Time series and Prediction

Analysis and forecast of monthly stock prices of CD Projekt S.A.

by Nikola Chmielewska

Abstract

In this project my main goal is to use time series of interest to analyze and forecast. I have chosen monthly stock prices of a Polish video game developer, publisher and distributor CD Projekt S.A. Firstly, I will introduce data and tell a little bit about important dates which were affecting this time series. Later I will conclude if this time series can be considered as a representation of stationary process. Next I will analyze correlogram and also use automatic criteria in order to identify test models. Then I will estimate those models using the training set and also check them, that is establish if residuals of those models can be considered as a realization of white noise process and also I will check whether the residuals are normally distributed or not. Finally I will forecast using those models and I will compare them to the real values of the test set. At the end I will summarize my observations.

1 Introducing the data

Let's first have a look at our data. This series is containing monthly stock prices of CD Projekt S.A. from January 2015 to December 2020. The currency of those prices is in Polish currency PLN. PLN to EUR converter is 0.22 on the day 20 January 2021.

Figure 1: Monthly stock prices of CD Projekt S.A from January 2015 to December 2020

The company was founded in May 1994 by Marcin Iwiński and Michał Kiciński, but it debuted only in 2011 on the stock market as a result of collaboration with Optimus (the computer business which was on a verge of bankruptcy). The years before 2015 were

not the best for the company and the prices remained practically at the same very low value (7.00PLN). That's why I am considering this time series starting from January 2015, because this is the most interesting period to analyze.

On May 2015 the company released *The Witcher 3: Wild Hunt* based on the book series made by Andrzej Sapkowski. But only after one year the company began to record growth. So what happened later? In May 2017 Netflix officially declared new series taking place in *The Witcher* world. The announcement of the series triggered an increase in the company's stock prices, while after the announcement of the official cast (in October 2018), prices began to fall. It was related to the controversy surrounding the actors who, according to the fans, did not fit into the world created by Andrzej Sapkowski. However, after the series premiere, which was in December 2019, the company's shares were still going up. In the December 2020 after releasing the most important game that was awaited by the whole gaming world, *Cyberpunk 2077*, prices fell sharply. Widespread criticism concerning technical underdevelopment, that caused numerous errors, despite many pre-release assurances that a full-fledged product will be available on the market, caused the shake throughout investors.

The plot below contains those important dates.

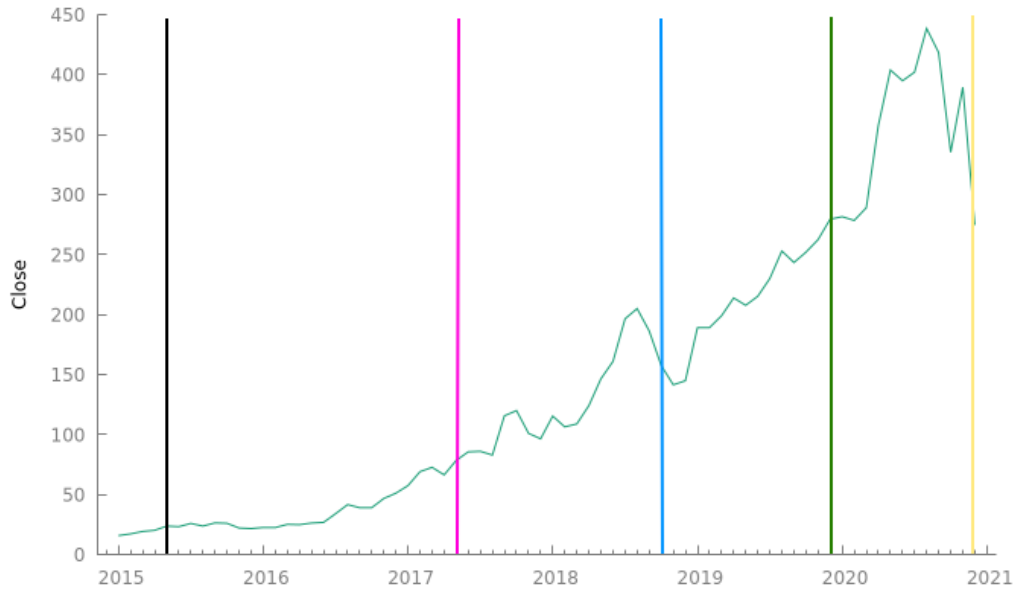

Figure 2: Monthly stock prices of CD Projekt S.A from January 2015 to December 2020, with important dates marked by vertical lines

Let's now concentrate on analyzing our time series using tools that I learned during the course. Firstly, I will consider whether this time series is a representation of a stationary process or not.

2 Stationary process or not?

Looking at the plot of our time series we can conclude that this clearly is not a representation of a stationary process. There is a trend and also by looking at the correlogram of this time series we can see that.

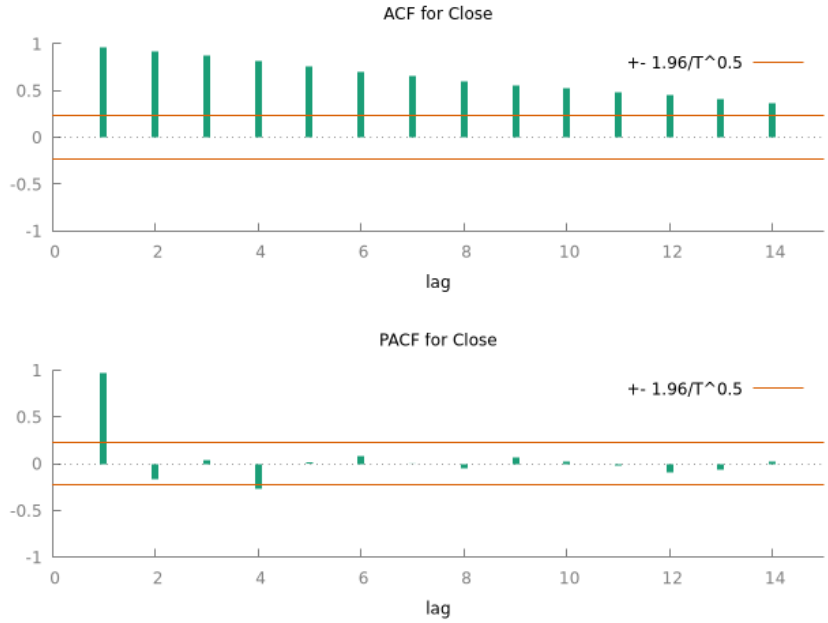

Figure 3: Correlogram of CD Projekt's S.A. stock prices

So now what we need to do is to transform it in a way to get the stationary process. I have decided to consider this time series as a seasonal time series because during forecast I obtained better forecasts when I removed seasonal component. My first attempt to obtain stationary process was using first difference of my observed values and then the seasonal difference. But although the expected value was 0, the variance was not constant. The plot below shows this result.

Figure 4: Transformed time series using first and seasonal differences

Clearly, in the above plot we can see that the variance is increasing over time. That's why now I am using log transformation to stabilize it and then both first and seasonal differences.

Figure 5: Transformed time series using log transformation, first and seasonal differences

In this situation we can see that the mean is equal to 0 and we can assume that the variance is more or less constant. That's why now, after all the transformations, this time series can be considered as a representation of a stationary process. So now we will consider correlogram of that transformed time series and also we will use automatic criteria in order to conclude test models.

3 Correlograms and Automatic criteria

Let's now have a look at the information of our transformed data.

```
Autocorrelation function for sd_d_l_Close
***, **, * indicate significance at the 1%, 5%, 10% levels
using standard error 1/T^0.5
  LAG      ACF          PACF         Q-stat. [p-value]
    1   0.0652        0.0652          0.2637  [0.608]
    2  -0.2569  **   -0.2623 **       4.4323  [0.109]
    3   0.0629        0.1091          4.6864  [0.196]
    4   0.0650       -0.0205          4.9629  [0.291]
    5  -0.2978  **   -0.2815 **      10.8752  [0.054]
    6  -0.0097        0.0680         10.8817  [0.092]
    7   0.2490  *     0.1125         15.1732  [0.034]
    8  -0.0468       -0.0589         15.3276  [0.053]
    9  -0.0450        0.0847         15.4735  [0.079]
   10   0.2310  *     0.1369         19.3945  [0.036]
   11   0.0096       -0.0357         19.4014  [0.054]
   12  -0.5190  ***  -0.4177 ***     40.0302  [0.000]
   13  -0.1521       -0.1414         41.8395  [0.000]
   14   0.1280       -0.0743         43.1493  [0.000]
```

And now on the correlogram of our transformed data.

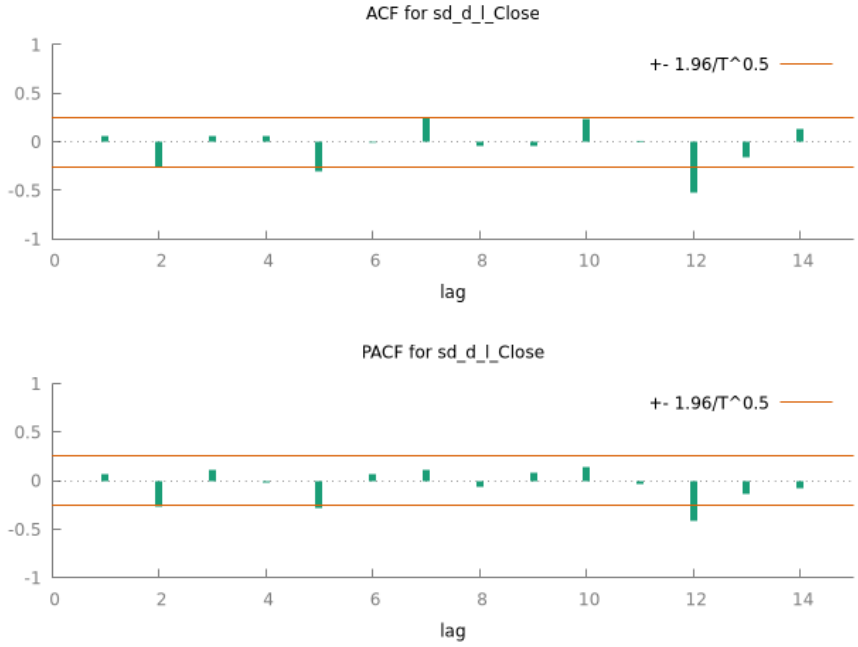

Figure 6: Correlogram of the transformed time series

Looking at this information we need to think a little which models will be appropriate to consider later. Let's look at partial auto-correlation function. We can see that the second coefficient is slightly different from zero. It can make sense to consider SARIMA$(0,1,2) \times (0,1,2)$. Let's look at what automatic criteria is saying.

```
? armax(3, 3, sd_d_l_Close, null, 0, 1, 0, 1, 0)
armax: successfully-1:p=0,q=0
================================================
 Information Criteria of ARMAX(p,q) for sd_d_l_Close
------------------------------------------------
 p, q          AIC           BIC           HQC
------------------------------------------------
 0, 0     -38.9057      -34.7506      -37.2837
 0, 1     -39.5207      -35.3656      -37.8987
 0, 2     -41.9442      -35.7116*     -39.5113
 0, 3     -40.1252      -31.8150      -36.8812
 1, 0     -39.0943      -34.9392      -37.4723
 1, 1     -39.5967      -33.3641      -37.1637
 1, 2     -40.0416      -31.7315      -36.7977
 1, 3     -40.2753      -29.8877      -36.2204
 2, 0     -41.7908      -35.5582      -39.3579
 2, 1     -40.8240      -32.5138      -37.5800
 2, 2     -44.3323*     -33.9447      -40.2774*
================================================
* indicates best models.
'9999.9999' suggests failures to estimate the models.
```

From what we can see I was right. So the models that I will consider will be SARIMA$(0,1,2)\times(0,1,2)$ and also SARIMA$(2,1,2)\times(2,1,2)$. Having in mind airline model I will also consider SARIMA$(0,1,1)\times(0,1,1)$.

4 Estimating and checking models

Firstly, let's notice that we have 72 observations which will be divided into two sets: the training set and the test set. The training set will contain 62 observations and to the test set will belong last 10 observations. Now I will estimate those models using the training set which contains dates from January 2015 to February 2020.

I will start my considerations with analyzing SARIMA$(0,1,2)\times(0,1,2)$.

4.1 SARIMA$(0,1,2)\times(0,1,2)$

By creating the model SARIMA$(0,1,2)\times(0,1,2)$ from the training set we are getting the following:

```
Function evaluations: 45
Evaluations of gradient: 17
Model 1: ARIMA, using observations 2016:02-2020:02 (T = 49)
Estimated using AS 197 (exact ML)
Dependent variable: (1-L)(1-Ls) l_Close
Standard errors based on Hessian
             coefficient   std. error      z      p-value
  ---------------------------------------------------------
  theta_1     0.0569287     0.140937     0.4039   0.6863
  theta_2    -0.243550      0.133694    -1.822    0.0685  *
  Theta_1    -1.23167       0.335676    -3.669    0.0002  ***
  Theta_2     0.513726      0.324166     1.585    0.1130
Mean dependent var   0.000659   S.D. dependent var   0.169314
Mean of innovations -0.000759   S.D. of innovations  0.090452
R-squared            0.986538   Adjusted R-squared   0.985641
Log-likelihood       38.18390   Akaike criterion    -66.36780
Schwarz criterion   -56.90870   Hannan-Quinn        -62.77903
                        Real  Imaginary    Modulus  Frequency
  -----------------------------------------------------------
  MA
    Root  1          -1.9128     0.0000     1.9128     0.5000
    Root  2           2.1465     0.0000     2.1465     0.0000
  MA (seasonal)
    Root  1           1.1988    -0.7138     1.3952    -0.0855
    Root  2           1.1988     0.7138     1.3952     0.0855
  -----------------------------------------------------------
```

Looking at p-value we can see that coefficient Theta_1 is significantly different from zero. Now let's have a look at the correlogram of residuals to consider if residuals can be a representation of white noise process.

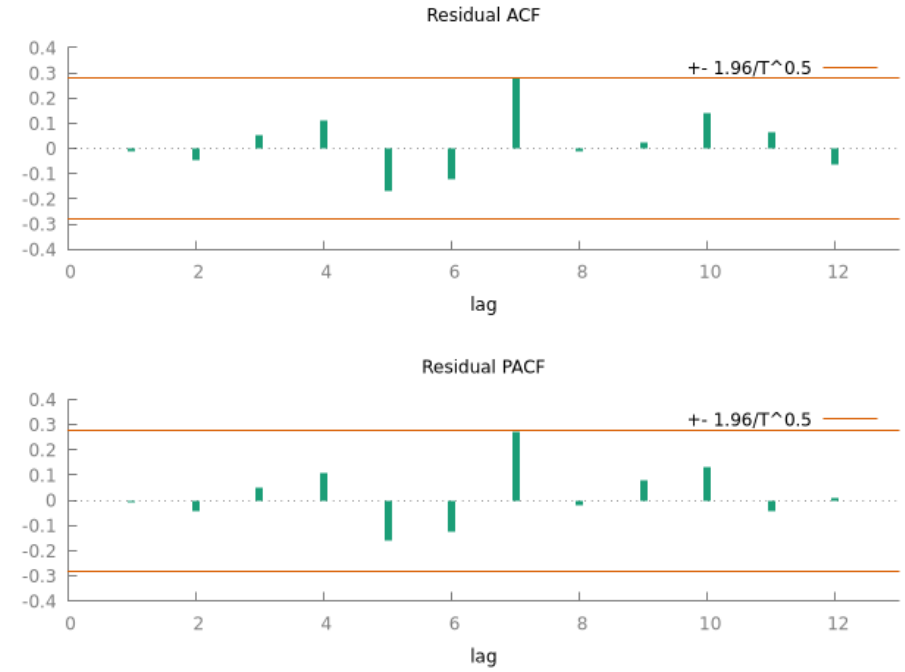

Figure 7: Correlogram of the residuals of SARIMA$(0,1,2) \times (0,1,2)$

We can consider residuals to be a representation of white noise process because we can't see coefficients that are significantly different from zero. Let's also test if the residuals are normally distributed.

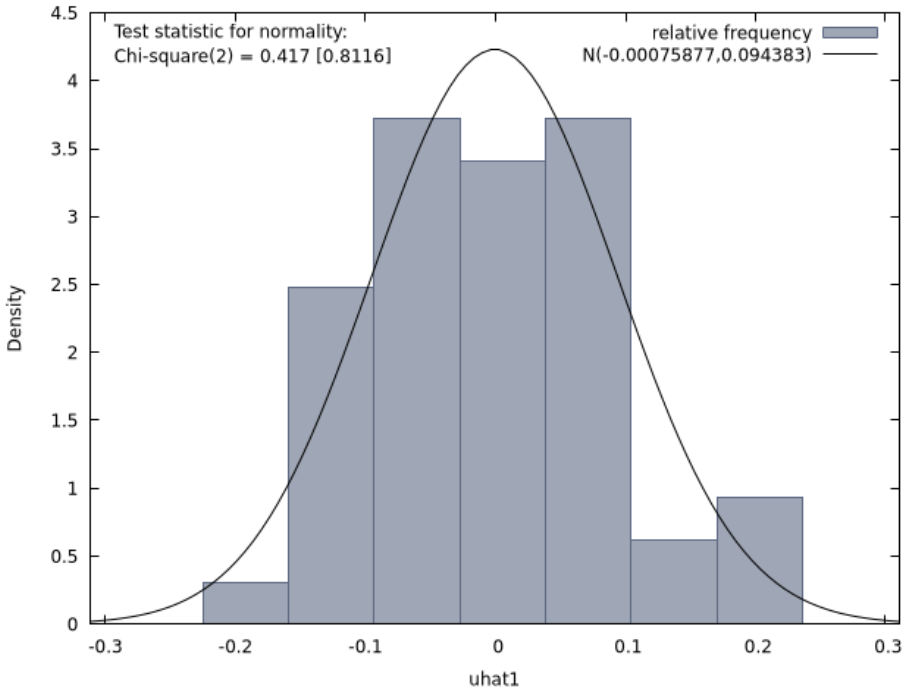

Figure 8: Normality of the residuals of SARIMA$(0,1,2) \times (0,1,2)$

From this histogram we can conclude that the residuals are normally distributed. Then

we can conclude that the residuals can be representation of Gaussian white noise process. So we checked assumptions that we needed to check to conclude if our model is correct. Now let's try to forecast.

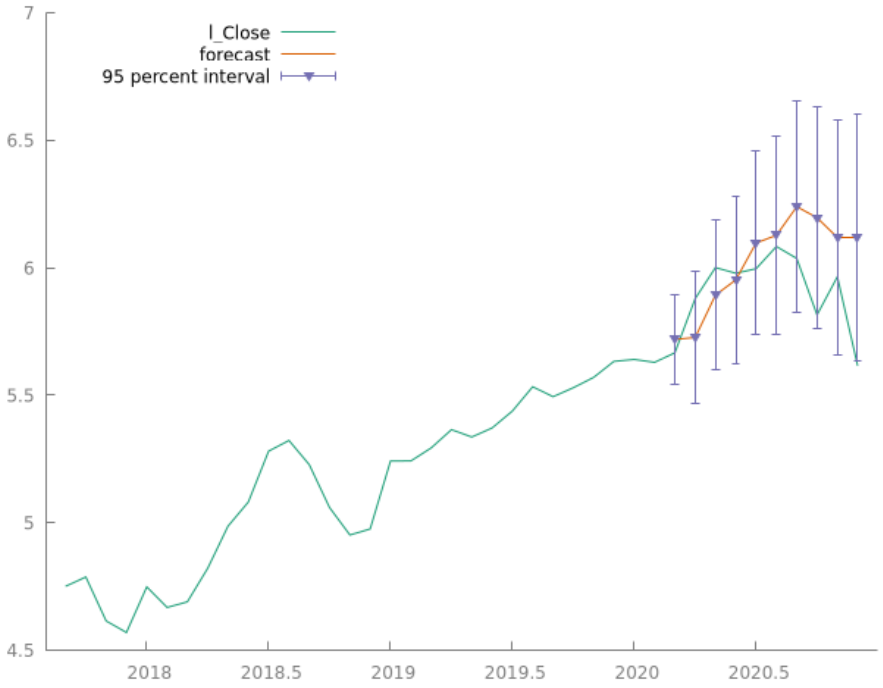

Figure 9: Forecast of SARIMA$(0,1,2) \times (0,1,2)$

We can see that this model is quite good. It didn't forecast too well but looking at the past values this forecasting is making sense. It predicted the drop down of stock prices at the moment of the biggest drop. Let's remember that Root Mean Squared Error in this case is equal to 0.22675.

4.2 SARIMA$(2,1,2) \times (2,1,2)$

Now we are considering SARIMA$(2,1,2) \times (2,1,2)$ and obtaining the following result.

```
Function evaluations: 130
Evaluations of gradient: 53
Model 2: ARIMA, using observations 2016:02-2020:02 (T = 49)
Estimated using AS 197 (exact ML)
Dependent variable: (1-L)(1-Ls) l_Close
Standard errors based on Hessian
             coefficient   std. error      z      p-value
  ---------------------------------------------------------
  phi_1       -0.477691     0.278712    -1.714    0.0865   *
  phi_2       -0.771382     0.170199    -4.532    5.84e-06 ***
  Phi_1       -0.146975     0.787533    -0.1866   0.8520
  Phi_2       -0.254412     0.561173    -0.4534   0.6503
  theta_1      0.578431     0.383737     1.507    0.1317
  theta_2      0.603307     0.265772     2.270    0.0232   **
```

```
  Theta_1     -1.11350      1.51098     -0.7369    0.4612
  Theta_2      0.999994     2.12631      0.4703    0.6381
Mean dependent var   0.000659   S.D. dependent var   0.169314
Mean of innovations -0.001223   S.D. of innovations  0.074471
R-squared            0.990771   Adjusted R-squared   0.989195
Log-likelihood       40.13797   Akaike criterion    -62.27594
Schwarz criterion   -45.24955   Hannan-Quinn        -55.81615
                        Real  Imaginary    Modulus  Frequency
  ------------------------------------------------------------
  AR
    Root  1          -0.3096    -1.0957     1.1386    -0.2938
    Root  2          -0.3096     1.0957     1.1386     0.2938
  AR (seasonal)
    Root  1          -0.2889    -1.9614     1.9826    -0.2733
    Root  2          -0.2889     1.9614     1.9826     0.2733
  MA
    Root  1          -0.4794    -1.1949     1.2875    -0.3107
    Root  2          -0.4794     1.1949     1.2875     0.3107
  MA (seasonal)
    Root  1           0.5568    -0.8307     1.0000    -0.1560
    Root  2           0.5568     0.8307     1.0000     0.1560
  ------------------------------------------------------------
```

We can see that most of the coefficients are not significantly different from zero. Let's now look at the correlogram of residuals.

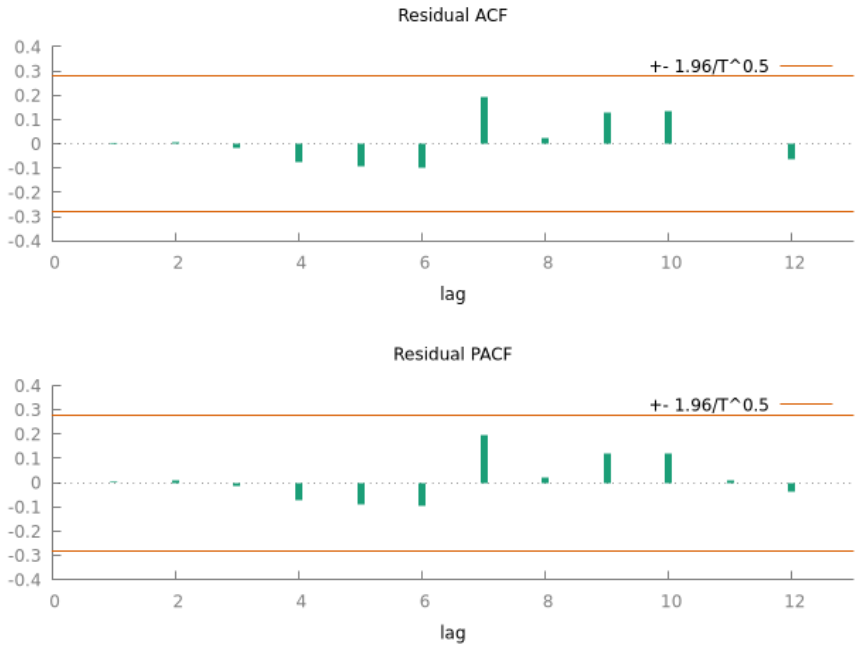

Figure 10: Correlogram of the residuals of SARIMA$(2,1,2) \times (2,1,2)$

We can conclude that they are representation of white noise process, because there is no residual that is significantly different from zero.

Let's check normality of the residuals.

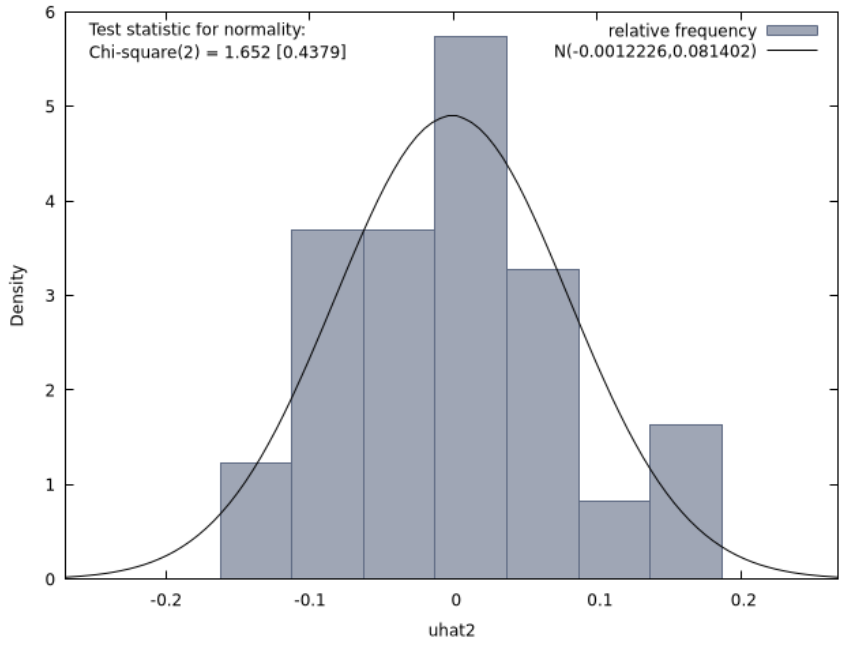

Figure 11: Normality of the residuals of SARIMA$(0, 1, 2) \times (0, 1, 2)$

We can also see that those residuals are normally distributed (we can conclude that looking at p-value) so they are representing Gaussian white noise process. Now we use forecasting tools to get:

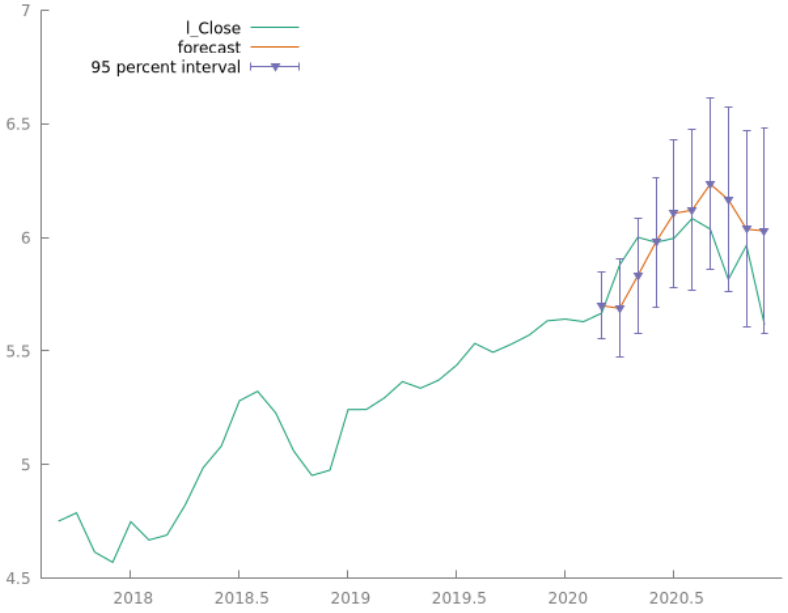

Figure 12: Forecast of SARIMA$(2, 1, 2) \times (2, 1, 2)$

And we see that it again predicted the fall of the prices as the model before. Still not good, but satisfactory. Let's keep in mind that Root Mean Squared Error equals 0.20448 (a bit better than the previous).

4.3 SARIMA$(0,1,1) \times (0,1,1)$

Finally let's consider now our last model, which is SARIMA$(0,1,1) \times (0,1,1)$. This is our result:

```
Function evaluations: 26
Evaluations of gradient: 11
Model 3: ARIMA, using observations 2016:02-2020:02 (T = 49)
Estimated using AS 197 (exact ML)
Dependent variable: (1-L)(1-Ls) l_Close
Standard errors based on Hessian
             coefficient   std. error      z      p-value
  ---------------------------------------------------------
  theta_1      0.217942     0.233655     0.9328   0.3509
  Theta_1     -1.00000      0.232788    -4.296    1.74e-05 ***
Mean dependent var   0.000659   S.D. dependent var   0.169314
Mean of innovations  0.000932   S.D. of innovations  0.100673
R-squared            0.982838   Adjusted R-squared   0.982472
Log-likelihood       33.19778   Akaike criterion    -60.39556
Schwarz criterion   -54.72010   Hannan-Quinn        -58.24230
                        Real  Imaginary    Modulus  Frequency
  -----------------------------------------------------------
  MA
    Root  1          -4.5884     0.0000     4.5884     0.5000
  MA (seasonal)
    Root  1           1.0000     0.0000     1.0000     0.0000
  -----------------------------------------------------------
```

We can see that only one parameter, Theta_1, is significantly different from zero. We are seeing it looking at p-value. Let's also give a look at the correlogram of the residuals.

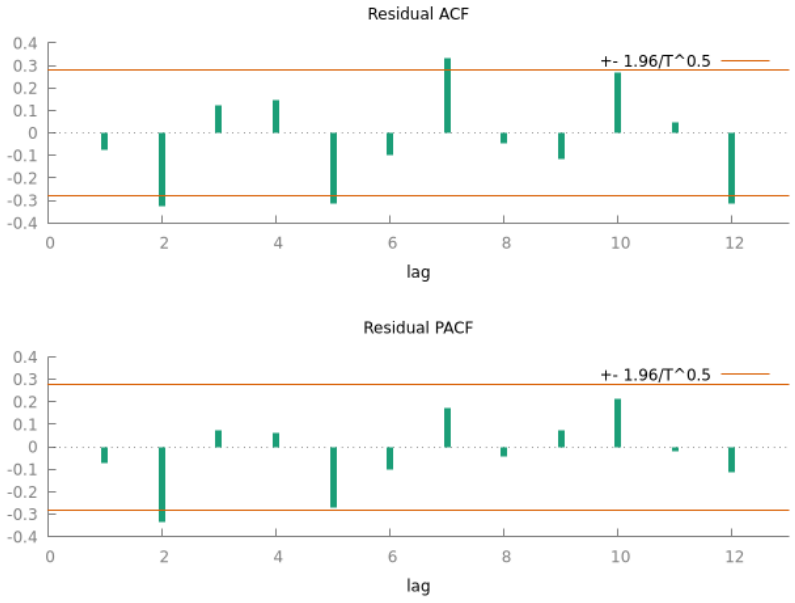

Figure 13: Correlogram of the residuals of SARIMA$(0,1,1) \times (0,1,1)$

In the correlogram we can see that there are coefficients which are significantly different from zero. So we probably can't say that the residuals of this model can be considered as a realization of a white noise process. Let's also have a look at histogram to check normality of the residuals.

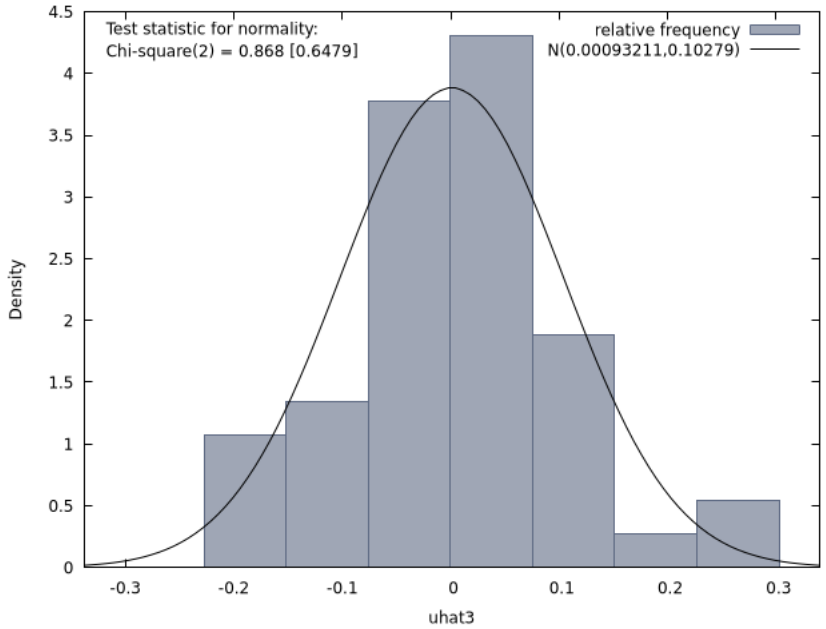

Figure 14: Normality of the residuals of SARIMA$(0, 1, 1) \times (0, 1, 1)$

We can see that p-value, equal to 0.6479, is saying to us that the residuals are normally distributed. But anyway let's look at a forecast.

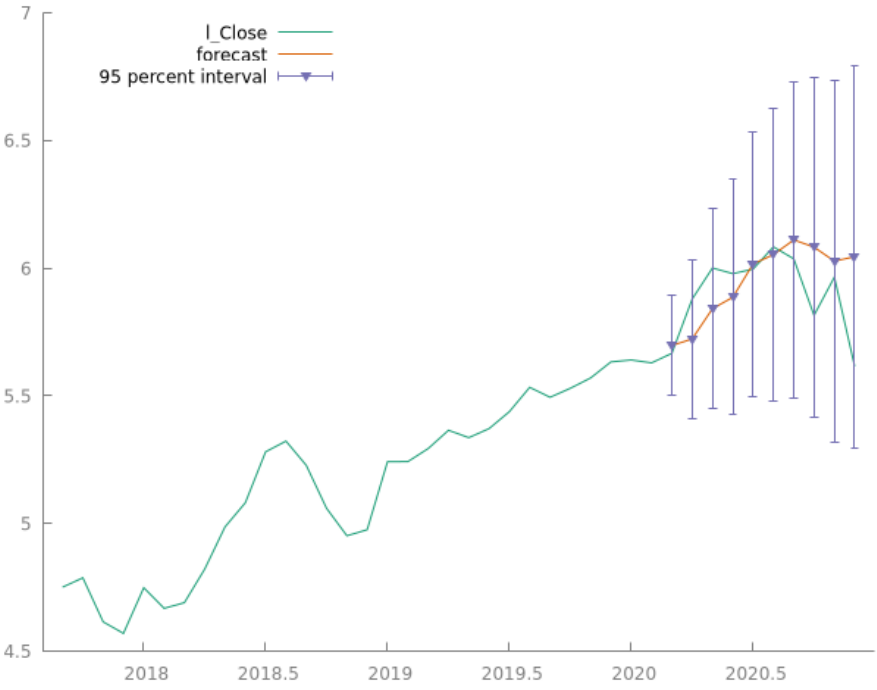

Figure 15: Forecast of SARIMA$(0, 1, 1) \times (0, 1, 1)$

It looks like this one is the best. There is a moment when the predicted values are practically the same as the real ones. Also Root Mean Squared Error is equal to 0.1805.

5 Summarize

From the analysis we did, we get very interesting results that will be presented here. First of all let's compare the plots of our forecasts.

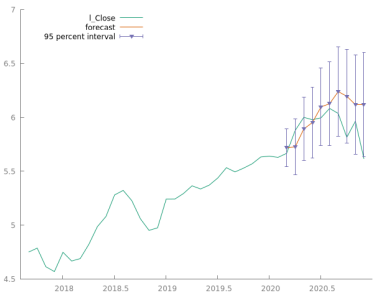

Figure 16: Forecast of SARIMA$(0,1,2) \times (0,1,2)$

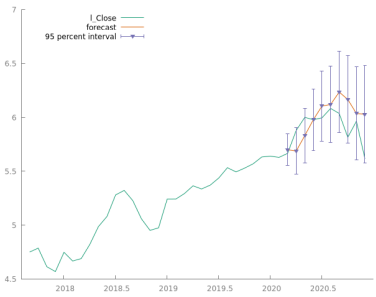

Figure 17: Forecast of SARIMA$(2,1,2) \times (2,1,2)$

Figure 18: Forecast of SARIMA$(0,1,1) \times (0,1,1)$

We can see that the first two models are really alike, because they forecast in a very similar way. The third model is the best because at some point the predicted values are practically the same as real stock prices. Let's also see the comparison of Root Mean Squared Error. The result is presented in the following table.

	Root Mean Squared Error
$SARIMA(0,1,2) \times (0,1,2)$	0.22675
$SARIMA(2,1,2) \times (2,1,2)$	0.20448
$SARIMA(0,1,1) \times (0,1,1)$	0.1805

Looking at the plots and taking the minimum of presented Root Mean Squared Errors, we can conclude that the best model is SARIMA$(0,1,1) \times (0,1,1)$, though the correlogram of the residuals is not truly presenting a realization of a white noise process. The next best model is $SARIMA(2,1,2) \times (2,1,2)$. Moreover the residuals of this model are realization of a white noise process.