UNIVERSITY OF LEICESTER

Department of Mathematics

HURST EXPONENTS AND DETRENDING FLUCTUATION ANALYSIS

by

OZLEM OZDEMIR

Student number: 129044313

Supervisor name: Dr Andrew Morozov

Submission date: 13/10/2017

Thesis submitted at the University of Leicester, Department of Mathematics in partial fulfilment of the requirements for the MSc degree in Financial Mathematics and Computation 2016/2017.

Abstract

In this work, we consider the main principles of using MFDFA analysis as the main method for estimation of the Hurst index H for financial time series. In detail, all assumptions relating to this type of analysis are considered. Along with this method of estimation the Hurst index, the S/R analysis and GPH methods of estimation are considered. The model examples showed that the estimation by these three methods are very close. Also, some deficiencies of MFDFA were revealed. Namely the high sensitivity to the deterministic components in the time series.

DECLARATION

All sentences or passages quoted in this project dissertation from other people's work have been specically acknowledged by clear cross referencing to author, work and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name:

Signed:

Date:

Acknowledgements

First, I am thankful to my supervisor Dr Andrew Morozov for his ideas as a starting point of my study and invaluable support. I am grateful to the help and support given to me by the staff at the University of Leicester. I would like to thank my mother, my father and my brother, for the priceless encouragement and support throughout my life. Also, I thank all my friends in Leicester who have shared this incredible journey with me.

Contents

Introduction **1**

1 The Hurst exponent and its analogs for time series **5**
1.1 Hurst index . . . 5
1.2 Detrended fluctuation analysis and multiracial detrended fluctuation analysis . . . 11
1.3 Analysis of the autocorrelation function of the time series and Hurst exponent 18
1.3.1 General classes of time series . . . 18
1.4 Autocorrelation function and 'memory' of time series. GPH estimation . . 21
1.4.1 Estimation of the parameter d and Geweke – Porter – Hudak test . 23

2 Deterministic 'elements' of the time series. Significance of the estimators **25**
2.1 Classical decomposition theorem and deleting non-random components . . 25
2.1.1 Deleting polynomial trend and seasonal component . . . 29
2.1.2 Outliers . . . 31
2.2 Tests . . . 32

3 Example of the calculation of the Hurst index **34**
3.1 Example 1. ANADOLU EFES BIRACILIK LTD . . . 34
3.2 Example 2. Five companies from UK . . . 36
3.3 Construction white noise and Brownian motion by model . . . 37

Conclusions **42**

References **43**

Appendix **47**
Appendix 1. Calculation for Turkish company . . . 47
Appendix 2. Calculation for five UK companies . . . 50
Appendix 3. Function MFDFA1 . . . 51
Appendix 4. Function MFDFA2 . . . 60
Appendix 5. Function DFA . . . 69
Appendix 6. Function GPH . . . 70

Appendix 7. Function PlotMFDFA 72
Appendix 8. Function White Noise and Brownian Motion 73

Introduction

Time series are one of the most used mathematical models for modeling and analysis of real processes in economy, econometrics, engineering, sociology [8, p. vii]. This popularity is due to many factors, the main ones which I consider two: relative ease of use of time series in comparison with other mathematical systems and the ability to qualitatively investigate the quantitative values of the real process - the function of the mean, autocorrelation function. The practicality of using time series is also confirmed by the abundance of works, which describes the basic principles of using time series for real processes (for example, [8, 26, 34]) and articles describing the use of these methods (for example, [3, 5, 24]). To form the main results we use the notation [8].

It should be noted that the study of the presence of long-term memory was tested in many works. For example, in work [16, p. 726] the stock indexes were studied [1] in the period from January 1982. The study is conducted for 200 months. The authors of this work proved that in many countries there is a presence of "long-term" memory, although in some countries the dependence is short-term [2]. Thus, it can be concluded that there is a long-term memory in global economic systems. In Ref. [11] an estimate of the parameter d for the rent of inflation in Spain is considered. The authors had shown that for this economic factor inherent is a long-term memory – estimates for the parameter d vary from 0.37 to 0.43, depending on the method chosen to construct the estimate [3].

In work [29] the securities market in Portugal is being considered. This work is interesting since the models in this study have trends. As noted in the conclusion of this work, Hurst's index[4] is more than 0.5. This means that the valuation of the parameter d is greater than 0, which indicates a long-term memory in the securities market in Portugal. Another interesting conclusion is the correlation between the presence of long-term memory and the reaction of

---

[1] Indices of shares are considered for USA, Japan, Germany, United Kingdom, Hong Kong, Taiwan, South Korea, Singapore and Australian.

[2] For example, in the UK and Australia, the dependence is short-line for 3 out of 4 ratings d – Table 2.

[3] In work a new method has been constructed for estimating the parameter d, which has some input parameters. The values of these parameters change the value of the estimate d.

[4] The Hurst index H is related to the exponent d by the relation:

$$H = d + 0.5$$

securities to shocks[5]. As noted, real processes with long memory are less affected by shocks. This is also confirmed by the mathematical conclusions made in Sections 1-4. Another important point of this work is a short description of the dependence of the presence of long memory ((meaning the values of the indicator H or d) and the presence of a trend in the model. As noted [29, p. 82], the assumption of a quadratic trend gives a worse estimate for the time series [6] . Thus, the presence of "long-term" memory is observed in local securities markets - in this case, Portugal.

The paper [7] is devoted to the study of the securities market for 12 countries in Europe[7]. As a result, the estimation of the parameter d for these countries varies from 0.0315 to 0.1201[7, p. 344], which indicates the presence of long-term memory. It should also be noted that the hypothesis testing is held $H_0 : d = 0$ vs $H_1 : d > 0$, as a result of which for each country we obtain a single-valued deviation H_0 [7, p. 344]. Thus, the authors conclude that the securities market in these countries is homogeneous and is better described using the ARFIMA model. Another practical observation can be made from Table 3: this table estimates the parameters of the ARFIMA model and as we see, different countries correspond to very different models[8]. On the basis of the conclusions drawn by the authors and Table 3, we can say that the authors of the work consider the models to be "close" (homogeneous) based on the value of the parameter d.

This work [6] is devoted to the study of the exchange rate for 5 nominal exchange rates against the US dollar [9] As a result of the study, it was found that all exchange rates (with the exception of the British pound) have a long-term memory, as evidenced by the results of the table 1 [6, p. 97]. In this table, all bets are valued by ARFIMA models with $d > 0$, but for the British pound it is impossible to reject the hypothesis $H_0 : d = 0$. Table 4[6, p. 100] analyzes for the first differences of these 5 nominal exchange rates. The results of the table indicate the existence of a long-term memory for all 5 time series. Also, for all time series this dependence is statistically significant – in all cases the hypothesis H_0 is rejected.

In the work [2] the prices for wheat in Pakistan in the period from 1902 to 2005 (annual data, item 1446) are considered. As a result of this study, it is shown that wheat prices in Pakistan are best approximated by ARIMA (1,2,2) model. As alternative models random walk, AR, MA, ARMA models are considered. The choice of the optimal model is carried out according to the information criterion of Akaike, described below.

The paper [30] is devoted to the study of the prices of wheat and rice [10]. According to this

[5] This indicates that the market gives a slow respond for the overall shocks.

[6] The Hurst exponent takes higher values because of the quadratic trend in the DFA method so it might not be a suitable choice.

[7] Austria, Denmark, Finland, France, Germany, Greece, Holland, Italy, Norway, Spain, Turkey, and UK. A more detailed description of the data can be found [7, p. 348].

[8] In some models there are no MA part values, in some – AR parts, in some – those and others. Thus, models describing the securities market in different countries can be described using ARFIMA(p,d,0), ARFIMA(0,d,q) and ARFIMA(0,d,0) models.

[9] British pound, Deutsche mark, Swiss frank, French frank and Japanese yen in period from January 1974 to December 1989. Weekly data collected from the Chicago Mercantile Exchange Yearbooks.

[10] Quarterly data are considered in the period from 1983 (1) to 2012 (4) – 120 observations. It should also be noted that not the original data but their logarithms are considered (see page 4).

work, the prices for wheat and rice can be determined using a stochastic linear model [11]. As a result, estimates of this model were obtained, and the presence of structural discontinuities in the data was established (see page 9). Although the work does not directly address the ARFIMA or ARIMA model, but the presence of a random walk β as a term in the model leads to the idea of a long-term dependence in the data, which the authors mention (see page 2).

The authors of [32] propose to investigate the prices of rye in Poland using ARIMA model [12]. The paper noted that the optimal model in this case is ARIMA (1,1,1) model. Thus, in this paper, is proved the existence of a long-term memory in time series, which describes the logarithm of the price of rye in the Poles.

The main task of this work, as noted above, is the detection of *long- memory* in time series based on the Hurst index H or the another indicator, which describe of the *long-memory effect* (for example d from ARFIMA(p,d,q) model). Looking ahead, we note that a long memory presupposes a "strong" relationship between the remote values of the time series, that is, the relationship between X_t and X_{t+h} for large h. The concept of strong dependence will be considered in the first chapter of this paper. This chapter will also consider the main classes of time series that we will use for research. In the same section, examples will be given that show the difference between the numeric characteristics for different classes of time series. Also in this section we consider the basic notation for a time series; the concept MA (∞) of the process is considered, by means of which it is possible to calculate the value of the autocorrelation function (ACF) quite simply; the concept of stationarity of the time series, which is necessary for the study of the ARFIMA model, is also considered. Two examples are given that allow one to understand the difference between the autocorrelation function for AR and ARFIMA processes.

The paper considers the main methods for evaluating the Hurst index, including DFA, MFDFA and GPH estimations. The first chapter relates to different techniques for estimating the Hurst index H. The second chapter examines tests by which one can check the stationary (or non-stationary) time series. Thus we confirm the estimates of the parameter H with results of these tests. In the third section, examples of Hurst index H estimation using different techniques are considered. We managed to show in the examples under consideration that all the methods discussed in the work give approximately the same results. It was also found that the presence of a seasonal component increases the value of H by about 10%.

To compare the obtained estimates of the Hurst index H with the of the MFDFA analysis, we will use the GPH estimates [15]. We note that there is a large amount of estimation of the Hurst index H, each of which is based on some properties of the time series X_t. For example, in [19] authors consider wavelet transform modulus maxima (WTMM) estimates are based on a wavelet analysis. This method (WTMM) is based on the fractal transformation of the time series itself. Consequently, in general this method uses other assumptions characteristics of the time

[11] This model has the form

$$y_t = \mu_t + \gamma_t + \psi_t + \sum_{j=1}^{h} d_{j,t} + \varepsilon_t,$$

where the trend μ_t has the representation $\mu_t = \mu_{t-1} + \beta_t + \zeta_t$, β_t – ARIMA process, ε_t, ζ_t – white noise.

[12] Weekly data are considered in the period from August 2010 to October 2013. We consider the logarithms of prices (Page 6)

series. The results of the estimation of the WTMM method is the dependence on the wavelet functio $\psi(x)$. Changing this base function

$$\psi(x)$$

can change the estimate of parameter H. We have chosen the GPH method, since it is based on the construction of the spectral density of the time series. This approach to estimating the parameter H is radically different from the approach proposed in MFDFA or WTMM methods.

Chapter 1

The Hurst exponent and its analogs for time series

In this chapter we consider the basic methods for estimation of the Hurst index H. It should be noted that there are a large number of works devoted to the estimation of this parameter. At present, the most common methods in this time are multifractal detrended fluctuation analysis (MFDFA) and Geweke and Porter-Hudak (GPH) methods. These methods are very different in their essence, as they are based on different characteristics of the time series: MFDFA is based on the fluctuation function $F(l;q)$, which constructed by summation of the time series X_t; GPH method is based on the spectral density $f(\lambda)$ of the time series. In the result, as we will see in Chapter 3, these methods give different estimates of the Hurst index H, but the conclusions about the availability of long-term memory in the time series using these two methods are identical. It should also be noted that another method, which was first applied to the estimation of the parameter H, was a R/S method, which is based on the analysis of the range of the time series.

1.1 Hurst index

Let's briefly consider the history of the problem of assessing the availability of long-term memory in time series (estimation of parameter H).

Standard Gaussian statistics work on the basis of several basic assumptions. The main of these assumptions is the assumption constructed on the central limit theorem (CLT). The central limit theorem indicates that when the number of trials increases, the limiting distribution of the random system will have a normal distribution. [1]. In the CLT events must be independent and identically distributed (that is, they should not affect each other and should have the same probability of occurrence). In the study of large complex systems, it is usually assumed that the system is normal, so it is possible to apply a standard statistical analysis. Note that this

[1]The standard normal distribution define by the density:

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

assumption greatly simplifies all subsequent calculations of the characteristics of the system, that is, a very important theoretical moment.

In practice, often the systems studied (from sunspots, average annual precipitation values to financial markets, time series of economic indicators) are not normally distributed or close to it. To analyze such systems Harold Edwin Hurst [2] [17] proposed the method of normalized range (RS analysis), the key parameter of which is the *Hurst exponent* (or *Hurst index*) H. The presence of such dependencies is very well tracked precisely in the time series, which is constructed with the help of stochastic difference equations.

This is the measure that is used in the analysis of time series: the greater the delay between two identical pairs of values in the time series, the lower the Hurst coefficient. As will be shown in this section, in the study of time series, it is more often resorted to an estimate not of the Hurst index H itself, but to some of its analogues[3]

$$d = H - 0.5,$$

which is calculated on the basis of the autocorrelation function of the time series. It should also be noted that the asymptotic behavior of the correlation function $\rho(h)$ allows us to conclude that the Hurst index H.

The basis of the statistical model of Hurst was the work of Albert Einstein [13] on Brownian motion, which is essentially a model of random walks of a particle. The essence of this theory is that the distance $R(T)$, which passes through the particle during the time T, increases in proportion to the square root of the time T, that is,

$$R(T) \approx T^{0.5}.$$

It is clear that this formula needs to be understood only formally, since it is not clear how to understand $\approx$. We rephrase the formula: the range of variation, $R(T)$, with a large number of tests is equal to the root of the number of trials, T. This is the formula Hurst took as the basis for proving that the Nile floods are not an accident.

To form his method, Hurst used the time series $x_1, ..., x_n$ of the river spill values. To evaluate the parameter H, the following algorithm is used, named in the following *method of the normalized span* or *R/S-analysis*:

1. Calculation of the mean $\overline{x} = \frac{1}{T}\sum_{i=1}^{T} x_t$ of the series $x_1, ..., x_T$.

[2]G.E. Herst is an outstanding British hydrologist who worked on the dam project on the Nile in Egypt. For the construction it was required to estimate the inflow of water and, accordingly, the need for outflow. It was initially assumed that the inflow of water is a random, stochastic process. However, Hurst studied the records of the Nile floods in nine centuries and found regularities in this process. This was the starting point in the study. It was found that spills more than average were changed by even larger spills. After that, the process changed its direction, and spills on the level less than the average were replaced even less. There are cycles with non-periodic duration.

[3]Index d often use in the AFRIMA(p,d,q) model, which is defined for time series X_t by relation

$$\Phi(L)(1-L)^d X - t = \Theta(L)\varepsilon_t.$$

2. Calculation of the standard deviation of the series, S by standard formula

$$S = \sqrt{\frac{1}{T-1}\sum_{i=1}^{T}(x_t - \overline{x})^2}.$$

3. The normalization of the series, by subtraction from each mean value, $y_i = x_i - \widehat{x}$, where $i = 1, ..., n$.

4. Creating a cumulative time series

$$z_t = \sum_{i=1}^{t} y_i,$$

where $t = 1, ..., T$.

5. Calculation of the magnitude of the cumulative time series or *range* of the time series x_t:

$$R(t) = max(z_1, ..., z_t) - min(z_1, ..., z_t).$$

6. The ratio of the cumulative time series by the standard deviation S are approximation of the Hurst index H:

$$c \cdot T^H \approx \frac{ER(T)}{S}$$

Hurst expanded the Einstein equation and led it to a more general form, resulting in the definition of the Hurst index

$$(R(T)/S(T)) = c \cdot T^H,$$

where c is a constant.

In the general case, the value R/S changes scale with increasing increment of time (horizon) T, according to the value of the degree of dependence, equal to H. Hurst took H for 0.5, if the spill process was accidental. In the course of observations, he found that $H = 0.91$! It turns out that the normalized sweep changes faster than the square root of time, that is, the system travels a greater distance than the probabilistic process. This fact was a prerequisite of the moment when it can be argued that the events of the past have a significant impact on the present and the future.

Based on Hurst's assumptions, we can give a more accurate mathematical definition of the Hurst exponent for the time series X_t.

Definition 1.1. *The Hurst exponent of the time series X_t is the quantity*

$$H = \lim_{t\to\infty} \frac{log(R(t)) - log(S(t))}{log(t)}, \quad (1.1)$$

where

$$R(t) = E\left(\max_{1\le i\le t} X_i - \min_{1\le i\le t} X_i\right), \quad (1.2)$$

$S(t)$ is the standard deviation of the process of time series X at time t.

We note that for a stationary (look formula (1.18)) time series $S(t) = const$, then formula (1.1) can be rewritten as follows:

$$H = \lim_{t \to \infty} \frac{log(R(t))}{log(t)}.$$

Subsequently, a methodology was developed to calculate the Hurst coefficient in application to financial and stock markets. This characteristic includes the normalization of data to a zero mean and a single standard deviation in order to compensate for the inflationary component. In other words, we are again dealing with R/S-analysis.

Let us consider in more detail the interpretation of the Hurst index for time series:

1. If the Hurst index is in the interval between 0.5 and 1 and differs from the expected value by two or more standard deviations, then the process is characterized by long-term memory. In other words, there is *persistence*. This means that, within a certain period of time, the following indicators strongly depend on the previous ones. Illustrative examples of a *persistent time series* are graphs of quotations of the most stable and authoritative companies.

2. The Hurst index, which differs from the expected absolute value by two or more standard deviations and assumes a value from 0 to 0.5, characterizes the *antipersistent* time series. The system changes faster than the random one, that is, it has frequent, but small changes. The *antipersistent process* is the graphs of the second-tier stock quotes.

3. If the Hurst index is 0.5 or its value differs from the expected value by less than two standard deviations, then the process is considered a random walk and the probability of short-term or long-term cyclic dependencies is minimal. In fact, this means that in trade one should not rely on technical analysis, since past values have little effect on the present. Here, the best solution will be a fundamental analysis.

We note that the third class of models is most popular at present because it involves processes with independent increments[4]. For these processes, it's enough to just build a process generator or a compensating operator. For first and second cases ($H \neq 0.5$), the construction of the generator is much more complicated, so there are few works devoted to the limit theorems for memory processes. The second type of random processes includes time series, which are the main object of our study. Note that for the time series, the condition for the independence of increments is not met, which complicates the analysis of non-numerical characteristics of these objects. This means that it's difficult to build a time series generator that contains all the information about it. In addition, time series often have the property of non-stationary (the variance goes to ∞). This fact leads to the finding of new methods for estimating of the parameters of these of random processes.

[4]Process $\xi(t)$ is called *process with independent increments*, if for any $0 = t_0 < t_1 < ... < t_n$ random variables

$$\xi(t_1) - \xi(t_0), \xi(t_2) - \xi(t_1), ..., \xi(t_n) - \xi(t_{n-1})$$

are independent.

Models examples of calculation Hurst exponent

Consider several examples for calculation Hurst exponent for the different systems – discrete and continuous. In this cases we consider only random processes without long-memory effect with Hurst exponent $H = 0.5$.

Model example 1.1. *Random walk. Consider random walk witch describe by standard Brownian motion $B(t)$.* In this case we can describe distribution of the minimum and maximum of the Brownian motion in the interval

$$m(t) = \min_{0 \le s \le t} B(s), M(t) = \max_{0 \le s \le t} B(s).$$

These distributions have next forms

$$p_{m(t)}(x) = \frac{2}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}}, x \le 0,$$

$$p_{M(t)}(x) = \frac{2}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}}, x \ge 0.$$

Using these formulas for the densities, not to hard calculate average values for the $M(t) - m(t)$, which we use for calculation average value of $R(t)$.

$$ER(t) = E(M(t) - m(t)) = \int_0^\infty 2x \frac{2}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}} dx = \frac{2\sqrt{2t}}{\sqrt{\pi}} \int_0^\infty y e^{-\frac{y^2}{2}} dy = \sqrt{\frac{8}{\pi}} \sqrt{t}.$$

Thus, the Brownian motion corresponds to the Hurst exponent $H = 0.5$. This is not surprising, since the very design of the Brownian process is based on an experiment without memory – the motion of a particle in a homogeneous environment. More detailed information on the construction of "fractal" (or "fractional") Brownian processes and the corresponding differential equations can be found in [22]. In this monograph detail describes the properties of fractional Wiener processes at different values of $H \in (0, 1)$. Some properties that are not typical for the standard Brownian process ($H = 0.5$) are described for the processes with long-memory $H > 0.5$.

Model example 1.2. *In this example we consider discrete white noise which is describe in the next way*

$$X_t = \xi_t, t = 1, 2, 3, ...$$

where ξ_t – independent random variables $N(0, 1)$.

In the work W. Feller [14] author consider asymptotical distribution of the value of the range $R(t)$ of the discrete process X_t. As a result, it was shown that the asymptotic behavior of the average value of the range $R(t)$ is follows

$$ER(t) = \sqrt{\frac{8}{\pi}} \sqrt{t}, t \to \infty.$$

Thus, this average for the range $R(t)$ of the random walks in the discrete case from this model example coincides with the mean for the spread for the continuous Wiener process from the previous example. This is understandable, since the standard Wiener process is a process with independent increments, that is, the values of the Wiener process can be represent in the following way

$$B_t = \left(B_{t_n} - B_{t_{n-1}}\right) + \left(B_{t_{n-1}} - B_{t_{n-2}}\right) + ... + \left(B_{t_1} - B_{t_0}\right),$$

where

$$0 = t_0 < t_1 < ... < t_{n-1} < t_n = t.$$

In this case increments $B_{t_i} - B_{t_{i-1}}, i = 1, 2, ..., n$ by condition $t_i - t_{i-1} = \frac{t}{n}$ can be rewritten in the term of the model example 1.2.

It should be noted that the use of R/S analysis requires the calculation of the range $R(t)$ and the variation $S(t)$ for very large values of the time t. For example, in the work [14], the author says that a reliable estimate of the Hurst exponent H can be find for large number of the observation $t > 10^3$. Thus, in the case of real time series estimation, for a fairly accurate estimate, at least 1000 values are needed. This assumption in turn does not allow the use of R/S analysis for time series with a small number of observations. That is, the R/S analysis is effective only in cases of a large number of observations T, in the case of a small number of observations, T estimate

$$\widehat{H}(t) = \frac{log(R(t)) - log(S(t))}{log(t)}$$

can be bias and far from the true value of the Hurst exponent H.

Another drawback of R/S analysis is the impossibility of calculating the mathematical expectation of the range of the $R(t)$. As we see, even in the case of a discrete random walk [14], it is impossible to accurately calculate $ER(t)$. Therefore, for more complex systems (for example, ARMA or GARCH models), the calculation of $ER(t)$ is impossible. In this case, we can use methods to approximate the numerical characteristics of a random process (for example, the Monte Carlo method), but as a result, an error due to the low value of T is added to the error of this method. Thus, the use of R/S analysis is rather cumbersome and gives a very large error if we do not evaluate the time series model itself. Therefore, estimating the parameter H using this method depends on the time series model itself. However, since the model is unknown, the estimation of the parameter H will depend on the model's estimation. These facts indicate the high error of this method.

As it was noted, the main work related to the estimation of the parameter P take their beginnings in the work of Einstein. Another approach to this problem was developed by the russian scientist A. Kolmogorov. In this way , the Hurst index H in general form for any processes (discrete and continuous) introduced by A.N. Kolmogorov first time. In [18] he first introduced a continuous Gaussian process with stationary increments and a self-similarity property. This means that for any $a > 0$ there exists a $b > 0$ such that

$$X(at) \approx^d bX(t).$$

It turns out that processes with zero mean have a special correlation function [22, p. VI]:

$$EX(t)X(s) = \frac{1}{2}\left(|s|^{2H} + |t|^{2H} - |t-s|^{2H}\right) \tag{1.3}$$

where $0 < H < 1$. Kolmogorov called this process as 'Wiener spiral'. And only then, after the publication of the article G.E Hurst [17], the H parameter was named the 'Hurst index'. At present, the theory of these processes with covariance function (1.3) is very well developed, which makes it possible to apply it to the analysis of time series, including the estimation of the Hurst index H.

The stochastic calculation of the processes of the aforementioned form comes from B. B. Mandelbrot and D.U. van Ness [20], where the integral moving average representation X is considered through the Wiener process on an infinite interval, called the 'fractional Brownian process'. Intensive study of the properties of the fractional Brownian process took place in the 1990s, with the onset of formalization through the fractional Brownian process of long-term processes such as finance, climatic and weather changes. These authors, like many others, have pointed out that a large number of time series of possessions have 'long-term memory' ($H > 0.5$). This applies primarily to large and 'stable' companies. In this case, when estimation of the parameter of the model of processes should take into account this fact, which leads to the appearance of Autoregressive fractionally integrated moving average (ARFIMA) models.

It turns out, as one would expect from the foregoing, that the fractional Brownian process possesses a long memory only when $H \in (1/2, 1)$. For $H \in (0, 1/2)$ it is a process with a short memory [22, p. VIII]. In this work, the properties of the fractal Wiener process and the methods for estimation of the parameter H for continuous random processes are considered. One of the main characteristics of this work is the proof of the limit theorems for fractal Brownian motion. Since time series can be considered a discrete analogue of fractal Brownian motion, then analyzes of the estimation of the parameter H have many common points for these two types of the random processes (discrete time series and continuous fractal Brownian motion).

1.2 Detrended fluctuation analysis and multiracial detrended fluctuation analysis

As we have noted before, assessments using the R/S analysis are rather rude and may lead to unreasonable conclusions. Many authors attempted to improve the estimation of the parameter H, which resulted in many methods for estimating the parameter H based on various characteristics of the time series.

One of the most commonly used methods for estimation Hurst exponent at this time is detrended fluctuation analysis (DFA). The reason for its popularity is its ease of using of this method. We describe the algorithm for using DFA, which is described in more detail in the work [27]. As we see below, this method does not contain any additional assumptions on numerical or other characteristics of the time series. This, on the one hand, makes it versatile when used for an arbitrary time series; on the other hand, as we will see below, this method is unstable to the presence of non-random components in time series (for example, the seasonal component). DFA method is based on the concept of the stochastic self-similarity of a random process, which was firs considered in the [18]:

Definition 1.2. *Random process X_t called stochastic self-similarity with index a, if for any*

$t_i \in R^1, i = 1, ..., n$ *and* $k > 0$

$$(X_{t_1}, X_{t_2}, ..., X_{t_n}) \sim^d k^{-a}(X_{kt_1}, X_{kt_2}, ..., X_{kt_n}), \tag{1.4}$$

where by $\sim^d$ *we define equality of the distribution of the corresponding random vectors.* Note that for the Brownian motion as well as random walk from the model example 1, have coefficient $\alpha = 0.5$. This means that these processes do not have long-term memory.

Consider now in details algorithm of the DFA analysis for the random processes [27]. It should be noted that some of the considerations of the DFA analysis are quite close to the R/S analysis, but the idea of estimation of the parameter H is different.

1. In this step, as in R/S analysis we calculate of the *single summation* process for the time series X_t – calculate the cumulative sum process which define in the next way

$$Y_t = \sum_{i=1}^{t} X_i - \overline{X},$$

   where $\overline{X}$ – average value if the time series. The idea of using of the process Y is sufficiently simple – the higher the value of H, the time series X_t at time e is more dependent on its history. This means that with increasing H it becomes more 'deterministic' (in the case of $H = 1$, the trajectories of the process of the process will be lines). So in the case $H = 1$ trajectory of the process Y_t are $kt(t+1)$, where k – some coefficient. This observation helps to determine the parameter H;

2. Separate interval of the time $t = 1, ..., T$ as $\frac{T}{l}$ equal (or approximal equal between them) subintervals. For each of these intervals, we compute a local polynomial trend of degree p. In the paper [27], as in many other works, it is assumed that the degree of the polynomial trend is 1, that is, a linear trend is considered. We denote this trend on the j-th interval by Y_t^j. As will be shown below, the presence of deterministic components can change the estimation of the model parameters, Including the estimation of the parameter H. Therefore, at this step, the polynomial trend for the time series is gradually removed. It should also be noted which does not remove the common trend from the time series over the entire study interval. This technique allows you to remove all deterministic component of the polynomial type and thus obtain only randomness in each interval. It should be noted that the disadvantage of this method is the impossibility of taking into account other types of deterministic components - seasonal components, deterministic factors, etc. In addition, the removal of the deterministic component in the study of time series is used in all work on the study of time series. This is due to the fact that the presence of these components leads to wrong estimates of the parameters of the model.

3. In this step, we calculate the sum of randomness from each subinterval, calculated in the previous step. The idea of this method is simple and to some extent reminiscent of the idea of the R/S method. After the trend is removed at each interval, the time series

$$U_t = Y_t - Y_t^j$$

are stationary (this is our assumption). So for this stationary time series $S(t) = const$ and we can not consider term $\log(S(t))$ (1.1). So, we compute on each subinterval the value of the variation of the random process Y_t with respect to the trend on a given interval and summarize these variations. As a result, we get the value

$$F(l) = \sqrt{\sum_{i=1}^{\frac{T}{l}} \frac{1}{n_i} \sum_{t=T_i} (Y_t - Y_t^j)^2}, \tag{1.5}$$

where n_i is the number of observations on the i -th interval, T_i is the set that contains the time values t from the i -th time interval. The function (1.5) is called a *fluctuation* or *fluctuation function*. Notice, that for different values of the degree of the polynomials Y_t^j we get different values of the fluctuation function $F(l)$. In the figure 1.1 we can see 3 types of trends – linear, quadratic and cubic.

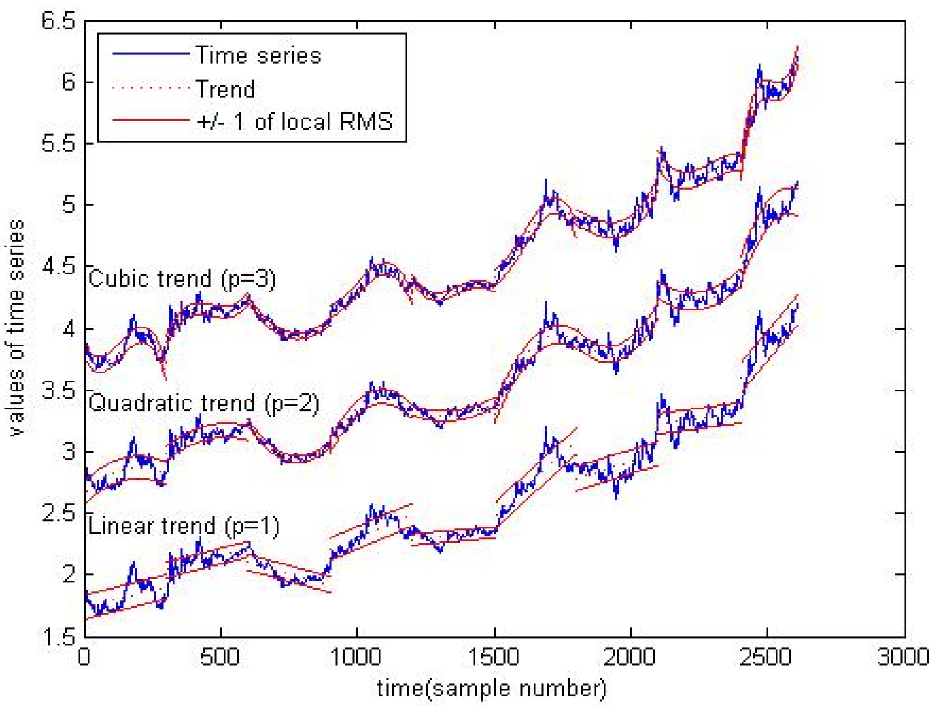

Figure 1.1: Three types of the trends for different degrees $p = 1, 2, 3$.

As we can see from this picture, different degrees of polynomial Y_t^j generate different fluctuation functions F. However, as we have established, the difference degrees of the Y_t^j ($m = 1, 2, 3$ is not significant for the estimation of the Hurst index H. Therefore, we will only consider the linear trend ($m = 1$), described in the bottom of Figure 1.1.

The final step of the DFA analysis is to estimation the parameter H based on the fluctuation function F. As shown in [14], the fluctuation function for random walks satisfies the relation

$F(l) \propto l^{0.5}$. In this way, the discrete random process of 'without memory' have Hurst index 0.5. It should also be noted that the coefficient of self-similarity is also equal to 0.5 too. Using these two facts, we can assume (our second assumption) that for random 'long memory' processes the estimation of Hurst index H is other. Using previous opinions, we get that stochastic self-similarity of the process we can calculate by relation

$$F(l) \propto l^{\alpha}, \tag{1.6}$$

where α determines the long-term memory of the process, that is, it determines the value of the Hurst exponent H. For a stationary time series, the value of the Hurst exponent H and the exponent of the growth of the fluctuation function are equal to:

$$H = \alpha. \tag{1.7}$$

Let us consider an example of estimation the fluctuation function values for the share price of the Turkish company ANADOLU EFES BIRACILIK LTD. To calculate the value of the coefficient α we use the method of least squares, as a result of which we obtain the estimate

$$\widehat{\alpha} = 1.47947436604495.$$

Note that in this case we have obtained the estimate $\widehat{\alpha} > 1$, which corresponds to the non-stationary time series. A similar result for this time sries with respect to non-stationarity is obtained on the basis of the augmented Dickey–Fuller (ADF) and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) tests. It should also be noted that in this case, the Hurst index is very close to 0.5 [5], therefore, to test the difference of 0.5, additional research is needed, namely to consider the hypothesis:

$$H_0 : \widehat{H} = 0.5 \text{ vs } \widehat{H} \neq 0.5.$$

Consider now the expansion of the detrended fluctuation analysis – multifractal detrended fluctuation analysis (MFDFA or MF-DFA). When using MFDFA analysis, then calculation if the formula (1.5) change as follow

$$F(l;q) = \sqrt[q]{\sum_{i=1}^{\frac{T}{l}} \left(\frac{1}{n_i} \sum_{t=T_i} (Y_t - Y_t^j)^2 \right)^{\frac{q}{2}}}, q \neq 0 \tag{1.8}$$

where q – parameter of the MFDFA. As we can see, for $q = 2$ we get classical DFA:

$$F(l) = F(l;2).$$

[5]For non-stationary time series, calculation of the parameter H have next

$$H = \alpha - 1.$$

Notice in the case $q = 0$ many author consider next value of the fluctuation function

$$F(l;0) = \lim_{q \to 0} F(l;0) = \exp\left\{ \sum_{i=1}^{\frac{T}{l}} log\left(\frac{1}{n_i} \sum_{t=T_i} (Y_t - Y_t^j)^2 \right) \right\}. \tag{1.9}$$

Next step of the estimation of the Hurst exponent $\widehat{H(q)}$ are estimation of the of the relation

$$F(l;q) \propto l^\alpha$$

or

$$\log_2 F(l;q) \propto \alpha \log_2 l.$$

The advantages of the MFDFA analysis are that at the same time we will break the estimate of the parameter H for many values of q at the same time. This mean, that we get not one estimation, but function $H(q)$. Thus, this approach allows us to build distributions of the $H(q)$ and different dependencies for this function.

So, using least squared estimation for this relation, we get estimation of the $H(q)$. In the figure (1.2) we can see three different sequences of the values $\log_2 F(l;q)$ for

$$l = [16, 32, 64, 128, 256, 512, 1024]$$

and linear regressions

$$\log_2 F(l;q) = a_q + \widehat{H(q)} \log_2 l,$$

where a_q – some constant in the linear regression.

In the next figure (1.2) we can see estimation of the fluctuation function for different values of the q. We build fluctuation for $q = (-0.5, 0, 05)$. As we see, for this values of the q we get different values of the Hurst index $H(q)$:

$$(1.7589; 1.5585; 1.3533).$$

For this values we can conclude, that estimation of the Hurst index $H(q)$ are near 1.5. This mean, than for share prise of the ANADOLU EFES BIRACILIK LTD are have 'long' memory ('long' memory effect). Unfortunately, in this case, we can only talk about the presence of long-term memory in the time series ($H > 0.5$). However, this information does not allow us to determine the optimal model for this time series. In the following sections, we define the time series models that we will use to forecasting of the time series.

Given the presence of the function $H(q)$, the question arises about the choice of the optimal (single) value of $H(q_0)$, which we will consider as an estimate of the parameter H. In the given case, the authors of the work [19] propose the use of multifractal density – the dependence of functions $H(q)$ and *multyfractal dimension* $D(q)$. As we can be seen in sub figure 2.2 of Figure 1.2, mode of the this density is 1.58. Similar result we from distribution of the $D(q)$, describe in the Figure 1.3.

Consider now in detail results from the figures 1.2 and 1.3. As we see with increasing of the parameter q from -5 to 5, the value of the estimation $H(q)$ also change. Therefore, the question

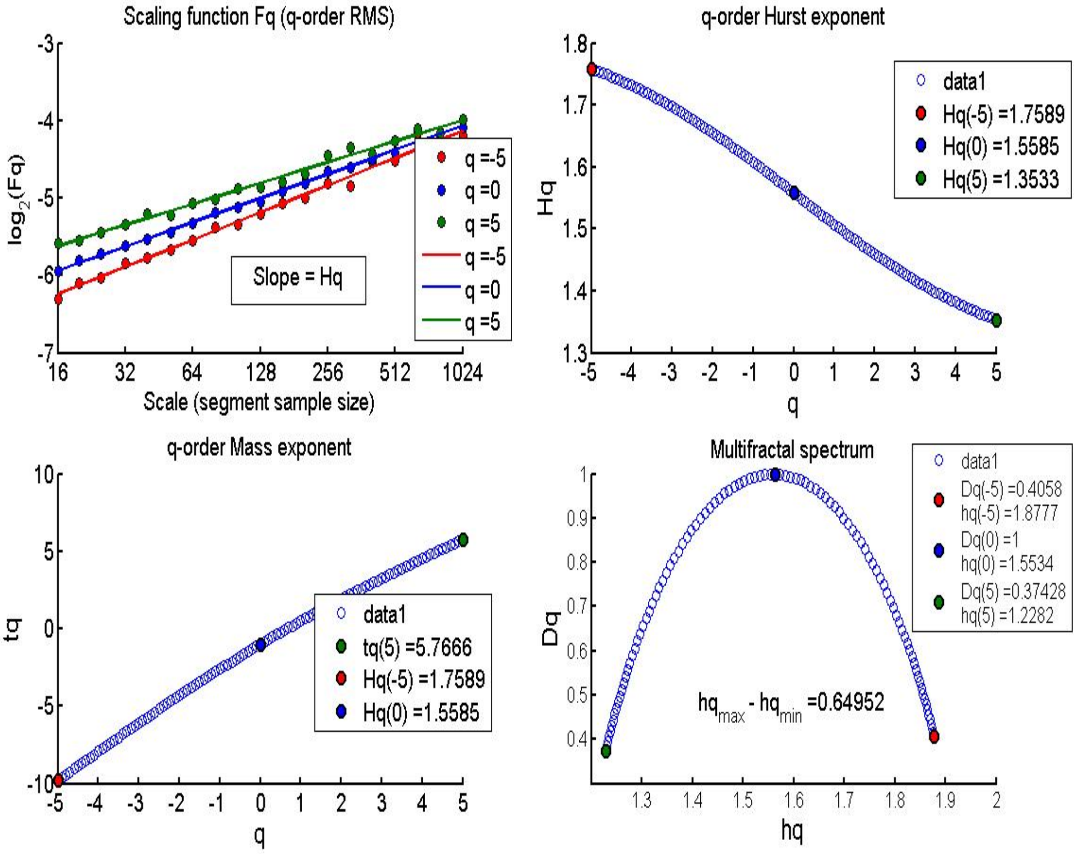

Figure 1.2: 1.1. Plot of fluctuation function $F(l)$ for the window width $l = 16, ..., 1024$ for different values of the parameters $q = [-5, 0, 5]$; 1.2. Estimation of the Hurst index i the interval [-5,5]; 2.1. Plot of the $\tau(q)$ for $q = [-5, 0, 5]$; 2.2. Plot of the $D(q)$.

arises about the dependence of the value of the estimation $\widehat{H(q)}$ on the parameter q. The answer to this question can be found in the work [19]. Authors of this work propose use relation between *scaling exponent* $\tau(q)$ and Hurst index $H(q)$. Notice first, the *scaling exponent* $\tau(q)$ we define from relation

$$Z_q(s) = \sum_{i=1}^{\frac{N}{s}} |p_s(i)|^q \propto s^{\tau(q)}, \tag{1.10}$$

where

$$p_s(i) = \sum_{j=s(i-1)+1}^{si} x_j.$$

Using last relation and definition of the Hurst index $H(q)$, we can conclude correctness next relation:

$$\tau(q) = qH(q) - 1. \tag{1.11}$$

Hence, if in (1.11), Hurst index $H(q)$ does not depend from q, then this relation are linear in the space $(q, \tau(q))$. As we see from figure 1.2 plot of the function $\tau(q)$ are almost line. This makes it possible to say that the Hurst index H is independent of the parameter q. Consider also multifractal dimension

$$D(q) = \frac{\tau(q)}{q-1}. \tag{1.12}$$

By our assumption $D(0) = 1$. Analogue result we get from last subfigure of the figure 1.2. So our assumption about independence of $H(q)$ from q are right. Consider the density of the mulifractal dimension $D(q)$, described in Figure 1.2, 1.3. As we see from this picture, this density is centered around point 1.58 (mode of this disribution). This conclusion also supplements the relation between $D(q)$ and $tau(q)$. Thus, the analysis shows that this time series is near long-term memories and the estimation of the parameter $H = 1.58$. In addition, according to the definition of the density of the mulifractal dimension $D(q)$, we can construct the confidence intervals for the parameter q – as result and for estimation of the Hurst exponent $\widehat{H}$.

From the next plot 1.3 we can see distribution of the $H(q)$ for $q \in [-5, 5]$. As we see this distribution have mean about 1.5 and mode about 1.58.

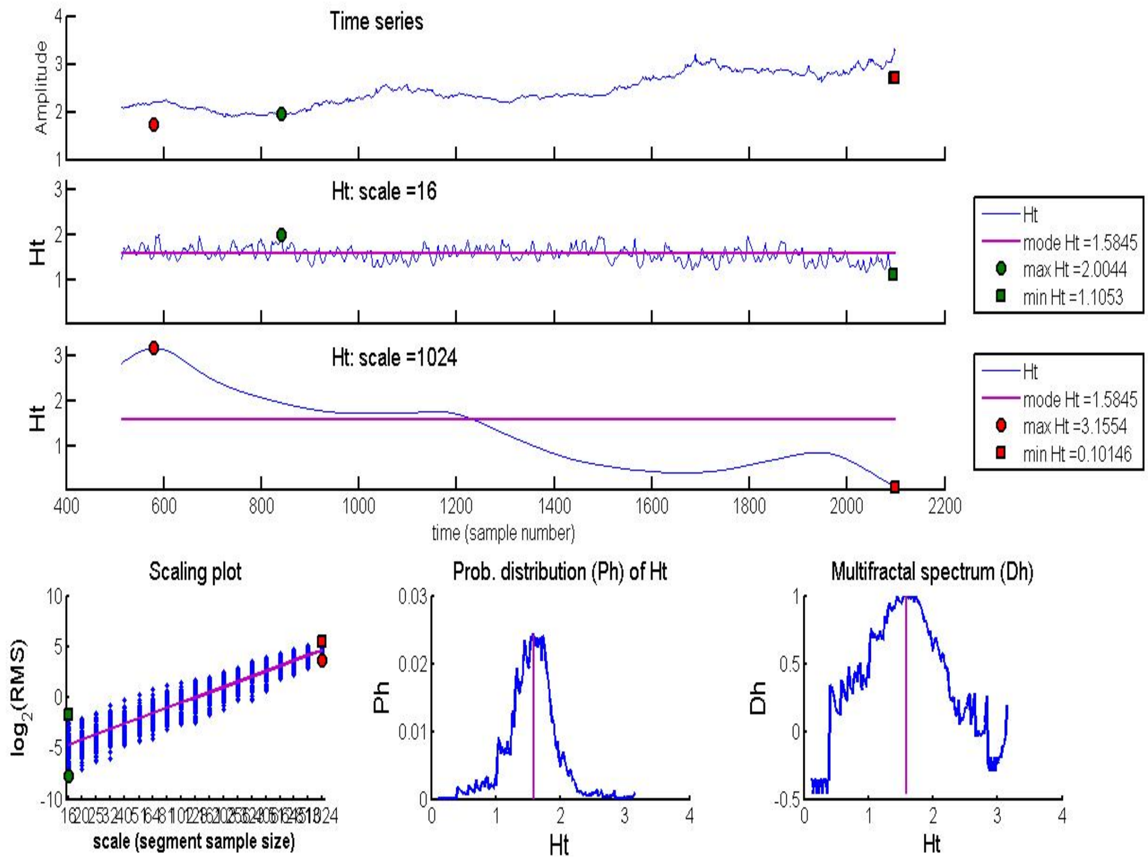

Figure 1.3: Distribution of the $H(q)$ and $D(q)$ in for the $q \in [-5, 5]$.

Also we see from figure 1.3 that average of root-mean-square(RMS) of the residuals $Y_t - Y_t^j$ from formula (1.5) are decreases with increasing width of the window q.

1.3 Analysis of the autocorrelation function of the time series and Hurst exponent

There is a large number of methods for estimating the parameter H. As we have seen from the previous point, the DFA (MFDFA) methods does not depend on the time series model itself. There are also methods in which the estimation of the parameter H depends on the time series model itself. Therefore, we will describe in this section the main models of time series that will be used by us in our study.

1.3.1 General classes of time series

As noted above, the main task of this paper is to consider time series models that most adequately describe some discrete random process describing the dynamics of stock price of the some financial company. To introduce the basic definitions, we will use the terminology of the book [8] and add some information from other sources, if necessary as needed. In our work, we will use linear discrete time series models [8, pp. 73, 158, 207]: ARMA, ARIMA and ARFIMA. Also consider the SARIMA model, which, as will be shown below, can not be used for our research. The data for the time series $x_t, t \geq 1$ can be described using the formula

$$(1-L)^d \Phi(L) X_t = \Theta(L) \varepsilon_t, \tag{1.13}$$

where L – lag operator which define by next formula

$$LX_t = X_{t-1};$$

$(1-L)^d$ – operator which describe fractional integration[6] of the process (1.13); Φ, Θ – polynomials which describe of the AR and MA parts of the process x_t (1.13) and these polynomials are next representation

$$\Phi(L) = 1 - \phi_1 L - \phi_2 L^2 - ... - \phi_p L^p, p \in \mathbb{N};$$

$$\Theta(L) = 1 + \theta_1 L + \theta_2 L^2 + ... + \theta_q L^q, q \in \mathbb{N};$$

$d \in \mathbb{R}$ – a quantity indicating how we will see on the page 22, the presence or absence of long-memory for the process. Concerning the random sizes $\varepsilon_n, n = 0, 1, 2, ...$, we assume the following

$$E\varepsilon_t = 0, Var\varepsilon_t = \sigma^2, cov(\varepsilon_n, \varepsilon_m) = 0, n \neq m, \tag{1.14}$$

that is, the random process ε_t is a white noise [8, p. 14].

Depending on the values of p, d, q, the time series (1.13) can be separated into the following subclasses[7]

[6] In essence, the study of this operator is the most important step in the study of the present of long-term memory in the time series.

[7] Note that these classes intersect and some of them generalize the others.

1. autoregressive process AR(p) with values $d = q = 0$;
2. moving-average process MA(q) with values $p = d = 0$;
3. autoregressive–moving-average process ARMA(p,q) with value $d = 0$;
4. autoregressive integrated moving average process ARIMA(p,d,q) with value $d \in \mathbb{N}$;
5. autoregressive fractionally integrated moving average process ARFIMA(p,d,q) with values $d \in \left(-\frac{1}{2}, \frac{1}{2}\right)$[8].

As we see, there is some nesting between the time series classes that we listed. For example, AR and MA processes are special cases of ARMA and ARIMA processes. Although these classes intersect, as we'll see below, some of the qualitative characteristics of these classes are very different. One of these characteristics is the autocorrelation function [8, p. 13], which is defined for time series (1.13) as follows[9]:

$$\rho(h) = \frac{E(X_t - EX_t)(X_{t+h} - EX_{t+h})}{\sqrt{Var(X_t)Var(X_{t+h})}}. \tag{1.15}$$

To calculate the autocorrelation function, a representation (if possible) of the process (1.13) in the form MA(∞) of the process [8, p. 75].

$$X_t = \sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i}. \tag{1.16}$$

ARMA (p, q) a process that admits a representation (1.16) is called a causal. Using the representation (1.16) and the property (1.14), it is easy to find the representation of the autocorrelation function of the process X_t [10]

$$\rho(h) = \frac{\sum\limits_{i=0}^{\infty} \psi_i \psi_{i+|h|}}{\sum\limits_{i=0}^{\infty} \psi_i^2}. \tag{1.17}$$

---

[8]In some works, for example [34, p.43], it is sometimes assumed that the parameter $d < \frac{1}{2}, d \neq 0, -1, -2...$

[9]In the general case, the autocorrelation function depends on the airs of the arguments - n, h, and is entered as follows:

$$\rho(t, h) = \frac{E(X_t - EX_t)(X_{t+h} - EX_{t+h})}{\sqrt{(}Var(X_t)Var(X_{t+h}))}.$$

But since the random process (1.13) is homogeneous, $\rho(t, t+h)$ depends only on the difference $t+h-t = h$.

[10]For the process X_{t+h} according to (1.16) we have

$$X_{t+h} = \sum_{i=0}^{\infty} \psi_i \varepsilon_{t+h-i}.$$

Using properties of the white noise ε_n (1.14) for $h \geq 0$ we get

$$\gamma(h) = E(X_t X_{t+h}) = E\left(\sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i} \sum_{j=0}^{\infty} \psi_j \varepsilon_{t+h-j}\right) = \sum_{t-i=t+h-j} \psi_i \psi_j E\varepsilon_{t-i}^2 +$$

Let's consider several examples of calculation of autocorrelation function. In example 1.3 we consider time series with short memory – ARMA(1,1) process with $d = 0$ (or $H = 0.5$). In the example 1.4 we consider ARFIMA(0,d,0) with $d > 0$ (or $H > 0.5$). As result, autocorrelation functions of these processes have difference limits properties.

Model example 1.3. *As an example of a time series with a short-term memory, consider the ARMA(1,1) process*

$$X_t - \phi X_{t-1} = \varepsilon_t + \theta\varepsilon_t, |\phi| < 1.$$

This process can be represented in the form of a MA (∞) process using the following decomposition[11]

$$X_t = \Phi^{-1}(L)\Theta(L)\varepsilon_t = \frac{1+\theta L}{1-\phi L}\varepsilon_t = (1+\theta L)\sum_{i=0}^{\infty}\phi^i L^i \varepsilon_t =$$

$$= \left[1 + \sum_{i=1}^{\infty} L^i(\phi^i + \theta\phi^{i-1})\right]\varepsilon_t = \varepsilon_t + \sum_{i=1}^{\infty}(\phi^i + \theta\phi^{i-1})\varepsilon_{t-i},$$

thus,

$$\psi_0 = 1, \psi_i = \phi^i + \theta\phi^{i-1}, i > 0.$$

Thus, using (1.17), we obtain representations for the autocorrelation function of the ARMA(1,1) process:

$$\rho(h) \approx \frac{\sum_{i=0}^{\infty}(\phi^i + \theta\phi^{i-1})(\phi^{i+|h|} + \theta\phi^{i+|h|-1})}{\sum_{i=0}^{\infty}(\phi^i + \theta\phi^{i-1})^2} =$$

$$= \frac{\sum_{i=0}^{\infty}\phi^{2i-2+|h|}(\phi+\theta)^2}{\sum_{i=0}^{\infty}\phi^{2i-2}(\phi+\theta)^2} = \frac{\sum_{i=0}^{\infty}\phi^{2i-2+|h|}}{\sum_{i=0}^{\infty}\phi^{2i-2}} = \phi^{|h|}.$$

Model example 1.4. *Consider now case of the ARFIMA*$(0, d, 0)$ *time series*

$$(1-L)^d X_t = \varepsilon_t, d \in (0, 0.5).$$

Note that in this case the process has a long-term memory, that is $H > 0.5$ by definition.

$$+\sum_{t-i\neq t+h-j}\psi_i\psi_j E(\varepsilon_{t-i}\varepsilon_{t+h-j}) = \sum_{i+h=j}\psi_i\psi_{i+h}\sigma^2 + 0.$$

Using the pairwise function $\gamma(h)$ [8, p.41], we obtain the equality

$$\gamma(h) = \sum_{i=0}^{\infty}\psi_i\psi_{i+|h|}\sigma^2.$$

Then for ACF we obtain the formula (1.17).

[11] In this calculation, we use the fact that the norm of the lag operator L is equal 1, thus $\|L\| = 1$.

We use the decomposition of the fraction $(1-L)^{-d}$ in a series [8, p. 246], we obtain

$$X_t = (1-L)^{-d}\varepsilon_t = \left(1 + \sum_{i=1}^{\infty} \frac{d(1-d)...(i-1-d)}{i!} L^i\right)\varepsilon_t =$$

$$= \varepsilon_t + \sum_{i=1}^{\infty} \frac{d(1-d)...(i-1-d)}{i!}\varepsilon_{t-i}.$$

Thus, we have a representation for the coefficients ψ_i from (1.17) is next

$$\psi_i = \frac{d(1-d)...(i-1-d)}{i!} = \frac{\Gamma(i+d)}{\Gamma(i+1)\Gamma(d)},$$

where Γ – gamma function[12]. Using this representation for the coefficients, we obtain

$$\rho(h) = \frac{\sum_{i=0}^{\infty}\psi_i\psi_{i+|h|}}{\sum_{i=0}^{\infty}\psi_i^2} = \frac{\sum_{i=0}^{\infty}\frac{(-d-1)...(-d-i)}{i!}\frac{(-d-1)...(-d-i-|h|)}{(i+|h|)!}}{\sum_{i=0}^{\infty}\left(\frac{(-d-1)...(-d-i)}{i!}\right)^2} =$$

$$= \frac{\sum_{i=0}^{\infty}\left[\left(\frac{(-d-1)...(-d-i)}{i!}\right)^2\frac{(-d-i-1)...(-d-i-|h|)}{(i+1)..(i+|h|)}\right]}{\sum_{i=0}^{\infty}\left(\frac{(-d-1)...(-d-i)}{i!}\right)^2}.$$

Using properties of the gamma function [4, p. 7], it can be shown that the relation for the autocorrelation function ARMA (0, d, 0) of the process

$$\rho(h) = O\left(h^{2d-1}\right) = O\left(h^{2H-2}\right), h \to \infty.$$

In examples 1.3 and 1.4, the calculation of the autocorrelation function for ARMA and ARFIMA models is considered. As we see, the asymptotic behavior of the autocorrelation functions for ARMA and ARFIMA models has a significant difference – Fig. 1.4. This difference will be identified with the concept of 'long' memory for ARFIMA models for some values of d. Before introducing the concept of 'long', let's consider one important property of time series – stationarity [8, p. 13], which plays a very important role in the study of time series. This property is important in that it indicates the static (stationary) nature of some characteristics of the time series. We will consider one of the most convenient notions of stationarity – weak stationarity. The time series X is called stationary if

$$EX_t = c = const, cov(X_t, X_{t+h}) = \gamma(h). \tag{1.18}$$

1.4 Autocorrelation function and 'memory' of time series. GPH estimation

In economics, econometrics, finance, the long process memory is a property that describes the probable dependence for the elements of the time series X_t and X_{t+h}. Since long memory implies

[12] Gamma function is define by relation $\Gamma(x) = \int_0^{\infty} t^{x-1}e^{-t}dt$.

the presence of significant autocorrelation between remote observations (that is, for large h), its presence leads to a large influence of time series values at distant instants of time. Mathematically, the presence of long-term memory in the time series x means the following [5, p. 10]:

1. Time series X_t is stationary, that is condition (1.18) are hold;

2. The series

$$\sum_{h=1}^{\infty} |\rho(h)| \tag{1.19}$$

   not convergence.

For the ARMA process, from the example 1.3 for $\|\phi\| < 1$, the process does not have a long-term memory, because the condition (1.19) is not satisfied. In this case

$$\sum_{h=1}^{\infty} |\rho(h)| = \sum_{h=1}^{\infty} |\phi^{|h|}| = \frac{1}{1-|\phi|} < \infty.$$

On the contrary, for the process from the example 1.4 with $d = 0.2$ satisfy second condition of the definition of the long-memory process:

$$\sum_{h=1}^{\infty} |\rho(h)| = \sum_{h=1}^{\infty} |h|^{2d-1} = \infty,$$

that is, this process has a long-term memory. For the stationarity of the process ARMA(0,d,0), the condition $2d - 2 < -1$ or $d < \frac{1}{2}$ is necessary from the example 1.4, since [8, p. 207] the coefficients from the representation (1.16) for this time series satisfy the relation

$$\psi_i \sim i^{d-1}.$$

In the figure 1.4, memory interpretation of ARMA $(1,1)$ processes with $\phi = 0.9, \theta = 0.3$ and ARFIMA $(0, 0.2, 0)$ processes is clearly visible. For example, for $h = 60$, the value of the autocorrelation function is 0.002 and 0.086, respectively. For the ARMA process, the time dependence of $n + 60$ of the same process at time n is approximately 0.2%, while at the same time for the ARFIMA process with Hurst index $H = 0.7$ this value is equal 8.7%. Thus, the ARFIMA model "remembers its own history" longer in comparison with the ARMA process.

Thus, the presence of long-term memory ($H > 0.5$) affects the asymptotic properties of the autocorrelation function $\rho(h)$. This observation, along with the assumptions about the model itself, makes it possible to use the GPH estimate as another method of the parameter H.

It should be noted that all stationary ARMA processes are considered processes with 'short memory' [34, p.134], since the relationship between the values of the X_t and X_{t+h} is very fast decreases with increasing $h \to \infty$ [13].

[13] The dependence of the autocorrelation function is implied

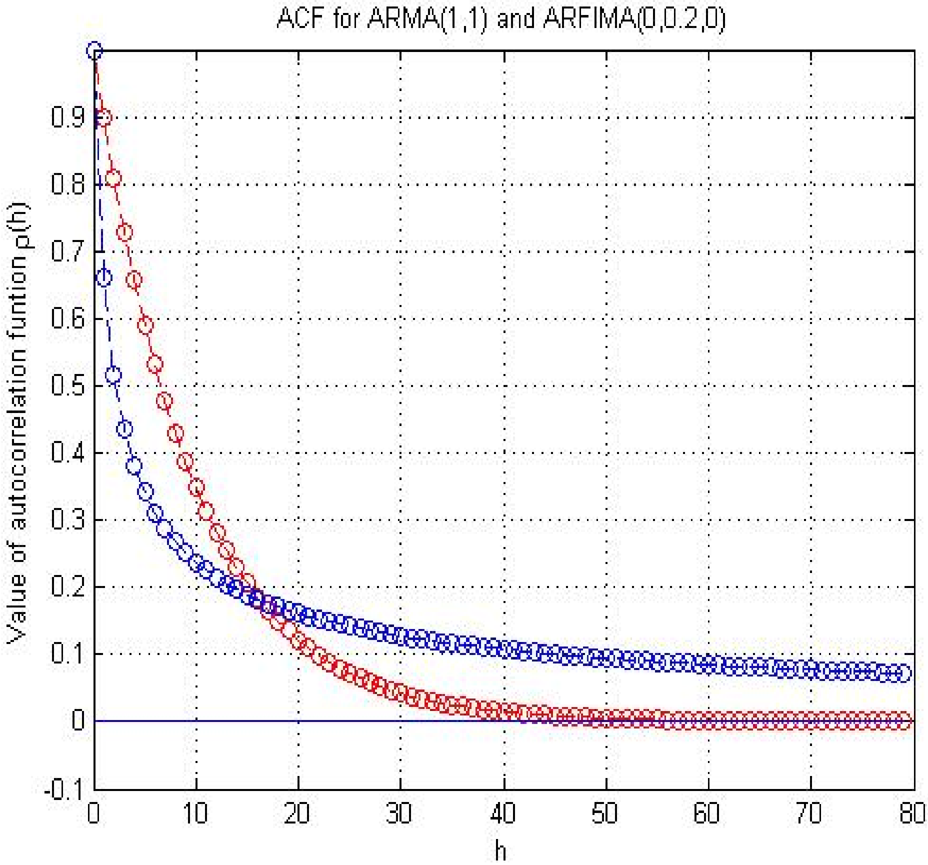

Figure 1.4: Graphic of the autocorrelation functions for ARMA(1,1) process with $\phi_1 = 0.9$, $\theta_1 = 0.3$ and ARFIMA(0,0.2,0) time series.

1.4.1 Estimation of the parameter d and Geweke – Porter – Hudak test

It should be noted that there are many methods for estimating the parameter d or Hurst exponent H. For example, in the work [7, p. 338, 339], we consider two 'classical' estimates of the parameter d by the Geweke and Porter-Hudak method (GPH method) and the Robinson's method. Similar estimates are used in [16]. In the work [11, eq. (2.1), (2.3)] uses Log autocovariance and Minimum distance autocovariance for estimating the parameter d, Matlab, as the main program in our study, uses the GPH method, so consider it in more detail. To this end, consider the two-step estimate of the parameter d [34, p. 82]. First, according to the assumption about the stationarity of the time series, the spectral density [34, p. 4] for the time series x_t can be written in the following form

$$f(\lambda) = f_0(\lambda) \left[2sin \left(\frac{\lambda}{2} \right) \right]^d,$$

where $f_0(\lambda)$ – spectral density of the white noise. Logarithmic this equation and using the ratio for the peridogramm $I(\lambda)$ [34, p.50]

$$I(\lambda) = \frac{1}{2\pi n}\left|\sum_{j=1}^{n} x_j e^{i\lambda_j}\right|^2,$$

where $\lambda_j = \frac{2\pi j}{n}$. In result we get equality

$$log(I(\lambda_j)) = log(f_0(0)) - d * log\left(2sin\left(\frac{\lambda_j}{2}\right)\right)^2 + log\left\{\frac{I(\lambda_j)\left(2sin\left(\frac{\lambda_j}{2}\right)\right)^{2d}}{f_0(0)}\right\}.$$

Using the least squares method for this regression, we obtain the GPH estimate of the parameter d

$$\widehat{d} = \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{m}(x_i - \overline{x})^2}, \tag{1.20}$$

where m – number of values of the λ_j,

$$x_j = log\left(2sin\left(\frac{\lambda_j}{2}\right)\right)^2, y_j = log(I(\lambda_j)).$$

So, using estimation (1.20), we get next estimation of the Hurst index H:

$$\widehat{H} = 0.5 + \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{m}(x_i - \overline{x})^2}.$$

To estimate (1.20), one can prove the asymptotic normality, that is, the convergence to the normal distribution of the normalized parameter $\widehat{d}$ (see [11, p. 83] or [29, p. 79]). According to these statements,

$$d \sim N\left(\widehat{d}, \frac{\pi^2}{24\sqrt{m}}\right). \tag{1.21}$$

Using this statement, it is easy to verify

$$H_0 : d = 0 \text{ vs } H_1 : d > 0.$$

This test is sometimes called the GPH test, because it is based on the normality statement (1.21), proved by Geweke and Porter-Hudak [15]

Chapter 2

Deterministic 'elements' of the time series. Significance of the estimators

In the previous Chapter, we assumed that the time series does not contain any deterministic (non-random) components. In this section we consider the main types of deterministic components. which may affect the estimation of parameters H or d. The basis of this review is the classical decomposition theorem of the time series and the concept of outliers.

Since the estimations of the parameter H depend on the deterministic components, so we specify methods for removing these deterministic components from time series. Note that the MFDFA method does not depend on the presence of the trend in the time series, since according to the MFDFA algorithm, the polynomial trend is removed when calculating fluctuation function $F(l;q)$. On the other hand, the seasonal component can not be deleted using MFDFA method.

In Chapter 3, it will be shown that the presence of a seasonal component changes the value of estimating of the parameter H. Therefore, the deletion of the deterministic components from time series is important step in the the accuracy of the estimation of the Hurst index H.

2.1 Classical decomposition theorem and deleting non-random components

In the previous section we considered the basic methods for estimating the Hurst index, which is not subject to deterministic components. But it often happens that the data considered in the investigation do not satisfy the representation (1.13) in advance, because some deterministic components are taken into account. The most important components are a polynomial trend and a seasonal component. According to the classical decomposition [8, p. 20] of the time series 1.13 there is a representation

$$X_t = p_t + s_t + y_t, \tag{2.1}$$

where s – seasonal component with period d ($s_t = s_{t+d}$); p_n – polynomial trend of the degree k

$$p_t = a_0 + a_1 t + ...a_k t^k; \tag{2.2}$$

y_t – random component with zero expectation. The importance of the decomposition components (2.1) in the following

1. the deterministic components of the time series p_t, s_t distort the estimates of the real values of the model parameters [8, p. 5]. This means that the estimation of the parameters of the random component y_n, which may also be a time series, and the time series X_t, leads to different results. This fact we will see when evaluating the parameters of ARIMA models. Thus, we will first remove the deterministic components p_t, s_t of the time series X_t or only the seasonal component s_t, and then estimate the model parameters for the random component y_t or $y_t + p_t$. Thus, the availability of a seasonal component can change the estimation of the real value of the Hurst exponent H. In the figure 2.1, we can see the presence of the seasonal component and the polynomial trend for the company's share price of the ANADOLU EFES BIRACILIK LTD.

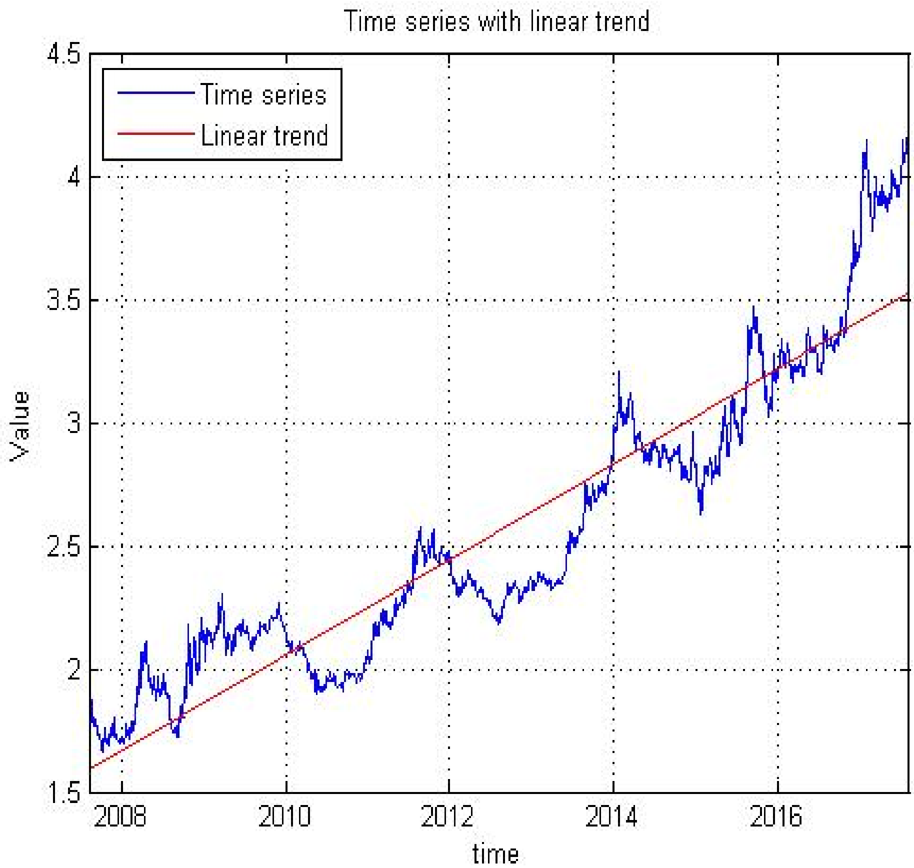

Figure 2.1: Graphic of the time series of the share price of ANADOLU EFES BIRACILIK LTD. and linear trend of this share in the period 08.08.2008 - 08.08.2017.

2. It should also be noted that in some works the decomposition (2.1) is used to cluster time series, which reflects some financial indicators. In [24], clustering is performed for 58

Mexican firms by trend, seasonal component and random component y_n, which in turn is estimated as AR (1) model. In [10], clustering is also considered, but with the Dirichlet process [33], which automatically determines the number of clusters.

As we see from the figure 2.2 in the time series, which describes the share price, all the components of the decomposition (2.1) are present, but as we can see a small seasonal component in absolute value. This means that the share price does not have a clearly defined seasonal component. On the contrary, the trend is the most pronounced term for this time series in absolute value. The random component has zero expectation and is more suitable as the sample generated by the model (1.13).

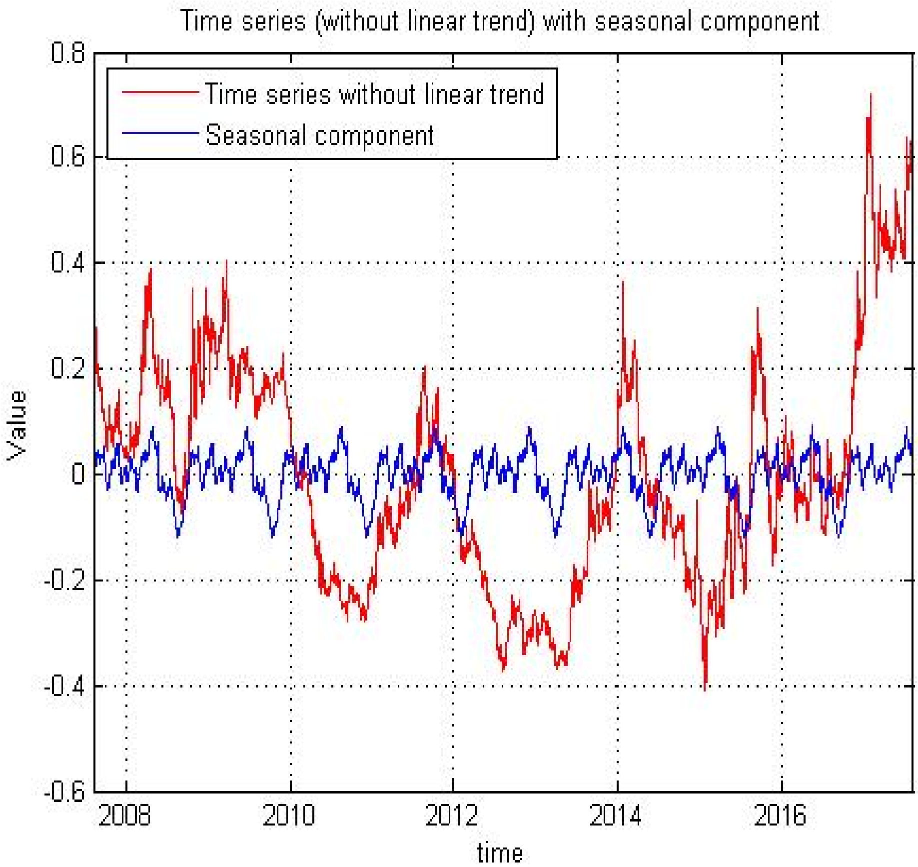

Figure 2.2: Graphic of the time series without linear trend and seasonal component in the period 08.08.2008 - 08.08.2017.

It should be noted that the decomposition of the time series should not be confused with SARMA or SARIMA models [1]. SARIMA$(p, d, q)(P, D, Q)_S$ model for the process X_t describe in the next form [8, p. 177]

$$\begin{gathered} X_t = (1 - L)^d (1 - L^S)^d Y_t, \\ \Phi(L)\Phi_1(L^S)Y_t = \Theta(L)\Theta_1(L^S)\varepsilon_t, \end{gathered} \tag{2.3}$$

[1]SARMA - seasonal ARMA model

where

$$\Phi_1(u) = 1 - \phi_1^1 u - ... - \phi_P^1 u^P,$$

$$\Theta_1(u) = 1 + \theta_1^1 u + ... + \theta_Q^1 u^Q.$$

Thus, the SARIMA model underpins some periodicity of the autocorrelation function of this process. We do not consider the representation of data using SARIMA models, since the autocorrelation functions for the considered time series do not satisfy conditions of periodically.

As you can see from the figure 2.3, the data is not described by SARIMA models. The figure 2.3 depicts the estimates of the autocorrelation function for the price of the share for unadjusted and adjusted (without linear trend and seasonal component) time series. As can be seen from these figures, the behavior of the autocorrelation function for rents is very different from the corresponding functions for the original series, but limit property of these time series are almost same.

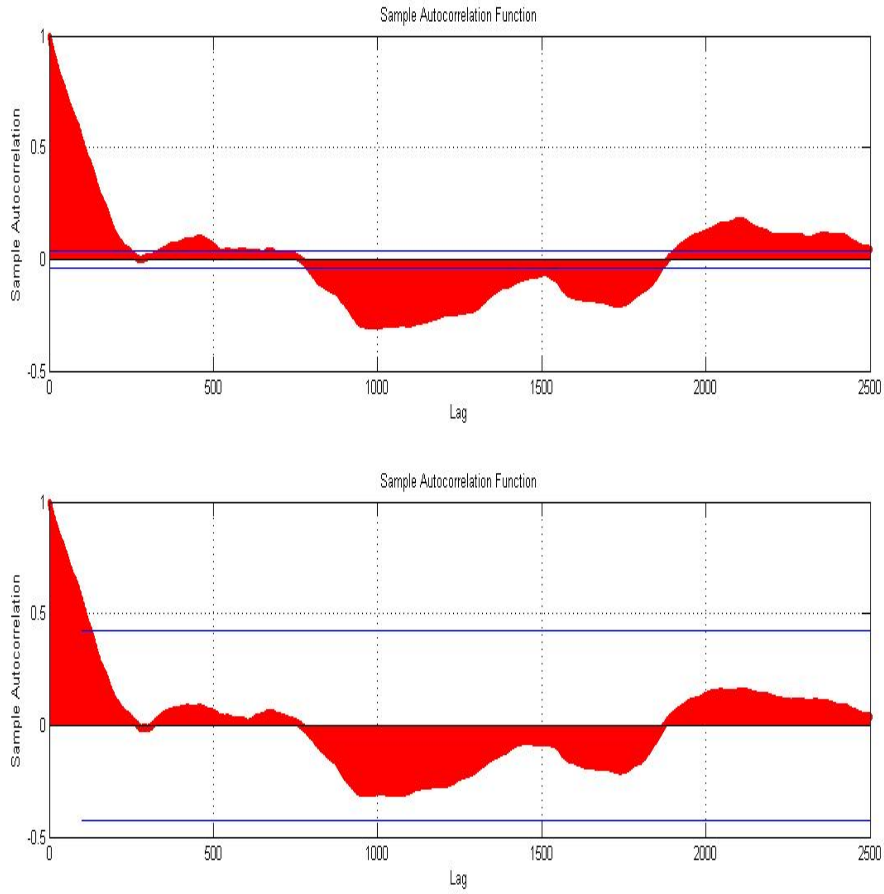

Figure 2.3: Graphic of the autocorrelation function for original time series (above figure) and adjusted time series without linear trend and seasonal component (below).

2.1.1 Deleting polynomial trend and seasonal component

Let us consider some methods for extracting a random component from the decomposition (2.1). We are interested in deleting the polynomial trend, which has a representation (2.2). Note that deleting a polynomial trend is not important for the DAF analysis, since this procedure itself presupposes the removal of the trend, but in the case of using other procedures, the presence of deterministic components (trend and seasonal components) is very important. You can delete a polynomial trend using [8, p.25]:

1. time series differences. In this case, we must consider $z_n^m = D^m x_n$ [2]. Using this operator, we obtain the following equalities

$$z_t^1 = X_t - x_{t-1} = p_t + s_t + y_t - (p_{t-1} + s_{t-1} + y_{t-1}).$$

   Using the representation (2.2), we get

$$p_t^1 = p_t - p_{t-1} = a_1 + a_2(t^2 - (t-1)^2)...a_k(t^k - (t-1)^k).$$

   Thus, using the process z^1, we remove the zero trend (drift) for the initial series X_t, in addition, the degree of the polynomial p_t^1 is one less than for, that is, $k-1$. As a result of using this method k times (the process z^k) we obtain a process without a polynomial trend $z_t^k = s_t^k + y_t^k$. This method is convenient to use in two cases: when the degree of polynomial drift k is known in advance or when this degree can be estimated.

2. estimate of coefficients of the polynomial (2.2) from the linear regression

$$X_t = a_0 + a_1 t + ...a_k t^k.$$

   This estimate gives the exact values of [3] the coefficients a_i , if the seasonal component is absent ($s_t = 0$). It should be noted that removing a trend with a small degree of $k = 1$ will not lead to a significant difference from the removal of the drift for our series, although the polynomials give a better approximation. For example, consider the trend of 6 degrees for stock:

$$X_t = 0 + 0t + 0 * t^2 + 1.353 * 10^{-8}t^3 - 1.786 * 10^{-11}t^4 + 8.0875 * 10^{-15}t^5 - 1.2265 * 10^{-18}t^6,$$

   where t – time in days with original in 08.08.2007. As we see, the coefficients of the trend decrease at large degrees. This is firstly due to the large dimensionality of the data ($N > 2000$) and, possibly, with a different (non-polynomial) trend structure. The figure 2.1 shows the time series with the polynomial trend at $k = 1$.

[2] The operator D has representation

$$DX_t = (1 - L)X_t = X_t - X_{t-1}.$$

[3] Here we have asymptotically exact values, that is, estimates of the parameters a_i will tend to their true values for $N \to \infty$, where N is the sample size.

3. smoothing the time series. According to this method, [8, p. 21] the trend of the time series has the form
$$p_n = \sum_{i=-q}^{q} \frac{x_{n+i}}{2q+1}. \tag{2.4}$$
This trend is not polynomial and this approximation is obtained from the law of large numbers [12, p. 50]. According to this law (the law of large numbers), the relation holds for $q \to \infty$:
$$\sum_{i=-q}^{q} \frac{y_{n+i}}{2q+1} \to Ey = 0.$$
For the seasonal component, the same equality holds [4]
$$\sum_{i=-q}^{q} \frac{s_{n+i}}{2q+1} \to 0.$$
This method of selecting a trend is very convenient for identifying the model, but less convenient when forecasting (forecasting) the time series, because it is impossible to extend the trend p_n for the future (for the time series at $N+1, N+2, ...$) [5].

To remove the seasonal component, we'll use the method described in [26, p. 154]. This method consists of the following. Let the time series $x_n, n = 1, ..., N$ have a period d, construct d dummy variables $d_n^i, i = 1, ..., d$:
$$d_n^i = \begin{cases} 1, n \vdots d = i, \\ 0, n \vdots d \neq i \end{cases} \tag{2.5}$$
Using these variables as independent we estimate the factors of regression
$$x_n = \sum_{i=1}^{d} \alpha_i d_n^i + u_n,$$
where u_n are errors. From this regression we obtain the seasonal component s_n of the expansion (2.1) in the form
$$s_n = \sum_{i=1}^{d} \alpha_i d_n^i. \tag{2.6}$$

[4]For the seasonal component with period d, $\sum_{i=1}^{d} s_{k+i} = K_1 = 0$. If $K_1 \neq 0$, then the value of this variable can be added with drift a_0 and get a new drift $a_0' = a_0 + K_1$. Then the seasonal component satisfies the relation $\sum_{i=1}^{d} s_{k+i} = 0$ и
$$\left|\sum_{i=-q}^{q} \frac{s_{n+i}}{2q+1}\right| = \left|\sum_{i=-q}^{-q+q\left[\frac{2q+1}{d}\right]} \frac{s_{n+i}}{2q+1} + \sum_{i=-q+1+q\left[\frac{2q+1}{d}\right]}^{q} \frac{s_{n+i}}{2q+1}\right| =$$
$$\left|\sum_{i=-q+1+q\left[\frac{2q+1}{d}\right]}^{q} \frac{s_{n+i}}{2q+1}\right| \leq \sum_{i=-q+1+q\left[\frac{2q+1}{d}\right]}^{q} \left|\frac{s_{n+i}}{2q+1}\right| \leq \frac{d*K}{2q+1}.$$

[5]For smoothing, they are needed to use the formula (2.4)

Note that during the removal of the seasonal component from the series, we will also delete the drift, that is, the modified variables (with the removed seasonal component and drift) will have the form

$$x_n^{SA} = x_n - \sum_{i=1}^{d} \alpha_i d_n^i - a_0. \tag{2.7}$$

Another method that can be used to remove the seasonal component is described in [8, p. 26]. According to this method, we must first remove the trend (in our case, delete only the drift) $x_n^{Wd} = x_n - a_0$, then the seasonal component of the series is calculated according to the formula

$$s_n = x_n^{Wd} - \frac{1}{d}\sum_{i=1}^{d} x_i^{Wd}, n = 1, .., d.$$

Removing that component from the time series x_n by the formula (2.7), we obtain the corrected time series x_n^{SA}. The disadvantage of this method is the construction of a seasonal component only for d time series values, therefore to remove the seasonal component we will use the first method with dummy components (2.5).

2.1.2 Outliers

In addition to the deterministic terms (see (2.1)), which, as we see below, affect the parameters of the time series, the same parameters are also affected by the outliers [5, p. 12]. These values can occur in real data for different reasons, for example, if there are structural discontinuities in the [3] time series. For a more detailed description of the relationship between structural discontinuities and the presence of long-term memory (Hurst exponent H), see [3, p. 10]. Here, in particular, it is noted that the presence of structural discontinuities in time series with short memory can lead to an incorrect estimate of the parameters d and H, that is, to the conclusion that there is a long-term memory in the time series. Thus, the identification of structural discontinuities in the time series is a very important stage of the study. Our task is to remove the outliers to minimize their influence on the model parameter estimates, including $\widehat{H}$ estimating the long memory of the process. Consider two methods for removing outliers.

The first is to replace the outliers with some combination of neighboring elements [8, p. 21], that is, smooth the outliers. For example, if we restrict ourselves to only two neighboring elements, then replacing the outlier x_n in the following form:

$$x_n = \frac{x_{n-1} + x_{n+1}}{2}.$$

The advantage of this method is its simplicity. The disadvantage of this method is that we do not make any assumptions about the real time series model.

The second method is to evaluate the time series values at time n for a model a priori calculated from the original data. For prediction and forecasting, we can use the predictor [8, p. 56, 88]. The algorithm for replacing outliers in this case is as follows:

1. We calculate the coefficients AR of the model using the method of least squares for time series values that do not include outliers;

2. We replace the values of the outliers with the predicted values (or forecast) for the model obtained in the first paragraph;

3. We define the basic model with new data.

These methods can reduce the impact of outsiders on model analysis.

Note that with a small number of outliers, we can discard them, using the assumption of the small effect of changes in a small part of the data [28, p. 127] to the pins, and replace them with adjacent values. Thus, according to this remark, we can use the replacement of outsiders:

$$x_n = x_{n-1},$$

where x_n is the value of the outliers for the time series.

2.2 Tests

One of the main methods of statistical research is the use of hypotheses. In our particular case, we need hypotheses that support the statistical significance of certain assumptions. Let's consider some basic hypotheses, which we will check in the next chapter. As noted in the Introduction, the analysis used the Gevek-Porter-Hudaka test [1] for long-term memory, the KPSS stationarity test [16, p. 726], the Dickey-Fuller test on the unit root [9], as well as the t Student test. Let's look briefly at each of these tests.

I. t-Student test This test is used to check the equality of the 0 coefficients of the regression model, in our case the ARMA or ARFIMA model. For this test hypotheses are considered

$$H_0 : \rho = 0 \text{ vs } H_1 : \rho \neq 0,$$

where ρ – some koefficient of the ARFIMA model (1.13). This test is performed automatically when calculating any regression in Matlab, as well as ARFIMA. Also, when using this test, confidence intervals are calculated for the corresponding coefficients.

II. Dickey – Fuller test (DF test)

As noted above, our main task is to check the availability of long-term memory in the time series. According to the definition of time series with long-term memory (see page 22). According to this definition, the time series must be stationary. In the case $d = 1$, the series (1.13) is non-stationary [8, p. 168]. Therefore, before using the estimate d, we first need to test the series for stationarity using the Dickey-Fuller test [9]. According to this test hypotheses are checked

$$H_0 : \rho = 1 \text{ vs } H_1 : \rho < 1,$$

where ρ – coefficient in AR(1) model [9, p. 1057]. As an alternative to this test, we can consider its extension to the AR (p) model – Augmented Dickey-Fuller test [31]. In this test, the hypothesis is the same as in the classical DF test. Thus, using this test to verify the existence of a unit root for the time series, which is equivalent to checking the stationarity of the time series. If the series is non-stationary, then the further evaluation of the parameter $d > 0$ does not make sense, since for $d > 0$ the series is stationary. For original data, the test results are recorded in Table 4. As we see all the rows except Antwepr, there are stationary

III. Kwiatkowski-Phillips-Schmidt-Shin and Phillip-Perron tests (KPSS, PP test)

Similar to the DF test, this test is designed to test the time series for stationarity cite KPSS92. An interesting approach is proposed in this paper: as a null hypothesis, the hypothesis of stationarity of the time series is considered against the hypothesis of the presence of a unit root in this time series [21, p. 160] [6]:

$$H_0: \text{ time series is stationary vs } H_1: \text{ unit root in time series.}$$

An alternative to using this test can be the Phillip – Perron test [25], which considers the same hypotheses as the DF test. The results of this test coincide with the corresponding results for the DF test for the original data.

[6]This work presents a clear test as the null hypothesis of stationarity versus the alternative of a unit root.

Chapter 3

Example of the calculation of the Hurst index

In this chapter, estimates of the Hurst index for the inspiration of the examples will be made. In the first example, we will consider the Turkish company – ANADOLU EFES BIRACILIK LTD, in the second - four British companies: ADMIRAL GROUP, ANGLO AMERICAN, ANTOFAGASTA, ASHTEAD GROUP, ASSOCIATED BRIT.FOODS. The estimation if the parameter H will be carried out using the MFDFA analysis and the GPH method. As will be shown, these methods give practically the same result in the evaluation of the parameter H. This fact tells us that the given time series have the Hurst index, which does not depend on the method of estimation.

3.1 Example 1. ANADOLU EFES BIRACILIK LTD

Estimation of the Hurst index for the shape price ANADOLU EFES BIRACILIK LTD in the period 08.08.2007 to 08.08.2017. Notice that we use original data without adjusting and deleting any components. Also, we don't use rent, but share price. For different technique we get next results:

Method of estimation	Estimatio of $\widehat{H}$
R/S analysis	1.4794
DFA analysis for unadjusted time series	1.521
MDDFA analysis for adjusted time series $q = -5$	1.7589
MDDFA analysis for adjusted time series $q = 0$	1.5585
MDDFA analysis for adjusted time series $q = 5$	1.3533
GPH estimation	1.463

As we see from this table, parameter estimates are fairly close. This means that the current time series have is a long memory, because estimation of the parameter $H > 0.5$. Using density of the multifractal density (figure 1.3) we get that estimation of the Hurst index by MFDFA is equal 1.5845, by R/S analysis – 1.4794, by GPH method – 1.463. To confirm this conclusion about long memory effect in time series, we consider the Dickey-Fuller and KPSS tests for checking stationarity of this time series. Results of these tests are next:

Test	p-value
ADF	0.7727
KPSS	0.01

As we see from last table, our time series are non stationary so it is possible to explain as ARFIMA(p,d,q) model with $d > 0$. These conclusions are consistent with the results of the estimation of the parameter P by means of three methods.

The second step of the comparison is to remove the seasonal component s_t from decomposition (2.1) and compare the results with the corresponding results for the original data. So, consider adjusted tine series without linear trend and seasonal component. For this time series we get next values of the estimation of the parameter H :

Method of estimation	Estimatio of $\widehat{H}$
R/S analysis	0.8649
DFA analysis for unadjusted time series	1.521
MDDFA analysis for adjusted time series $q = -5$	1.615
MDDFA analysis for adjusted time series $q = 0$	1.4635
MDDFA analysis for adjusted time series $q = 5$	1.2803
GPH estimation	1.4867

As we can see from this table, there is a big difference between the estimates of the Hurst index for the original time series of adjusted time series (without linear trend and seasonal component). Thus, we can conclude that the deleting of the deterministic component changes (decreases) the Hurst index. It is strange, because for high values of the Hurst index H, the time series becomes more and more similar to the deterministic function. Plot for the functions $H(q)$, $\tau(q)$ and $D(q)$ we can see in the next figure 3.1.

For adjusted time series, as and for original time series, from figures we can se than $H(q)$ statistically insignificant depend from parameter q. This mean, that

$$\widehat{H(q)} = const$$

for any q. For example mode of the H is equal 1.5296 for the adjusted time series and equal 1.5845 for real time series. Also we get difference result for other method

	Unadjusted time seires	Adjusted time seires
R/S analysis	1.4794	0.8649
MDDFA	1.5845	1.5296
GPH estimation	1.463	1.4867

As we see from this table, exist significance difference between estimation for adjusted and unadjusted data.

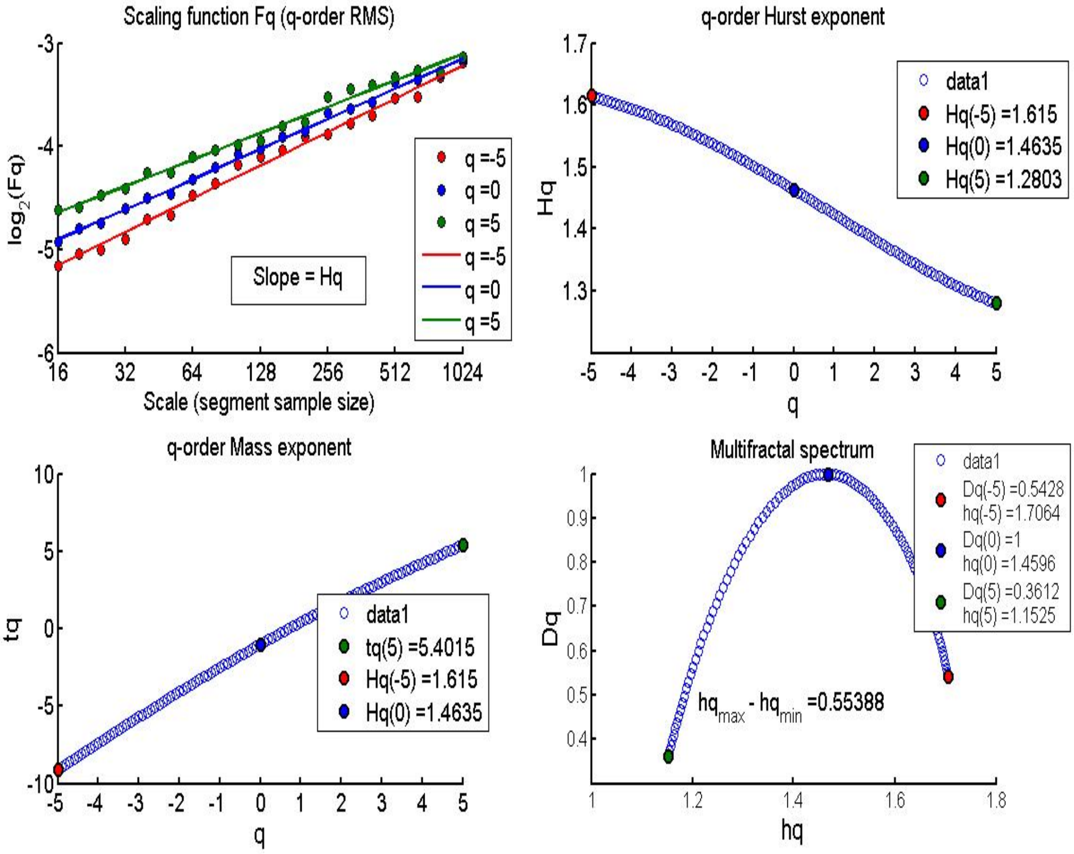

Figure 3.1: Plots of the $H(q)$, $\tau(q)$, $D(q)$ for $q \in [-5, 5]$ for adjusted time series.

3.2 Example 2. Five companies from UK

In this example we consider five companies from UK[1]. Using unadjusted data for this five companies we get next results of the estimation of the Hurst index H for each company

Company	$q = -5$	$q = 0$	$q = 5$	GPH	ADF test	KPSS test
ADMIRAL GROUP	1.6449	1.5289	1.3863	1.6733	0.3276	0.01
ANGLO AMERICAN	1.7717	1.5694	1.4078	1.6218	0.4607	0.1
ANTOFAGASTA	1.7042	1.5292	1.3700	1.5795	0.6531	0.01
ASHTEAD GROUP	1.7339	1.6240	1.4990	1.5184	0.7068	0.01
ASSOCIATED BRIT.FOODS	1.7887	1.5801	1.4190	0.9741	0.5017	0.01

First consider ADF and KPSS test. Using these tests, we get that all five time series are not stationary. This mean that for all of these time series we can‘t reject hypothesis about non-stationarity of time series. Now consider estimation of the Hurst index H. As we see two

[1] ADMIRAL GROUP, ANGLO AMERICAN, ANTOFAGASTA, ASHTEAD GROUP, ASSOCIATED BRIT.FOODS in the period from 28.09.2007 to 29.09.2017

methods (MFDFA analysis and GPH estimation) give same result. So, we can conclude that all five companies have long-memory, because estimation of $H > 0.5$ for all five companies in each case. In general, we see that all time series have a long-term memory. This means that these companies are less responsive to shocks in the market, that is, the price of shares of these companies is more resistant to shocks in the market.

3.3 Construction white noise and Brownian motion by model

After evaluating the Hurst index H, we can simulate the corresponding prices assuming that their dynamics are described by the Brownian motion with the Hurst index H calculated in example 1, 2 or white noise. The results for the Turkish company ANADOLU EFES BIRACILIK LTD are as follows – see next two figures:

As we see from these figure 3.2 , approximation by white noise and Brownian motion are not good for ANADOLU EFES BIRACILIK LTD.

For other example for White noise and Brownian Motion, Turkey exchange rate data over the period of 08/08/2007 - 08/08/2017 is used. The results can be seen in the following figures.

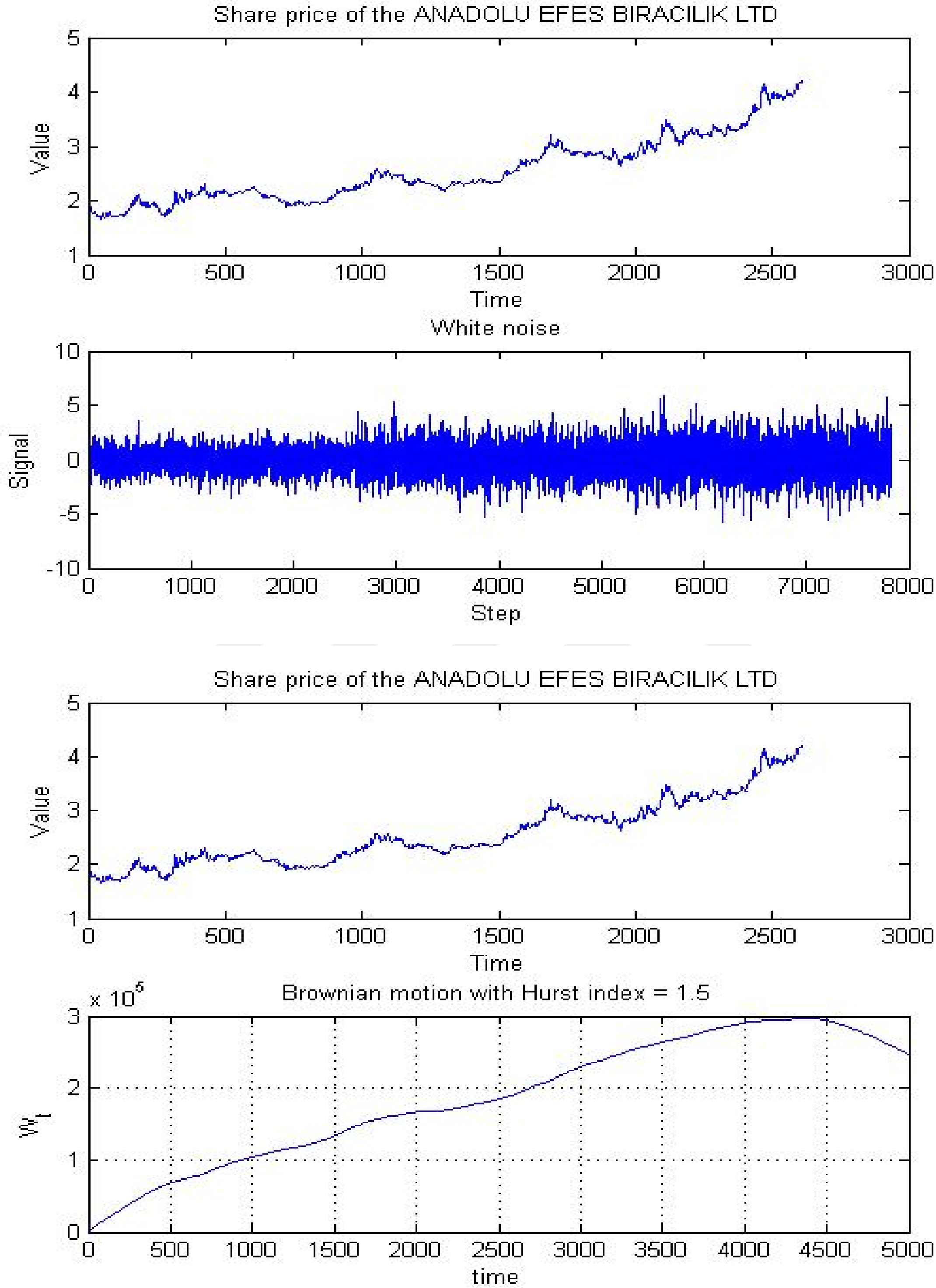

Figure 3.2: Evaluation of the white noise and Brownian motion by data for ANADOLU EFES BIRACILIK LTD

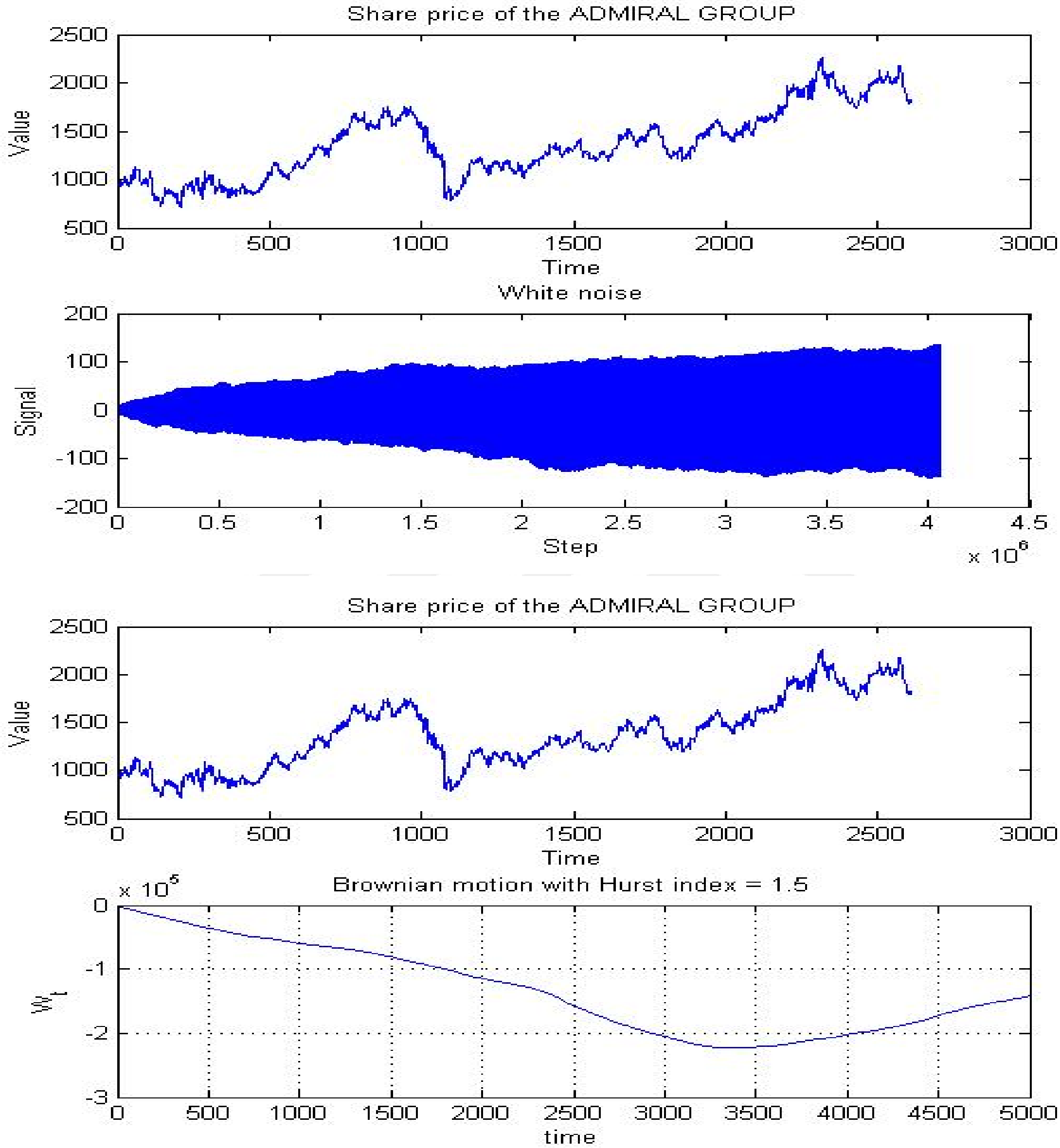

Figure 3.3: Evaluation of the white noise an Brownian motion by data for ADMIRAL GROUP

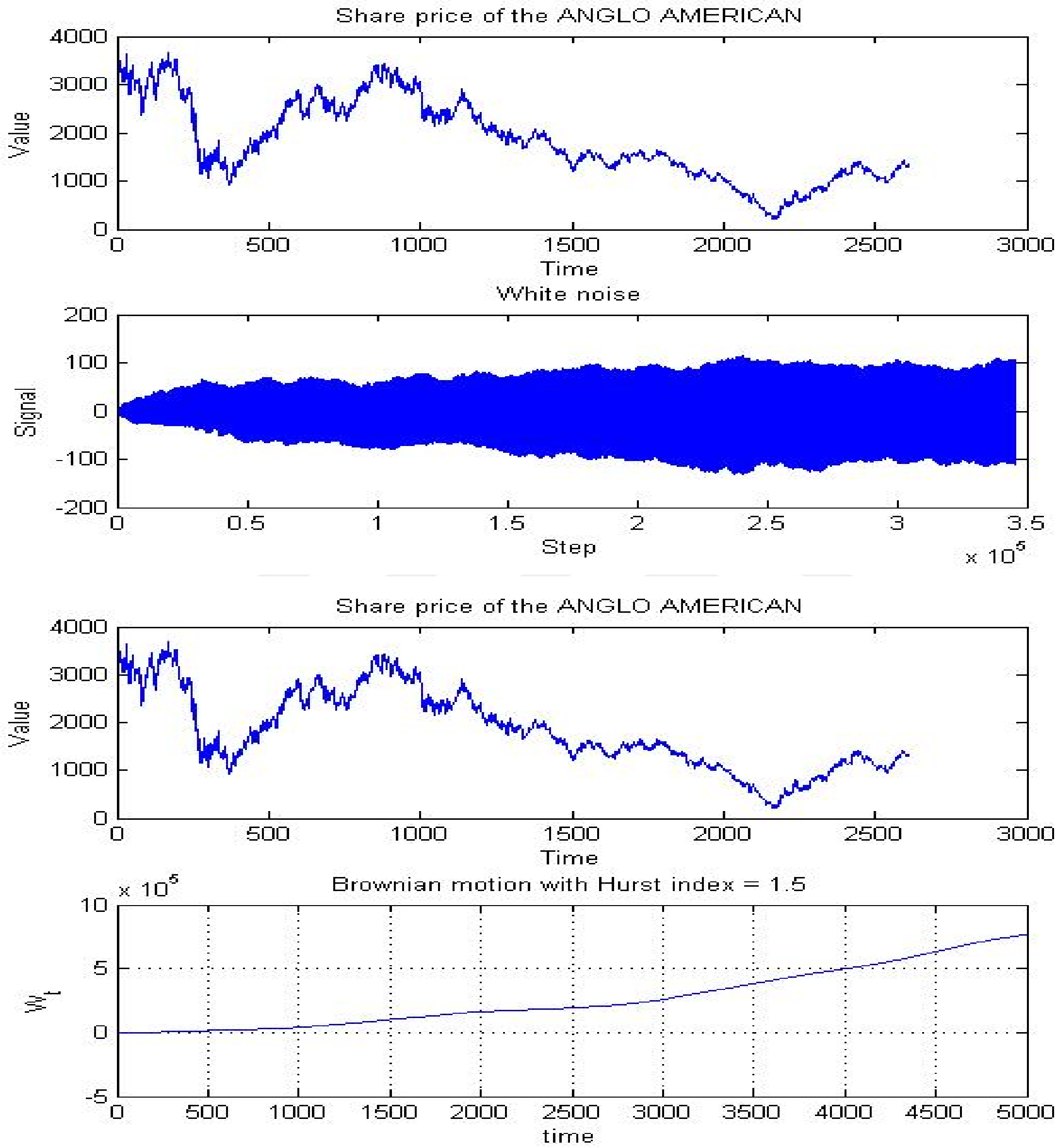

Figure 3.4: Evaluationof the white noise an Brownian motion by datafor ANGLO AMERICAN

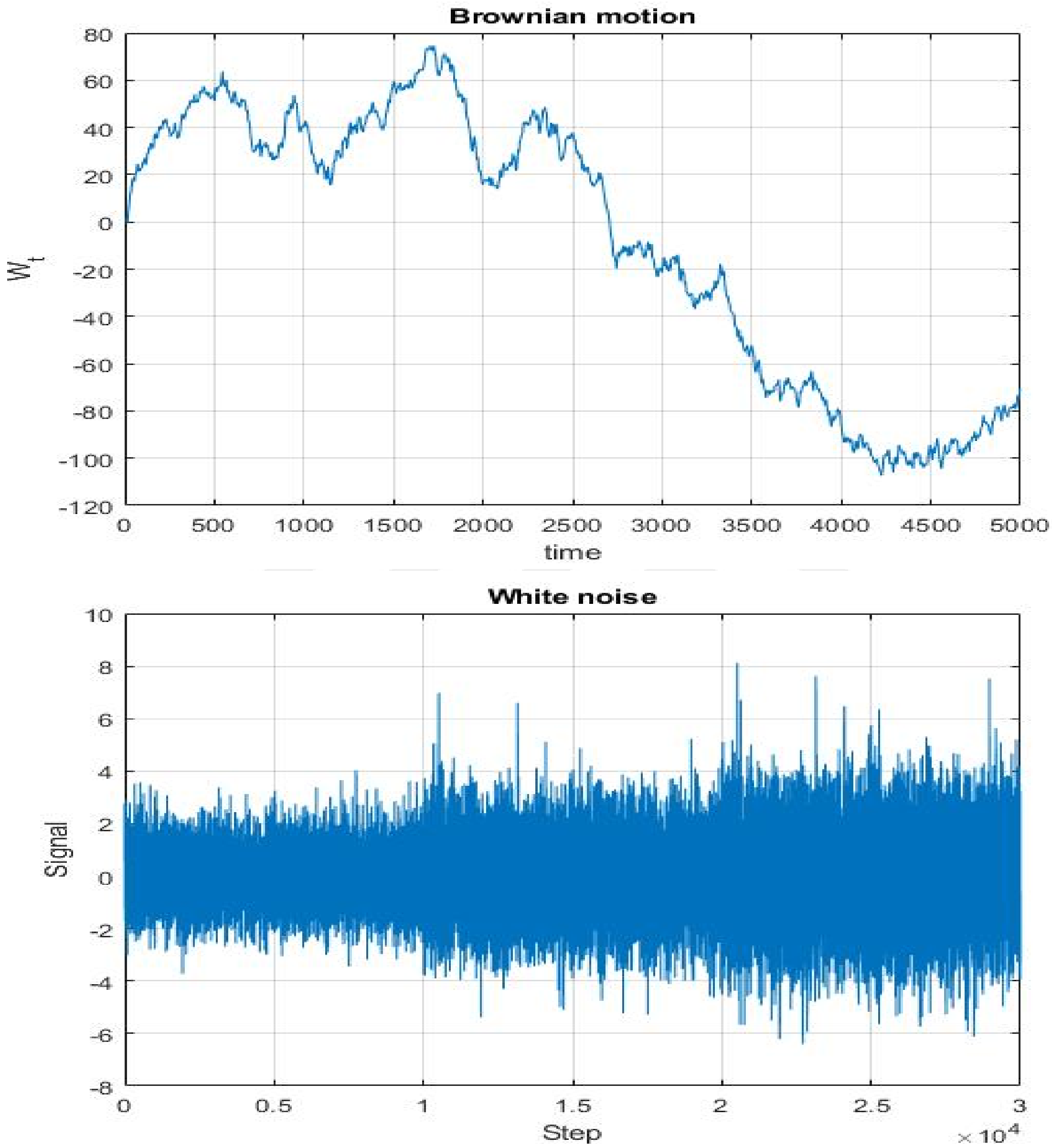

Figure 3.5: Evaluation of the white noise an Brownian motion by data for TURKEY EXCHANGE RATE DATA

Conclusions

The study of the presence of 'long-term' memory in the time series is very large for research, since it allows us to construct adequate assumptions about the real model of the time series. Consideration of the estimation of the Hurst index H began to be considered in the works of Hurst [17] and Kolmogorov [18]. At present, the most powerful method of estimating parameter H is MFDFA [19], which is based on the fluctuation function. Currently, there are many methods for estimation of the Hurst index. In the work, it was shown, in the author's opinion, the most important:

1. R/S estimation;
2. DFA estimation;
3. MDFA estimation;
4. GPH estimation.

It was shown in the work that there is a rather large difference between the estimates of H for different methods. This result indicates different assumptions for different estimation methods. In addition, it was established that the presence of deterministic components (polynomial trend and seasonal component) increases the estimation of the parameter H. As can be seen from tables 1 and 3, this increase is approximately 10%. Thus, it is quite important to conclude that the time series have been separate by deterministic and random terms before the study of the presence of long-term memory – estimation of the parameter H. As noted in the work, the presence of a trend can not influence the assessment of H using the MFDFA method, but the presence of the periodic component s_t increases the estimation of the Hurst index H due to the appearance of 'additional fluctuations' in the fluctuation function $F(l;q)$ for any q. The multifractal dimension of the time series was described in the work for the determination of the Hurst index (the only one form function $H(q)$). With the help of this function (see (1.6)) and its density (see, for example, Figure 1.3), with show that estimation of the Hurst index is uniquely determined. In addition, using the scaling exponent $\tau(q)$, we show that the Hurst index is a constant[2].

In chapter 3 there was an analysis of one Turkish company and four companies from the United Kingdom. As a result, it was found that for all these companies the Hurst index is approximately 1.5. This means that the companies we have selected have long-term memory, that is, they react less to the shocks in the market.

[2]This mean, that changes of the function $H(q)$ is insignificant respect to the changes of the q

Thus, our research makes it possible to verify the fact that the MFDFA analysis is one of the most powerful method for estimation of the Hurst index H. However, his main disadvantage in our study is the sensitivity to deterministic components that differ from polynomial trends. Therefore, important steps in the further development of the MFDFA analysis are the consideration of these deterministic components themselves.

Bibliography

[1] Agiakloglou, C., Newbold, P. and Wohar, M. (1992) Bias in an estimator of the fractional difference parameter, *Journal of Time Series Analysis, 14, 235 – 246*

[2] Amin, M., Amanullah, M. and Akbar A. (2014) Time series modeling foe forecasting wheat production of Pakistan, *The Journal of Animal & Plant Sciences, 24(5), Page: 1444-1451.*

[3] Auea A., Horvath L. (2013) Structural breaks in time series, *Journal of time series analysis.*

[4] Artin, E. (1965) The gamma functions. *Holt, Rinehart and Winston.*

[5] Baillic, R.T. (1996) Long memory processes and fractional integration in econometrics, *Journal of Econometrics* 73(59), 5-59.

[6] Cheung, Yin-Wong (1993) Long memory in foreign exchange rates,*Journal of Business and Economic Statistics,* 11 , 1, 93-101.

[7] Christodoulou-Volos, C., Siokis, F.M., Zikos, S. (2013) Is there long term memory in Uropean stock market?, *IJER, Vol. 10, No. 2, 335-348.*

[8] Brockwell P.J., Davis R.A. (2016) Introduction to Time Series and Forecasting, *3rd Edition, Springer International Publishing Switzerland.*

[9] Dickey, D. and Fuller, W. (1981) Likelihood ratio statistics for autoregressive time series with a unit root, *Econometrica, 49, 1057 – 1072*

[10] Dahl, D. B. (2006). Model-based clustering for expression data via a Dirichlet process mixture model. Bayesian Inference for Gene Expression and Proteomics (Cambridge University Press), 201-218.

[11] Delgado, M.A. and Robinson P.M. (1993) NEW METHODS FOR THE ANALYSIS OF LONG MEMORY TIME SERIES: APPLICATION TO SPANISH INFLATION, *Statistics and Econometrics Series.*

[12] Durrett R. (2013) Probability: Theory and Examples, *4th editions, Cambridge University Press.*

[13] Einstein A. (1926) Investigations on the Theory of the Brownian Movement. London, Methuen & Co., 124 p.

[14] Feller, W. (1965) The assymptotic distribution of the range of the sum of independent random variables

[15] Geweke, J., Porter-Hudak, S. (1983) The Estimation and Application of Long Memory Time Series Models, Journal of Time Series Analysis 4, 221-237.

[16] Henry, O.T.(2002) Long memory in stock international evidence, *Applied Financial Economics, 12 , 725-729.*

[17] Hurst H.E. (1951) Long-Term Storage Capacity of Reservoirs // Transactions of the American Society of Civil Engineers, Vol. 116, Issue 1, P. 770-799.

[18] Kolmogorov, A.N. (1940) The Wiener spiral and some other interesting curves in Hilbert space // *Dokl. Akad. Nauk SSSR, 26*, 115–118.

[19] Kantelhardt, J.W., Zschiegner, S.A., Koscielny-Bunde, E., Havlin,S., Bunde,A., and Stanley, H.E.(2002). Multifractal detrended fluctuation analysis of non stationary time series. *Physica A*, Vol. 316, 87–114.

[20] Mandelbrot, B.B., van Ness J.W.(1968) Fractional Brownian motions, fractional noices and applications // *SIAM Review, 10*, 422–437.

[21] Kwiatkowski, D., Phillips, P.C.B., Schmidt, P., Shin, Y. (1992) Testing the null hypothesis of stationarity against the alternative of a unit root, *Journal of Econometrics. 54 (1-3)*, 159-178.

[22] Mishura, Yu. (2008) Stochastic Calculus for Fractional Brownian Motion and Related Processes.

[23] Morris, C.C., Stark, R.M. (2016) FUNDAMENTALS OF CALCULUS. *John Wiley & Sons, Inc.*

[24] Nieto-Barajas, L.E., Contreras-Cristan, A. (2014) A Bayesian nonparametric approach for time series clustering. Bayesian Analysis 9(1), 147-170.

[25] Phillips, P.C.B.; Perron, P. (1988) Testing for a Unit Root in Time Series Regression, *Biometrika. 75 (2)*, 335?346.

[26] Pedace, R. (2013) Econometrics for Dammies, *Willey.*

[27] Peng, C.K.; et al. (1994). Mosaic organization of DNA nucleotides. *Phys. Rev. E.*, V. 49, 1685–1689.

[28] Prado, R., West, M. (2010) Time series: modeling, computation and interference, *CRC Press.*

[29] Rege, S., Martin, S. G. (2011) Portuguese Stock Market: A Long-Memory Process?. *Verslas: teorija ir praktika 12:1, 75-84.*

[30] Rezitis, A.N., Ntinou, A.G. and Pachis D.N. (2015) Investigating the international prices of wheat and rice, *Agricultural and Food Economics, 3, 16.*

[31] Said, S.E., Dickey, D.A. (1984). Testing for Unit Roots in Autoregressive-Moving Average Models of Unknown Order. *Biometrika. 71 (3)*, 599?607

[32] Szczepanik, M., Domaga?a, D. (2014) Modeling and forecasting rye price using ARIMA model, *Electronic Journal of Polish agricultural universities, Volume 17, Issue 4.*

[33] Teh Y.W., Jordan M.I., Beal M.J., Blei D.M. Hierarchical Dirichlet Processes

[34] Wilfredo, P. (2007) Long-memory time series: Theory and Methods. *WILEY-INTERSCIENCE A JOHN WILEY & SONS, INC., PUBLICATION.*

Appendix

Appendix 1.

```
clc
clear
warning('off','all')
A = xlsread('TURKEY_DATA.xlsx','EXCHANGE RATE Matlab','B:B');
L=2500;
l = 100;
d = 300; % period of time series
T = length(A);
q = 0.1:0.1:10;
%q = 1:1:3;


%% Calculation of the fluctuation function
scmin=16;
scmax=1024;
scres=19;
exponents=linspace(log2(scmin),log2(scmax),scres);
scale=round(2.^exponents);
q=linspace(-5,5,101);
m=1;

[Hq1,tq1,hq1,Dq1,Fq1] = MFDFA1(A,scale,q,m,1);
[Ht1,Htbin1,Ph1,Dh1] = MFDFA2(A,scale,m,1);

%% GPH estimation
gph(A)

%% Tests of the stationarity and homoscedasticy
[hDF, pDF] = adftest(A, 'model','TS','lags',0:3);
[hKPSS, pKPSS] = kpsstest(A, 'lags',0:3);
```

```
%% R/S analisys
y = A - mean(A);
z = cumsum(y);
S = sqrt(var(A));
R = max(z)-min(z);

H_RS = (log(R)-log(S))/log(T);

%% Deletind linear trend and seasonal component
time = 1:T;
M1 = LinearModel.fit(time, A);
Trend = table2array(M1.Coefficients(1,1))+table2array(M1.Coefficients(2,1))*time

startDate = datenum('08-08-2007');
endDate = datenum('08-08-2017');
xData = linspace(startDate,endDate,T);

figure;
plot(xData,A(:,1));
hold on;
grid on;
plot(xData,Trend,'r');
datetick('x','yyyy')
xlim([min(xData) max(xData)])
hold off;
title('Time series with linear trend')
xlabel('time')
ylabel('Value')
legend('Time series','Linear trend', 'Location', 'northwest')
TS = A(:,1) - Trend';

%% Delete seasonal term
Dummy = zeros(T,d);

for i=1:d
    Dummy(i:d:T,i) = 1;
end;
MS = fitlm(Dummy, TS, 'Intercept' ,false);

Seasoal = zeros(T,1);
```

```
for i=1:d
    Seasoal = Seasoal + table2array(MS.Coefficients(i,1))*Dummy(:,i);
end;

figure
plot(xData,TS,'r');
hold on
plot(xData,Seasoal, 'b')
datetick('x','yyyy')
xlim([min(xData) max(xData)])
hold off;
title('Time series (without linear trend) with seasonal component')
xlabel('time')
ylabel('Value')
grid on;
legend('Time series without linear trend','Seasonal component', 'Location', 'nort

TS_Adj = TS - Seasoal;

%% Build autocorrelation function
subplot(2,1,1)
autocorr(TS, 2500)
subplot(2,1,2)
autocorr(TS_Adj, 2500, 'b')


%% Autocorrelation function
phi = 0.9;
theta = 0.3;
d=0.2;
ARMA=[];
ARFIMA = [];
N=80;
for i = 1:N
    ARMA(i) = phi^(i-1);
    ARFIMA(i) = i^(2*d-1);
end;
figure
plot(0:(N-1),ARMA, 'r--o');
grid on;
hold on;
plot(0:(N-1),ARFIMA, 'b--o');
```

```
plot(0:(N-1),0*(0:(N-1)));
hold off;
title('ACF for ARMA(1,1) and ARFIMA(0,0.2,0)')
xlabel('h')
ylabel('Value of autocorrelation funtion \rho(h)')
ylim([-0.1 1]);

%% Checking after deleting deterministic coefficients

[Hq1,tq1,hq1,Dq1,Fq1] = MFDFA1(TS_Adj,scale,q,m,1);
[Ht1,Htbin1,Ph1,Dh1] = MFDFA2(TS_Adj,scale,m,1);

%% R/S estimation
A = TS_Adj;
y = A - mean(A);
z = cumsum(y);
S = sqrt(var(A));
R = max(z)-min(z);
H_RS = (log(R)-log(S))/log(T);

%% GPH estimation
gph(A)

%% Brownian mmotion
BM(A, 1.5, 'ANADOLU EFES BIRACILIK LTD');
WN(A, 1.5, 'ANADOLU EFES BIRACILIK LTD');

PlotMFDFA(A, 400) ;
gph([1,2, 5, 2,5])
```

Appendix 2.

```
clc
clear
warning('off','all')
Data = xlsread('UK_data.xlsx','B6:F2616');
%A = xlsread('TURKEY_DATA.xlsx','EXCHANGE RATE Matlab','B:B');

[n,m] = size(Data);

Result = zeros(m,6);
for k = 1:m
```

```
  display(['Iteration ', num2str(k)])
%% Calculation of the fluctuation function
A = Data(:,k);
 scmin=16;
 scmax=1024;
 scres=19;
 exponents=linspace(log2(scmin),log2(scmax),scres);
 scale=round(2.^exponents);
 q=linspace(-5,5,101);
m=1;

 [Hq1,~,~,~,~] = MFDFA1(A,scale,q,m,0);
 Result(k,1:3) = Hq1([1,51, 101]);
 %[Ht1,Htbin1,Ph1,Dh1] = MFDFA2(A,scale,m,0);

%% GPH estimation
 Result(k,4) = gph(A);

%% Tests of the stationarity and homoscedasticy
 [~, pDF] = adftest(A, 'model','TS','lags',3);
 [~, pKPSS] = kpsstest(A, 'lags',3);
 Result(k,5:6) = [pDF, pKPSS];

 end;

BM(Data(:,1), 1.5, 'ADMIRAL GROUP');
BM(Data(:,2), 1.5, 'ANGLO AMERICAN');
%%
WN(Data(:,1), 1.5, 'ADMIRAL GROUP');
WN(Data(:,2), 1.5, 'ANGLO AMERICAN');

 Result
```

Appendix 3.

```
 function [Hq,tq,hq,Dq,Fq] = MFDFA1(signal,scale,q,m,Fig)
% Multifractal detrended fluctuation analysis (MFDFA)
%
% [Hq,tq,hq,Dq,Fq]=MFDFA(signal,scale,q,m,Fig);
%
% INPUT PARAMETERS---------------------------------------------------------
%
```

```
% signal:        input signal
% scale:         vector of scales
% q:             q-order that weights the local variations
% m:             polynomial order for the detrending
% Fig:           1/0 flag for output plot of Fq, Hq, tq and multifractal
%                spectrum (i.e. Dq versus hq).
%
% OUTPUT VARIABLES---------------------------------------------------------
%
% Hq:            q-order Hurst exponent
% tq:            q-order mass exponent
% hq:            q-order singularity exponent
% Dq:            q-order dimension
% Fq:            q-order scaling function
%
% EXAMPLE-----------------------------------------------------------------
%
% load fractaldata
% scmin=16;
% scmax=1024;
% scres=19;
% exponents=linspace(log2(scmin),log2(scmax),scres);
% scale=round(2.^exponents);
% q=linspace(-5,5,101);
% m=1;
% signal1=multifractal;
% signal2=monofractal;
% signal3=whitenoise;
% [Hq1,tq1,hq1,Dq1,Fq1]=MFDFA1(signal1,scale,q,m,1);
% [Hq2,tq2,hq2,Dq2,Fq2]=MFDFA1(signal2,scale,q,m,1);
% [Hq3,tq3,hq3,Dq3,Fq3]=MFDFA1(signal3,scale,q,m,1);
%-------------------------------------------------------------------------
warning off
X=cumsum(signal-mean(signal));
if min(size(X))~=1||min(size(scale))~=1||min(size(q))~=1;
    error('Input arguments signal, scale and q must be a vector');
end
if size(X,2)==1;
   X=transpose(X);
end
if min(scale)<m+1
   error('The minimum scale must be larger than trend order m+1')
```

```
end
for ns=1:length(scale),
    segments(ns)=floor(length(X)/scale(ns));
    for v=1:segments(ns),
        Index=((((v-1)*scale(ns))+1):(v*scale(ns)));
        C=polyfit(Index,X(Index),m);
        fit=polyval(C,Index);
        RMS_scale{ns}(v)=sqrt(mean((X(Index)-fit).^2));
    end
    for nq=1:length(q),
        qRMS{nq,ns}=RMS_scale{ns}.^q(nq);
        Fq(nq,ns)=mean(qRMS{nq,ns}).^(1/q(nq));
    end
    Fq(q==0,ns)=exp(0.5*mean(log(RMS_scale{ns}.^2)));
end
for nq=1:length(q),
    C = polyfit(log2(scale),log2(Fq(nq,:)),1);
    Hq(nq) = C(1);
    qRegLine{nq} = polyval(C,log2(scale));
end
tq = Hq.*q-1;
hq = diff(tq)./(q(2)-q(1));
Dq = (q(1:end-1).*hq)-tq(1:end-1);

%OUTPUT FIGURE--------------------------------------------------------------
if Fig==1,
   qindex=[1,round(length(q)/2),length(q)];
   qindex2=[1,round((length(q)-1)/2),length(q)-1];

   %Variable settings----------
   qstart1=q(qindex(1));
   qmid1=q(qindex(2));
   qstop1=q(qindex(3));
   Hqstart1=Hq(qindex(1));
   Hqmid1=Hq(qindex(2));
   Hqstop1=Hq(qindex(3));
   tqstart1=tq(qindex(1));
   tqmid1=tq(qindex(2));
   tqstop1=tq(qindex(3));
   hqstart2=hq(qindex2(1));
   hqmid2=hq(qindex2(2));
   hqstop2=hq(qindex2(3));
```

```
Dqstart1=Dq(qindex2(1));
Dqmid1=Dq(qindex2(2));
Dqstop1=Dq(qindex2(3));
for nq=1:length(qindex),
    qRegFit(nq,:)=qRegLine{qindex(nq)};
end
X1=log2(scale);
YMatrix1=[log2(Fq(qindex,:));qRegFit];
X2=q;
Y1=Hq;
Y3=tq;
X4=hq;
Y5=Dq;
%-------------------------------------------
 q_end=num2str(max(q));
 if length(find(q==0))==1
    q_middle=0;
 else
    qm0=min(q)+((max(q)-min(q))/2);
    qm1=[q(find(q<qm0, 1, 'last' )),q(find(q>qm0, 1 ))];
    qm2=qm1-qm0;
    midind= qm2==min(qm2);
    q_middle=qm2(midind(1));
 end
 q_mid=num2str(q_middle);
 q_start=num2str(min(q));

 % Create figure
 figure1 = figure('PaperSize',[20.98 29.68],'Color',[1 1 1]);

 scaleInd=floor(min(log2(scale))):ceil(max(log2(scale)));
 for ns=1:length(scaleInd),
     scaletick{ns}=num2str(2^scaleInd(ns));
 end
 Fqind=floor(log2(min(min(Fq)))):ceil(log2(max(max(Fq))));
 for nf=1:length(Fqind),
     Fqtick{nf}=num2str(Fqind(nf));
 end

 % Create subplot
 subplot1 = subplot(2,2,1,'Parent',figure1,...
     'YTickLabel',Fqtick,...
```

```
    'XTickLabel',scaletick ,...
    'XTick',scaleInd ,...
    'LineWidth',2,...
    'FontSize',14);
hold(subplot1,'all');

% Create multiple lines using matrix input to plot
plot1 = plot(X1,YMatrix1,'Parent',subplot1);
set(plot1(1),'MarkerFaceColor',[1 0 0],'MarkerEdgeColor',[0 0 0],...
    'Marker','o',...
    'LineStyle','none',...
    'Color',[1 0 0],...
    'DisplayName',strcat('q = ',num2str(min(q))));
set(plot1(2),'MarkerFaceColor',[0 0 1],'MarkerEdgeColor',[0 0 0],...
    'Marker','o',...
    'LineStyle','none',...
    'Color',[0 0 1],...
    'DisplayName',strcat('q = ',num2str(q_middle)));
set(plot1(3),'MarkerFaceColor',[0 0.498 0],'MarkerEdgeColor',[0 0 0],...
    'Marker','o',...
    'LineStyle','none',...
    'Color',[0 0.498 0],...
    'DisplayName',strcat('q = ',num2str(max(q))));
set(plot1(4),'LineWidth',2,'Color',[1 0 0],'DisplayName',strcat('q = ',num2s
set(plot1(5),'MarkerFaceColor',[1 0 0],'MarkerEdgeColor',[1 1 1],...
    'LineWidth',2,...
    'Color',[0 0 1],...
    'DisplayName',strcat('q = ',num2str(q_middle)));
set(plot1(6),'LineWidth',2,'Color',[0 0.498 0],'DisplayName',strcat('q = ',nu

% Create xlabel
xlabel('Scale (segment sample size)','FontSize',14);

% Create ylabel
ylabel('log_2(Fq)','FontSize',14);

% Create title
title('Scaling function Fq (q-order RMS)','FontSize',14);

qInd=floor(min(q)):ceil(max(q));
for nq=1:length(qInd),
    qtick{nq}=num2str(qInd(nq));
```

```
end

% Create subplot
subplot2 = subplot(2,2,2,'Parent',figure1,...
    'XTickLabel',qtick,...
    'XTick',qInd,...
    'LineWidth',2,...
    'FontSize',14);
hold(subplot2,'all');

% Create plot
plot(X2,Y1,'Parent',subplot2,'Marker','o','LineStyle','none',...
    'Color',[0 0 1]);

% Create xlabel
xlabel('q','FontSize',16);

% Create ylabel
ylabel('Hq','FontSize',16);

% Create title
title('q-order Hurst exponent','FontSize',14);

% Create plot
plot(qstart1,Hqstart1,'Parent',subplot2,'MarkerFaceColor',[1 0 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[1 0 0],...
    'DisplayName',strcat('Hq(',num2str(min(q)),') = ',num2str(Hq(q==min(q))))

% Create plot
plot(qmid1,Hqmid1,'Parent',subplot2,'MarkerFaceColor',[0 0 1],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0 1],...
    'DisplayName',strcat('Hq(',num2str(q_middle),') = ',num2str(Hq(q==q_midd
```

```
% Create plot
plot(qstop1,Hqstop1,'Parent',subplot2,'MarkerFaceColor',[0 0.498 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0.498 0],...
    'DisplayName',strcat('Hq(',num2str(max(q)),') = ',num2str(Hq(q==max(q))))

% Create subplot
subplot3 = subplot(2,2,3,'Parent',figure1,...
    'XTickLabel',qtick,...
    'XTick',qInd,...
    'LineWidth',2,...
    'FontSize',14);
hold(subplot3,'all');

% Create plot
plot(X2,Y3,'Parent',subplot3,'Marker','o','LineStyle','none',...
    'Color',[0 0 1]);

% Create xlabel
xlabel('q','FontSize',16);

% Create ylabel
ylabel('tq','FontSize',16);

% Create title
title('q-order Mass exponent','FontSize',14);

% Create plot
plot(qstop1,tqstop1,'Parent',subplot3,'MarkerFaceColor',[0 0.498 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0.498 0],...
    'DisplayName',strcat('tq(',num2str(max(q)),') = ',num2str(tq(q==max(q))))
```

```
% Create plot
plot(qstart1,tqstart1,'Parent',subplot3,'MarkerFaceColor',[1 0 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[1 0 0],...
    'DisplayName',strcat('Hq(',num2str(min(q)),') = ',num2str(Hq(q==min(q)))

% Create plot
plot(qmid1,tqmid1,'Parent',subplot3,'MarkerFaceColor',[0 0 1],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0 1],...
    'DisplayName',strcat('Hq(',num2str(q_middle),') = ',num2str(Hq(q==q_midd

% Create subplot
subplot4 = subplot(2,2,4,'Parent',figure1,'LineWidth',2,'FontSize',12);
% Uncomment the following line to preserve the X-limits of the axes
% xlim(subplot4,[0.2 1.8]);
hold(subplot4,'all');

% Create plot
plot(X4,Y5,'Parent',subplot4,'Marker','o','LineStyle','none',...
    'Color',[0 0 1]);

% Create xlabel
xlabel('hq','FontSize',16);

% Create ylabel
ylabel('Dq','FontSize',16);

% Create title
title('Multifractal spectrum','FontSize',14);

% Create plot
plot(hqstart2,Dqstart1,'Parent',subplot4,'MarkerFaceColor',[1 0 0],...
    'MarkerEdgeColor',[0 0 0],...
```

```
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[1 0 0],...
    'DisplayName',[strcat('Dq(',num2str(min(q)),') = ',num2str(Dq(q==min(q)))

% Create plot
plot(hqmid2,Dqmid1,'Parent',subplot4,'MarkerFaceColor',[0 0 1],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0 1],...
    'DisplayName',[strcat('Dq(',num2str(q_middle),') = ',num2str(Dq(q==q_mid

% Create plot
plot(hqstop2,Dqstop1,'Parent',subplot4,'MarkerFaceColor',[0 0.498 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0.498 0],...
    'DisplayName',[strcat('Dq(',num2str(max(q)),') = ',num2str(Dq(find(q==ma

% Create legend
%legend1 = legend(subplot1,'show');
%set(legend1,'Position',[0.4174 0.6072 0.09288 0.2055]);
% Create legend
legend1 = legend(subplot1,'show');
set(legend1,'Position',[0.4126 0.6072 0.1024 0.2055]);

% Create legend
legend2 = legend(subplot3,'show');
set(legend2,'Position',[0.3414 0.1576 0.1458 0.1714]);

% Create legend
legend3 = legend(subplot2,'show');
set(legend3,'Position',[0.8171 0.7456 0.1528 0.1714]);
```

```
    % Create legend
    legend4 = legend(subplot4,'show');
    set(legend4,'Position',[0.8338 0.2436 0.1354 0.2764]);

    % Create textbox
    annotation(figure1,'textbox',[0.2726 0.638 0.1111 0.05249],...
        'String',{'Slope = Hq'},...
        'HorizontalAlignment','center',...
        'FontSize',14,...
        'FitBoxToText','off');

    % Create textbox
    annotation(figure1,'textbox',[0.6355 0.1669 0.2013 0.05518],...
        'String',{strcat('hq_m_a_x - hq_m_i_n = ',num2str(max(hq)-min(hq)))},...
        'HorizontalAlignment','center',...
        'FontSize',14,...
        'FitBoxToText','off',...
        'LineStyle','none');

end
```

Appendix 4.

```
function [Ht,Htbin,Ph,Dh] = MFDFA2(signal,scale,m,Fig)
% Multifractal detrended fluctuation analysis (MFDFA). This Matlab function
% estimate the multifractal spectrum Dh directly without q-order statistics.
%
% [Ht,Htbin,Ph,Dh] = MFDFA2(signal,scale,m,Fig);
%
% INPUT PARAMETERS---------------------------------------------------------
%
% signal:        input signal
% scale:         vector of scales
% m:             polynomial order for the detrending
% Fig:           1/0 flag for output plot of F0, Ht, Ph, and Dh.
%
% OUTPUT VARIABLES---------------------------------------------------------
%
% Ht:            Time evolution of the local Hurst exponent
% Htbin:         Bin senters for the histogram based estimation of Ph and Dh
% Ph:            Probability distribution of the local Hurst exponent Ht
% Dh:            Multifractal spectrum
```

```
%
% EXAMPLE————————————————————————————————————————
%
% load fractaldata
% scale=[7,9,11,13,15,17];
% m=2;
% signal1=multifractal;
% signal2=monofractal;
% signal3=whitenoise;
% [Ht1,Htbin1,Ph1,Dh1] = MFDFA2(signal1,scale,m,1);
% [Ht2,Htbin2,Ph2,Dh2] = MFDFA2(signal2,scale,m,1);
% [Ht3,Htbin3,Ph3,Dh3] = MFDFA2(signal3,scale,m,1);
%————————————————————————————————————————————
warning off
X=cumsum(signal-mean(signal));
if min(size(X))~=1||min(size(scale))~=1;
    error('Input arguments signal or scale must be a 1D vector');
end
if size(X,2)==1;
   X=transpose(X);
end
if min(scale)<m+1
   error('The minimum scale must be larger than trend order m+1')
end
disp('--please wait for the results--');

scmin=10;
scmax=length(signal)/10;
scres=20;
exponents=linspace(log2(scmin),log2(scmax),scres);
scale0=round(2.^exponents);

for ns=1:length(scale0),
    segments(ns)=floor(length(X)/scale0(ns));
    for v=1:segments(ns),
                Index0=((((v-1)*scale0(ns))+1):(v*scale0(ns)));
                C0=polyfit(Index0,X(Index0),m);
                fit0=polyval(C0,Index0);
                RMS0{ns}(v)=sqrt(mean((X(Index0)-fit0).^2));
    end
    Fq0(ns)=exp(0.5*mean(log(RMS0{ns}.^2)));
end
```

```
halfmax=floor(max(scale)/2);
Time_index=halfmax+1:length(X)-halfmax;
for ns=1:length(scale),
    halfseg=floor(scale(ns)/2);
    for v=halfmax+1:length(X)-halfmax;
        Index=v-halfseg:v+halfseg;
        C=polyfit(Index,X(Index),m);
        fit=polyval(C,Index);
        RMS{ns}(v)=sqrt(mean((X(Index)-fit).^2));
    end
    F(ns)=exp(0.5*mean(log(RMS{ns}.^2)));
end
C = polyfit(log2(scale0),log2(Fq0),1);
Regfit = polyval(C,log2(scale));
Hq0=C(1);
maxL=length(Time_index);
for ns=1:length(scale);
        RMSt=RMS{ns}(Time_index);
        resRMS=Regfit(ns)-log2(RMSt);
        logscale=log2(maxL)-log2(scale(ns));
    Ht(ns,:)=resRMS./logscale + Hq0;
end
Ht_row=Ht(:);
BinNumb= round(sqrt(length(Ht_row)));
[freq,Htbin]=hist(Ht_row,BinNumb);
Ph=freq./sum(freq);
Ph_norm=Ph./max(Ph);
Dh=1-(log(Ph_norm)./log(mean(diff(Htbin))));

%OUTPUT FIGURE-------------------------------------------------------------------
if Fig==1
   for ns=1:length(scale);
       RMSt1(ns,:)=RMS{ns}(Time_index);
   end
   Ht_mode=mean(Htbin(Ph==max(Ph)));
   numb1=find(Ht(1,:)==max(Ht(1,:)));
   numb2=find(Ht(end,:)==max(Ht(end,:)));
   numb3=find(Ht(1,:)==min(Ht(1,:)));
   numb4=find(Ht(end,:)==min(Ht(end,:)));

   %Figure variables--------------------------
```

```
X1=Time_index;
Y1=signal(Time_index);
X2=Time_index(numb1);
Y2=signal(numb1);
X3=Time_index(numb3);
Y3=signal(numb3);
X4=Time_index(numb2);
Y4=signal(numb2);
X5=Time_index(numb4);
Y5=signal(numb4);
YMatrix2=[Ht(1,:);Ht_mode.*ones(size(Time_index))];
YMatrix3=[Ht(end,:);Ht_mode.*ones(size(Time_index))];
YMatrix4=[log2(RMSt1(1,numb1)),log2(RMSt1(1,numb3))];
YMatrix5=[log2(RMSt1(end,numb2)),log2(RMSt1(end,numb4))];
YMatrix6=[log2(RMSt1(:,1:floor(length(RMSt1)/100):end)),log2(F'),Regfit'];
Y6=Ht(1,numb1);
Y7=Ht(1,numb3);
Y8=Ht(end,numb2);
Y9=Ht(end,numb4);
log2scale=log2(scale);
%-------------------------------------------
for ns=1:length(scale),
    scaletick{ns}=num2str(scale(ns));
end

% Create figure
figure1 = figure('PaperSize',[20.98 29.68],'Color',[1 1 1]);

% Create axes
axes1 = axes('Parent',figure1,'XTickLabel',{'','','','','','','','',''},...
    'Position',[0.08139 0.7673 0.7259 0.1577],...
    'LineWidth',2,...
    'FontSize',11);
hold(axes1,'all');

% Create plot
plot(X1,Y1,'Parent',axes1);

% Create plot
plot(X2,Y2,'Parent',axes1,'MarkerFaceColor',[0 0.498 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
```

```
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0.498 0]);

% Create multiple lines using matrix input to plot
plot(X3,Y3,'Parent',axes1,'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','square',...
    'LineWidth',2,...
    'LineStyle','none',...
    'MarkerFaceColor',[0 0.498 0],...
    'Color',[0 0.498 0]);

% Create plot
plot(X4,Y4,'Parent',axes1,'MarkerFaceColor',[1 0 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[1 0 0]);

plot(X5,Y5,'Parent',axes1,'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','square',...
    'LineWidth',2,...
    'LineStyle','none',...
    'MarkerFaceColor',[1 0 0],...
    'Color',[1 0 0]);

% Create ylabel
ylabel('Amplitude','FontSize',12);

% Create axes
axes2 = axes('Parent',figure1,'XTickLabel',{'','','','','','','','',''},...
    'Position',[0.08139 0.5872 0.7259 0.1577],...
    'LineWidth',2,...
    'FontSize',12);
ylim(axes2,[min(min(Ht))-0.05 max(max(Ht))+0.05]);
hold(axes2,'all');
```

```
% Create multiple lines using matrix input to plot
plot2 = plot(X1,YMatrix2,'Parent',axes2);
set(plot2(1),'DisplayName','Ht');
set(plot2(2),'LineWidth',2,'Color',[0.749 0 0.749],...
    'DisplayName',strcat('mode Ht = ',num2str(Ht_mode)));

% Create plot
plot(X2,Y6,'Parent',axes2,'MarkerFaceColor',[0 0.498 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0.498 0],...
    'DisplayName',strcat('max Ht = ',num2str(max(Ht(1,:)))));

% Create plot
plot(X3,Y7,'Parent',axes2,'MarkerFaceColor',[0 0.498 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','square',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0.498 0],...
    'DisplayName',strcat('min Ht = ',num2str(min(Ht(1,:)))));

% Create ylabel
ylabel('Ht','FontSize',16);

% Create axes
axes3 = axes('Parent',figure1,'Position',[0.08139 0.4045 0.7259 0.1577],...
    'LineWidth',2,...
    'FontSize',12);
ylim(axes3,[min(min(Ht))-0.05 max(max(Ht))+0.05]);
hold(axes3,'all');

% Create multiple lines using matrix input to plot
plot3 = plot(X1,YMatrix3,'Parent',axes3);
set(plot3(1),'DisplayName','Ht');
set(plot3(2),'LineWidth',2,'Color',[0.749 0 0.749],...
    'DisplayName',strcat('mode Ht = ',num2str(Ht_mode)));
```

```
% Create plot
plot(X4,Y8,'Parent',axes3,'MarkerFaceColor',[1 0 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[1 0 0],...
    'DisplayName',strcat('max Ht = ',num2str(max(Ht(end,:)))));

% Create plot
plot(X5,Y9,'Parent',axes3,'MarkerFaceColor',[1 0 0],...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'Marker','square',...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[1 0 0],...
    'DisplayName',strcat('min Ht = ',num2str(min(Ht(end,:)))));

% Create xlabel
xlabel('time (sample number)','FontSize',12);

% Create ylabel
ylabel('Ht','FontSize',16);

% Create axes
axes4 = axes('Parent',figure1,'XTickLabel',scaletick,...
    'XTick',log2scale,...
    'Position',[0.08139 0.06326 0.2134 0.2234],...
    'LineWidth',2,...
    'FontSize',12);
xlim(axes4,[log2scale(1)-0.05 log2scale(end)+0.05]);
hold(axes4,'all');

% Create multiple lines using matrix input to plot
plot4 = plot(log2scale,YMatrix6,'Parent',axes4,'Marker','.','LineStyle','none
    'Color',[0 0 1]);
set(plot4(length(YMatrix6)-1),'MarkerFaceColor',[0.749 0 0.749],'MarkerEdgeC
    'MarkerSize',8,...
    'Marker','o',...
    'LineWidth',2,...
```

```
    'Color',[0.749 0 0.749]);
set(plot4(length(YMatrix6)),'LineWidth',2,'Color',[0.749 0 0.749],'Marker','r
    'LineStyle','-');

% Create multiple lines using matrix input to plot
plot5 = plot(log2scale(1),YMatrix4,'Parent',axes4,'MarkerFaceColor',[0 0.498
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[0 0.498 0]);
set(plot5(1),'Marker','o');
set(plot5(2),'Marker','square');

% Create multiple lines using matrix input to plot
plot6 = plot(log2scale(end),YMatrix5,'Parent',axes4,'MarkerFaceColor',[1 0 0
    'MarkerEdgeColor',[0 0 0],...
    'MarkerSize',8,...
    'LineWidth',2,...
    'LineStyle','none',...
    'Color',[1 0 0]);
set(plot6(1),'Marker','o');
set(plot6(2),'Marker','square');

% Create xlabel
xlabel('scale (segment sample size)','FontWeight','bold','FontSize',12);

% Create ylabel
ylabel('log_2(RMS)','FontSize',14);

% Create title
title('Scaling plot','FontSize',14);

% Create axes
axes5 = axes('Parent',figure1,'Position',[0.383 0.06326 0.2134 0.2234],...
    'LineWidth',2,...
    'FontSize',12);
hold(axes5,'all');

% Create plot
plot(Htbin,Ph,'Parent',axes5,'LineWidth',2);
```

```
% Create plot
plot(mean(Htbin(Ph==max(Ph))).*ones(2,1),[min(Ph),max(Ph)],'Parent',axes5,'Li

% Create xlabel
xlabel('Ht','FontSize',16);

% Create ylabel
ylabel('Ph','FontSize',16);

% Create title
title('Prob. distribution (Ph) of Ht','FontSize',14);

% Create axes
axes6 = axes('Parent',figure1,'Position',[0.6907 0.06326 0.2134 0.2234],...
    'LineWidth',2,...
    'FontSize',12);
hold(axes6,'all');

% Create plot
plot(Htbin,Dh,'Parent',axes6,'LineWidth',2);

% Create plot
plot(mean(Htbin(Ph==max(Ph))).*ones(2,1),[min(Dh(isinf(Dh)==0)),max(Dh)],'Pa

% Create xlabel
xlabel('Ht','FontSize',14);

% Create ylabel
ylabel('Dh','FontSize',16);

% Create title
title('Multifractal spectrum (Dh)','FontSize',14);

% Create legend
legend1 = legend(axes2,'show');
set(legend1,'Position',[0.8215 0.6036 0.1589 0.1175],'LineWidth',2);

% Create legend
legend2 = legend(axes3,'show');
set(legend2,'Position',[0.8215 0.4314 0.1589 0.1175],'LineWidth',2);

% Create textbox
```

```
    annotation(figure1,'textbox',[0.2935 0.9004 0.2342 0.04172],...
        'String',{'Time series'},...
        'FontSize',14,...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure1,'textbox',[0.2953 0.7135 0.2342 0.04172],...
        'String',{strcat('Ht: scale = ',num2str(scale(1)))},...
        'FontSize',14,...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure1,'textbox',[0.2953 0.5307 0.2342 0.04172],...
        'String',{strcat('Ht: scale = ',num2str(scale(end)))},...
        'FontSize',14,...
        'FitBoxToText','off',...
        'LineStyle','none');
end
```

Appendix 5

```
function output1=DFA(DATA,win_length,order)

      N=length(DATA);
      n=floor(N/win_length);
      N1=n*win_length;
      y=zeros(N1,1);
      Yn=zeros(N1,1);

        fitcoef=zeros(n,order+1);
             mean1=mean(DATA(1:N1));
        for i=1:N1
                   y(i)=sum(DATA(1:i)-mean1);
        end
          y=y';
              for j=1:n
                    fitcoef(j,:)=polyfit(1:win_length,y(((j-1)*win_length+1):j*win_len
              end
              for j=1:n
                    Yn(((j-1)*win_length+1):j*win_length)=polyval(fitcoef(j,:),1:win_
```

```
            end

            sum1=sum((y'-Yn).^2)/N1;
            sum1=sqrt(sum1);
            output1=sum1;
end
```

Appendix 6

```
function [d, nobs, tasy, sigasy, tols, sigols] = gph (series, incl, excl)
if nargin < 1
        error('The GPH regression requires a time series input.')
elseif prod(size(series)) == 1
        error('The GPH regression cannot be run on a scalar.')
end

if nargin < 2
        incl = 0.5;
elseif isempty(incl)
        incl = 0.5;
end

if nargin < 3
        excl = 0;
elseif isempty(excl)
        excl = 0;
end

if incl >= 1
        error('The regression can only be run on fewer frequencies than there are
end

if excl < 0
        error('Input argument EXCL cannot be negative.')
end

lincl = length(incl);
lexcl = length(excl);

if lincl > 1 & lexcl > 1
        error('Only one of input arguments INCL and EXCL can be a vector, not bot
elseif sum(excl >= incl)
```

```
        error('There are more frequencies excluded than there are included.')
end

incl(1:lincl*lexcl) = incl;   % If one of EXCL and INCL is a vector, this transfo
excl(1:lincl*lexcl) = excl;   % other into a vector of equal values for each elem

% Take the Discrete Fourier Transform, thus obtaining the complex vector DFT (see
dft     = fft(series(:));
n       = length(dft);
dft(1) = [];

% The following FOR loop runs the GPH regression for each parameter in EXCL or IN
for i = 1 : lincl*lexcl
first(i)= ceil (n^excl(i));
last(i) = floor(n^incl(i));
nobs(i) = last(i) - first(i) + 1;

% ... the Fourier frequencies (harmonic ordinates) of the sample, defined as (see
freq     = 2*pi*(first(i):last(i)) / n;

% ... and the corresponding "power" observations of the periodogram, i.e. the squ
% of the magnitude
of DFT:
power    = abs(dft(first(i):last(i))).^2;

y        = log(power);
X        = [ones(nobs(i),1), log(4*sin(freq'/2).^2)];
beta     = X \ y;

d(i)     = - beta(2);

seasy    = sqrt(diag(inv(X'*X)) * pi^2/6 );
tasy(i) = d(i)/seasy(2);

if nargout > 4
        resid    = y - X*beta;
        seols    = sqrt(diag(inv(X'*X)) * resid'*resid/(nobs(i)-2));
        tols(i) = d(i)/seols(2);
        end
end

if exist('tcdf.m','file')
```

```
        sigasy = min(1-tcdf(tasy, nobs-1), tcdf(tasy, nobs-1))*2;
        if nargout > 4
                sigols = min(1-tcdf(tols, nobs-1), tcdf(tols, nobs-1))*2;
        end
else
        sigasy(1:lincl*lexcl) = NaN;
        if nargout > 4
        sigols(1:lincl*lexcl) = NaN;
        end
end

function dft = sft (series, n)
N      = length(series);
series = series(:);

if length(series) ~= N
        error('This Fourier transformation can only be performed on vector input
end

if nargin == 1
n = N;
end

series(N+1:n) = 0;
dft    (1:n,:) = 0;

% Computation for each element K in DFT (i.e. k from 1 to n):
for k = 1 : n
dft(k) = exp(-i*2*pi*(k-1)*(0:n-1)/n) * series(1:n);
end
```

Appendix 7

```
function  PlotMFDFA(X,l)
% X - time series
% l - length of simple width
N = length(X);
s = floor(N/l);
I = 1:N;
plot(I,X);
hold on;
for v=1:(s+1)
```

```
          if v<s+1
           Index=((v-1)*l+1):(v*l);
          else
           Index=((v-1)*l+1):N;
          end
          C=polyfit(Index,X(Index)',1);
          fit=polyval(C,Index);
          RMS=sqrt(mean((X(Index)-fit').^2));
          plot(Index, fit+(1-1), 'r:');
          plot(Index, fit+(1-1)+RMS, 'r');
          legend('Time series', 'Trend ', '+/- 1 of local RMS', 'Location','northw
          plot(Index, fit+(1-1)-RMS, 'r');
end
          plot(X+1);
          plot(X+2);
          %grid on;
          for m = 2:3
          for v=1:(s+1)
                  if v<s+1
                  Index=((v-1)*l+1):(v*l);
                  else
                  Index=((v-1)*l+1):N;
                  end
                  C=polyfit(Index,X(Index)',m);
                  fit=polyval(C,Index);
                  RMS=sqrt(mean((X(Index)-fit').^2));
                  plot(Index, fit+(m-1)+RMS, 'r');
                  plot(Index, fit+(m-1)-RMS, 'r');
                  plot(Index, fit+(m-1), 'r:');
          end
end;
text(20,2.4,'Linear trend (p=1)');
text(20,3.4,'Quadratic trend (p=2)');
text(20,4.4,'Cubic trend (p=3)');
hold off;
xlabel ('time(sample number)')
ylabel ('values of time series')

end
```

Appendix 8

```
% White Noise
function WN(Data, H, names )
A = Data;

q = round(max(A)-min(A));
[n,m] = size(A);
length = 100;
X = cumsum(randn(q,length));
B=reshape(X',1,q*length);

%% The output noise signal
figure;
subplot(2,1,1)
plot(A);
title(['Share price of the ', names]);
xlabel('Time')
ylabel('Value')

subplot(2,1,2)
plot(1:q*length,B);
title('White noise');
xlabel('Step')
ylabel('Signal')

end

% Brownian Motion

function BM(Data, H, names )
h=H;
t=5000;
n=10;
M=1000;

if (h<0 && h>1) % Error check
    error('Hurst parameter should be lie on the interval [0,1]')
end

k=nan(n+1,1);k(1) = 1;n=M;delta=1:n;
k(delta+1) = 0.5*((delta+1).^(2*h) - 2*delta.^(2*h) + (delta-1).^(2*h));
k=[k; k(end-1:-1:2)]; % the first row of the matrix
lambda=real(fft(k))/(2*n); % eigenvalues
```

```
W=fft(sqrt(lambda).*complex(randn(2*n,1),randn(2*n,1))); %output function
W = n^(-h)*cumsum(real(W(1:n+1))); % scaling
W=t^h*W;
 td=(0:n)/n;
 td=td*t; % scale of time

 figure;
 subplot(2,1,1)
 plot(Data);
 title(['Share price of the ', names]);
 xlabel('Time')
 ylabel('Value')

 subplot(2,1,2)
 plot(td,W);
 title(['Brownian motion with Hurst index = ', num2str(H)]);
 grid on
 xlabel('time')
 ylabel('W_t')

 end


% White noise for Turkey Exchange Rate
 clc
 clear

A = xlsread('TURKEY_DATA.xlsx','EXCHANGE RATE Matlab','B:B');

q = round(max(A)-min(A));
 length = 10000;
X = cumsum(randn(q,length));
B=reshape(X',1,q*length);

% The output noise signal
 plot(1:q*length,B);
 title('White noise');
 xlabel('Step')
 ylabel('Signal')
 grid on

% Brownian motion for Turkey Exchange Rate
```

```
clc
clear

h=0.5;
t=5000;
n=10;
M=1000;

if (h<0 && h>1) % Error check
        error('Hurst parameter should be lie on the interval [0,1]')
end

k=nan(n+1,1);k(1) = 1;n=M;delta=1:n;
k(delta+1) = 0.5*((delta+1).^(2*h) - 2*delta.^(2*h) + (delta-1).^(2*h));
k=[k; k(end-1:-1:2)]; % the first row of the matrix
lambda=real(fft(k))/(2*n); % eigenvalues
W=fft(sqrt(lambda).*complex(randn(2*n,1),randn(2*n,1))); %output function
W = n^(-h)*cumsum(real(W(1:n+1))); % scaling
W=t^h*W;
td=(0:n)/n;
td=td*t; % scale of time
plot(td,W);
title('Brownian motion');
grid on
xlabel('time')
ylabel('W_t')
```

document