**ARIMA-GARCH HW**

*1.a) Compute and plot the log price xt and the log return rt . Comment on the two plots (how volatile the data are, volatility clustering, outliers etc)*

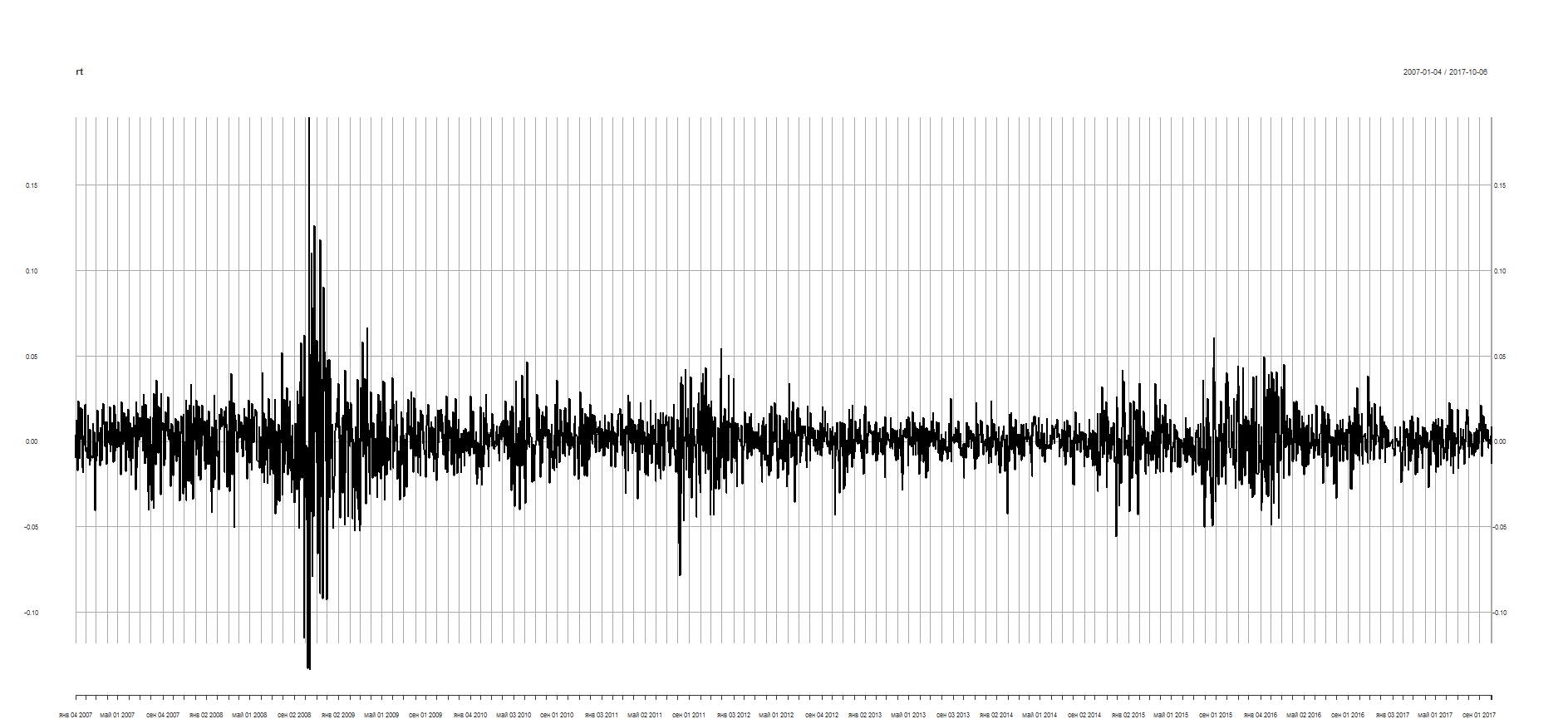
I used a daily data from Yahoo Finance for Chevron Corporation. Dataset contains for stock prices from 2007-01-03 up to now. Dataset checked for NA-values and it not found – it contains 16266 observations. At image 1 shown log-price. As shown at plot price grew from 3.9 to 4.8. Price have 2 price drops – in 2008-2009 and 2015. But can be observed a steady growth.

Log-returns shown at image 2. According plot, the most volatile period was in two periods: 2008-2009 and 2015. The biggest and smallest return was at the end of 2008. Nowadays returns is not high volatile.

*Image 1. Log-price*



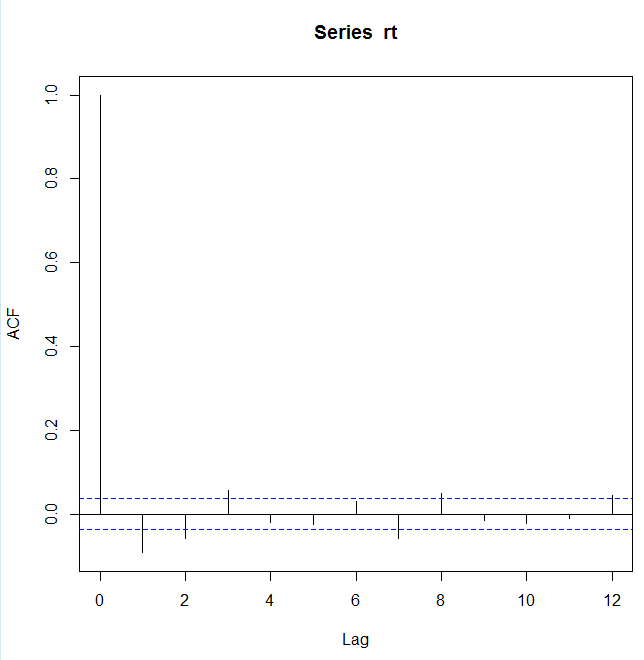
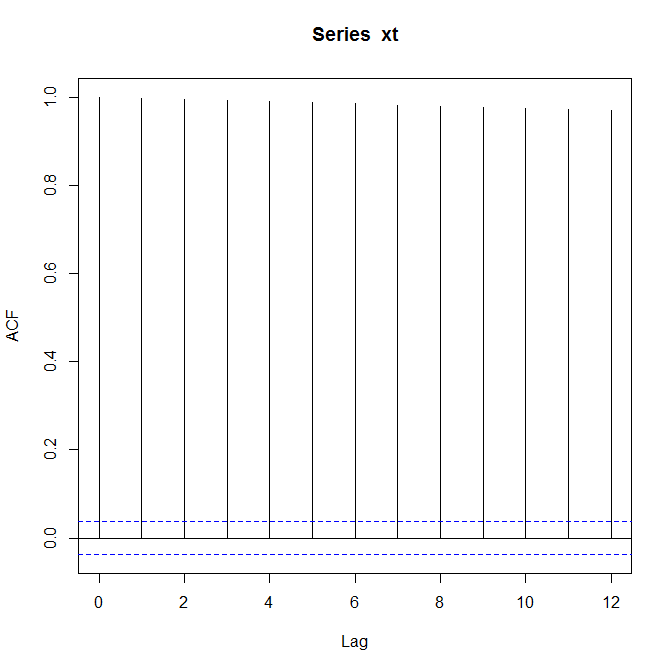
*Image 2. Log-return*



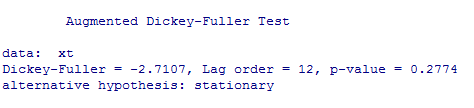
*1.b) Compute and plot the first 12 lags of ACF of xt . Comment on the plot. Based on the ACF, is there a unit root in xt dataset? Why?*

According image 3, the dataset has a unit root, because first lags equal to 1. All 12 lags is confidence. At image 3 also shown ACF for log-returns – according it, dataset have 3 significant lags (because it higher than significance interval). According plot, xt is not stationary time series.

*Image 3. ACF for xt and rt*



The ADF test with lag = 12 and drift has a test statistic -2.7107 with p-value 0.2774. The unit-root hypothesis (unit root is present in a time series sample) cannot be rejected at the 5% level.



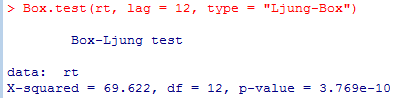
*1.c) Consider the time series for rt. Perform the Ljung-Box test for m = 12. Draw a conclusion and justify it with the statistical language, i.e., in terms of the critical region or p-value.*

H0: the correlations in the population from which the sample is taken are 0.

H1: the data are not independently distributed; the exhibit serial correlation.

p.value < 0.05 and we should reject the H0. It means that some coefficient in AR-model is not equal 0.

*Image 4. Box-Ljung*

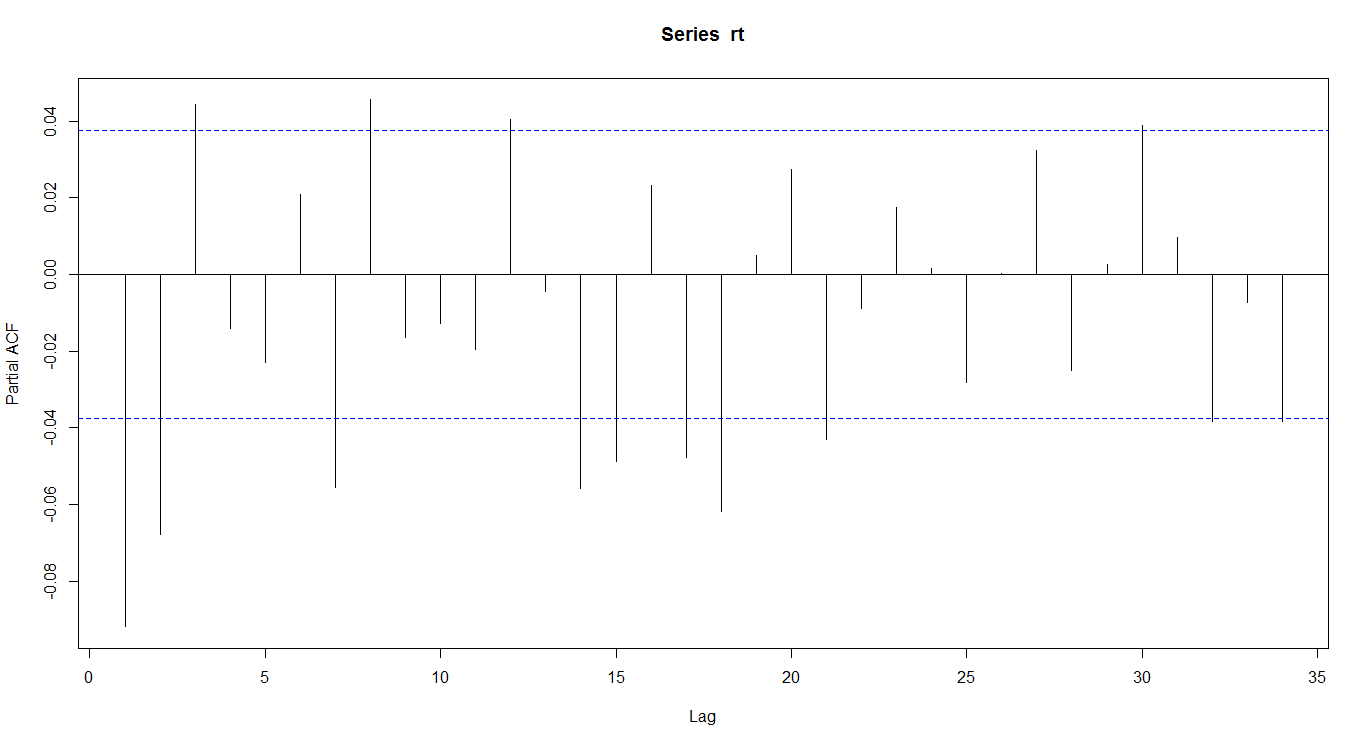


*1.d) Use the command ar(rt,method=’mle’,order.max=20) to specify the order of an AR model for rt . Use the PACF and AIC criteria (ar() and pacf() commands). Compare both approaches.*

According pacf model (Image 5) we should use model with 3 lags, because 4-th lag is non-significant (fully in confidence interval).

Because PACF for some lags is more than confidence interval (7,8,12,14,15,17,18), AIC criteria said use high-level model with 8 lags

*Image 5. PACF for log-returns*



Using AIC criteria the best model is AR(3):

AR(1): AIC = -14415.64

AR(2): AIC = -14426.2

AR(3): AIC = -14429.59

AR(4): AIC = -14428.13

AR(5): AIC = -14427.56

AR(6): AIC = -14426.75

AR(7): AIC = -14433.08

**AR(8): AIC = -14436.73**

AR(9): AIC = -14435.47

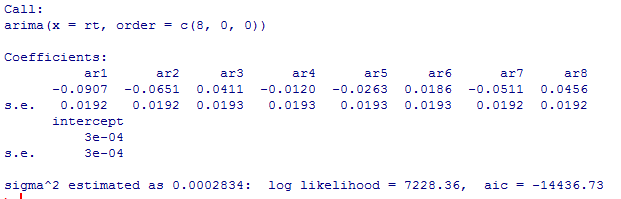
AR(10): AIC = -14433.91

PACF – AR(8), AIC – AR(8). Chosen AR(8).

*1.e) Build an AR model for rt . Plot the time series of the residuals, ACF and p-values of the Ljung-Box test (command tsdiag()). Perform the Ljung-Box test of the residuals by hand adjusting the degrees of freedom for the number of the model parameters (see [2], p.66). Is the model adequate? Why? Refine the model by eliminating all estimates with t-ratio less than 1.645 and check the new model as described above. Is the new model adequate? Why? Write down the final model.*

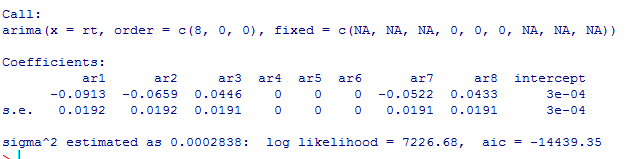
In the model not all coefficients are significant (see Image 6). We should reject 4, 5 and coefficients.

*Image 6. AR(3) coefficients*



Model after refining non-significant coefficients is shown at image 7.

*Image 7. AR(8) coefficients without non-significant coefficients*

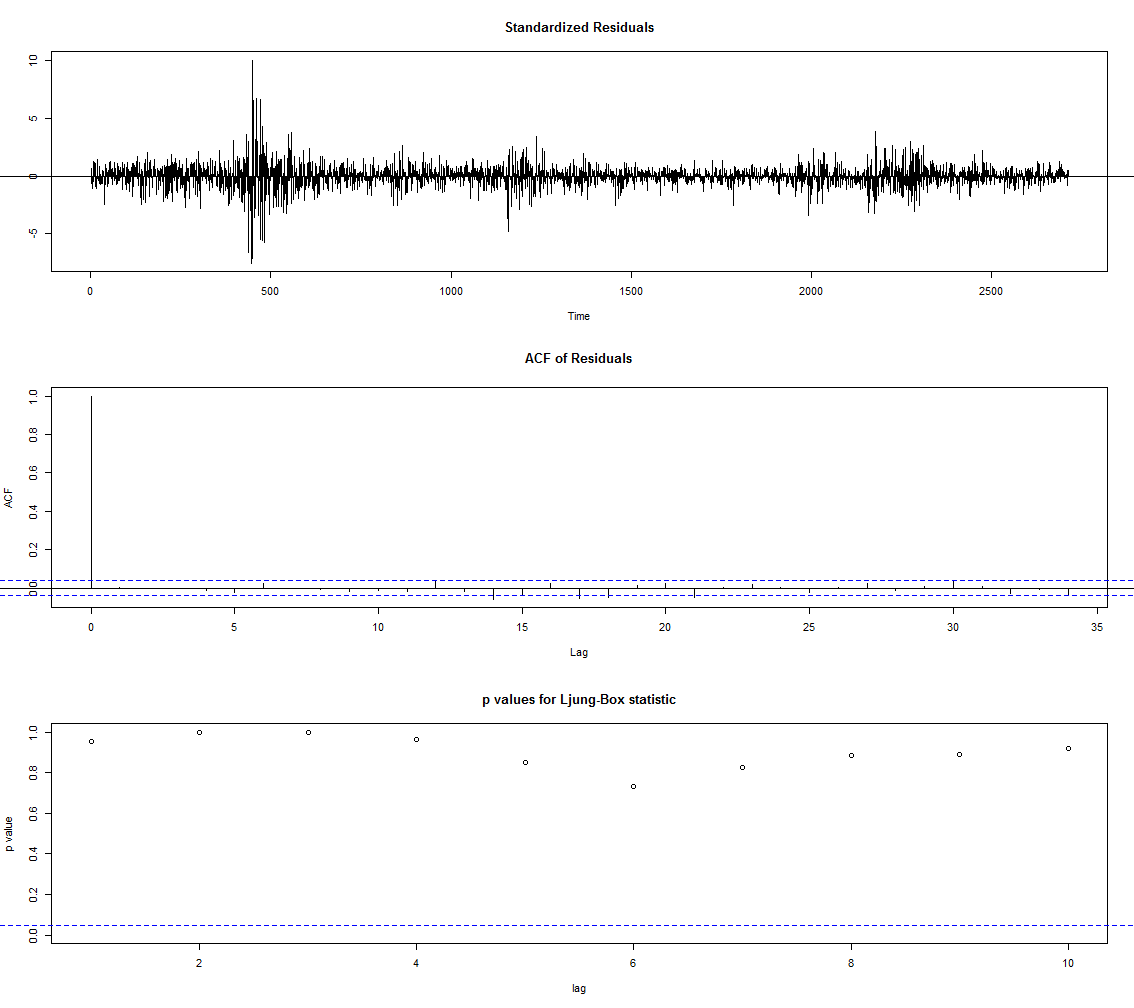


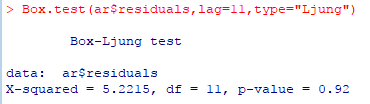
Because p-value in Ljung-Box test is high, we should not reject H0 (correlation is equal to zero) and errors is white noise. The model is adequate in modeling the dynamic linear dependence of data.

Final model is X = 0.0003 – 0.0913 Xt-1 – 0.0659 Xt-2 + 0.0446 Xt-3 -0.0522 Xt-7 + 0.0433 Xt-8,

where X – log return.

*Image 8. Standardized Residual, ACF of Residuals, p-values for log-returns AR(8) without non-significant coefficients*

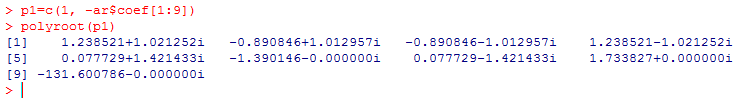




*1.f) Does the model imply existence of a cycle? Why? If the cycles are present, compute the average length of these cycles.*

I found 3 cycles in the data. Average lengths of cycle are: 9.112056, 2.741185, 4.144124.

*Image 8. Polyroot for the model*



*1.g) Use the fitted AR model to compute 1-step to 4-step ahead forecasts of rt at the forecast origin corresponding to the last observed date of the time series. Also, compute the corresponding 95% interval. Plot these results.*

With AR(3) model I made prediction for 9 observations in the future. It shown at image 9 (also contain 95% (light grey) and 80% (light blue) confidence interval). 1-step to 4-step prediction in table 1 (95% confidence interval).

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

2713 -1.451851e-03 -0.02304061 0.02013691 -0.03446901 0.03156531

2714 -4.260392e-05 -0.02172124 0.02163604 -0.03319722 0.03311202

2715 1.132773e-03 -0.02058141 0.02284696 -0.03207621 0.03434176

2716 -3.232882e-04 -0.02207099 0.02142441 -0.03358352 0.03293695

2717 1.377681e-03 -0.02037033 0.02312569 -0.03188303 0.03463839

2718 -5.533319e-04 -0.02230170 0.02119503 -0.03381459 0.03270792

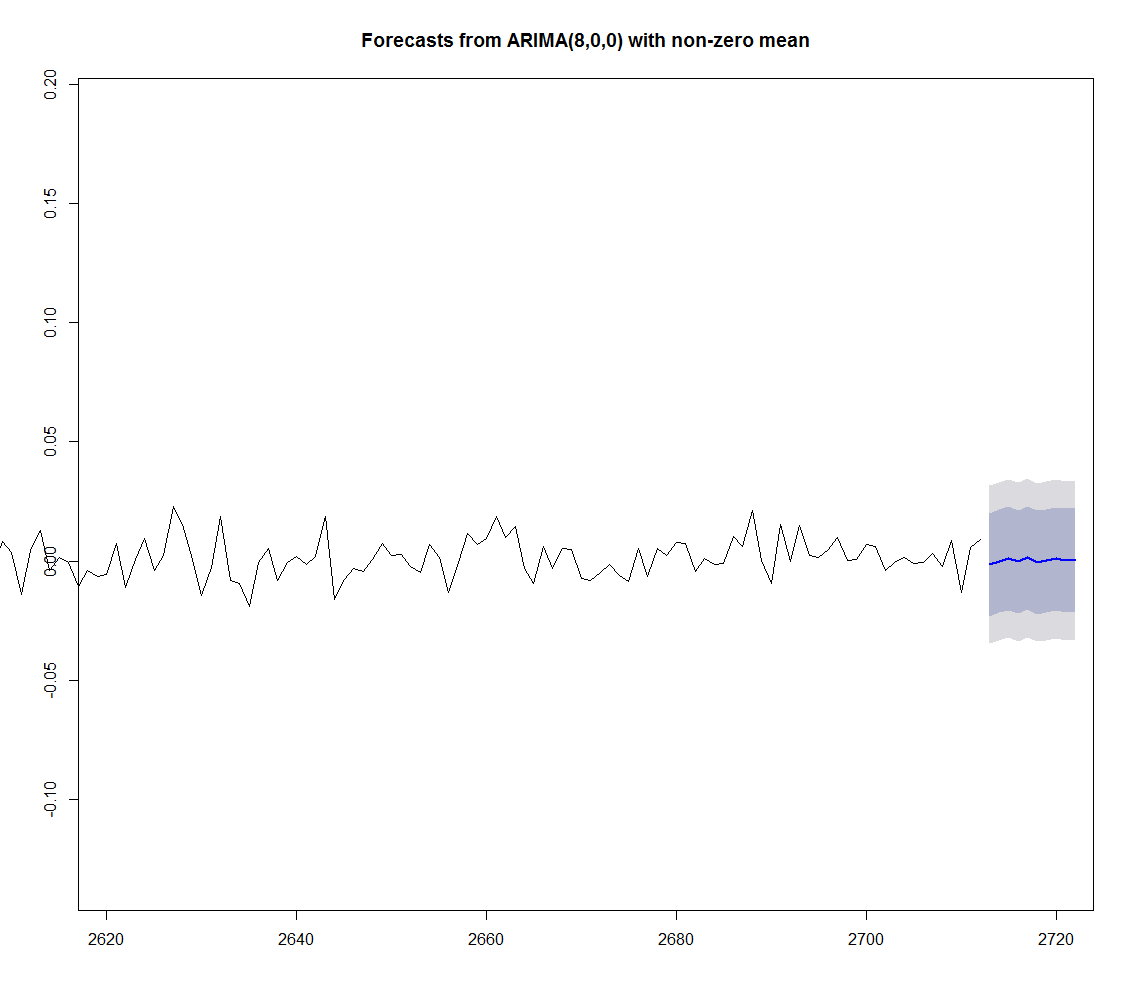
2719 8.737936e-05 -0.02166111 0.02183587 -0.03317406 0.03334882

2720 9.365297e-04 -0.02084127 0.02271433 -0.03236975 0.03424281

2721 1.954780e-04 -0.02161161 0.02200256 -0.03315558 0.03354654

2722 2.355013e-04 -0.02157163 0.02204263 -0.03311563 0.03358663

*Image 9. Prediction by AR(3)-model*



*Table 1. 4 step prediction*

|  |  |
| --- | --- |
| 1 step | 2 step |
| 3 step | 4 step |

*2.a) Consider a MA model for rt: Choose the order of such model. Support your choice with the ACF plot*

According image 10, ACF is not null in 4 lags: 0, 1, 2, 3, 7, 8. We should use 3 lags for MA(8)-model.

*2.b) Build the model. Refine it by removing coefficients estimates with t-ratio less than 1.645. Write down the fitted model.*

The lowest AIC in low-ordered model is AIC for model MA(8). According ACF and t-ratio, MA coefficient for 4, 5, 6 lags are equal to zero.

Fitted model shown at Image 11:

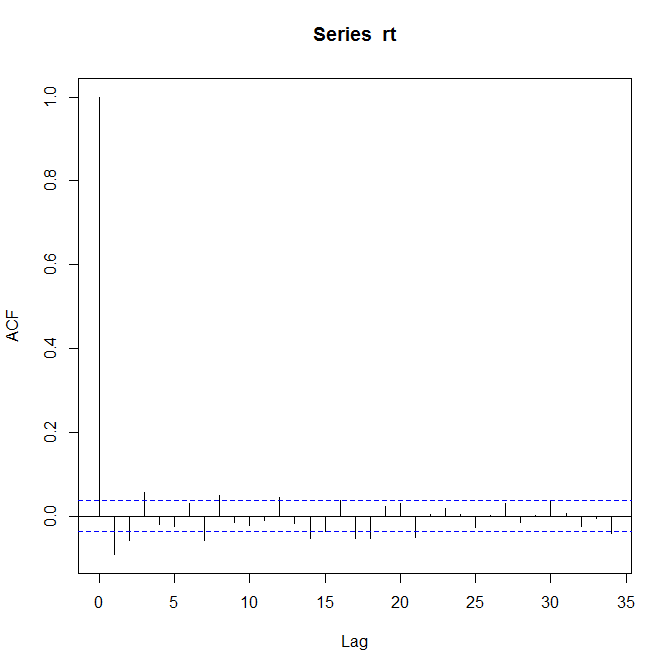
X = 0.0003 – 0.0885 Bt-1 – 0.0585 Bt-2 + 0.0466 Bt-3 -0.0564 Bt-7 + 0.0432 Bt-8,

X – log-return, B – moving average

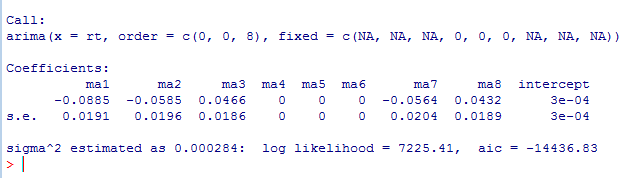
*2.c) Compute the Ljung-Box statistic of the residuals of the fitted MA model. Is there serial correlation in the residuals? Why?*

Because p-value for all lags for Ljung-Box statistic bigger then 0.05, we should not reject H0 (correlation is equal to zero) and errors is white noise. See at Image 12.

*Image 10. ACF for rt*



*Image 11. Fitted model for MA(8)*



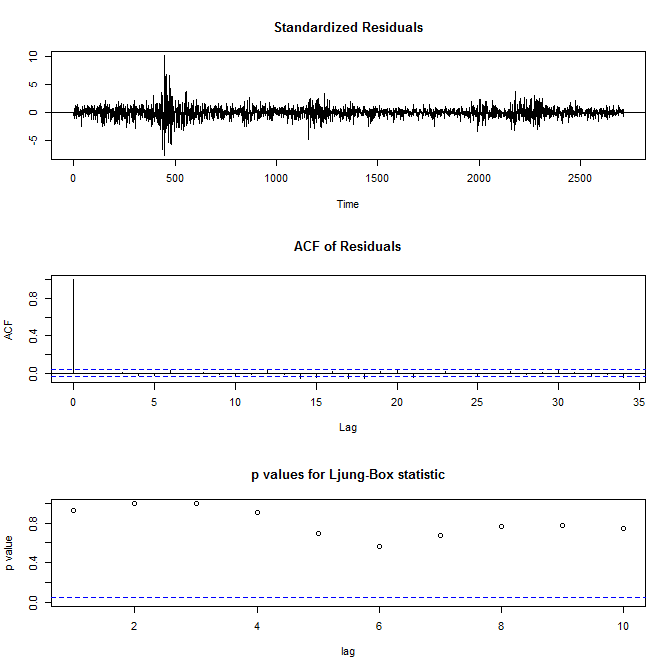
*2.d) Consider the in-sample fits of the AR model of Problem 1 and the MA model. Which model is preferred? Why?*

The choice between the two models is based on different criteria: AIC, log likelihood, sigma^2 (see table 2). According table, AR(8) model is better the MA(8).

*Table 2. AIC, log likelihood, sigma^2 for AR(8) and MA(8)*

|  |  |
| --- | --- |
| AR(8) | MA(8) |
| sigma^2 estimated as 0.0002838:  log likelihood = 7226.68,  aic = -14439.35 | sigma^2 estimated as 0.000284:  log likelihood = 7225.41,  aic = -14436.83 |

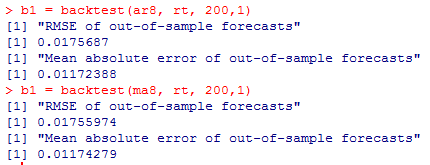
*Image 12. tsdiag for MA(8)*



*2.e) Use backtest at some forecast origin with horizon h = 1 to compare the two models. Indicate clearly the parameters of such backtesting (the estimation and forecasting subsamples, forecast origin and so on). Which model is preferred? Why?*

I used the backtest for log-return dataset for 200 lags and fitted AR and MA models. According RMSE AR(8) better MA(8), according mean absolute error of out-of-sample forecasts MA(8) better AR(8). In 2.d problem I chosen AR model. After this step I will choose AR(8) model because RMSE in this model lower (image 13).

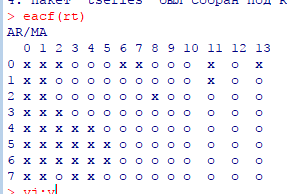
*Image 13. Backtest for AR(8) and MA(8) models*



*3.a) Yet again, focus on the log return series rt of the asset from Problem 1. Build an ARMA model including. Choosing the order of the model.*

According EACF the best model is ARMA(2,2). According AIC the best model is ARMA(2,3):

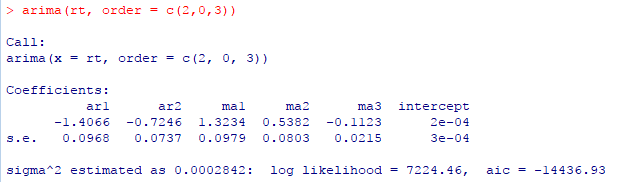
AIC = -14436,93. I will choose ARMA(2,3) model.



*3.b) Writing down the model.*

Coefficients for log-return is shown at image 14. All coefficients are significant.

*Image 14. ARMA(2,3)*



*3.c) Checking the model for adequacy by analyzing the residuals.*

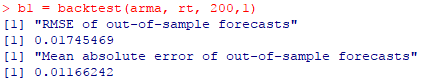
According p-value for Ljung-Box statistic (Image 16) we should reject H0 for high-level lags (> 12), because p-value < 0.05. It means, that correlation in residuals is not equal to zero. It means that residuals have some information. But according the first plot at image 16, standardized residuals quite close to white noise.

To sum up, the model is quite close to be adequate, but it can be improvement.

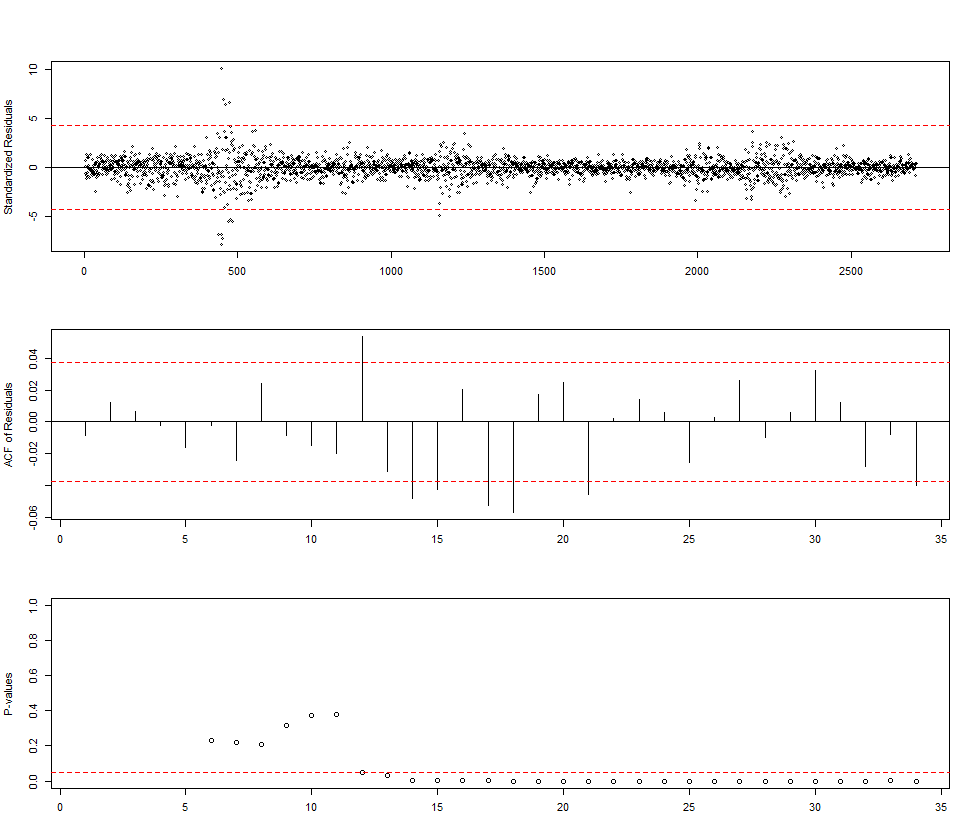
*3.d) Backtesting and comparing the model with those of Problems 1 and 2.*

Comparing ARMA(2,3) backtest result (Image 15) with backtest results for AR(8) and MA(8), the best model is ARMA(2,3) because RMSE and mean absolute error are lowest.

*Image 15. Backtest for ARMA(2,3)*



*Image 16. tsdiag() for ARMA(2,3)*



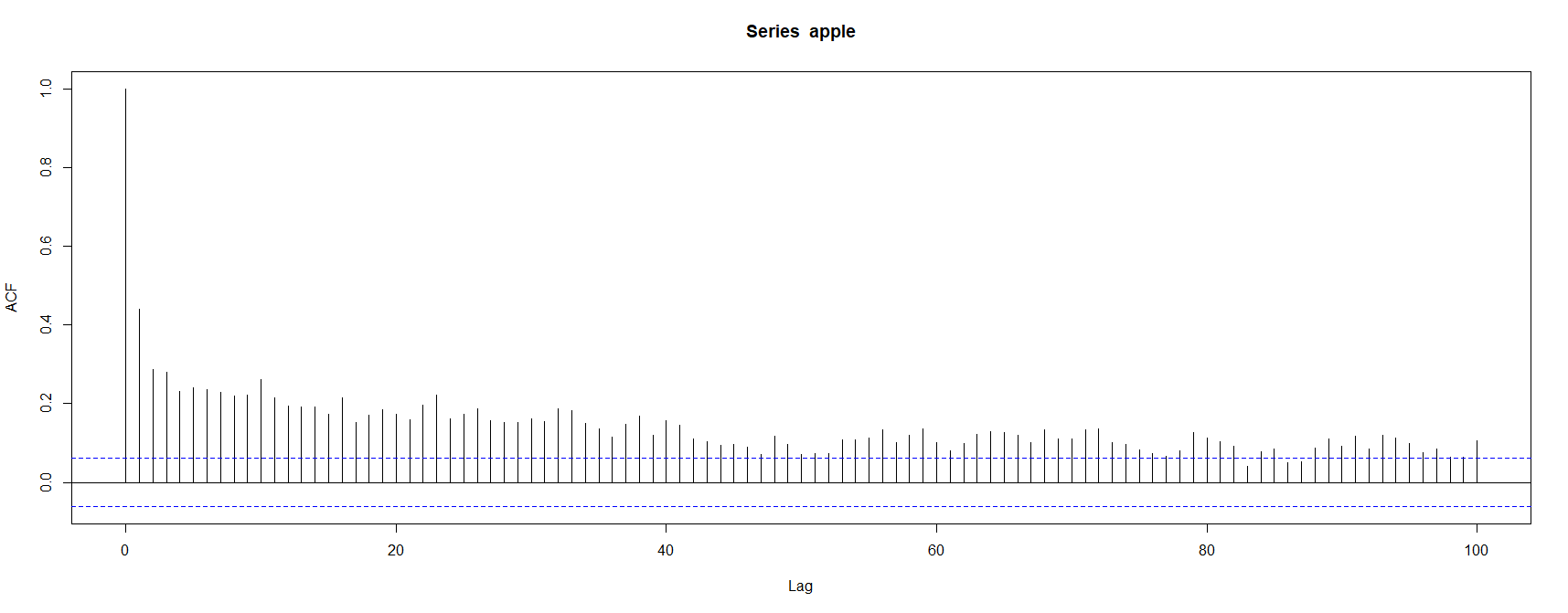
*4. Consider the daily range (daily high minus daily low) of a “blue chip” stock (Apple, CocaCola*

*etc.) for the last 4 years. Compute the first 100 lags of ACF of this series. Is there*

*evidence of long-range dependence? Explain!*

For the last 4 year 1008 observations in the dataset. ACF for first 100 lags is shown at Image 17.

*Image 17. ACF for Apple daily range*



Because, the sample ACF of a time series is not large in magnitude, but decays slowly, then the series have long memory.

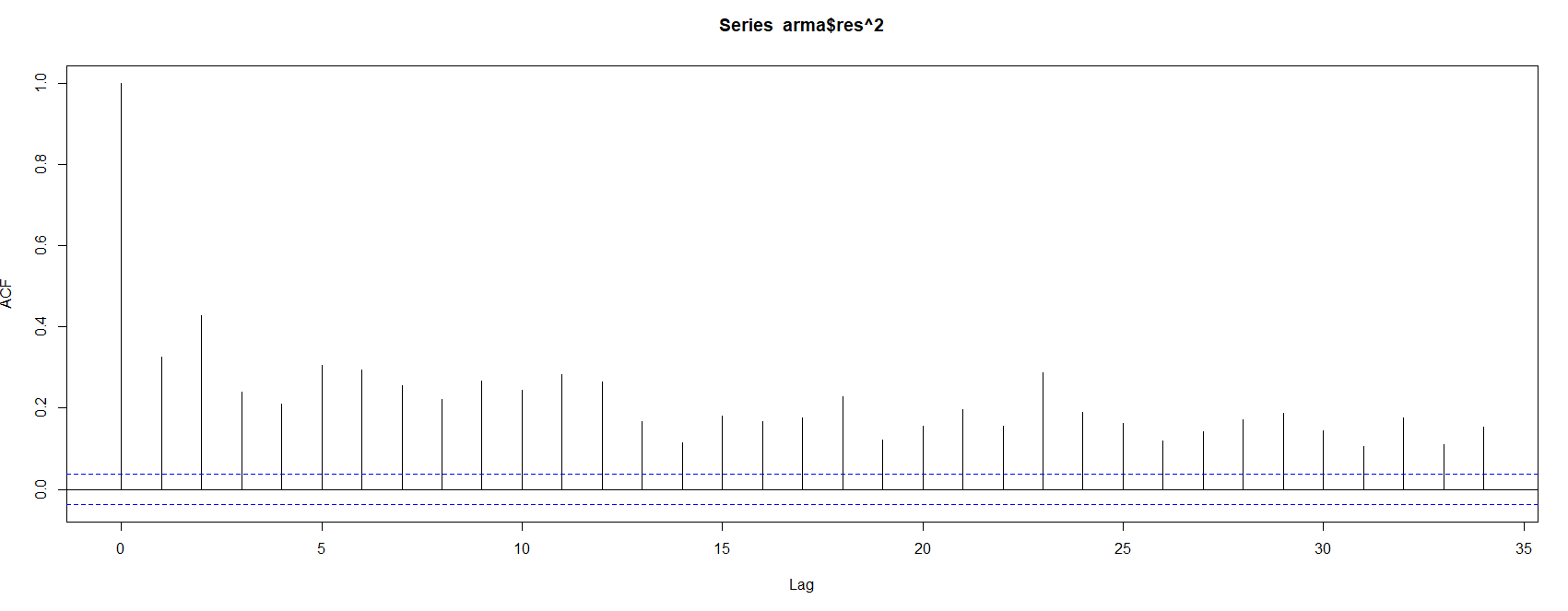
*5.a) Consider the log return series rt of the asset from Problem 1. Build an appropriate ARMA model.*

According Problem 3, the best ARMA model is ARMA(2,3). Coefficients shown at image 14.

*5.b) Test the residuals for the ARCH effect.*

Because residuals^2 are highly correlated (ACF bigger confidence interval), ARCH-effect there is a place to be. ACF for residuals^2 is shown at image 18.

*Image 18. ACF for residuals^2*



*5.c) Fit an ARMA-GARCH Gaussian model to the data*

I checked different order for GARCH part of model by AIC criteria. According table 3, the best model is GARCH(3,1) with ARMA(2,3)

*Table 3. AIC for GARCH(p,q)*

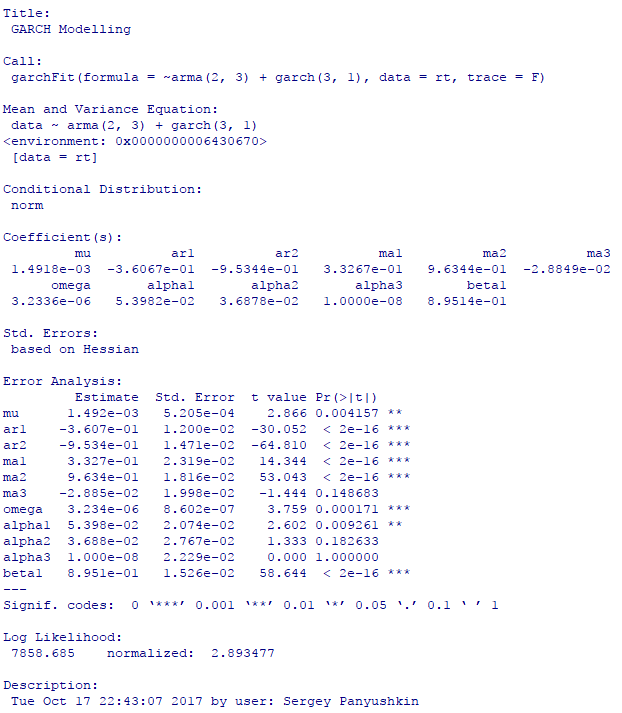
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p\q | 0 | 1 | 2 | 3 |
| 1 | -5.497679 | -5.776758 | -5.77594 | -5.775136 |
| 2 | -5.62115 | -5.776972 | -5.776888 | -5.77881 |
| 3 | -5.661551 | **-5.778855** | -5.776079 | -5.778126 |

Model is shown at image 19. Some coefficients not significant: ma3, alpha2, alpha3.

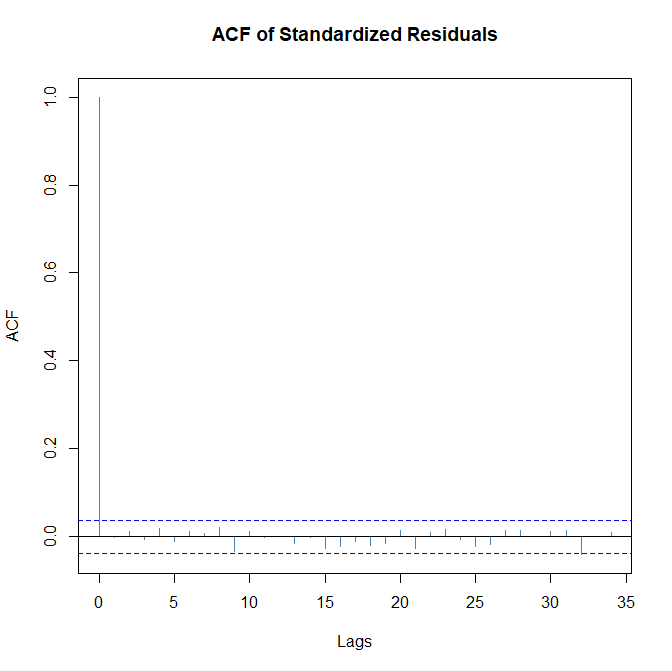
*5.d) Check the model by analyzing standardized residuals.*

I plotted the ACF of Standardized residuals (Image 20). According plot, all lags is not significant (in confidence interval). It means, that GARCH model’s order specified correct. Same situation in squared residuals (Image 21).

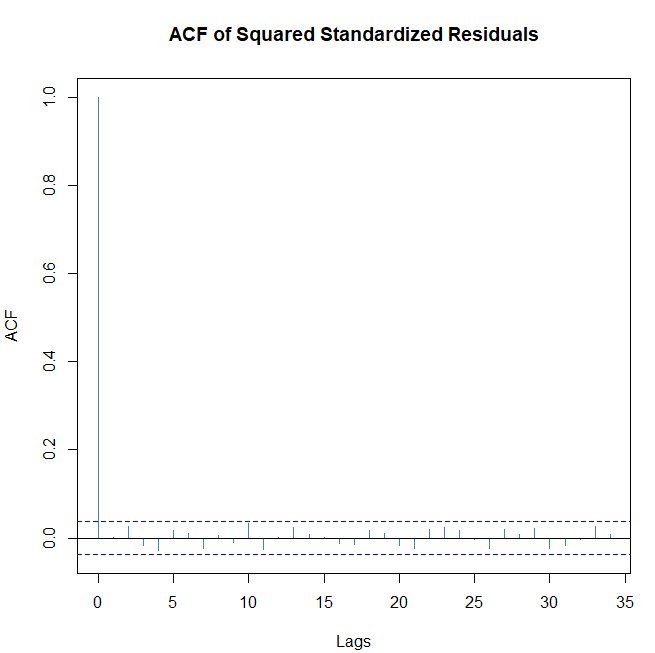
*Image 19. ARMA(2,3) - GARCH(3,1)*



*Image 20. ACF of Standardized Residuals ARMA(2,3) - GARCH(3,1)*



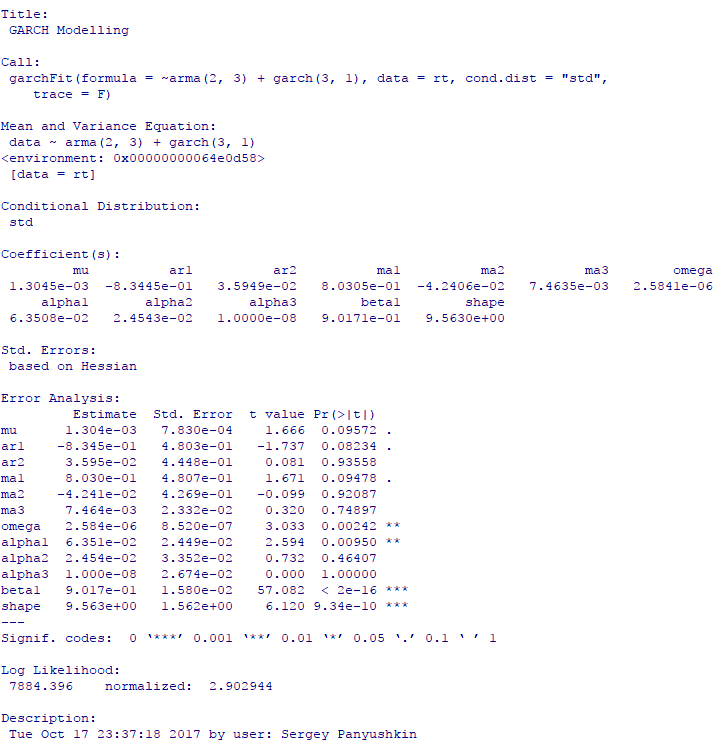
*Image 20. ACF of Squared Standardized Residuals ARMA(2,3) - GARCH(3,1)*



*5. e) Rebuild and check the model using Student t innovations.*

In the Student t innovations-model more coefficients is not significant (ar2, ma3, alpha2, alpha3) see Image 21. Log likelihood bigger than in normal distribution. AIC is better than in normal distribution model -5.79.

*Image 21. Student t innovations model*

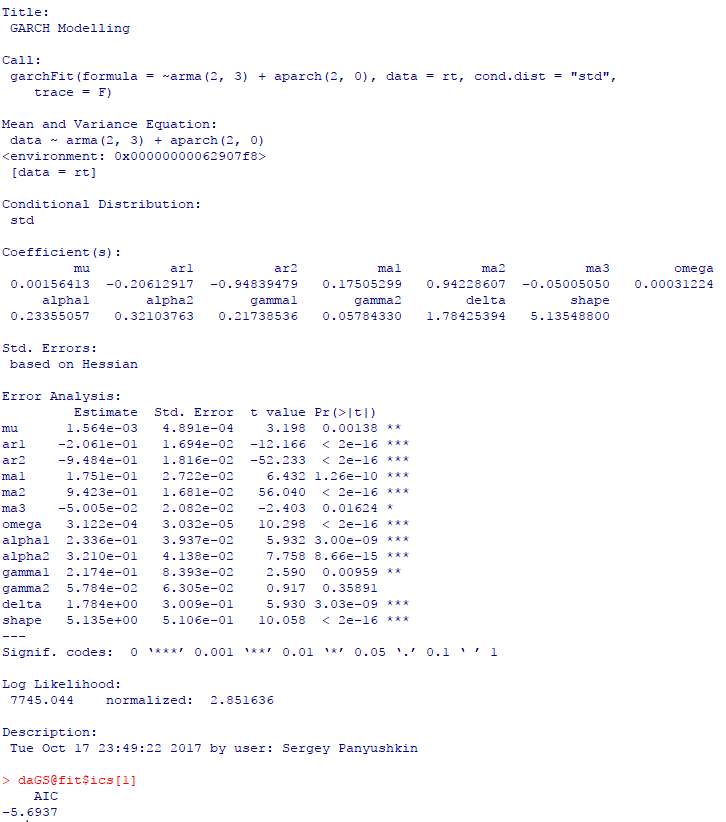


ACF of Squared Standardized Residuals and ACF of Standardized Residuals is not differ from normal distribution. Residuals are white noise.

*5. f) Build and check an ARMA-APACRH model (order=2).*

In the APARCH model 1 coefficient is not significant – gamma2. This model has small log likelihood – 7635, and big AIC -5.6937. Used std distribution, because AIC Likelihood for it better (for normal distribution AIC = 7651,674).

*Image 22. ARMA(2,3) – APARCH(2) model*



*5. g) Make and plot forecasts based on the above models.*

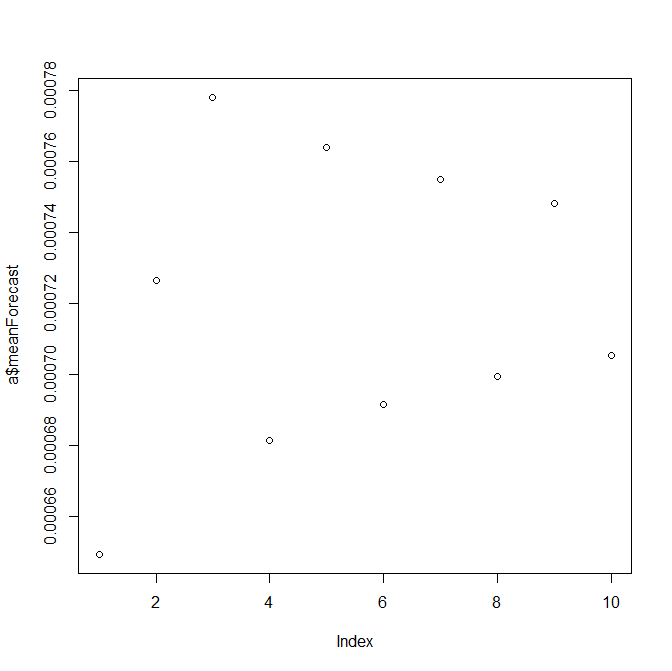
According AIC criteria, the TOP of models with GARCH is:

1. ARMA(2,3) - GARCH(3,1) Student t innovations. AIC: **-5.79.**
2. ARMA(2,3) – GARCH(3,1). AIC: **-5.778855.**
3. ARMA(2,3) – APARCH(2) . AIC: **-5.6937.**

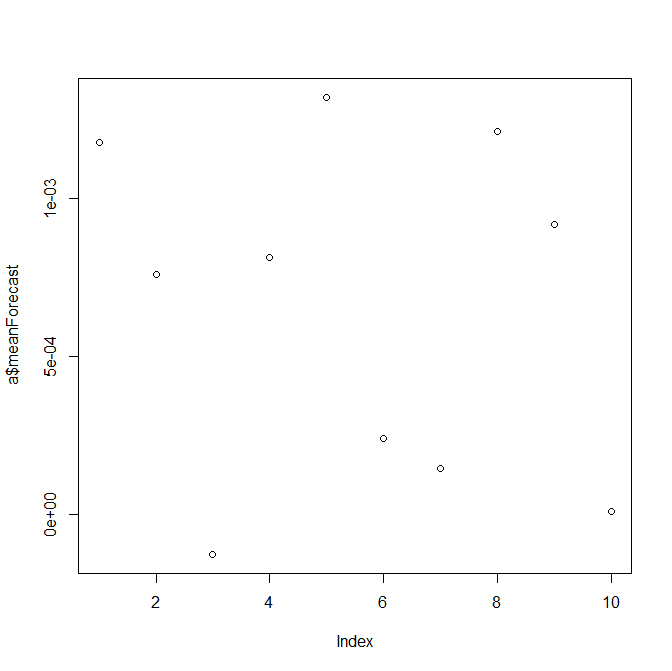
Table 4. Mean Forecast for models

|  |  |  |
| --- | --- | --- |
| ARMA(2,3) - GARCH(3,1) Student t innovations (Img.23) | ARMA(2,3) – GARCH(3,1) (Img.24) | ARMA(2,3) – APARCH(2) (Img.25) |
| 0.0006492323  0.0007265588  0.0007780953  0.0006813353  0.0007639296  0.0006915301  0.0007549132  0.0006994204  0.0007480051  0.0007054685 | 1.176576e-03  7.594844e-04  -1.262266e-04  8.132484e-04  1.318883e-03  2.407838e-04  1.475262e-04  1.209065e-03  9.151182e-04  9.020978e-06 | 0.0004119390  0.0007888188  0.0006411501  0.0006838617  0.0008151058  0.0007475451  0.0006370001  0.0007238609  0.0008107966  0.0007104984 |

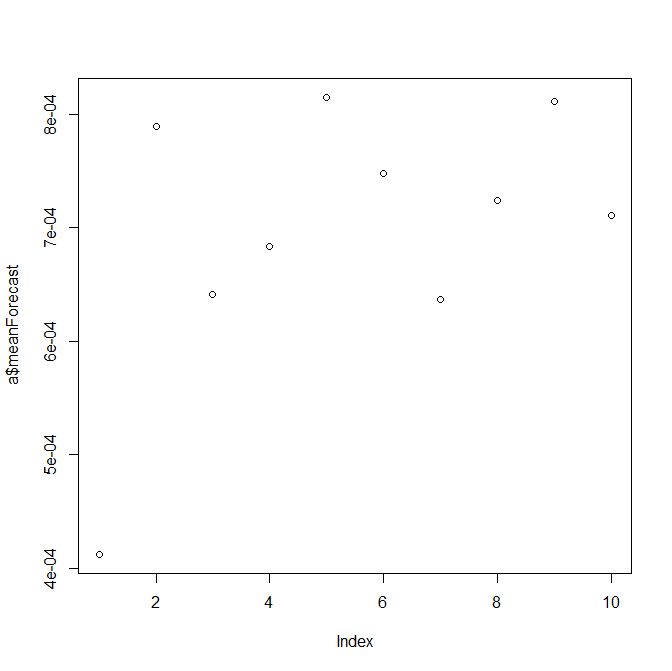
*Image 23. Forecast for ARMA(2,3) - GARCH(3,1) Student t innovations model*



*Image 24. ARMA(2,3) – GARCH(3,1) model*



*Image 25. ARMA(2,3) – APARCH(2)*



**BLACKBOX HW**

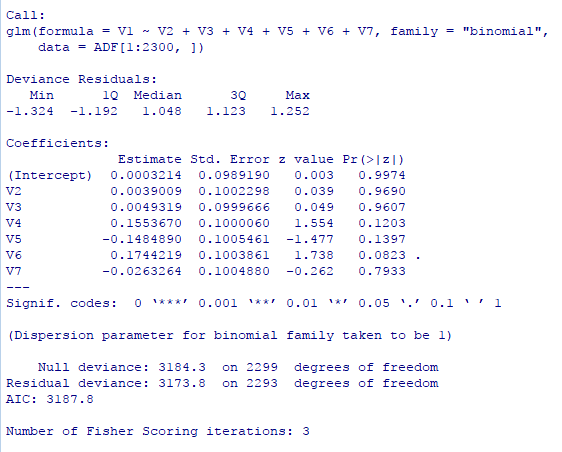
*1.a) Divide the data set into the training and forecast subsets. Clearly state this partition in your report.*

Dataset of 2714 observations was splitted Into 2 dataset, one of them contain 2300 observations (training) and another – 414. 414 observations (15%) is enough for getting statistically significant results.

*1.b) Fit a linear logistic regression model for P(Dt = 1) using Dt−i , Mt−i , i = 1, 2, 3 as explanatory variables. Use only the training subset for estimation. Discuss statistical significance of the coefficients. Refine the model if needed.*

According the image 26, only one coefficient is significant (with 0.1 significant level). Other coefficients are equal zero. In the future analysis, I will use only 1 coefficient: V6.

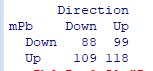
*Image 26. Logistic model coefficients significance*



*1.c) Using the model, make predictions for the forecast subset. Specify the threshold you apply. Compute the forecast error.*

The threshold is 0.5 – if probability > 0,5, it means price will grow. If <=, price will fall. According image 27, our model have accuracy = 49,758%.

*Image 27. Forecasting for test-dataset*

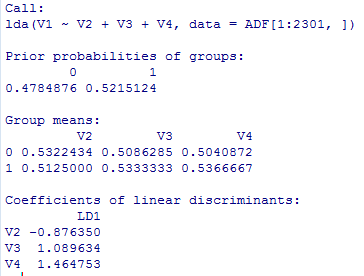


*1.d) Apply the linear discriminant analysis instead of log regression in items 1a-1c. Compute*

*the forecast error.*

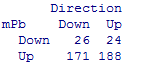
LDA coefficients calculated on training dataset (2300 observations). The coefficients can be found at Image 28.

*Image 28. LDA model*



According image 29, our LDA-model have accuracy = 52.32%. This result better than in logistic regression.

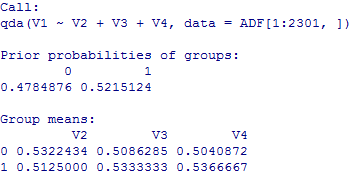
*Image 29. Forecasting for test-dataset LDA*



*1. e) Apply the quadratic discriminant analysis instead of log regression in items 1a-1c. Compute the forecast error.*

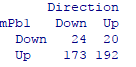
QDA coefficients calculated on training dataset (2300 observations). The coefficients can be found at Image 30.

*Image 30. QDA model*



According image 31, our QDA-model have accuracy = 52.81%. This result better than in logistic regression and better than in LDA-model.

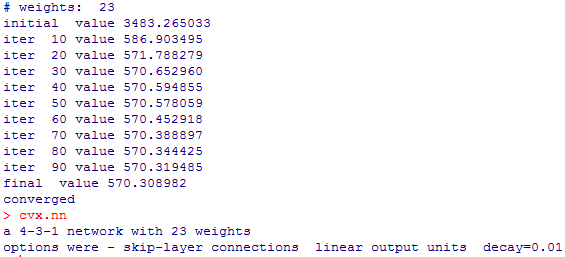
*Image 31. Forecasting for test-dataset QDA*



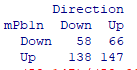
*1. f) Employ a 4-3-1 look-forward neural network with direct link for P(Dt = 1) instead of log regression in items 1a-1c. Build a model for two (i = 1, 2) lagged variables D, M. Compute the forecast error.*

Final value for neural network is 570.308982. Network has 4-3-1 specification. See Image 32. According image 33, accuracy for Neural network is 51.37%

*Image 32. Neural network specification*



*Image 33. Forecasting for test-dataset Neural network*



*1. g) Compare the predictive power (i.e, forecast error) of methods 1b,1d,1e,1f.*

The most balanced characteristic for models is F1 score (2\*precision\*accuracy/(precision+accuracy)). The QDA model is the best model from all (according all measurements, see table 5).

*Table 5. Measurement for model’s relevancies*

|  |  |  |  |
| --- | --- | --- | --- |
| **Logistic regression** | **LDA** | **QDA** | **Neural network** |
|  |  |  |  |
| *Accuracy: 49.76%* | *Accuracy: 52.32%* | ***Accuracy: 52.81%*** | *Accuracy: 51.37%* |
| *Recall: 51.98%* | *Recall: 52.36%* | ***Recall: 52.60%*** | *Recall: 55.08%* |
| *Specificity: 47.05%* | *Specificity: 52%* | ***Specificity: 54.54%*** | *Specificity: 46.77%* |
| *Precision: 54.37%* | *Precision: 88.67%* | ***Precision: 90.56%*** | *Precision: 69.01%* |
| *F1 score: 0.5314* | *F1 score: 0.6631* | ***F1 score: 0.6654*** | *F1 score: 0.6126* |

2.a) Consider the trading model based on technical indicators (as discussed in class on October 4, 2017); also, see [2], Chapter 3: (a) Choose and download the market data. Remember, the data must be unique

Downloaded daily data for CVX from Yahoo. Dataset contains 2718 observations.

*2.b) Specify the training and forecasting subsets.*

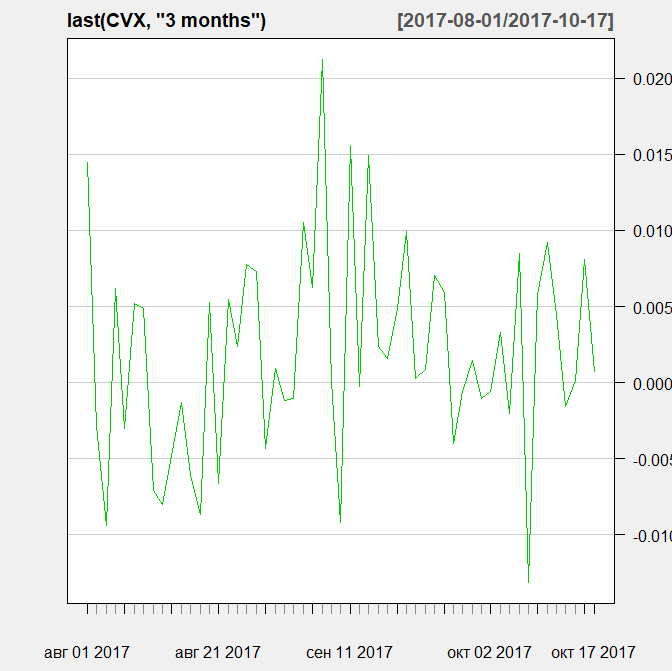
Dataset of 2714 observations was splitted Into 2 dataset:

* train dataset contains data from '2008-01-10' to '2016-12-31',
* test dataset contain data from '2017-01-01' to '2017-10-15'.

*2.c) Using the template code, play with the model. Obtain the significant regressors under random forest approach.*

Data for the last 3 months is shown at image 34.

*Image 33. CVX data for last 3 months*



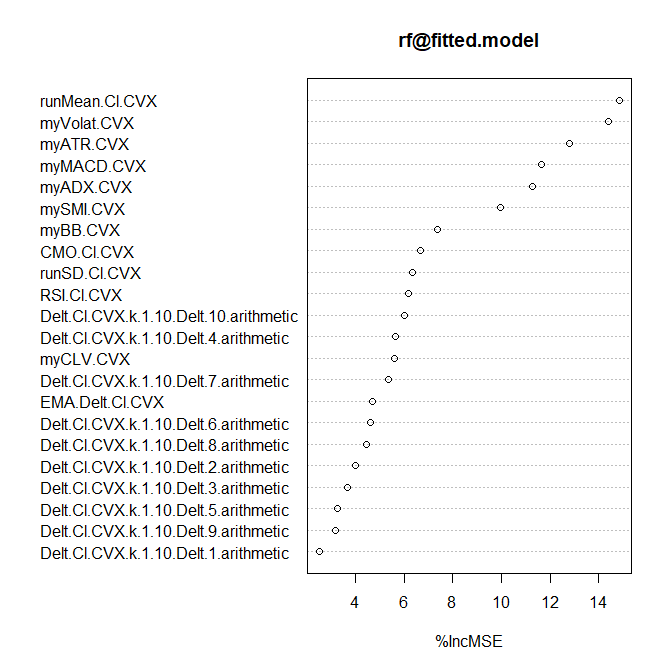
Using Random Forest the most significant regressors can be specified. The most significant regressor is running mean for Close price CVX, second - selected volatility estimators – this two regressors contain around 30% information (%IncMSE). Another 4 regressors contain 10-12% each:

* ATR - average true range (ATR) is a Welles Wilder's style moving average of the TR
* MACD - is a special case of the general oscillator applied to price
* ADX - Directional Movement Index
* SMI - Stochastic Momentum Indicator

All regressors is shown at Image 34.

*2.d) Try neural networks, support vector machines, regressive splines for prediction. Obtain the tabular summary of the results. Compare the predictive power.*

*Image 34. Regressors significance*



Function for neural network returns the values of Precision and Recall for the buy, sell and sell+buy signals. The measurements for this neural network, support vector machines, support vector machines are shown at table 6. The measurement for the support vector machines is shown at image 36. The result for multidimension regression scrumming is shown at Image 37.

*Table 6. Measurements for different models*

|  |  |  |
| --- | --- | --- |
| *Neural network measurement* | *Support vector machines* | *Multidimension regression scrumming* |
|  |  |  |

The best model for total s+b precision and recall is the first – neural network.

*2.e) Also, provide the fragments of the time series with the true and predicted signals in visual or tabular form.*

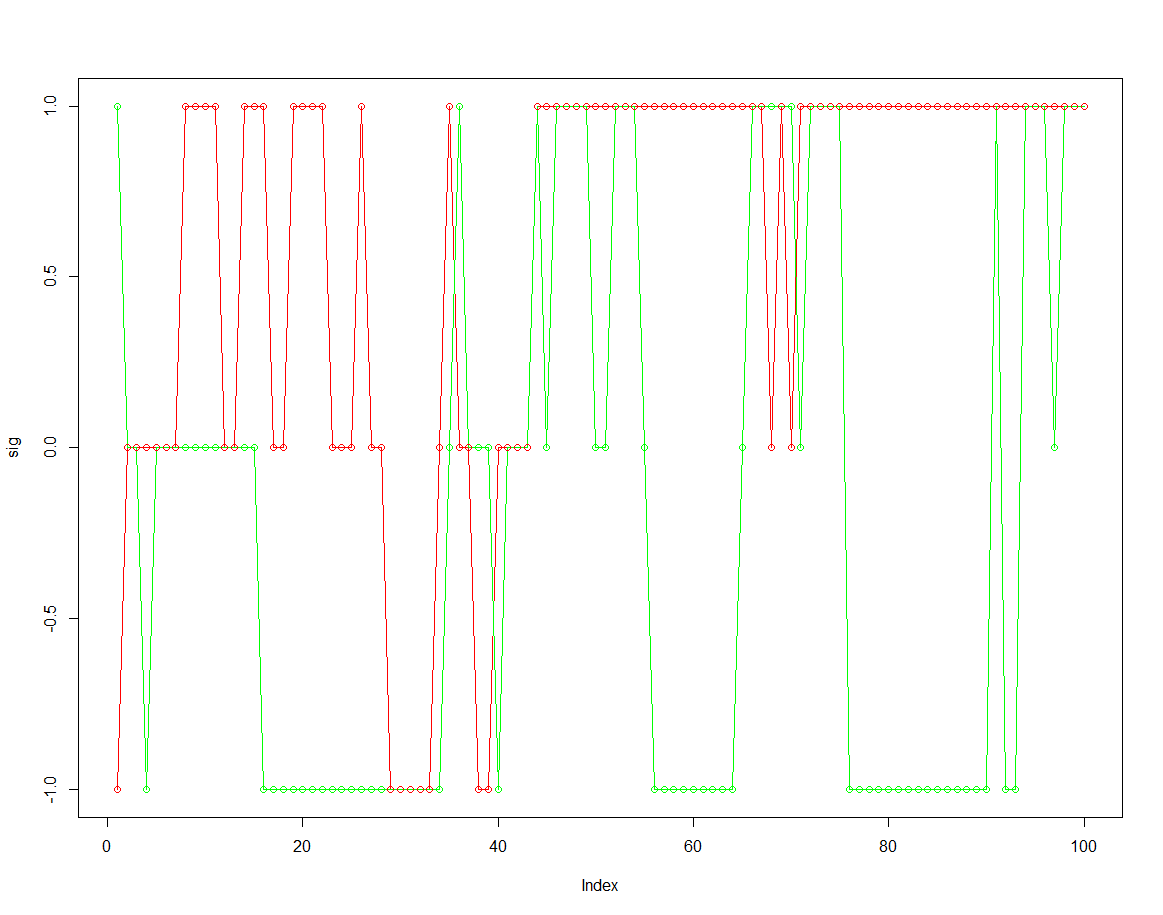
Plotted graphics for the predicted values (Images 35-37):

Green – true,

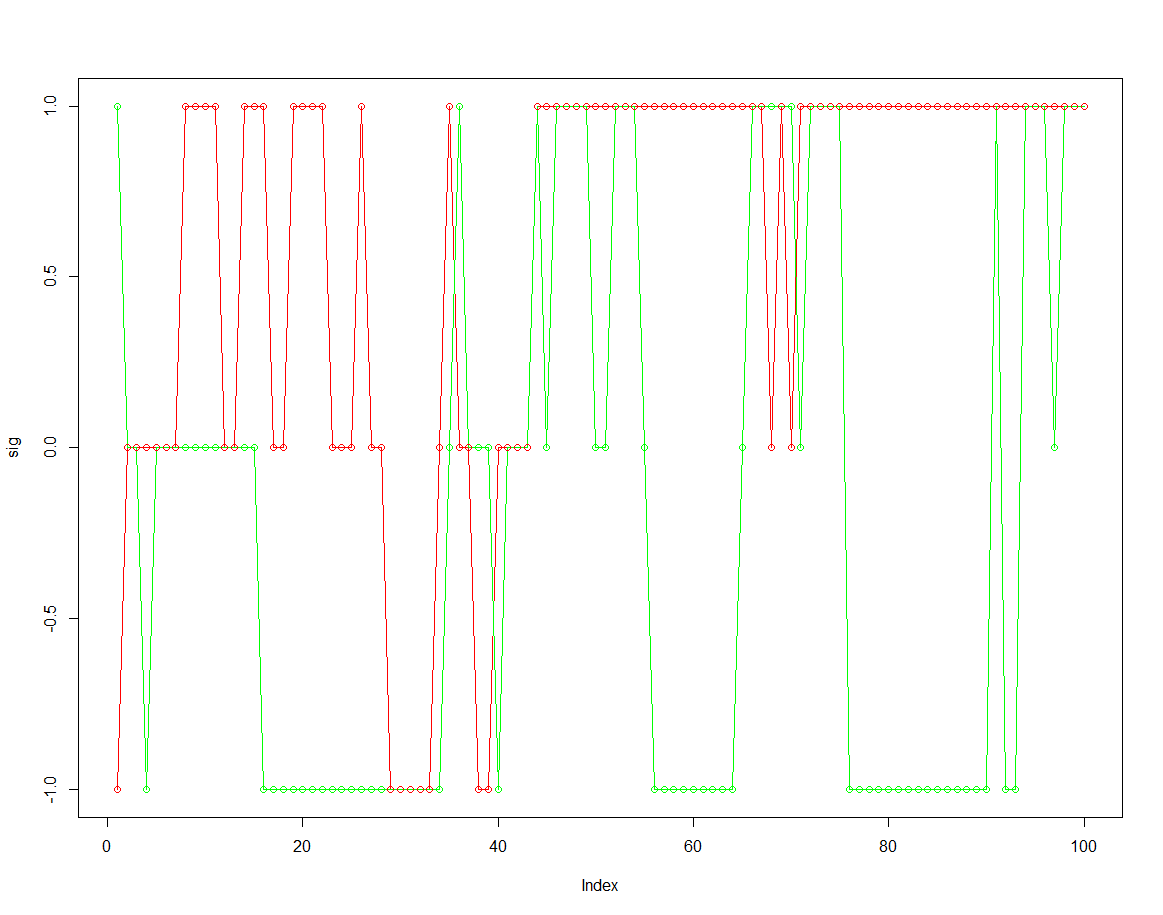
Red – predicted.

Also, 1 = buy, -1 = sell, 0 = hold.

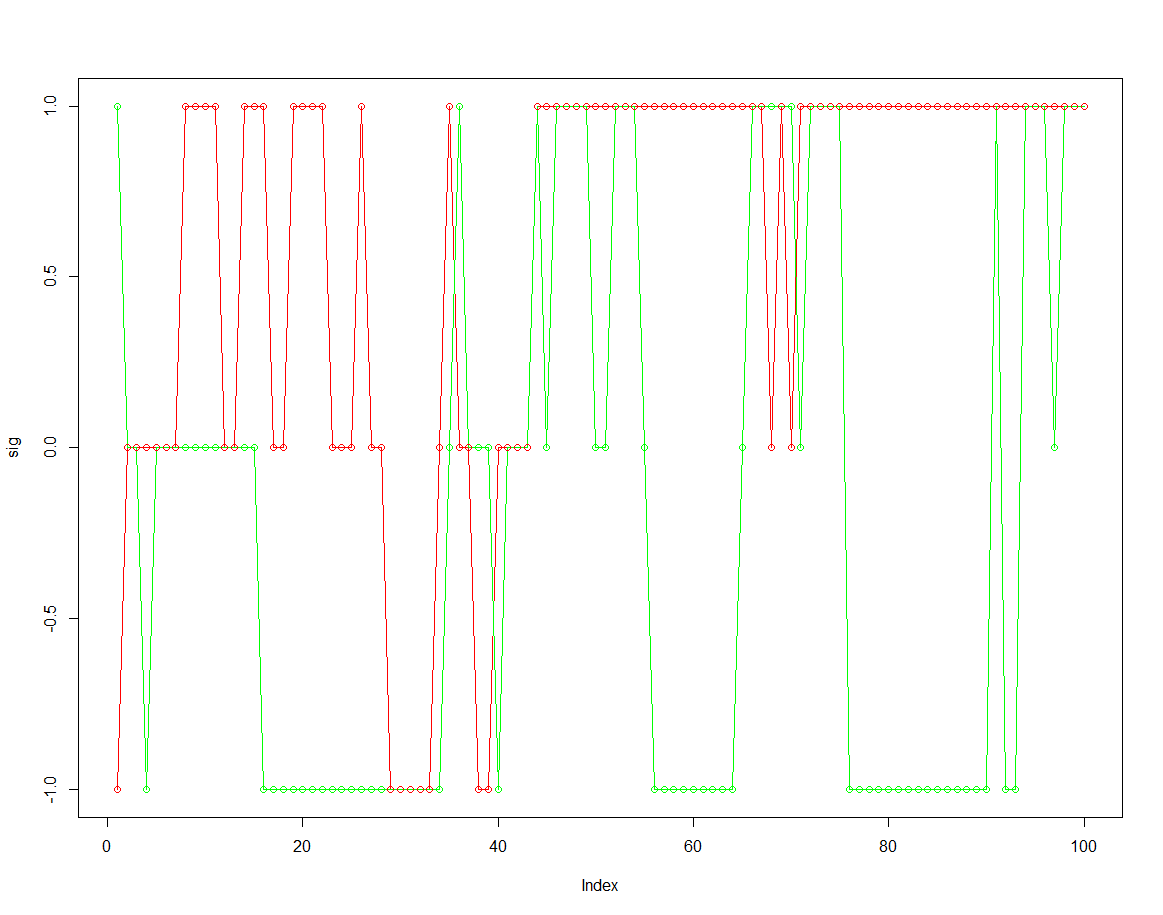
*Image 35. Plot for neural network measurement*



*Image 36. Plot for support vector machines prediction*



*Image 37. Plot for multidimension regression scrumming prediction*



**Appendix 1. R code for the first homework**

getSymbols("CVX", src="yahoo")

length(CVX)

CVX = na.omit(CVX)

length(CVX)

rt =diff(log(Ad(CVX)))

xt = log(Ad(CVX))

plot(xt)

plot(rt)

acf(rt, lag = 12)

acf(xt, lag = 12)

adf.test(xt, k = 12)

Box.test(rt, lag = 12, type = "Ljung-Box")

rt = na.omit(rt)

pacf(rt)

da1 = ar(as.numeric(rt),method="mle",order.max=20)

names(da1)

da1$aic

arima(rt, c(3,0,0))

ar = arima(rt, c(3,0,0))

ar = arima(rt, c(8,0,0), fixed = c(NA, NA, NA, 0, 0, 0, NA, NA, NA))

tsdiag(ar)

p1=c(1, -ar$coef[1:9])

polyroot(p1)

2\*pi/acos(-1.113921/Mod(-1.113921-2.21038i))

daP = forecast(ar)

daP$upper

plot(daP, xlim = c(2621,2720))

daP = forecast(ar, level = 80, h = 1)

plot(daP, xlim = c(2621,2720))

daP = forecast(ar, level = 80, h = 5)

plot(daP, xlim = c(2621,2720))

daP = forecast(ar, level = 95, h = 5)

plot(daP, xlim = c(2700,2720))

daP = forecast(ar, level = 95, h = 1)

plot(daP, xlim = c(2700,2720))

daP = forecast(ar, level = 95, h = 2)

plot(daP, xlim = c(2700,2720))

daP = forecast(ar, level = 95, h = 3)

plot(daP, xlim = c(2700,2720))

daP = forecast(ar, level = 95, h = 4)

plot(daP, xlim = c(2700,2720))

acf(rt)

arima(rt, order = c(0,0,8))

ma8 = arima(rt, order = c(0,0,8), fixed = c(NA, NA, NA, 0, 0, 0, NA, NA, NA))

tsdiag(ma8)

ar8 = arima(rt, order = c(8,0,0), fixed = c(NA, NA, NA, 0, 0, 0, NA, NA, NA))

source("backtest.R")

b1 = backtest(ar8, rt, 200,1)

b1 = backtest(ma8, rt, 200,1)

eacf(rt)

arma = arima(rt, order = c(2,0,3))

tsdiag(arma)

source("backtest.R")

b1 = backtest(arma, rt, 200,1)

getSymbols("AAPL", src="yahoo")

apple = Hi(AAPL) - Lo(AAPL)

apple = apple[1710:2717]

acf(apple, lag = 100)

acf(arma$res^2)

daG=garchFit(~arma(2,3)+garch(3,0),data = rt, trace=F)

plot(daG)

str(daG)

daG@fit$ics[1] – для всех моделей по очереди выводим AIC

daG=garchFit(~arma(2,3)+garch(3,1),data = rt, trace=F)

plot(daG)

acf(arma$res^2)

daGS =garchFit(~arma(2,3)+garch(3,1),data = rt, trace=F, cond.dist = "std")

daT =garchFit(~arma(2,3)+aparch(2,0),data = rt, trace=F)

daT =garchFit(~arma(2,3)+aparch(2,0),data = rt, trace=F, cond.dist = "std")

a = predict(daG)

a$meanForecast

plot(a$meanForecast)

a = predict(daGS)

a$meanForecast

plot(a$meanForecast)

a = predict(daT)

a$meanForecast

plot(a$meanForecast)

**Appendix 2. R code for the second homework**

getSymbols("^GSPC", src = "yahoo")

getSymbols("CVX", src="yahoo")

N1 = length(CVX)

CVX = diff(log(Ad(CVX)))

GSPC = diff(log(Ad(GSPC)))

idx=c(1:N1)[CVX>0]

jdx=c(1:N1)[CVX<=0]

idxM=c(1:N1)[GSPC>0]

jdxM=c(1:N1)[GSPC<=0]

N=length(CVX)

cvx=rep(0,N)

cvx[idx]=1

cvx[jdx]=0

spm=rep(0,N)

spm[idxM]=1

spm[jdxM] = 0

A=matrix(0,N-3,7)

A[,1]=cvx[4:N]

A[,2]=cvx[3:(N-1)]

A[,3]=cvx[2:(N-2)]

A[,4]=cvx[1:(N-3)]

A[,5]=spm[3:(N-1)]

A[,6]=spm[2:(N-2)]

A[,7]=spm[1:(N-3)]

ADF=as.data.frame(A)

m1=glm(V1~V2+V3+V4+V5+V6+V7,data=ADF[1:2600,],family="binomial")

m1=glm(V1~V6,data=ADF[1:2300,],family="binomial")

mP=predict(m1,ADF[2301:2714,],type="response")

mPb=rep(0,length(mP))

mPb[mP>0.5]="Up"

mPb[mP<=0.5]="Down"

Direction=rep("Down",length(A[2301:2714]))

Direction[A[2301:2714]>0]="Up"

table(mPb,Direction)

A=matrix(0,N-3,4)

A[,1]=cvx[4:N]

A[,2]=cvx[3:(N-1)]

A[,3]=cvx[2:(N-2)]

A[,4]=cvx[1:(N-3)]

ADF=as.data.frame(A)

mLDA=lda(V1~V2+V3+V4,data=ADF[1:2300,])

mPLDA=predict(mLDA,ADF[2301:2709,])

mPb=rep(0,length(mPLDA$class))

mPb[mPLDA$class == 1]="Up"

mPb[mPLDA$class == 0]="Down"

Direction=rep("Down",length(A[2301:2709]))

Direction[A[2301:2709]>0]="Up"

table(mPb,Direction)

mQDA=qda(V1~V2+V3+V4,data=ADF[1:2300,])

mPQDA=predict(mQDA,ADF[2301:2709,])

mPb1=rep(0,length(mPQDA$class))

mPb1[mPQDA$class == 1]="Up"

mPb1[mPQDA$class == 0]="Down"

Direction=rep("Down",length(A[2301:2709]))

Direction[A[2301:2709]>0]="Up"

table(mPb1,Direction)

A=matrix(0,N-4,5)

A[,1]=cvx[5:N]

A[,2]=cvx[4:(N-1)]

A[,3]=cvx[3:(N-2)]

A[,4]=cvx[2:(N-3)]

A[,5]=cvx[1:(N-4)]

cvx.nn=nnet(head(A[,2:5],2300),head(A[,1], 2300),size=3,linout=T,skip=T,maxit=10000,decay=1e-2,reltol=1e-7,abstol=1e-7,range=1.0)

PredNN = predict(cvx.nn,tail(A[,2:5], 409))

mPb1n=rep(0,length(PredNN))

mPb1n[PredNN >0.5]="Up"

mPb1n[PredNN <= 0.5]="Down"

Direction=rep("Down",length(A[2301:2709]))

Direction[A[2301:2709]>0]="Up"

table(mPb1n,Direction)

T.ind <- function(quotes, tgt.margin = 0.02, n.days = 10) {

v <- apply(HLC(quotes), 1, mean)

r <- matrix(NA, ncol = n.days, nrow = NROW(quotes))

for (x in 1:n.days) r[, x] <- Next(Delt(v, k = x), x)

x <- apply(r, 1, function(x) sum(x[x > tgt.margin | x <

-tgt.margin]))

if (is.xts(quotes))

xts(x, time(quotes))

else x

}

candleChart(last(CVX, "3 months"), theme = "white", TA = NULL)

avgPrice <- function(p) apply(HLC(p), 1, mean)

addAvgPrice <- newTA(FUN = avgPrice, col = 1, legend = "AvgPrice")

addT.ind <- newTA(FUN = T.ind, col = "red", legend = "tgtRet")

addAvgPrice(on = 1)

addT.ind()

myATR <- function(x) ATR(HLC(x))[, "atr"]

mySMI <- function(x) SMI(HLC(x))[, "SMI"]

myADX <- function(x) ADX(HLC(x))[, "ADX"]

myAroon <- function(x) aroon(x[, c("High", "Low")])$oscillator

myBB <- function(x) BBands(HLC(x))[, "pctB"]

myChaikinVol <- function(x) Delt(chaikinVolatility(x[, c("High", "Low")]))[, 1]

myCLV <- function(x) EMA(CLV(HLC(x)))[, 1]

myEMV <- function(x) EMV(x[, c("High", "Low")], x[, "Volume"])[, 2]

myMACD <- function(x) MACD(Cl(x))[, 2]

myMFI <- function(x) MFI(x[, c("High", "Low", "Close")], x[, "Volume"])

mySAR <- function(x) SAR(x[, c("High", "Close")])[, 1]

myVolat <- function(x) volatility(OHLC(x), calc = "garman")[, 1]

data(C)

library(randomForest)

data.model <- specifyModel(T.ind(CVX) ~ Delt(Cl(CVX),k=1:10) + myATR(CVX) + mySMI(CVX)

+ myADX(CVX)+ myBB(CVX)+ myCLV(CVX)+CMO(Cl(CVX))+EMA(Delt(Cl(CVX)))+myVolat(CVX)

+ myMACD(CVX)+ RSI(Cl(CVX)) +runMean(Cl(CVX))+runSD(Cl(CVX)))

set.seed(1234)

rf <- buildModel(data.model,method='randomForest',

training.per=c(start(CVX),index(CVX["2012-12-31"])),

ntree=50, importance=T)

varImpPlot(rf@fitted.model, type = 1)

imp <- importance(rf@fitted.model, type = 1)

rownames(imp)[which(imp > 8)]

data.model <- specifyModel(T.ind(CVX) ~ mySMI(CVX)+CMO(Cl(CVX))

+ runMean(Cl(CVX)))

Tdata.train <- as.data.frame(modelData(data.model,

data.window=c('2008-01-10','2016-12-31')))

Tdata.eval <- na.omit(as.data.frame(modelData(data.model,

data.window=c('2017-01-01','2017-10-15'))))

Tform <- as.formula('T.ind.CVX~ .')

# requires package DMwR

set.seed(1234)

library(nnet)

norm.data <- scale(Tdata.train)

nn <- nnet(Tform, norm.data[1:600, ], size = 10, decay = 0.01,

maxit = 1000, linout = T, trace = F)

norm.preds <- predict(nn, norm.data[601:1200, ])

preds <- unscale(norm.preds, norm.data)

sigs.nn <- trading.signals(preds, 0.02, -0.02)

true.sigs <- trading.signals(Tdata.train[601:1200, "T.ind.CVX"],

0.02, -0.02)

sigs.PR(sigs.nn, true.sigs)

library(e1071)

sv <- svm(Tform, Tdata.train[1:100, ], gamma = 0.001, cost = 100)

s.preds <- predict(sv, Tdata.train[101:200, ])

sigs.svm <- trading.signals(s.preds, 0.1, -0.1)

true.sigs <- trading.signals(Tdata.train[101:200, "T.ind.CVX"],

0.1, -0.1)

sigs.PR(sigs.svm, true.sigs)

library(earth)

e <- earth(Tform, Tdata.train[1:100, ])

e.preds <- predict(e, Tdata.train[101:200, ])

sigs.e <- trading.signals(e.preds, 0.1, -0.1)

true.sigs <- trading.signals(Tdata.train[101:200, "T.ind.CVX"],

0.1, -0.1)

sigs.PR(sigs.e, true.sigs)

plot(sig, type = "l", "col" = "red")

lines(true\_sig, col = "green")

points(true\_sig)

points(true\_sig, col = "green")

points(sig, col = "red")