МАТЕМАТИЧНІ ТА ІНФОРМАЦІЙНІ МОДЕЛІ В ЕКОНОМІЦІ

УДК 519.246.8

B. KOTYRA, A. KRAJKA

THE ANALYSIS OF WIG20 STOCK INDEX IN R: A CASE STUDY

Анотація. *У представлені роботі коротко наведені методи аналізу часових рядів. Ці методи дозволяють розробити різноманітні моделі часових рядів (розкладання, ARIMA, метод Фур'є, експонентне згладжування та GARCH). Точність отриманих моделей можна перевірити за допомогою нев'язок (невеликі відхилення, стаціонарні, корелювання та некорелювання) або шляхом верифікації прогнозів (це не представлене у даному дописі). Також не розглядаються багато методів інтелектуального аналізу даних, які можуть бути застосовані до фондового індексу часових рядів, наприклад, нейронні мережі та генетичні алгоритми.*
Ключові слова: *R мова, біржове котирування, WIG 20, Фур'є-аналіз, ARIMA, GARCH, CENSUS.*

Аннотация. *В данной работе коротко представлены методы анализа временных рядов. Эти методы позволяют разработать различные модели временных рядов (разложение, ARIMA, метод Фурье, сглаживание по экспоненте и GARCH). Точность полученных моделей можно проверить с помощью невязок (небольшие отклонения, стационарные, коррелированные и некоррелированные) или путем верификации прогнозов (что не будет здесь представлено). Мы опускаем также множество методов интеллектуального анализа данных, которые могут быть применены к фондовому индексу временных рядов, такие как нейронные сети и генетические алгоритмы.*
Ключевые слова: *R язык, биржевые котировки, WIG20, Фурье-анализ, ARIMA, GARCH, CENSUS.*

Abstract. *In this short note we would like to show the basic methods of analyzing time series. This methods leads us to the different models of time series (decomposition, ARIMA, Fourier techniques, exponentially smoothing and GARCH). The correctness of the models obtained may be verified by behavior of residuals (small variance, stationary, uncorrelated, normally distributes) or by verifying the predictions. This second method not will be discussed here. We omit the lot of data mining methods, which may be applied to the stock index time series, such as neural networks and genetic algorithms.*
Keywords: *R language, stock quotes, WIG20, Fourier analysis, ARIMA, GARCH, CENSUS.*

Introduction

Having been invited to the first number of Mathematical Modeling in the Economy journal we would like to present the classical basic sequence of proceeding with time series taken from Polish WIG20 stock index. The observation was taken from the server BOŚ: ftp.bossa.pl. We work with quotations running every 30 minutes which are given in ASCII file. We present here the decomposition on trend and seasonal term ([3]), the ARIMA model ([1, 15]), Fourier transformation

techniques ([16, 17]), exponential smoothing techniques ([11, 8, 10, 9]) and GARCH models ([7, 4])

We investigate these data using the R language environment, a widely used free environment for statistical and data mining analysis. We think, that the R environment is the best tool for statistical and data mining modeling process. R is available from the url: http://cran.r-project.org. The installation procedure is intuitive and easy.

There are different methods to prepare data for use in R, but the easiest one is to prepare data in a spreadsheet in Excel, save this spreadsheet as CSV-file (cf. Table 1) and import by the following command read.csv("D:/wig20_m30.csv", header=T, dec=",", sep=";").

Table 1 – Data taken to analysis

DATE	TIME	OPEN	HIGH	LOW	CLOSE
20001117	103000	1614	1623	1614	1623
20001117	110000	1623	1627	1623	1624
20001117	113000	1624	1628	1622	1628
20001117	120000	1628	1631	1624	1624
20001117	123000	1624	1630	1624	1629
20001117	130000	1629	1634	1629	1633

In the R language we import the required libraries, show the direction with data ("D:/Data"), import a CSV format file "wig20_m30.csv" and transform this OPEN column into time series object (library ts). The frequency 3555 was taken from computations of the average number of observations in every year.

Listing 1 – Introduction

```
library(quadprog)
library(zoo)
library(tseries)
library(forecast)
library(FinTS)
library(fGarch)
library(e1071)
library(nortest)
library(MASS)
setwd("D:/Data")
dane<-read.table("wig20_30.csv", sep = ";", header = T)
dd < −ts(dane$OPEN, start=1, freq=3555)
```

1. Data transformation

It is known, that operations of logarithm on time series and differentiations (we replace the given series $\{x_k, k \geq 1\}$ on $\{\log(x_k) - \log(x_{k-1}), k \geq 1\}$ putting $x_o = 1$, eliminate autocorrelations and nonstationarity of wide class time series. Here and in what follows $\log(x) = \log_e(x)$ denotes the natural logarithm of x.

Another method of elimination of autocorrelations is Box-Cox transformation $\{f_\lambda(x_k), k \geq 1\}$ where function f is defined by

$$f_\lambda(x) = \begin{cases} \frac{x^\lambda - 1}{\lambda}, & \text{if } \lambda > 0, \\ \log(x), & \text{if } \lambda = 0. \end{cases}$$

The following fragment of R-code (Listing 2) presents, how we should choose λ in the Box-Cox transformation and how we should evaluate the degree of differentiations.

Listing 2 – Data transformation

```
par(mfcol = c(2,1))
boxcox(dd~time(dd))
boxcox(dd~time(dd), lambda=seq(-0.1, 0.5, 0.01))
lambda<-0.2

adf.test(dd, alternative="stationary")$p.value
kpss.test(dd)
adf.test(log(dd), alternative="stationary")$p.value
kpss.test(log(dd))
adf.test(diff(log(dd), differences=1), altern="stationary")$p.value
kpss.test(diff(log(dd), difference=1))
dif1<-diff(log(dd), differences=1)

dif2<-(dd^lambda-1)/lambda
adf.test(dif2, alternative="stationary")$p.value
kpss.test(dif2)
adf.test(log(dif2), alternative="stationary")$p.value
kpss.test(log(dif2))
adf.test(diff(log(dif2), differences=1), altern="stationary")$p.value
kpss.test(diff(dif2, differences=1))
dif2<-diff(log(dif2), differences=1)
```

Analyzing the graph of functions, in order to find maximum, we see that $\lambda = 0.2$ is the good choice for our time series (cf. Figure 1).

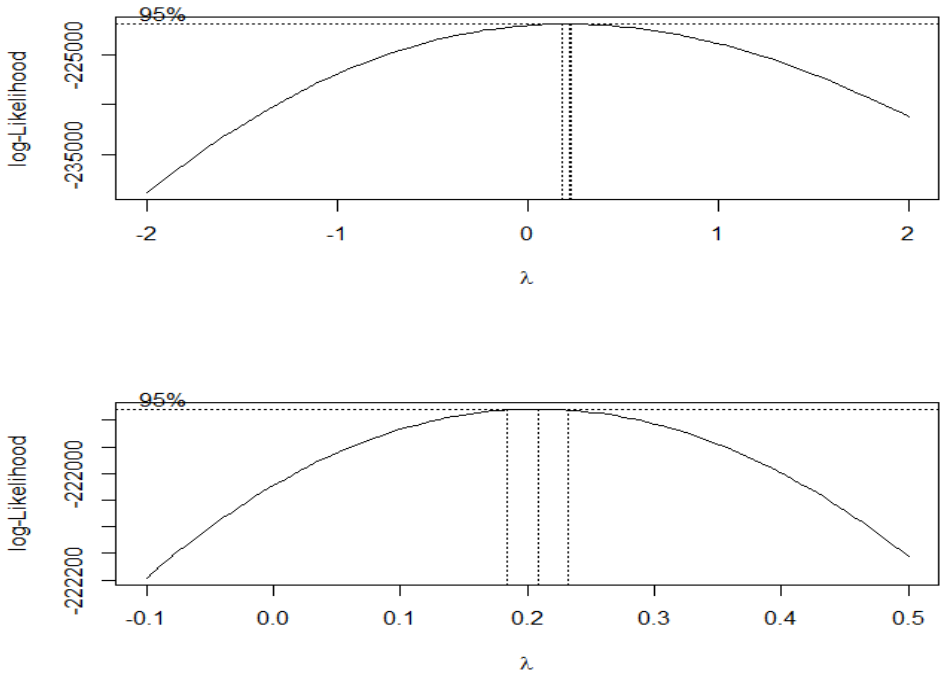

Figure 1 – Searching λ for Box-Cox transformation

For the differentiation purpose we use two statistics:

- Augmented Dickey-Fuller (ADF) is a test for a unit root in a time series sample (H_o - the series has unit root),
- Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for the evaluation stationarity (H_o - the series is nonstationary).

Taken the significance level $\alpha = 0.05$ we accept the hypothesis that time series is stationary and time series has not unit root, when computed p in ADF test is less than α and in KPSS test is greater than α. We accept such transformation. The sequential values of ADF and KPSS tests are presented in Table 2.

Table 2 – The result of ADF and KPSS tests

Transformation	ADF	KPSS
dd	0.7201	0.01
log(dd)	0.7058	0.01
diff(log(dd), difference=1)	0.01	0.1
dif2	0.7162	0.01
log(dif2)	0.7020	<0.05
diff(log(dif2, difference=1)	0.01	0.1

Thus we see that transformations:

$$dif1_k = \log(x_k) - \log(x_{k-1}),$$

$$dif2_k = \log\left(\frac{x_k^{0.2} - 1}{0.2}\right) - \log\left(\frac{x_{k-1}^{0.2} - 1}{0.2}\right), k = 2,3,\ldots$$

are possible best. All further computations in R were made both for dd time series as well as for dif1 and dif2 time series, although we describe here the results which deal with dd series, only, because the others turned out to be similar.

2. Census decomposition

The classical approach to time series is the decomposition of time series on the trend, seasonal fluctuations and the rest. The decomposition is corrected, when the rest is behavior as "white noise", i.e. it is normally distributed, uncorrelated, stationary with the possible small standard deviation. In the R library stats the seasonal decomposition is produced by the command stl. The procedure is described in [3]. Listing 3 produces three stl objects with the period 3555.

Listing 3 – Census

```
res0<-stl(dd, s.window='periodic')
res1<-stl(dif1, s.window='periodic')
res2<-stl(dif2, s.window='periodic')
```

The Figure 2, presenting the decomposition terms, is produced by the command plot(res0).

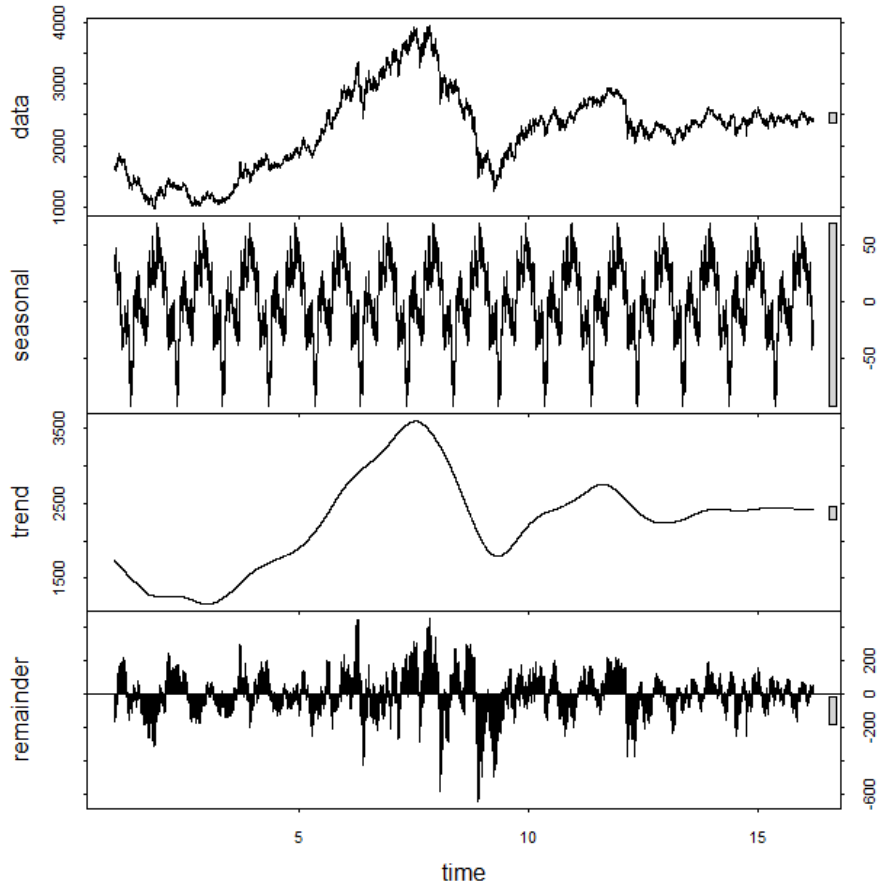

Figure 2 – Census decomposition

The problem how to choose the best period for the seasonal component is discussed in section 5.

3. ARIMA

Model ARIMA(p,d,q) assumes that the time series $\{y_t, t \geq 1\}$ has forms

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \cdots \theta_q B^q) e_t \quad (1)$$

where $c, \{\phi - i, 1 \leq i \leq p\}, \{\theta_i, 1 \leq i \leq q\}$ are real numbers and B denotes the differentiation operator, i.e. $Bx_t = x_t - x_{t-1}, B^2 x_t = BX_t - BX_{t-1} = x_t - 2x_{t-1} + x_{t-2}, \ldots$ Here and in what follows $\{e_t, t \geq 1\}$ denotes the sequence of independent identically distributed random variables drawn from standard normal distribution. The parameters p, d, q are chosen as a result of the analysis of Listing 4.

Listing 4 – ARIMA

```
wynAR=data.frame(i=c(0), d=c(0), j=c(0), AIC=c(0))
for (i in 0:3) {
for (d in 0:3) {
for (j in 0:3) {
  if (i+d+j>0) {
    wynAR<-rbind(wynAR, c(i,d,j,
      Arima(dd, order=c(i,d,j))$aicc))
}}}}
```

We choose the methods with the minimal AIC (Akaike information criterion) coefficient. For any statistical model, the AIC value is

$$AIC = 2k - 2log(L),$$

where k is the number of parameters in the model, and L is the maximized value of the likelihood function for the model. AIC stands for the compromise between complexity and quality of the model. Lower coefficients than 38527 are summarized in Table 3.

Table 3 – AIC coefficient

i	d	j	AIC
2	0	3	385256.9
2	0	2	385263.1
0	1	1	385268.1
0	1	1	385268.1
1	1	0	385268.4
1	1	0	385268.4
2	1	0	385269.5
0	1	2	385269.5
0	1	2	385269.5

3	0	2	385269.6

This leads us to the model AR0<-arima(dd, order=c(2,0,3)) and similarly

Listing 5 – ARIMA

```
AR0<-arima(dd, order=c(2,0,3))$residuals
AR1<-arima(dif1, order=c(0,0,1))$residuals
AR2<-arima(dif2, order=c(0,0,1))$residuals
```

The ARIMA model obtained for the dd has computed coefficients $ar_1 = 0.0593, ar_2 = 0.9402, ma_1 = 0.9150, ma_2 = -0.0282, ma_3 = 0.0039, c = 2245.9831$ thus the series $\{x_t, t \geq 1\}$ is approximated by

$$(1 - 0.0593B - 0.9402B^2)y_t = 2245.9831 + (1 + 0.9150B - 0.0282B^2 + 0.0039B^3)e_t$$

or equivalently

$$\begin{aligned} 0.0005y_t = \\ = 2245.9831 - 1.9397y_{t-1} + 0.9402y_{t-2} + 11.8868e_t - 0.8703e_{t-1} - 0.0399e_{t-2} \\ - 0.0039e_{t-3}. \end{aligned} \tag{2}$$

Equation (2) allows us to compute sequential values of $\{y_t, t \geq 3\}$ assuming knowledge $y_1 = x_1, y_2 = x_2$. The results of ARIMA approximations may be observed in Figure 3 produced by tsdiag command.

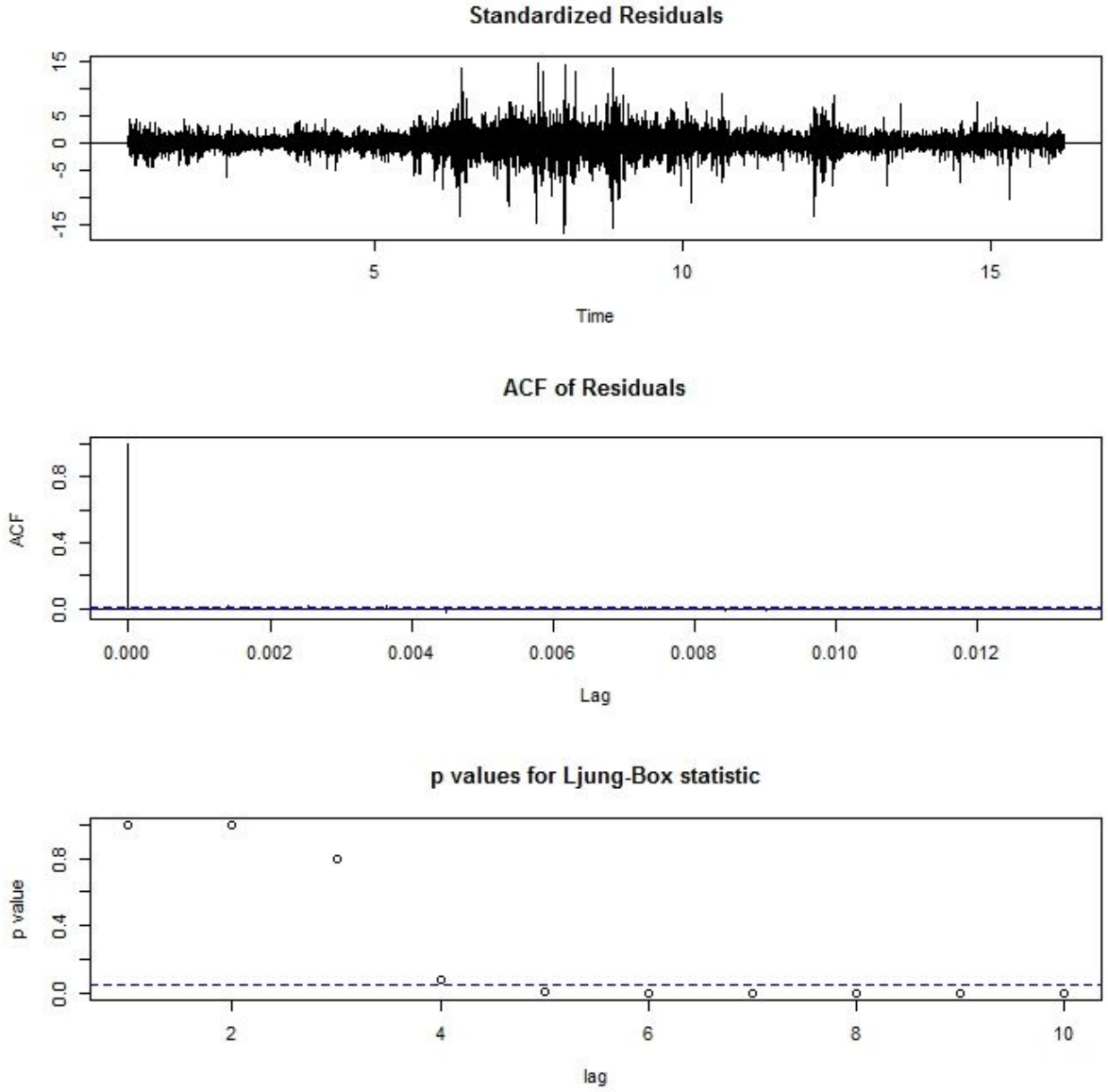

Figure 3 – ARIMA(2,0,3) model

The acf and similar pacf command product the values of autocorrelations and partial autocorrelations of residuals of models. The presented here figure of acf is good. Box-Ljung test (cf. [2,13]) investigates whether any of a group of autocorrelations of a time series are different from zero (null hypothesis - all autocorrelations equal 0). In our result the autocorrelations of one two and three order are zero, the remaining ones are nonequal. This problem may be inappropriately chosen period.

In order to choose the parameters of the ARIMA model we may use the command auto.arima(dd) too.

4. Fourier analysis - identification of season

A Fourier Transform converts a wave from the time domain into the frequency domain. Formally, it maps the sequence $\{x_k, 1 \leq k \leq N\}$ into the complex sequence $\{X_k, 1 \leq k \leq N\}$ by the formula:

$$X_{k+1} = \sum_{n=0}^{N-1} x_{n+1} e^{-2\pi i k n / N}, 1 \leq k \leq N,$$

where $i = \sqrt{-1}$. The inverse mapping may be counted by

$$x_{k+1} = \sum_{n=0}^{N-1} X_{n+1} e^{2\pi i k n/N}, 1 \le k \le N.$$

To perform Fourier Transform in R we use fft(x) command whereas the inverse transformation may be obtained by fft(X, inverse=TRUE)/length(X) because the inverse series is non normalized. The big values of $|X_k|$ suggest the existence of period k. If we have some values of periods $k_1, k_2, \ldots k_l$, say, then the true period is equal lowest common multiply of numbers $k_1, k_2, \ldots k_l$. Therefore we write in R functions gcd, lcm, Gcd, Lcm - the great common divisor and lowest common multiply two (gcd, lcm) and arbitrary (Gcd, Lcm) real numbers. Function TFrq makes Fast Fourier Transformation, sorts obtained modulus of complex numbers in a decreasing order and packs all to returned data frame.

Listing 6 – Fourier procedures

```
gcd <- function(a,b) ifelse (b==0, a, gcd(b, a %% b))
lcm <- function(a,b) ifelse (b==0, 0, a*b/gcd(b, a %% b))
Gcd <- function(a) {
      La<-rep(a[1],length(a))
      for (l in 2:length(a)) La[l]<-gcd(La[l-1], a[l])
      return(La)
                }
Lcm <- function(a) {
      La<-rep(a[1],length(a))
      for (l in 2:length(a)) {
          La[l]<-La[l-1]*a[l]/gcd(La[l-1], a[l]) }
      return(La)
                }
TFrq <- function(danet,k,l) {
  N<-length(danet)
  XTv<-fft(danet)
  XT<-Mod(XTv)
  XT1<-time(XT)
  YT<-sort(as.numeric(XT), index.return=T, decreasing=T)
  YTv<-YT$x[YT$ix<N/l][1:k]
  YTn<-YT$ix[YT$ix<N/l][1:k]
  if (k==1) return(data.frame("valuesMOD"=YTv[1],"values"=XTv[YTn[1]],
           "numbers"=XT1[YTn[1]],"num"=YTn[1],"NWW"=YTn[1])) else  {
          return(data.frame("valuesMOD"=YTv,"values"=XTv[YTn],
            "numbers"=XT1[YTn],"num"=YTn,"NWW"=Lcm(YTn))) }
                  }
```

Using TFrq(dd,10,2) and observed obtained results allows us to make a conclusion, that good period for dd is 60 whereas for dif1 and dif2 is 15264. We correct the freq option in definitions of time series and repeat Census procedure.

Listing 7 – Fourier analysis

```
dd1<-ts(dd,freq=60)
dif1<-ts(dif1, freq=15264)
dif2<-ts(dif2, freq=15264)
CENF0<-stl(dd1, s.window='periodic')$time.series[,3]
CENF1<-stl(dif1, s.window='periodic')$time.series[,3]
CENF2<-stl(dif2, s.window='periodic')$time.series[,3]
```

5. Exponential smoothing methods

There are other methods of decomposition of time series. In the library forecast there is described class ets which allows us to do exponential smoothing state space model (cf. [11, 8, 10]) At first, however, we must evaluate three-character string identifying method. The first letter denotes the error type ("A", "M" or "Z"); the second letter denotes the trend type ("N","A","M" or "Z"); and the third letter denotes the season type ("N","A","M" or "Z"). In all cases, "N"=none, "A"=additive, "M"=multiplicative and "Z"=automatically selected. So, for example, "ANN" is simple exponential smoothing with additive errors, "MAM" is multiplicative Holt-Winters' method with multiplicative errors, and so on. If parameter damped is TRUE, we use a damped trend (either additive or multiplicative). The ets without parameters with except time series allows us to choose the better model. The following session produces the best models for our time series

Listing 8 – ETS model

```
ets(dd)    # model M,Md,N
ets(dif1)   # model A,N,N
ets(dif2)   # model A,Ad,N
mod1<-ets(dd, model="MMN", damped=TRUE)
mod2<-ets(dif1, model="ANN", damped=FALSE)
mod3<-ets(dif2, model="AAN", damped=TRUE)
EXP0<-mod1$residuals
EXP1<-mod2$residuals
EXP2<-mod3$residuals
```

In computations there arises a problem with seasonality, which should be smaller than 24 (in our examples are greater). In consequence the elimination of seasonality was omitted. The decomposition we observe on diagram by plot (mod1) (cf. Figure 4) whereas the basic diagnostic may be obtained by tsdiag (mod1).

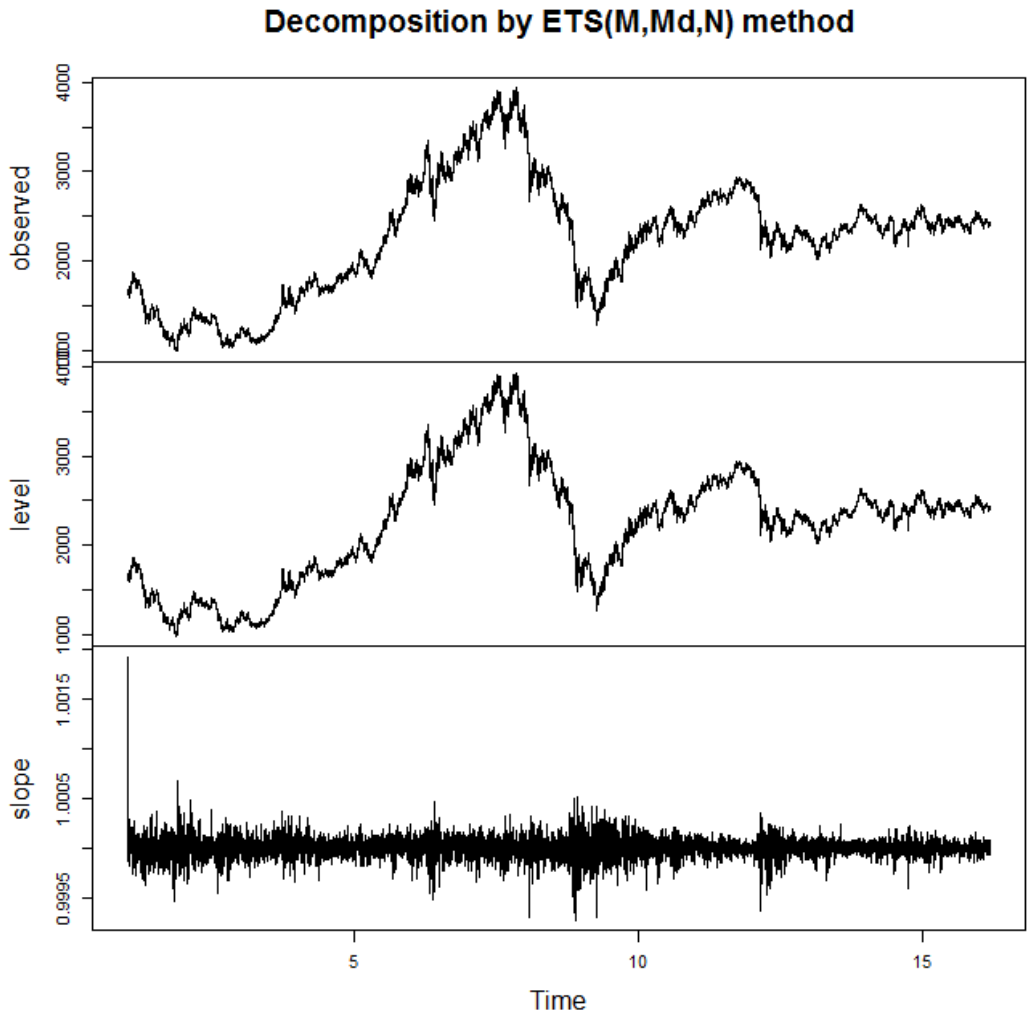

Figure 4 – ETS model

6. Comparison

To sum up we have four models: CENSUS (CEN), ARIMA (AR), CENSUS with period obtained by Fourier analysis (CENF) and exponential smoothing model (EXP). By commands (cf. Listing 9)

Listing 9 – Comparison

```
plot_colors <- c("blue","red","green","yellow")
max_y<-max(max(CEN0), max(AR0), max(CENF0), max(EXP0))
par(mfcol=c(2,2))
plot(CEN0, type="o", col=plot_colors[1])
plot(AR0, type="o", pch=22, lty=2, col=plot_colors[2])

plot(CENF0, type="o", pch=23, lty=3, col=plot_colors[3])
plot(EXP0, type="o", pch=4, lty=5, col=plot_colors[4])
```

we produce the residuals of considered methods (see Figure 5).

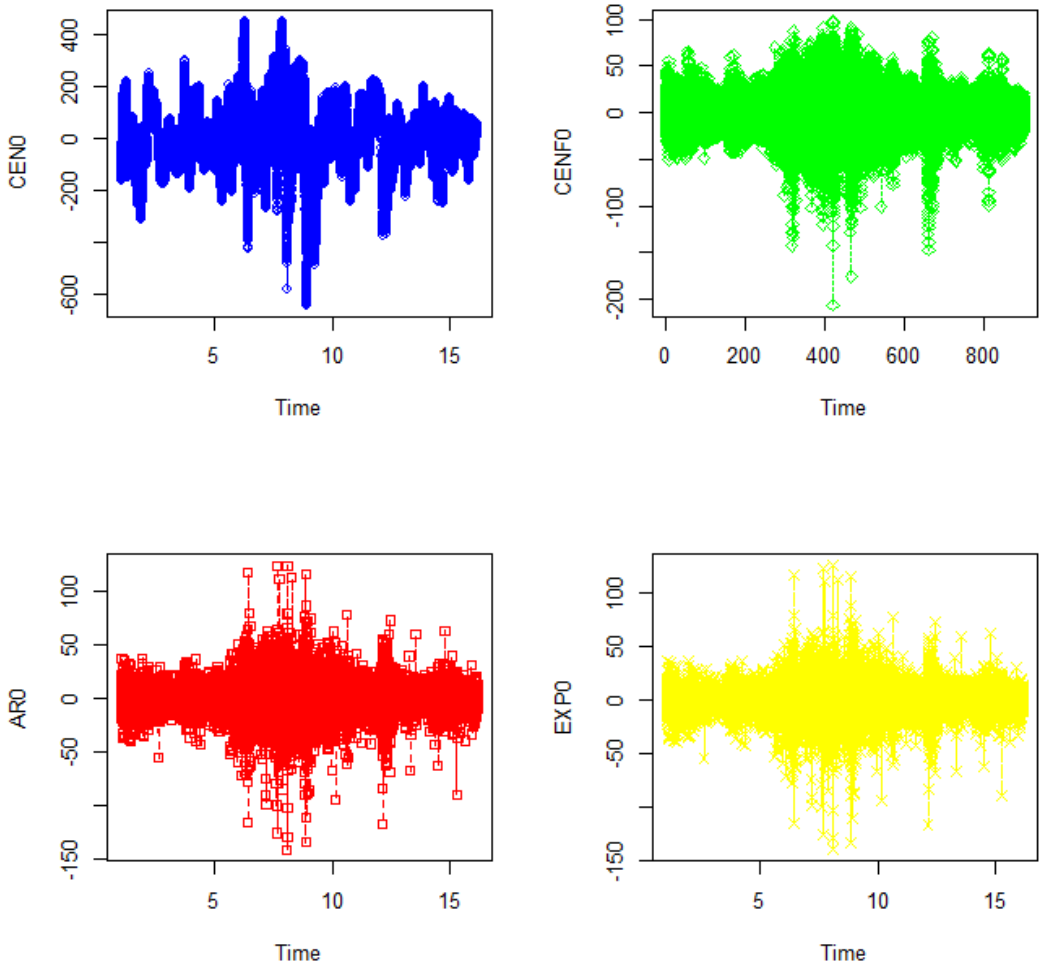

Figure 5 – Comparison residuals of CEN, AR, CENF and EXP methods

Additionally the tests of Shapiro-Wilks and Jarque-Bera show that all residuals are not normal. The analysis of standard deviations of residuals (120.6,8.6, 18.5,8.6 for CEN, AR, CENF, EXP, respectively) leads us to conclusion that ARIMA i EXP methods are better than both CENSUS. But all methods don't work "well". The tests results are bad.

7. GARCH models

It seems that the reason for the conclusion of previous section is volatility clustering Volatility clustering — the phenomenon of there being periods of relative calm and periods of high volatility — is a seemingly universal attribute of market data. There is no universally accepted explanation of it. GARCH (Generalized AutoRegressive Conditional Heteroskedasticity) models volatility clustering. It does not explain it. GARCH (m,r) model assumes that the process $\{y_t, t \geq 1\}$ satisfies

$$y_t = \sigma_t e_t,$$

$$\sigma_t^2 = \alpha_o + \sum_{i=1}^{m} \alpha_i y_{t-i}^2 + \sum_{i=1}^{r} \beta_j \sigma_{t-i}^2$$

for $t = m+1, m+2, \ldots$. Contrary to ARIMA and EXP models we assume that variance of process $\{y_t, t \geq 1\}$ is not constants and dependent of time. We denote this variance by σ_t^2. In R we ask if the time series has GARCH structure by ArchTest(dd) with null hypothesis that considered series have not GARCH structure. For all the three considered time series $dd, dif1, dif2$ we get $p - value < 2.2 * 10^{-16}$ such that we reject null hypothesis and use GARCH methods for modeling WIG behaviour. In the library FGarch we use garch.fit. We write function to test opitmal parameters for GARCH:

Listing 10 – GARCH parameters testing

```
fGAR <- function(danet) {
  gwyn<-data.frame(Ni=c(0), Nj=c(0), NAIC=c(0.0), NBIC=c(0.0),
      NSIC=c(0.0), NHQIC=c(0.0), BLjungX=c(0.0), BLdf=c(0),
      BLpvalue=c(0.0), ShaX=c(0.0), Shapvalue=c(0.0),
      ADX=c(0.0), ADpvalue=c(0.0), LillX=c(0.0), Lillpvalue=c(0.0),
      SFX=c(0.0), SFpvalue=c(0.0), SD=c(0.0))
for (i in 1:4) {
for (j in 0:6) {
     form<-as.formula(paste("danet~garch(",i,",",j,")",sep=""))
          wd<-garchFit(form)
     wd1<-AutocorTest(residuals(wd))
          wm1<-shapiro.test(residuals(wd))
          wm2<-ad.test(residuals(wd))
          wm3<-lillie.test(residuals(wd))
          wm4<-sf.test(residuals(wd))
     gwyn<-rbind(gwyn, c(i,j,wd@fit$ics[[1]],wd@fit$ics[[2]],
       wd@fit$ics[[3]],wd@fit$ics[[4]],wd1$statistic[[1]],
       wd1$parameter[[1]], wd1$p.value, wm1$statistic[[1]],
       wm1$p.value, wm2$statistic[[1]], wm2$p.value,
       wm3$statistic[[1]], wm3$p.value, wm4$statistic[[1]],
       wm4$p.value, sd(residuals(wd))))
      }}
   return(gwyn)
    }
```

obtaining the best GARCH models garch(1,3) for all the three series. Creating GARCH models and using method summary we see that in the case of time series dd the residuals are not normal (Shapiro-Wilk test and Jarque-Bera test) but all autocorrelations are equal to zero (Ljung-Box test). The model is as follows:

$$y_t = -0.0038 + 0.1906 y_{t-1} + \sigma_t e_t,$$

$$\sigma_t^2 = 0.0113 + 0.1906 e_{t-1}^2 + 0.3882 \sigma_{t-1}^2 + 0.3765 \sigma_{t-3}^2,$$

and by method plot we may obtain the 13 diagnostic plots (for eg. cf. Figure 6).

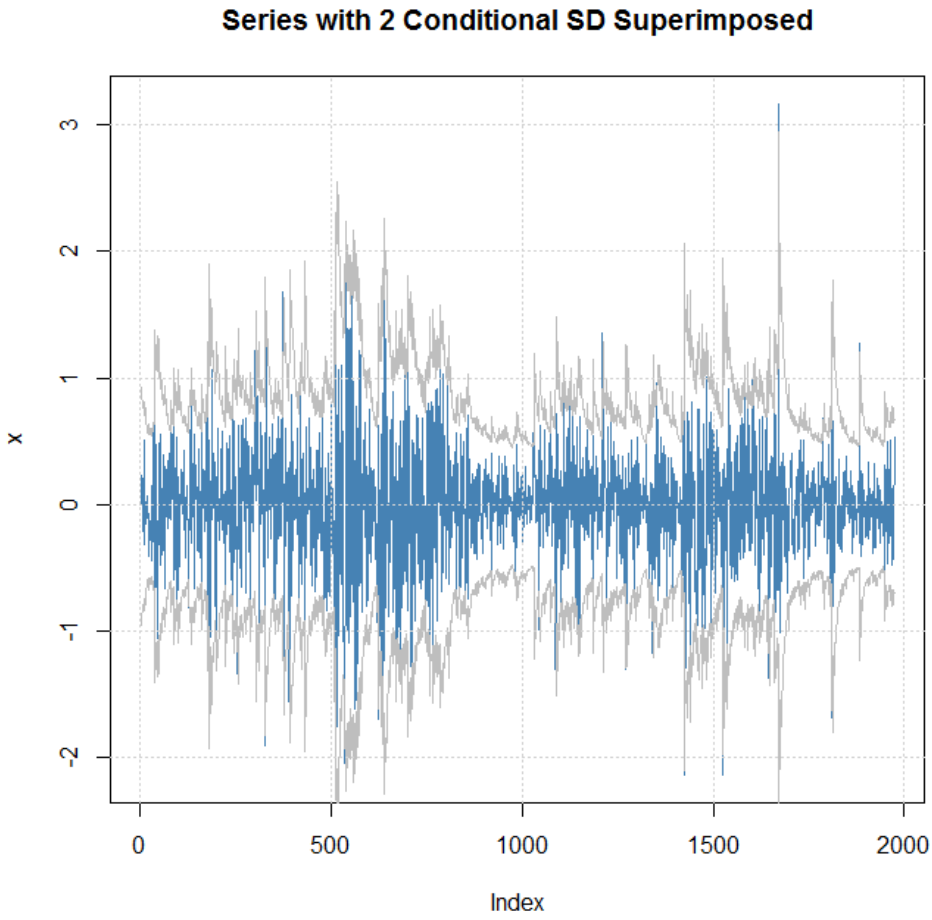

Figure 6 – GARCH(1,3) model

REFERENCES

1. Asteriou, D., Hall, S.G. (2011). ARIMA Models and the Box-Jenkins Methodology. Applied Econometrics (Second ed.). Palgrave MacMillan.
2. Box, G. E. P. and Pierce, D. A. (1970), Distribution of residual correlations in autoregressive-integrated moving average time series models. Journal of the American Statistical Association, 1509 – 1526.
3. Cleveland R.B., Cleveland W.S., McRae J.E., Terpenning I. (1990), STL: A Seasonal-Trend Decomposition Procedure Based on Loess, Journal of Official Statistics, Vol. 6, No. 1., 3 – 73.
4. Engle R.F. (1982), Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. Econometrica. (4), s. 987 – 1007.
5. Handbook of Financial Time Series, T.G. Andersen, R.A. Davis, J-P. Kreiss and T. Mikosh (eds), Springer, New York 2009.
6. Harvey, A. C. Time Series Models. 2nd Edition, Harvester Wheatsheaf, NY, 1993.
7. Harvey A., Ruiz E., Shephard N. (1994), Multivariate Stochastic Variance Models, Review of Economic Studies, , 247–264.
8. Hyndman, R.J., Akram, Md., and Archibald, B. (2008) "The admissible parameter space for exponential smoothing models". Annals of Statistical Mathematics, 60(2), 407 – 426.
9. Hyndman R.J., Khandakar Y., Automatic Time Series Forecasting: The forecast Package for R, Journal of Statistical Software, July 2008, Volume 27, Issue 3. http://www.jstatsoft.org/
10. Hyndman, R.J., Koehler, A.B., Ord, J.K., and Snyder, R.D. Forecasting with exponential smoothing: the state space approach, Springer-Verlag 2008. http://www.exponentialsmoothing.net.

11. Hyndman, R.J., Koehler, A.B., Snyder, R.D., and Grose, S. (2002), A state space framework for automatic forecasting using exponential smoothing methods, International J. Forecasting, (3), 439 – 454.
12. Kleiber Ch., Zeileis A., Applied Econometrics with R, 2008 Springer Science+Business Media LNC.
13. Ljung, G. M. and Box, G. E. P. (1978), On a measure of lack of fit in time series models. Biometrika , 297–303.
14. McLeod A. I., Hao Yu, Krougly Z.L., Algorithms for Linear Time Series Analysis: With R Package, Journal of Statistical Software, December 2007, Volume 23, Issue 5. http://www.jstatsoft.org/
15. Mills, T.C. Time Series Techniques for Economists. Cambridge University Press 1990.
16. Percival, D, B., Walden, A.T. Spectral Analysis for Physical Applications. Cambridge University Press 1993.
17. Walker J.S. Fast Fourier Transforms, CRC Press, Second ed. 1996.

Стаття надійшла до редакції 01.12.2014