regarding the JPM and INTC stocks, we can say that the application of ARIMA processes generates forecasts that in the short term are quite accurate; in fact, the models are able to capture the dynamic structure of the data and the future trend of the stock prices. In the medium term, however, the spread between real data and forecasted data is too large to allow us to say that the forecast is reliable.

From the plots we can notice that the spread increases over time.

As for the WMT stock we can say that the forecast obtained by the estimation of the ARIMA process is not reliable in order to predict the future evolution of the stock price and to develop a strategy based on this forecast procedure.

The application of ARCH/GARCH processes to generate forecasts about the future values of the conditional volatility of the three stocks is not accurate. In fact the previsions are not enough reliable to develop a strategy based on the study of the future volatility of the equity securities.

Multivariate Analysis

This part of the thesis concerns the hypothesis for which two equity securities that operate in the same industry could be cointegrated.

This means that the two stocks are tied by a Long-run equilibrium relationship and they share a common stochastic trend in price movements.

So we test the hypothesis that a cointegration relationship exists between the ADBE and ADSK stocks.

First we apply the Phillips-Ouliaris cointegration test, which confirms the hypothesis of cointegration between the two stochastic variables; then we study the residuals of their linear combinations.

On the basis of the Granger's Representation Theorem we estimate the Error Correction Model, so we are able to analyze both the long-term and the short-term dynamic structure of the relationship.

Then we estimate the Vector Error Correction Model through the application of the Johansen cointegration test.

This allow us to jointly analyze the variables and their dynamic evolution over time, studying the long-run equilibrium and the movements of the mean-reverting deviations.

From the results of our analysis we can say that there is a cointegration relationship between the two stocks, and for this reason we can represent the relationship using the Error Correction Model. The ADBE and ADSK stocks are tied in a long-run relationship and they share a common stochastic trend.

We can exploit this relationship developing a “pairs trading” strategy, but we have to be aware of the fact that, even if we know that the future movements of the prices of the series are tied together, we do not know when the short-term disequilibrium is going to be reabsorbed. Moreover, the strategy can be unsuccessful due to the fact that this analysis does not take into consideration the transaction costs of the trading operations.

In the last part of the thesis we model the data of the two stochastic series with a Vector Autoregressive (VAR) process. This process describes the structure of the variables as a system in which each series is formulated as a function of their own lagged values and the lagged values of the other variables.

This allow us to study the relationship and the interdependences between the variables.

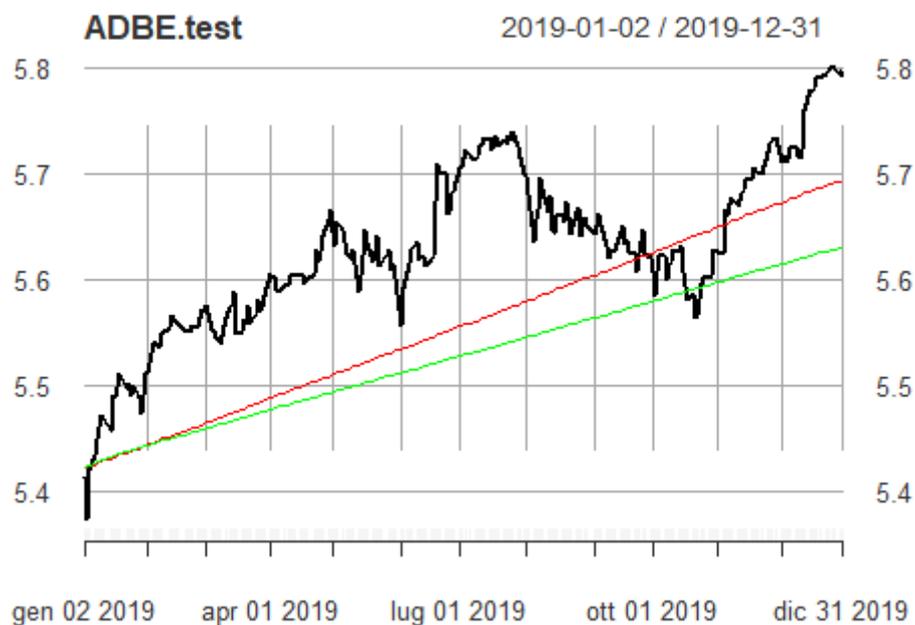
In fact, once we estimate the VAR process, we can apply the Impulse Response Function and the Granger Causality test.

Through the estimation of the VAR process, we are able to make prevision about the future values of the stock price.

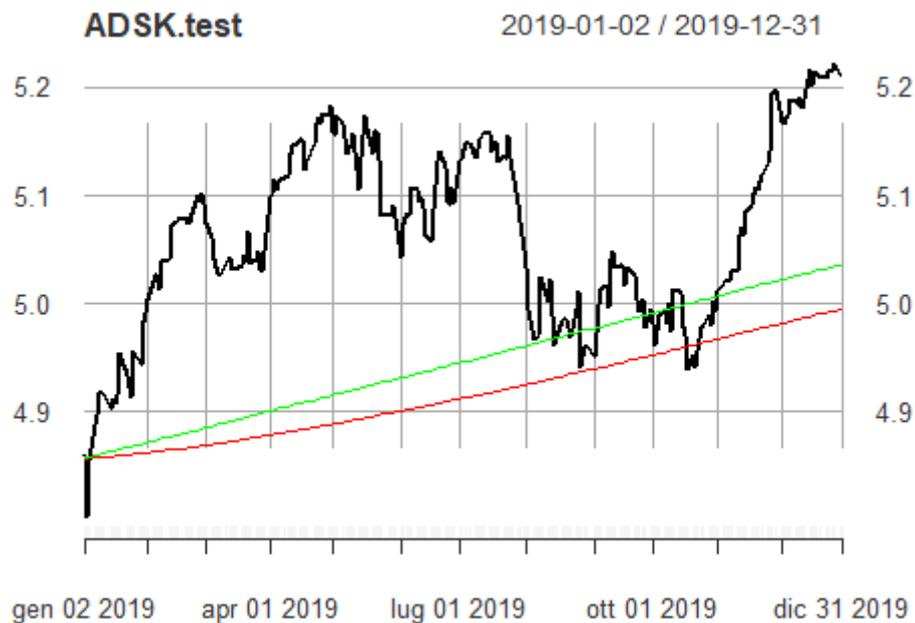
Once we compute the future values of the variables through the forecast procedure, we decide to test the hypothesis which states that the forecast obtained by the estimation of a VAR model is more accurate than the forecast produced by an univariate ARMA process.

The ARIMA processes are estimated using the function “*auto.arima*” in R.

We start forecasting the values of the stock ADBE; as we can see from the plot below, the forecast obtained by the VAR process (red line) is closer to the real data than the values produced by the ARIMA process (green line).



Then we compute the forecast of the future values of the ADSK stock. If we observe the plot below, we can notice that this time the values produced by the ARIMA process (green line) are closer to the real prices of the stock.



Analyzing the forecast results obtained by the estimation of a VAR process, we can say that the forecast procedure is able to capture the growth trend of the prices of the two stocks, especially as regards the future prices of the ADBE stock; we can use the VAR process to develop statistical hypothesis about the future evolution of the values of the time series.

However the spread between the real data and the values generated by the forecast is too large to allow us to use this procedure to develop trading strategies.

From the table below we are able to compare the forecast procedures through the accuracy measures; we can notice that, with respect to the ADBE stock, the forecast produced by the application of the VAR process is more accurate than the one produced by the univariate ARIMA model, while, regarding the ADSK stock, the forecast obtained by the ARIMA model is more precise. These results are consistent with the graphical analysis.

	MAE	MSE	RMSE	MAPE
ADBE	0.07787924	0.007954996	0.08919079	0.01380318
ADBE.arima(3,1,2)	0.10091695	0.012491302	0.11176450	0.01785418
ADSK	0.14829433	0.028943939	0.17012918	0.02904907
ADSK.arima(0,1,0)	0.12298297	0.021445682	0.14644344	0.02407930

Analyzing these results, we cannot reject or validate the hypothesis, due to the fact that in one case the application of a VAR process produces more accurate forecast, on the other case it produces less accurate prevision.

BIBLIOGRAPHY

- Alexander C., Giblin I., Weddington W. III, *Cointegration and Asset Allocation: a new active hedge fund strategy*, ISMA Centre Discussion Paper Series in Finance, 2002
- Bollerslev T., *Generalized autoregressive conditional heteroskedasticity*, Journal of Econometrics, vol. 31, pages 307-327, 1986
- Box G.E.P., Jenkins G.M., *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, 1976
- Brooks C., *Introductory Econometrics for Finance*, 3rd Ed., Cambridge University Press, Cambridge, 2014
- Cerchi M., Havenner A., *Cointegration and Stock Prices*, Journal of Economic Dynamics and Control, vol. 12, pages 333-346. North-Holland, 1988
- Enders W., *Applied Econometric Time Series*, 4th Ed., Wiley, New York, 2004
- Engle R.F., *Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation*, Econometrica, vol. 50, pages 987-1007, 1982
- Engle R.F., *GARCH 101: The use of ARCH/GARCH Models in Applied Econometrics*, Journal of Economic Perspectives, vol. 15, no. 4, pages 157-168, 2001
- Engle R.F., Granger C.W.J., *Cointegration and error correction: representation, estimation and testing*, Econometrica, vol. 55, pages 251-276, 1987
- Engle R.F., Yoo B.S., *Forecasting and testing in Co-Integrated systems*, Journal of Econometrics, vol. 35, pages 143-159, 1987
- Engsted T., Johansen S., *Granger's Representation Theorem and Multicointegration*, EUI Working Paper ECO, no. 97/15, 1997
- Fama E.F., French K.R., *Permanent and temporary components of Stock prices*, Journal of Political Economy, vol. 96, no. 2, 1988, pages 246-273
- Granger C.W.J., *Some properties of time series data and their use in econometric model specification*, Journal of Econometrics, vol. 16, pages 121-130, 1981
- Hamilton J.D., *Time Series Analysis*, Princeton University Press, Princeton, 1994
- Johansen S., *Statistical Analysis of Cointegration Vectors*, Journal of Economic Dynamics and Control, vol. 12, pages 231-254. North-Holland, 1988
- Johnston J., DiNardo J., *Econometrics Methods*, 4th Ed., McGraw-Hill, New York, 1997
- Ljung G.M., Box G.E.P., *On a measure of lack of fit in time series models*, Biometrika, vol. 65, pages 297-303, 1978

- Lucchetti R., *Appunti di Analisi delle Serie Storiche*, 2015
- Mandelbrot, B., *The variation of certain speculative prices*, Journal of Business, vol. 36, pages 394-419, 1963
- Pfaff B., *Analysis of Integrated and Cointegrated Time Series with R*, Springer, New York, 2006
- Phillips P.C.B., *Understanding spurious regressions in Econometrics*, Journal of Econometrics, vol. 33, pages 311-340, 1986
- Phillips P.C.B., Ouliaris S., *Asymptotic properties of residual based tests for cointegration*, Econometrica, vol. 58, pages 165-193, 1990
- Phillips P.C.B., Perron P., *Testing for a unit root in time series regression*, Biometrika, vol. 75, pages 335-346, 1988
- Pole A., *Statistical Arbitrage*, Wiley, New York, 2007
- Ruppert D., *Statistics and Data Analysis for Financial Engineering*, Springer, New York, 2011
- Stock J.H., Watson M.W., *Testing for Common Trends*, Journal of the American Statistical Association, Vol. 83, No. 404, pages 1097-1107, 1987
- Tsay R.S., *Analysis of Financial Time Series*, 2nd Ed., Wiley, New York, 2005
- Verbeek M., *Econometria*, Zanichelli, Bologna, 2006
- Vidyamurthy G., *Pairs Trading: Quantitative methods and analysis*, Wiley, New York, 2004

APPENDIX

```
## R CODES
library(MASS)
library(Metrics)
library(quantmod)
library(forecast)
library("xlsx")
library(tseries)
library(timeSeries)
library(dplyr)
library(fGarch)
library(rugarch)
library(rmgarch)
library(urca)
library(lmtest)
library(car)
library(vars)

#Univariate Analysis
tickers= c("JPM","INTC","WMT")
ntickers= length(tickers)
tick<-tickers
for (i in 1:ntickers) {
  tick[i]<- getSymbols(tickers[i], from="2009-01-01", to= "2020-01-01")}
# ARIMA PROCESSES
## JPM
JPM.price<- JPM$JPM.Adjusted
JPM.log<- log(JPM.price)
JPM.returns<- diff(JPM.log)[-1]
plot(JPM.log)
plot(JPM.returns, main="JPM.returns")
plot(JPM.returns^2, ylim=c(0,0.07), main="JPM.squared returns")
basicStats(JPM.returns)
# samples per stima modello
JPM.log.20092015<- JPM.log["20090101/20150101"] #sample per forecast 2015
JPM.log.20102016<- JPM.log["20100101/20160101"] #sample per forecast 2016
JPM.log.20112017<- JPM.log["20110101/20170101"] #sample per forecast 2017
JPM.log.20122018<- JPM.log["20120101/20180101"] #sample per forecast 2018
JPM.log.20132019<- JPM.log["20130101/20190101"] #sample per forecast 2019
# stationarity sub-sample
pp.test(JPM.log.20092015)
pp.test(JPM.log.20102016)
pp.test(JPM.log.20112017)
pp.test(JPM.log.20122018)
pp.test(JPM.log.20132019)
# samples for test
JPM.log.20152016<- JPM.log["20150101/20160101"]
n20152016<- length(JPM.log.20152016)
JPM.log.20162017<- JPM.log["20160101/20170101"]
```

```

n20162017<- length(JPM.log.20162017)
JPM.log.20172018<- JPM.log["20170101/20180101"]
n20172018<- length(JPM.log.20172018)
JPM.log.20182019<- JPM.log["20180101/20190101"]
n20182019<- length(JPM.log.20182019)
JPM.log.20192020<- JPM.log["20190101/20200101"]
n20192020<- length(JPM.log.20192020)
# stationarity & integration
JPM.returns.20092015<- diff(JPM.log.20092015)[-1]
pp.test(JPM.log.20092015)
pp.test(JPM.returns.20092015)
JPM.returns.20102016<- diff(JPM.log.20102016)[-1]
pp.test(JPM.log.20102016)
pp.test(JPM.returns.20102016)
JPM.returns.20112017<- diff(JPM.log.20112017)[-1]
pp.test(JPM.log.20112017)
pp.test(JPM.returns.20112017)
JPM.returns.20122018<- diff(JPM.log.20122018)[-1]
pp.test(JPM.log.20122018)
pp.test(JPM.returns.20122018)
JPM.returns.20132019<- diff(JPM.log.20132019)[-1]
pp.test(JPM.log.20132019)
pp.test(JPM.returns.20132019)
# estimation models
JPM.autoarima.20092015<- auto.arima(JPM.log.20092015)
summary(JPM.autoarima.20092015)
pp.test(JPM.autoarima.20092015$residuals)
Box.test(JPM.autoarima.20092015$residuals, type="Ljung-Box")
JPM.autoarima.20102016<- auto.arima(JPM.log.20102016)
summary(JPM.autoarima.20102016)
pp.test(JPM.autoarima.20102016$residuals)
Box.test(JPM.autoarima.20102016$residuals, type="Ljung-Box")
JPM.autoarima.20112017<- auto.arima(JPM.log.20112017)
summary(JPM.autoarima.20112017)
pp.test(JPM.autoarima.20112017$residuals)
Box.test(JPM.autoarima.20112017$residuals, type="Ljung-Box")
JPM.autoarima.20122018<- auto.arima(JPM.log.20122018, trace=T)
summary(JPM.autoarima.20122018)
pp.test(JPM.autoarima.20122018$residuals)
Box.test(JPM.autoarima.20122018$residuals, type="Ljung-Box")
JPM.autoarima.20132019<- auto.arima(JPM.log.20132019)
summary(JPM.autoarima.20132019)
pp.test(JPM.autoarima.20132019$residuals)
Box.test(JPM.autoarima.20132019$residuals, type="Ljung-Box")
# test for accuracy
JPM.forecast.20152016<- forecast(JPM.autoarima.20092015, h=n20152016)
JPM.forecast.20152016.mean<-JPM.forecast.20152016$mean
JPM.dataframe.20152016 <- data.frame(exp(JPM.log.20152016), exp(JPM.forecast.20152016.mean))
names(JPM.dataframe.20152016)<- c("JPM.Adjusted", "arima(3,1,1)")
JPM.forecast.20152016.mean.xts<- xts(JPM.forecast.20152016.mean, order.by = index(JPM.log.20152016))
plot(exp(JPM.log.20152016), main="JPM forecast 2015", ylim=c(44,66))
lines(exp(JPM.forecast.20152016.mean.xts), col="red")

```

```

JPM.f.20152016.MAE<- mae(JPM.log.20152016,JPM.forecast.20152016.mean.xts)
JPM.f.20152016.MSE<- mse(JPM.log.20152016,JPM.forecast.20152016.mean.xts)
JPM.f.20152016.RMSE<- rmse(JPM.log.20152016,JPM.forecast.20152016.mean.xts)
JPM.f.20152016.MAPE<- mape(JPM.log.20152016,JPM.forecast.20152016.mean.xts)
JPM.accuracy.20152016<- data.frame(JPM.f.20152016.MAE,
                                   JPM.f.20152016.MSE,
                                   JPM.f.20152016.RMSE,
                                   JPM.f.20152016.MAPE)

JPM.forecast.20162017<- forecast(JPM.autoarima.20102016, h=n20162017)
JPM.forecast.20162017.mean<-JPM.forecast.20162017$mean
JPM.dataframe.20162017<- data.frame(exp(JPM.log.20162017), exp(JPM.forecast.20162017.mean))
names(JPM.dataframe.20162017)<- c("JPM.Adjusted", "arima(1,1,1)")
JPM.forecast.20162017.mean.xts<- xts(JPM.forecast.20162017.mean, order.by = index(JPM.log.20162017))
plot(exp(JPM.log.20162017) , main="JPM forecast 2016", ylim=c(44,81))
lines(exp(JPM.forecast.20162017.mean.xts), col="red")
JPM.f.20162017.MAE<- mae(JPM.log.20162017,JPM.forecast.20162017.mean.xts)
JPM.f.20162017.MSE<- mse(JPM.log.20162017,JPM.forecast.20162017.mean.xts)
JPM.f.20162017.RMSE<- rmse(JPM.log.20162017,JPM.forecast.20162017.mean.xts)
JPM.f.20162017.MAPE<- mape(JPM.log.20162017,JPM.forecast.20162017.mean.xts)
JPM.accuracy.20162017<- data.frame(JPM.f.20162017.MAE,
                                   JPM.f.20162017.MSE,
                                   JPM.f.20162017.RMSE,
                                   JPM.f.20162017.MAPE)

JPM.forecast.20172018<- forecast(JPM.autoarima.20112017, h=n20172018)
JPM.forecast.20172018.mean<-JPM.forecast.20172018$mean
JPM.dataframe.20172018<- data.frame(exp(JPM.log.20172018), exp(JPM.forecast.20172018.mean))
names(JPM.dataframe.20172018)<- c("JPM.Adjusted", "arima(1,1,1)")
JPM.forecast.20172018.mean.xts<- xts(JPM.forecast.20172018.mean, order.by = index(JPM.log.20172018))
plot(exp(JPM.log.20172018), main="JPM forecast 2017", ylim=c(74,106))
lines(exp(JPM.forecast.20172018.mean.xts), col="red")
JPM.f.20172018.MAE<- mae(JPM.log.20172018,JPM.forecast.20172018.mean.xts)
JPM.f.20172018.MSE<- mse(JPM.log.20172018,JPM.forecast.20172018.mean.xts)
JPM.f.20172018.RMSE<- rmse(JPM.log.20172018,JPM.forecast.20172018.mean.xts)
JPM.f.20172018.MAPE<- mape(JPM.log.20172018,JPM.forecast.20172018.mean.xts)
JPM.accuracy.20172018<- data.frame(JPM.f.20172018.MAE,
                                   JPM.f.20172018.MSE,
                                   JPM.f.20172018.RMSE,
                                   JPM.f.20172018.MAPE)

JPM.forecast.20182019<- forecast(JPM.autoarima.20122018, h=n20182019)
JPM.forecast.20182019.mean<-JPM.forecast.20182019$mean
JPM.dataframe.20182019<- data.frame(exp(JPM.log.20182019), exp(JPM.forecast.20182019.mean))
names(JPM.dataframe.20182019)<- c("JPM.Adjusted", "arima(0,1,0)")
JPM.forecast.20182019.mean.xts<- xts(JPM.forecast.20182019.mean, order.by = index(JPM.log.20182019))
plot(exp(JPM.log.20182019), main="JPM forecast 2018", ylim=c(84,131))
lines(exp(JPM.forecast.20182019.mean.xts), col="red")
JPM.f.20182019.MAE<- mae(JPM.log.20182019,JPM.forecast.20182019.mean.xts)
JPM.f.20182019.MSE<- mse(JPM.log.20182019,JPM.forecast.20182019.mean.xts)
JPM.f.20182019.RMSE<- rmse(JPM.log.20182019,JPM.forecast.20182019.mean.xts)
JPM.f.20182019.MAPE<- mape(JPM.log.20182019,JPM.forecast.20182019.mean.xts)
JPM.accuracy.20182019<- data.frame(JPM.f.20182019.MAE,
                                   JPM.f.20182019.MSE,
                                   JPM.f.20182019.RMSE,

```

```

JPM.f.20182019.MAPE)
JPM.forecast.20192020<- forecast(JPM.autoarima.20132019, h=n20192020)
JPM.forecast.20192020.mean<-JPM.forecast.20192020$mean
JPM.dataframe.20192020<- data.frame(exp(JPM.log.20192020), exp(JPM.forecast.20192020.mean))
names(JPM.dataframe.20192020)<- c("JPM.Adjusted", "arima(0,1,0)")
JPM.forecast.20192020.mean.xts<- xts(JPM.forecast.20192020.mean, order.by = index(JPM.log.20192020))
plot(exp(JPM.log.20192020), main="JPM forecast 2019", ylim=c(89,141))
lines(exp(JPM.forecast.20192020.mean.xts), col="red")
JPM.f.20192020.MAE<- mae(JPM.log.20192020,JPM.forecast.20192020.mean.xts)
JPM.f.20192020.MSE<- mse(JPM.log.20192020,JPM.forecast.20192020.mean.xts)
JPM.f.20192020.RMSE<- rmse(JPM.log.20192020,JPM.forecast.20192020.mean.xts)
JPM.f.20192020.MAPE<- mape(JPM.log.20192020,JPM.forecast.20192020.mean.xts)
JPM.accuracy.20192020<- data.frame(JPM.f.20192020.MAE,
                                   JPM.f.20192020.MSE,
                                   JPM.f.20192020.RMSE,
                                   JPM.f.20192020.MAPE)
names(JPM.accuracy.20152016)<- c("MAE","MSE","RMSE","MAPE")
row.names(JPM.accuracy.20152016)<- "JPM.arima(3,1,1).2015"
names(JPM.accuracy.20162017)<- c("MAE","MSE","RMSE","MAPE")
row.names(JPM.accuracy.20162017)<- "JPM.arima(1,1,1).2016"
names(JPM.accuracy.20172018)<- c("MAE","MSE","RMSE","MAPE")
row.names(JPM.accuracy.20172018)<- "JPM.arima(1,1,1).2017"
names(JPM.accuracy.20182019)<- c("MAE","MSE","RMSE","MAPE")
row.names(JPM.accuracy.20182019)<- "JPM.arima(0,1,0).2018"
names(JPM.accuracy.20192020)<- c("MAE","MSE","RMSE","MAPE")
row.names(JPM.accuracy.20192020)<- "JPM.arima(0,1,0).2019"
JPM.accuracy<- rbind(JPM.accuracy.20152016,
                    JPM.accuracy.20162017,
                    JPM.accuracy.20172018,
                    JPM.accuracy.20182019,
                    JPM.accuracy.20192020)
JPM.accuracy
# estimation and forecast with whole sample
JPM.log.20092019<- JPM.log["20090101/20190101"]
pp.test(JPM.log.20092019)
#JPM.log.20192020
JPM.autoarima.20092019<- auto.arima(JPM.log.20092019)
summary(JPM.autoarima.20092019) #arima(3,1,1)
adf.test(JPM.autoarima.20092019$residuals)
Box.test(JPM.autoarima.20092019$residuals, type="Ljung-Box")
JPM.forecast.20192020.whole<- forecast(JPM.autoarima.20092019, h=n20192020)
JPM.forecast.20192020.wholemean<-JPM.forecast.20192020.whole$mean
JPM.dataframe.20192020.whole<- data.frame(exp(JPM.log.20192020),
exp(JPM.forecast.20192020.wholemean))
names(JPM.dataframe.20192020.whole)<- c("JPM.Adjusted", "arima(3,1,1)")
JPM.forecast.20192020.wholemean.xts<- xts(JPM.forecast.20192020.wholemean, order.by =
index(JPM.log.20192020))
plot(exp(JPM.log.20192020), main="JPM forecast 2019 whole sample", ylim=c(89,141))
lines(exp(JPM.forecast.20192020.wholemean.xts), col="red")
JPM.f.20192020.whole.MAE<- mae(JPM.log.20192020,JPM.forecast.20192020.wholemean.xts)
JPM.f.20192020.whole.MSE<- mse(JPM.log.20192020,JPM.forecast.20192020.wholemean.xts)
JPM.f.20192020.whole.RMSE<- rmse(JPM.log.20192020,JPM.forecast.20192020.wholemean.xts)

```

```

JPM.f.20192020.whole.MAPE<- mape(JPM.log.20192020,JPM.forecast.20192020.wholemean.xls)
JPM.accuracy.20192020.whole<- data.frame(JPM.f.20192020.whole.MAE,
      JPM.f.20192020.whole.MSE,
      JPM.f.20192020.whole.RMSE,
      JPM.f.20192020.whole.MAPE)
names(JPM.accuracy.20192020.whole)<- c("MAE","MSE","RMSE","MAPE")
row.names(JPM.accuracy.20192020.whole)<- "JPM.arima(3,1,1).wholesample"
rbind(JPM.accuracy, JPM.accuracy.20192020.whole)

# ARCH/GARCH PROCESSES
## JPM
## TEST WITH SAMPLE 2019
JPM.sample<-JPM.log["20090101/20190101"] #prima parte su cui fai il modello
JPM.test<- JPM.log["20190101/20200101"] #seconda parte per forecast
JPM.sample.ts<-ts(JPM.sample)
nforecast<- nrow(JPM.test)
JPM.samplediff<- diff(JPM.sample.ts)
JPM.diff.acf<- acf(JPM.samplediff)
JPM.diff.pacf<- pacf(JPM.samplediff)
# estimation GARCH models
spec.JPM.sgarch12<- ugarchspec(JPM.samplediff,
      variance.model =list(model = "sGARCH", garchOrder = c(1, 2)),
      mean.model = list(armaOrder = c(3, 1)),
      distribution.model = "std")
ugfit.JPM.sgarch12= ugarchfit(spec=spec.JPM.sgarch12, data=JPM.samplediff)
ugfit.JPM.sgarch12
JPM.res.sgarch12<- residuals(ugfit.JPM.sgarch12)
pp.test(JPM.res.sgarch12)
spec.JPM.sgarch13<- ugarchspec(JPM.samplediff,
      variance.model =list(model = "sGARCH", garchOrder = c(1, 3)),
      mean.model = list(armaOrder = c(3, 1)),
      distribution.model = "std")
ugfit.JPM.sgarch13= ugarchfit(spec=spec.JPM.sgarch13, data=JPM.samplediff)
ugfit.JPM.sgarch13
JPM.res.sgarch13<- residuals(ugfit.JPM.sgarch13)
pp.test(JPM.res.sgarch13)
spec.JPM.egarch15<- ugarchspec(JPM.samplediff,
      variance.model =list(model = "eGARCH", garchOrder = c(1, 5)),
      mean.model = list(armaOrder = c(0, 0)),
      distribution.model = "std")
ugfit.JPM.egarch15= ugarchfit(spec=spec.JPM.egarch15, data=JPM.samplediff)
ugfit.JPM.egarch15
JPM.res.egarch15<- residuals(ugfit.JPM.egarch15)
pp.test(JPM.res.egarch15)
spec.JPM.aparch13<- ugarchspec(JPM.samplediff,
      variance.model =list(model = "fGARCH", garchOrder = c(1, 3),
      submodel="APARCH"),
      mean.model = list(armaOrder = c(3, 1)),
      distribution.model = "std")
ugfit.JPM.aparch13= ugarchfit(spec=spec.JPM.aparch13, data=JPM.samplediff)
ugfit.JPM.aparch13
JPM.res.aparch13<- residuals(ugfit.JPM.aparch13)

```

```

pp.test(JPM.res.aparch13)
JPM.sgarch.12IC<- infocriteria(ugfit.JPM.sgarch12)
JPM.sgarch.13IC<- infocriteria(ugfit.JPM.sgarch13)
JPM.egarch.15IC<- infocriteria(ugfit.JPM.egarch15)
JPM.aparch.13IC<- infocriteria(ugfit.JPM.aparch13)
JPM.garchmodelsIC.df<- data.frame(JPM.sgarch.12IC,JPM.sgarch.13IC,JPM.egarch.15IC,JPM.aparch.13IC)
names(JPM.garchmodelsIC.df)<-
c("JPM.sGARCH(1,2)","JPM.sGARCH(1,3)","JPM.eGARCH(1,5)","JPM.APARCH(1,3)")
JPM.garchmodelsIC.df<- t(JPM.garchmodelsIC.df)
JPM.garchmodelsIC.df
### FORECAST
### sgarch(1,2)
garchforecast.JPM.sgarch12=ugarchforecast(ugfit.JPM.sgarch12, n.ahead=nforecast, n.roll=0, out.sample =
0)
garchforecast.JPM.sgarch12 #series=expec returns, sigma=conditional standard deviation
JPM.returns30<- tail(JPM.returns, n=nforecast)
returnsforecast.JPM.sgarch12<- garchforecast.JPM.sgarch12@forecast$seriesFor
# comparison sigma
sigmaforecast.JPM.sgarch12<- garchforecast.JPM.sgarch12@forecast$sigmaFor
JPM.sigma<- (JPM.returns30- mean(JPM.returns))^2
JPM.sigma<- sqrt(JPM.sigma)
mean(JPM.sigma)
mean(sigmaforecast.JPM.sgarch12)
#measures of accuracy for returns
MAE.JPM.sgarch12<- mae(JPM.returns.tail,returnsforecast.JPM.sgarch12)
MSE.JPM.sgarch12<- mse(JPM.returns.tail,returnsforecast.JPM.sgarch12)
RMSE.JPM.sgarch12<- rmse(JPM.returns.tail,returnsforecast.JPM.sgarch12)
dfaccuracy.JPM.sgarch12<- data.frame(MAE.JPM.sgarch12,
MSE.JPM.sgarch12,
RMSE.JPM.sgarch12)
row.names(dfaccuracy.JPM.sgarch12)<- "JPM.sgarch(1,2) returns"
names(dfaccuracy.JPM.sgarch12)<- c("MAE","MSE","RMSE")
dfaccuracy.JPM.sgarch12
#measures of accuracy for sigma
MAE.JPM.sgarch12.sigma<- mae(JPM.sigma,sigmaforecast.JPM.sgarch12)
MSE.JPM.sgarch12.sigma<- mse(JPM.sigma,sigmaforecast.JPM.sgarch12)
RMSE.JPM.sgarch12.sigma<- rmse(JPM.sigma,sigmaforecast.JPM.sgarch12)
dfaccuracy.JPM.sgarch12.sigma<- data.frame(MAE.JPM.sgarch12.sigma,
MSE.JPM.sgarch12.sigma,
RMSE.JPM.sgarch12.sigma)
row.names(dfaccuracy.JPM.sgarch12.sigma)<- "JPM.sgarch(1,2) sigma"
names(dfaccuracy.JPM.sgarch12.sigma)<- c("MAE","MSE","RMSE")
dfaccuracy.JPM.sgarch12.sigma
returnsforecast.JPM.sgarch12.xts<- xts(returnsforecast.JPM.sgarch12, order.by = index(JPM.returns.tail))
plot(JPM.returns.tail, main="JPM forecast 2019 sGARCH(1,2)")
lines(returnsforecast.JPM.sgarch12.xts, col="red")
sigmaforecast.JPM.sgarch12.xts<- xts(sigmaforecast.JPM.sgarch12, order.by = index(JPM.returns.tail))
plot(JPM.sigma, main="JPM sigma forecast 2019 sGARCH(1,2)")
lines(sigmaforecast.JPM.sgarch12.xts, col="red")
### sgarch(1,3)
garchforecast.JPM.sgarch13=ugarchforecast(ugfit.JPM.sgarch13, n.ahead=nforecast, n.roll=0, out.sample =
0)

```

```

garchforecast.JPM.sgarch13 #series=expec returns, sigma=conditional standard deviation
JPM.returns.tail<- tail(JPM.returns, n=nforecast)
returnsforecast.JPM.sgarch13<- garchforecast.JPM.sgarch13@forecast$seriesFor
# comparison sigma
sigmaforecast.JPM.sgarch13<- garchforecast.JPM.sgarch13@forecast$sigmaFor
JPM.sigma<- (JPM.returns.tail- mean(JPM.returns))^2
JPM.sigma<- sqrt(JPM.sigma)
mean(JPM.sigma)
mean(sigmaforecast.JPM.sgarch13)
#measures of accuracy for returns
MAE.JPM.sgarch13<- mae(JPM.returns.tail,returnsforecast.JPM.sgarch13)
MSE.JPM.sgarch13<- mse(JPM.returns.tail,returnsforecast.JPM.sgarch13)
RMSE.JPM.sgarch13<- rmse(JPM.returns.tail,returnsforecast.JPM.sgarch13)
dfaccuracy.JPM.sgarch13<- data.frame(MAE.JPM.sgarch13,
                                     MSE.JPM.sgarch13,
                                     RMSE.JPM.sgarch13)
row.names(dfaccuracy.JPM.sgarch13)<- "JPM.sgarch(1,3) returns"
names(dfaccuracy.JPM.sgarch13)<- c("MAE","MSE","RMSE")
dfaccuracy.JPM.sgarch13
#measures of accuracy for sigma
MAE.JPM.sgarch13.sigma<- mae(JPM.sigma,sigmaforecast.JPM.sgarch13)
MSE.JPM.sgarch13.sigma<- mse(JPM.sigma,sigmaforecast.JPM.sgarch13)
RMSE.JPM.sgarch13.sigma<- rmse(JPM.sigma,sigmaforecast.JPM.sgarch13)
dfaccuracy.JPM.sgarch13.sigma<- data.frame(MAE.JPM.sgarch13.sigma,
                                           MSE.JPM.sgarch13.sigma,
                                           RMSE.JPM.sgarch13.sigma)
row.names(dfaccuracy.JPM.sgarch13.sigma)<- "JPM.sgarch(1,3) sigma"
names(dfaccuracy.JPM.sgarch13.sigma)<- c("MAE","MSE","RMSE")
dfaccuracy.JPM.sgarch13.sigma
returnsforecast.JPM.sgarch13.xts<- xts(returnsforecast.JPM.sgarch13, order.by = index(JPM.returns.tail))
plot(JPM.returns.tail, main="JPM forecast 2019 sGARCH(1,3)")
lines(returnsforecast.JPM.sgarch13.xts, col="red")
sigmaforecast.JPM.sgarch13.xts<- xts(sigmaforecast.JPM.sgarch13, order.by = index(JPM.returns.tail))
plot(JPM.sigma, main="JPM sigma forecast 2019 sGARCH(1,3)")
lines(sigmaforecast.JPM.sgarch13.xts, col="red")
### egarch(1,5)
garchforecast.JPM.egarch15=ugarchforecast(ugfit.JPM.egarch15, n.ahead=nforecast, n.roll=0, out.sample =
0)
garchforecast.JPM.egarch15 #series=expec returns, sigma=conditional standard deviation
JPM.returns.tail<- tail(JPM.returns, n=nforecast)
returnsforecast.JPM.egarch15<- garchforecast.JPM.egarch15@forecast$seriesFor
# comparison sigma
sigmaforecast.JPM.egarch15<- garchforecast.JPM.egarch15@forecast$sigmaFor
JPM.sigma<- (JPM.returns.tail- mean(JPM.returns))^2
JPM.sigma<- sqrt(JPM.sigma)
mean(JPM.sigma)
mean(sigmaforecast.JPM.egarch15)
#measures of accuracy for returns
MAE.JPM.egarch15<- mae(JPM.returns.tail,returnsforecast.JPM.egarch15)
MSE.JPM.egarch15<- mse(JPM.returns.tail,returnsforecast.JPM.egarch15)
RMSE.JPM.egarch15<- rmse(JPM.returns.tail,returnsforecast.JPM.egarch15)
dfaccuracy.JPM.egarch15<- data.frame(MAE.JPM.egarch15,

```

```

MSE.JPM.egarch15,
RMSE.JPM.egarch15)
row.names(dfaccuracy.JPM.egarch15)<- "JPM.egarch(1,5) returns"
names(dfaccuracy.JPM.egarch15)<- c("MAE","MSE","RMSE")
dfaccuracy.JPM.egarch15
#measures of accuracy for sigma
MAE.JPM.egarch15.sigma<- mae(JPM.sigma,sigmaforecast.JPM.egarch15)
MSE.JPM.egarch15.sigma<- mse(JPM.sigma,sigmaforecast.JPM.egarch15)
RMSE.JPM.egarch15.sigma<- rmse(JPM.sigma,sigmaforecast.JPM.egarch15)
dfaccuracy.JPM.egarch15.sigma<- data.frame(MAE.JPM.egarch15.sigma,
MSE.JPM.egarch15.sigma,
RMSE.JPM.egarch15.sigma)
row.names(dfaccuracy.JPM.egarch15.sigma)<- "JPM.egarch(1,5) sigma"
names(dfaccuracy.JPM.egarch15.sigma)<- c("MAE","MSE","RMSE")
dfaccuracy.JPM.egarch15.sigma
returnsforecast.JPM.egarch15.xts<- xts(returnsforecast.JPM.egarch15, order.by = index(JPM.returns.tail))
plot(JPM.returns.tail, main="JPM forecast 2019 eGARCH(1,5)")
lines(returnsforecast.JPM.egarch15.xts, col="red")
sigmaforecast.JPM.egarch15.xts<- xts(sigmaforecast.JPM.egarch15, order.by = index(JPM.returns.tail))
plot(JPM.sigma, main="JPM sigma forecast 2019 eGARCH(1,5)")
lines(sigmaforecast.JPM.egarch15.xts, col="red")
### aparch(1,3)
garchforecast.JPM.aparch13=ugarchforecast(ugfit.JPM.aparch13, n.ahead=nforecast, n.roll=0, out.sample =
0)
garchforecast.JPM.aparch13 #series=expec returns, sigma=conditional standard deviation
JPM.returns.tail<- tail(JPM.returns, n=nforecast)
returnsforecast.JPM.aparch13<- garchforecast.JPM.aparch13@forecast$seriesFor
# comparison sigma
sigmaforecast.JPM.aparch13<- garchforecast.JPM.aparch13@forecast$sigmaFor
JPM.sigma<- (JPM.returns.tail- mean(JPM.returns))^2
JPM.sigma<- sqrt(JPM.sigma)
mean(JPM.sigma)
mean(sigmaforecast.JPM.aparch13)
#measures of accuracy for returns
MAE.JPM.aparch13<- mae(JPM.returns.tail,returnsforecast.JPM.aparch13)
MSE.JPM.aparch13<- mse(JPM.returns.tail,returnsforecast.JPM.aparch13)
RMSE.JPM.aparch13<- rmse(JPM.returns.tail,returnsforecast.JPM.aparch13)
dfaccuracy.JPM.aparch13<- data.frame(MAE.JPM.aparch13,
MSE.JPM.aparch13,
RMSE.JPM.aparch13)
row.names(dfaccuracy.JPM.aparch13)<- "JPM.aparch(1,3) returns"
names(dfaccuracy.JPM.aparch13)<- c("MAE","MSE","RMSE")
dfaccuracy.JPM.aparch13
#measures of accuracy for sigma
MAE.JPM.aparch13.sigma<- mae(JPM.sigma,sigmaforecast.JPM.aparch13)
MSE.JPM.aparch13.sigma<- mse(JPM.sigma,sigmaforecast.JPM.aparch13)
RMSE.JPM.aparch13.sigma<- rmse(JPM.sigma,sigmaforecast.JPM.aparch13)
dfaccuracy.JPM.aparch13.sigma<- data.frame(MAE.JPM.aparch13.sigma,
MSE.JPM.aparch13.sigma,
RMSE.JPM.aparch13.sigma)
row.names(dfaccuracy.JPM.aparch13.sigma)<- "JPM.aparch(1,3) sigma"
names(dfaccuracy.JPM.aparch13.sigma)<- c("MAE","MSE","RMSE")

```

```

dfaccuracy.JPM.aparch13.sigma
returnsforecast.JPM.aparch13.xts<- xts(returnsforecast.JPM.aparch13, order.by = index(JPM.returns.tail))
plot(JPM.returns.tail, main="JPM forecast 2019 APARCH(1,3)")
lines(returnsforecast.JPM.aparch13.xts, col="red")
sigmaforecast.JPM.aparch13.xts<- xts(sigmaforecast.JPM.aparch13, order.by = index(JPM.returns.tail))
plot(JPM.sigma, main="JPM sigma forecast 2019 APARCH(1,3)")
lines(sigmaforecast.JPM.aparch13.xts, col="red")
### RESULTS
JPM.forecast.df<- data.frame(JPM.returns.tail,returnsforecast.JPM.sgarch12,returnsforecast.JPM.sgarch13,
returnsforecast.JPM.egarch15,returnsforecast.JPM.aparch13)
names(JPM.forecast.df)<- c("JPM.Adjusted", "sgarch(1,2)", "sgarch(1,3)", "egarch(1,5)", "aparch(1,3)")
tail(JPM.forecast.df) # comparison returns
JPM.sigmacomparison<- data.frame(JPM.sigma, sigmaforecast.JPM.sgarch12,sigmaforecast.JPM.sgarch13,
sigmaforecast.JPM.egarch15, sigmaforecast.JPM.aparch13)
names(JPM.sigmacomparison)<- c("JPM.Adjusted", "sgarch(1,2)", "sgarch(1,3)", "egarch(1,5)", "aparch(1,3)")
tail(JPM.sigmacomparison)
rbind(dfaccuracy.JPM.sgarch12,dfaccuracy.JPM.sgarch13,dfaccuracy.JPM.egarch15,
dfaccuracy.JPM.aparch13)
rbind(dfaccuracy.JPM.sgarch12.sigma,dfaccuracy.JPM.sgarch13.sigma,dfaccuracy.JPM.egarch15.sigma,
dfaccuracy.JPM.aparch13.sigma)

#Multivariate Analysis
tickers= c("ADBE","ADSK")
ntickers= length(tickers)
tick<-tickers
for (i in 1:ntickers) {
tick[i]<- getSymbols(tickers[i], from="2010-01-01", to= "2020-01-01")}
ADBE.price<- ADBE$ADBE.Adjusted
ADSK.price<- ADSK$ADSK.Adjusted
ADBE.log<- log(ADBE.price)
ADSK.log<- log(ADSK.price)
plot(ADBE.log, main="Cointegration Plot")
lines(ADSK.log, col="red", lwd=2)
addLegend("topleft", on=0,
legend.names = c("ADBE.log","ADSK.log"),
bty="o",
fill=c("black","red"))
#cointegration test
ADBE_on_ADSK<- data.frame(ADBE.log, ADSK.log)
po.test(ADBE_on_ADSK)
ADSK_on_ADBE<- data.frame(ADSK.log, ADBE.log)
po.test(ADSK_on_ADBE)
# regressions
r1<- lm(ADBE.log~ADSK.log)
summary(r1)
resd1<- resid(r1)
residuals.test<- ADBE.log -r1$coefficients[2]*ADSK.log -r1$coefficients[1]
mean(resd1)
pp.test(residuals.test)
plot(residuals.test)
z1=resd1[-1]
dADBE.log<- diff(ADBE.log)[-1]

```

```

dADSK.log<- diff(ADSK.log)[-1]
ecm1=lm(dADBE.log~dADSK.log +z1 -1)
summary(ecm1)
ecm1
r2<- lm(ADSK.log~ADBE.log)
summary(r2)
resd2<- resid(r2)
residuals.test2<- ADSK.log -r2$coefficients[2]*ADBE.log -r2$coefficients[1]
mean(resd2)
pp.test(residuals.test2)
plot(residuals.test2)
z2=resd2[-1]
ecm2=lm(dADSK.log~dADBE.log +z2 -1)
summary(ecm2)
ecm2
# johansen coint test & VECM
ADBE.ADSK<- data.frame(ADBE.log, ADSK.log)
ADBE.ADSK.vecm1 <- ca.jo(ADBE.ADSK, type = "trace", ecdet ="const")
summary(ADBE.ADSK.vecm1) #r<= 1 HO not rejected
vecm.r1 <- cajorls(ADBE.ADSK.vecm1, r = 1)
vecm.r1
# VAR
varmodel1<- VAR(ADBE.ADSK, p=1, type="const")
summary(varmodel1)
plot(irf(varmodel1, n.ahead=30))
causality(varmodel1, cause="ADBE.Adjusted")
causality(varmodel1, cause="ADSK.Adjusted")
# forecast VAR
ADBE.sample<- ADBE.log["20100101/20190101"]
ADSK.sample<- ADSK.log["20100101/20190101"]
ADBE.test<- ADBE.log["20190101/20200101"]
ADSK.test<- ADSK.log["20190101/20200101"]
ADBE.ADSK.sample<- data.frame(ADBE.sample, ADSK.sample)
ADBE.ADSK.test<- data.frame(ADBE.test, ADSK.test)
nforecast<- nrow(ADBE.test)
varmodel.sample<- VAR(ADBE.ADSK.sample, p=1, type="const")
summary(varmodel.sample)
forecast.values<- predict(varmodel.sample, n.ahead= nforecast)
ADBE.forecast<- as.vector(forecast.values$fcst[1])
ADSK.forecast<- as.vector(forecast.values$fcst[2])
ADBE.comparison<- data.frame(ADBE.test, ADBE.forecast)
ADBE.forecast.xts<- xts(ADBE.comparison[,2], order.by = index(ADBE.test))
plot(ADBE.test, main="ADBE forecast with VAR")
lines(ADBE.forecast.xts, col="red")
ADSK.comparison<- data.frame(ADSK.test, ADSK.forecast)
ADSK.forecast.xts<- xts(ADSK.comparison[,2], order.by = index(ADSK.test))
plot(ADSK.test, main="ADSK forecast with VAR")
lines(ADSK.forecast.xts, col="red")
MAE.ADBE<- mae(ADBE.test,ADBE.forecast.xts)
MSE.ADBE<- mse(ADBE.test,ADBE.forecast.xts)
RMSE.ADBE<- rmse(ADBE.test,ADBE.forecast.xts)
MAPE.ADBE<- mape(ADBE.test,ADBE.forecast.xts)

```

```

dfaccuracy.ADBE<- data.frame(MAE.ADBE,
                             MSE.ADBE,
                             RMSE.ADBE,
                             MAPE.ADBE)
row.names(dfaccuracy.ADBE)<- "ADBE"
names(dfaccuracy.ADBE)<- c("MAE","MSE","RMSE","MAPE")
MAE.ADSK<- mae(ADSK.test,ADSK.forecast.xts)
MSE.ADSK<- mse(ADSK.test,ADSK.forecast.xts)
RMSE.ADSK<- rmse(ADSK.test,ADSK.forecast.xts)
MAPE.ADSK<- mape(ADSK.test,ADSK.forecast.xts)
dfaccuracy.ADSK<- data.frame(MAE.ADSK,
                             MSE.ADSK,
                             RMSE.ADSK,
                             MAPE.ADSK)
row.names(dfaccuracy.ADSK)<- "ADSK"
names(dfaccuracy.ADSK)<- c("MAE","MSE","RMSE","MAPE")
rbind(dfaccuracy.ADBE,dfaccuracy.ADSK)
# comparison with ARIMA process
ADBE.autoarima<- auto.arima(ADBE.sample)
summary(ADBE.autoarima)
pp.test(ADBE.autoarima$residuals)
Box.test(ADBE.autoarima$residuals, type="Ljung-Box")
ADSK.autoarima<- auto.arima(ADSK.sample)
summary(ADSK.autoarima)
pp.test(ADSK.autoarima$residuals)
Box.test(ADSK.autoarima$residuals, type="Ljung-Box")
ADBE.arima.forecast<- forecast(ADBE.autoarima, h=nforecast)
ADBE.arima.forecast.mean<-ADBE.arima.forecast$mean
ADBE.dataframe.arima <- data.frame(ADBE.test, ADBE.arima.forecast.mean)
names(ADBE.dataframe.arima)<- c("ADBE.log", "arima(3,1,2)")
ADBE.arima.forecast.mean.xts<- xts(ADBE.arima.forecast.mean, order.by = index(ADBE.test))

ADBE.arima.MAE<- mae(ADBE.test,ADBE.arima.forecast.mean.xts)
ADBE.arima.MSE<- mse(ADBE.test,ADBE.arima.forecast.mean.xts)
ADBE.arima.RMSE<- rmse(ADBE.test,ADBE.arima.forecast.mean.xts)
ADBE.arima.MAPE<- mape(ADBE.test,ADBE.arima.forecast.mean.xts)
ADBE.arima.accuracy<- data.frame(ADBE.arima.MAE,
                                 ADBE.arima.MSE,
                                 ADBE.arima.RMSE,
                                 ADBE.arima.MAPE)
row.names(ADBE.arima.accuracy)<- "ADBE.arima(3,1,2)"
names(ADBE.arima.accuracy)<- c("MAE","MSE","RMSE","MAPE")
ADSK.arima.forecast<- forecast(ADSK.autoarima, h=nforecast)
ADSK.arima.forecast.mean<-ADSK.arima.forecast$mean
ADSK.dataframe.arima <- data.frame(ADSK.test, ADSK.arima.forecast.mean)
names(ADSK.dataframe.arima)<- c("ADSK.log", "arima(3,1,2)")
ADSK.arima.forecast.mean.xts<- xts(ADSK.arima.forecast.mean, order.by = index(ADSK.test))
ADSK.arima.MAE<- mae(ADSK.test,ADSK.arima.forecast.mean.xts)
ADSK.arima.MSE<- mse(ADSK.test,ADSK.arima.forecast.mean.xts)
ADSK.arima.RMSE<- rmse(ADSK.test,ADSK.arima.forecast.mean.xts)
ADSK.arima.MAPE<- mape(ADSK.test,ADSK.arima.forecast.mean.xts)
ADSK.arima.accuracy<- data.frame(ADSK.arima.MAE,

```

```
        ADSK.arima.MSE,  
        ADSK.arima.RMSE,  
        ADSK.arima.MAPE)  
row.names(ADSK.arima.accuracy)<- "ADSK.arima(0,1,0)"  
names(ADSK.arima.accuracy)<- c("MAE","MSE","RMSE","MAPE")  
plot(ADBE.test)  
lines(ADBE.forecast.xts, col="red")  
lines(ADBE.arima.forecast.mean.xts, col="green")  
plot(ADSK.test)  
lines(ADSK.forecast.xts, col="red")  
lines(ADSK.arima.forecast.mean.xts, col="green")  
rbind(dfaccuracy.ADBE,  
      ADBE.arima.accuracy,  
      dfaccuracy.ADSK,  
      ADSK.arima.accuracy)
```