**Project Report**

**Applied Time Series Analysis**



**Participants**

Name, Surname: **Evgeniy Petrov**  Student ID: **3137211**

Name, Surname: **Jonas Bezler** Student ID: **3140390**

Submission Date: **05.12.2022**

Grade:

# **Exercise Descriptions**

**Exercise 1**

In the csv file dataset<your group number>.csv you have saved a data set, but you forgot the theoretic process you’ve used for generating it. Can you reproduce the formula using your advanced time series skills?

**Exercise 2**

You’ll also find two stock price data

1. Load the data sets to R and perform an adequate preprocessing. Test for missing values and outliers. We’re using the close prices.
2. Which of the two stock prices would you consider as riskier? And why? Elaborate!
3. Choose one of the stocks (any preferences?), and compute the 95% VaR for 4 months in advance. How reliable is this number? Any criticism? Any possible solutions?

**Exercise 3**

Again choose one of the two stock price data. We intend to gamble a bit and expect increasing prices. A European call option would be the right tool for it. We’ve briefly discussed the option during the lecture. Let me please define it again. Let’s assume a strike price **S** which is 10% above the last noted price **P.** Let’s denote the last observed price by . Then the European Call option value for a so called time-to-maturity of **k** is given as

Please estimate the option value using price simulations for a k of 60.

*Hint: You need a simulation of the absolute prices, but you probably simulate the differences or log-differences. How to reconstruct the absolute prices from them?. Simply consider if then . So if you have simulations for the returns you can reconstruct simulations for the absolute prices.*

# **Exercise Solutions**

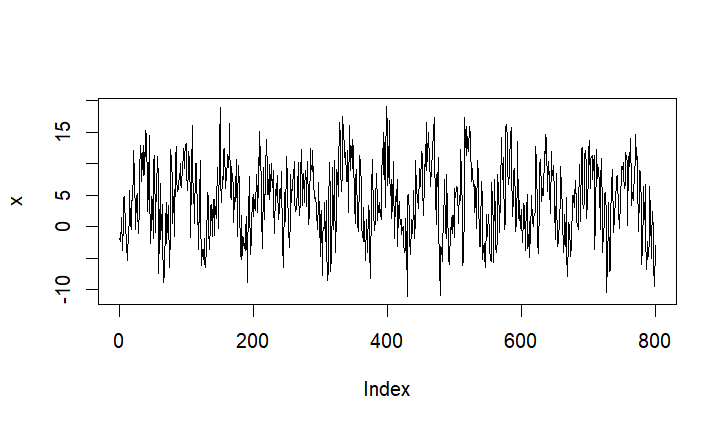
# \* - this symbol means check the corresponding R source code (listed as an appendix)

# **Exercise 1**

\*First, we load up the raw data set into R:

\*Check for NAs -> there are **none**

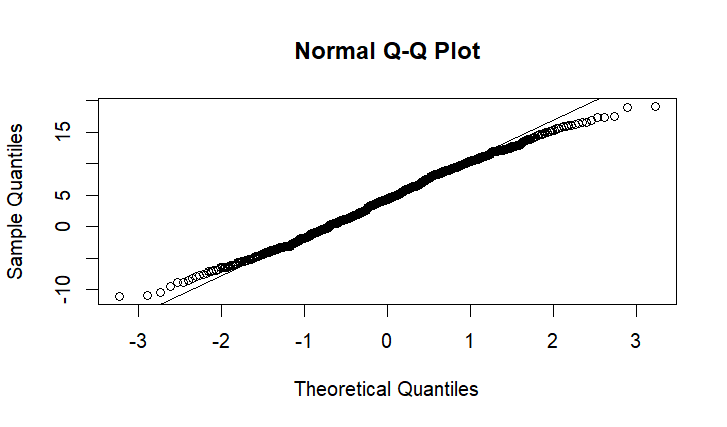
\*Then we plot the data set and we notice that there is a periodic pattern (**Figure 1.1**)



**Figure 1.1**

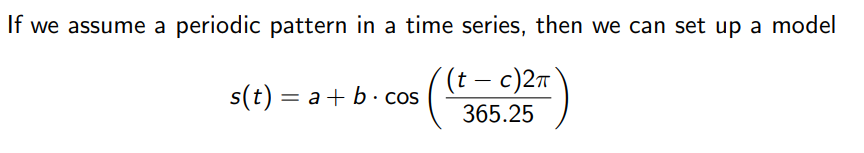
\*We use the **qqnorm** , **qqline** (and additionally **jbTest**), in order to check for the type of the distribution of the dataset (**Figure 1.2**)

\*A conclusion can be made that the dataset is **Gaussian distributed**

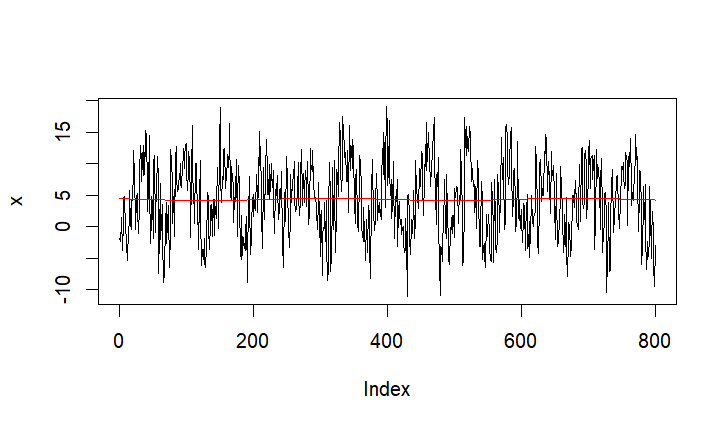


**Figure 1.2**

As we said, by initially visualizing **Figure 1.1** we assumed a periodic pattern, then by definition:



\*For our model we are going to try to fit the **sine function** model instead (**Figure 1.4**)



**Figure 1.4**

\* Using the sine function we cannot see a precise fitting because there is too frequently changing amplitude, because this model cannot handle changes amplitude or period (e.g. cold winter, long summer) (**Figure 1.4**).

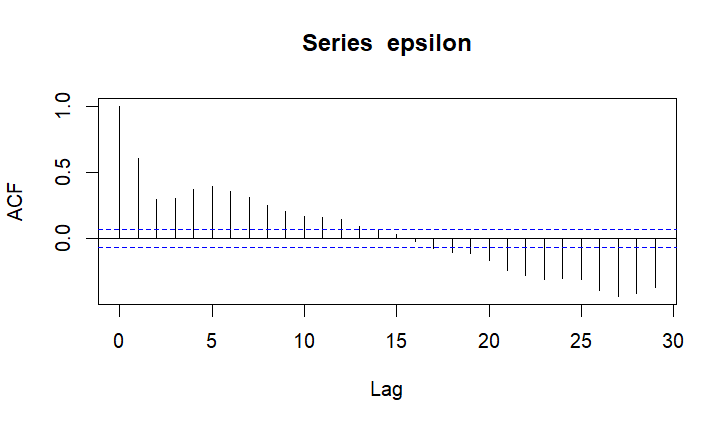
\* The red line indicates the current trend

\*Then we detrend the data



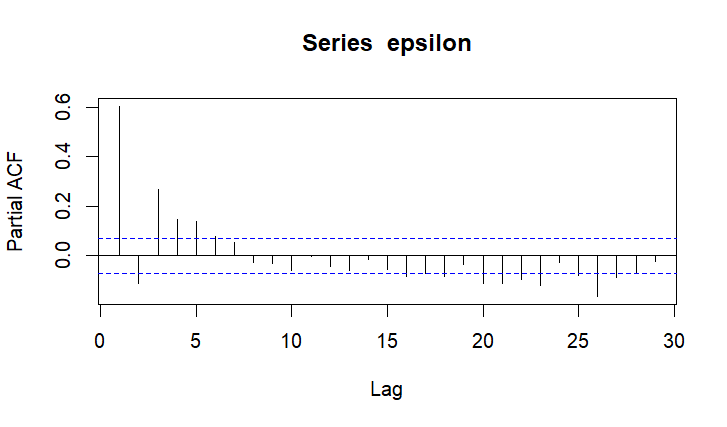
\*We can say that this data does not look like a perfect i.i.d. model

\*As we use the **ACF**, we can notice again a seasonal (periodic) pattern (**Figure 1.5**)



**Figure 1.5**

\*Also, we are going to use the **PACF** in order to check which AR process is appropriate (**Figure 1.6**)



**Figure 1.6**

We notice **more than 5 significant lag values** (the ones exceeding the blue horizontal dash line), but we will go with **AR(5) model**, that means that only five days are the taken into account which is not ideal, but it will work for basic simulation purposes

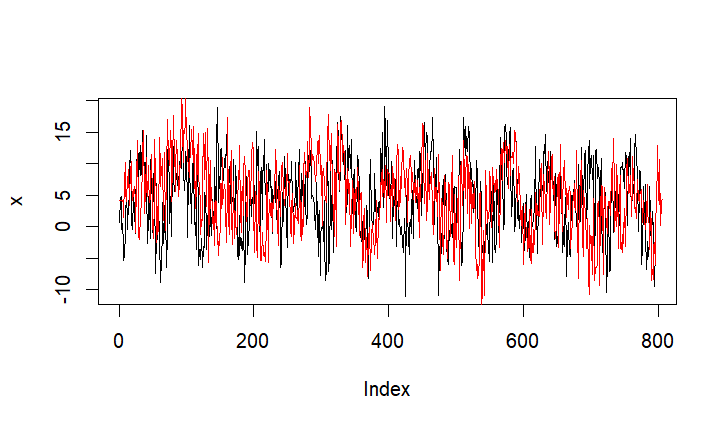
\*We compute **LSE** (Least Square Estimation) for the AR(5) model along with the residuals and its standard deviation

\*Then we calculate the future trend (**see code**)

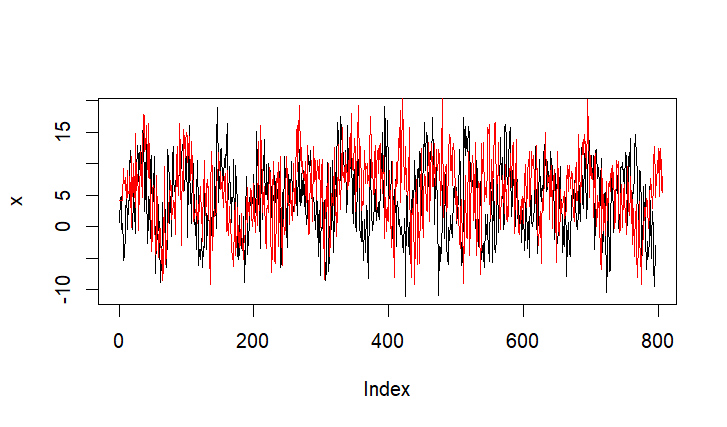
\*Then we simulate the AR(5) process

\*We dont forget to add the future trend to our simulated AR(5) process

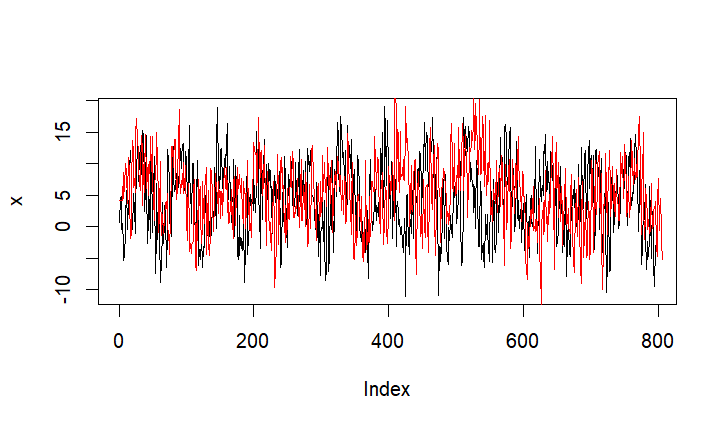
\*Then we plot a few simulations to see what the result is **(Figure 1.7 - Figure 1.9)**:



**Figure 1.7**



**Figure 1.8**



**Figure 1.9**

**Conclusion**: As we continuously run new simulations, we see that we are actually g**etting closer to forecasting the actual data set** (look at the red line). The reason why we do not see a perfect simulation is because of **randomness** and because of the l**imitations of the chosen AR(5) model**. If we want to improve our simulations, we should consider using a different model like AR(6), AR(6) + AR(1), or an ARMA model.

# **Exercise 2**

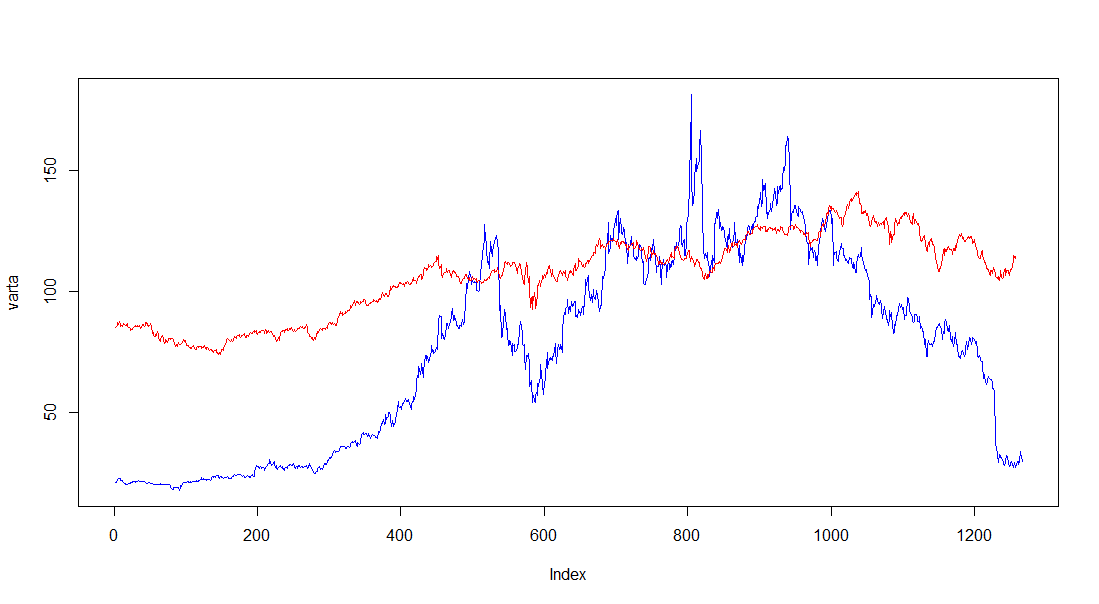
# \***Load** in the .csv datasets into R

\***Check** both data sets for NAs --> **no NAs**

\***Plot** both stock prices for initial comparison (Figure 2.1)

**Nestle** -> **red line**

**Varta** -> **blue line**



**Figure 2.1**

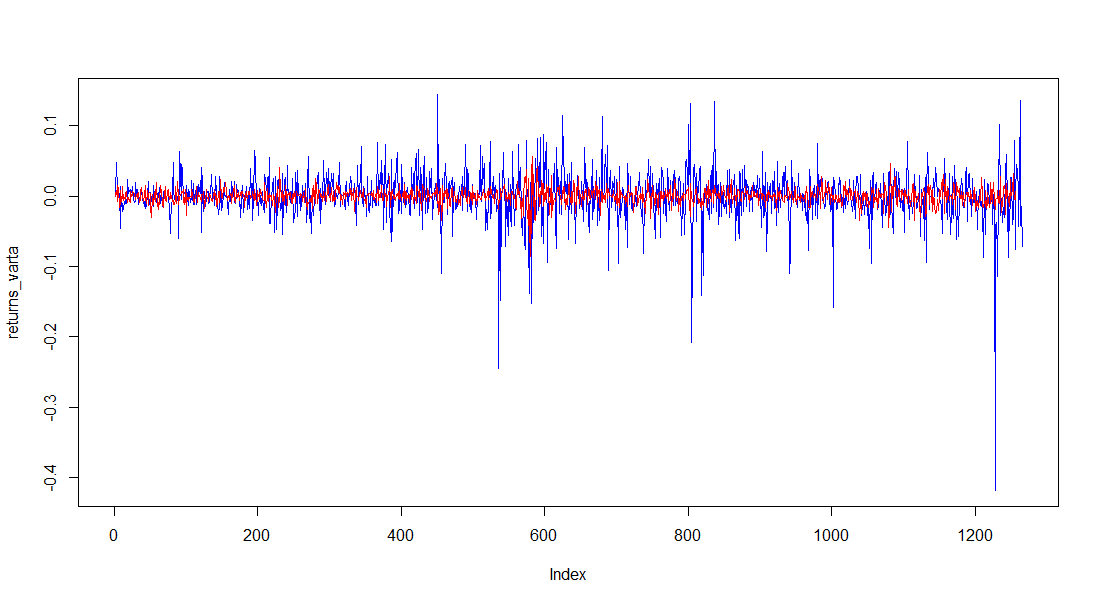
nestle

varta

# By looking at the plot (**Figure 2.1**) we can initially see that **Varta stock prices** appear to be **fluctuative** and therefore are **riskier than Nestles stock prices**, but now we have to prove this point theoretically as well:

\*We **detrend the data** by computing the **log-differences** and after that we **check for any outliers**

\*In **Figure 2.2** we see that such significant outliers are existing and they that have to be removed



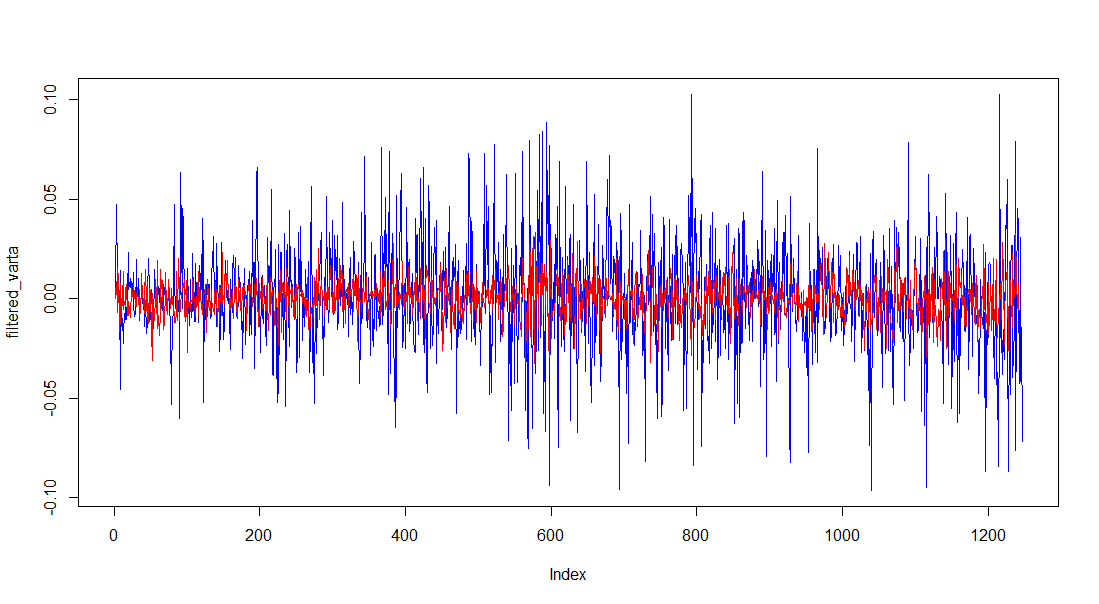
**Figure 2.2**

\*We use the **rule of thumb** and we choose both boundary to be equal to **3** since we do not want to be that strict regarding outliers



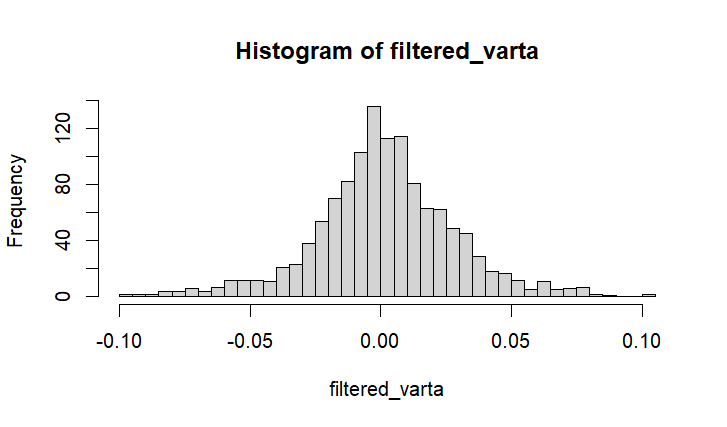
\*We filter the data by removing outliers from the both data sets

**Figure 2.3:** Displays both filtered stock prices (data without outliers)

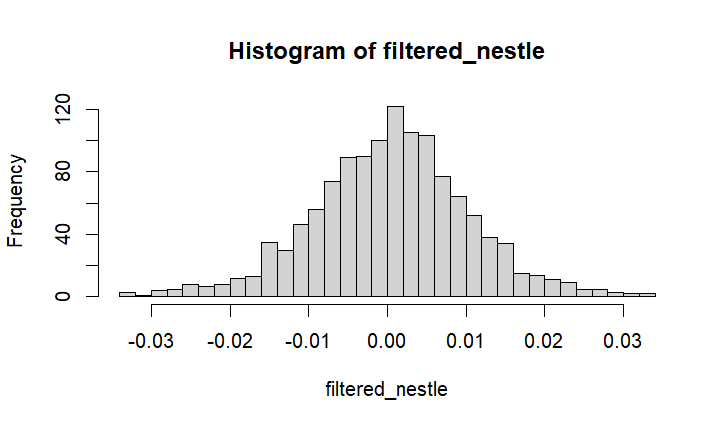


**Figure 2.3**

After taking a careful look at the **filtered stock prices (Figure 2.3)** and the corresponding **histograms** (**Histogram 2.1, Histogram 2.2**), we can see that **returns of Varta are more volatile** and have a higher standard deviation (**0.027\***) than the returns of nestle (sd = **0.01\***) which concludes that Varta appears to be riskier.

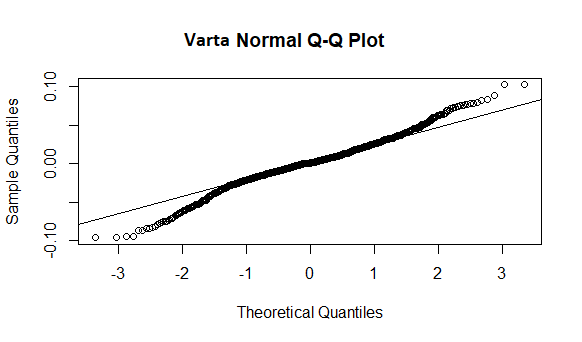


**Histogram - 2.1**

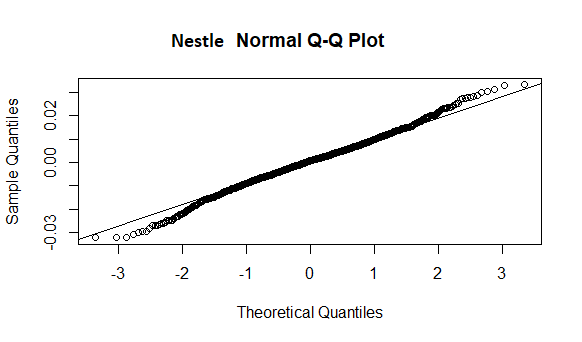


**Histogram - 2.2**

\*To give a forecast for the stock prices we first have to check their distribution. Since the **qqnorm** and **qqline** plots **(Figure 2.4 and 2.5)** are **S-Shaped** (**leptokurtic** distribution) in both instances, we can reason with a **Student-T distribution**.



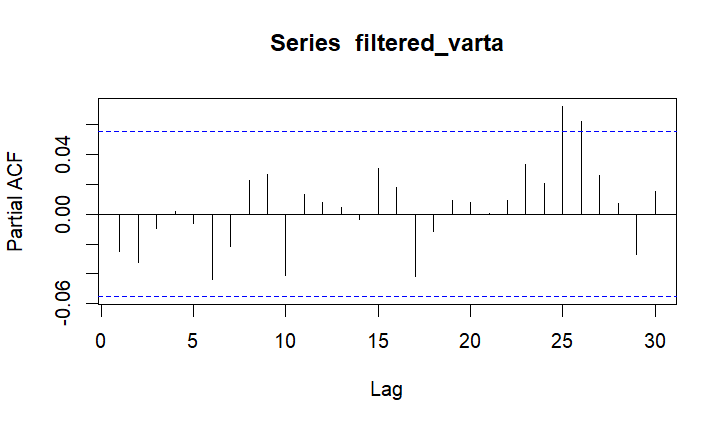
**Figure 2.4**



F**igure 2.5**

\*After looking at the **standard deviation** we choose the **Varta** option since it is higher risk, therefore we can expect higher profits. After calculating **VaR\_emperical** for the **5% quantile** we get output of **-0.0473** which means there is a possible profit of roughly **4.73 %**. We also calculate the **VaR\_theoretical** (**-0.0484**). Therefore, we choose to “stay” with Varta for further simulations.

\*We plot the **PACF (Figure 2.6)** so we can see significant **lag coefficients**

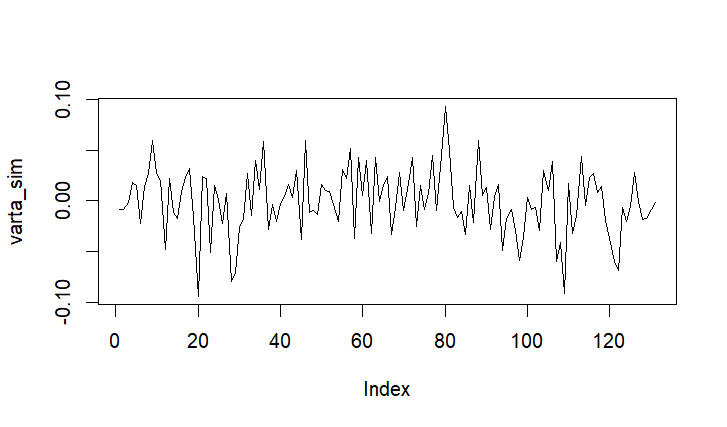


**Figure 2.6**

After analyzing this figure, we notice there are **only two significant lag values (25 and 26)**, therefore we are going to use an **AR(2) model** for further simulations

\*We **generate the AR(2) process** for **125 days** + **27 lag days** using the corresponding “phi” values retrieved from the PACF variable.

\*After **plotting** this **simulated 4 month values**, we notice that there is a **lag** (as expected of 27 days). Then we **cut it off** and get the resulting plot (of our actual simulation, **Figure 2.7**):



**Figure 2.7**

\*Additionally we used the **PACF** again in order prove that we have used the **AR(2) model** with the corresponding lag coefficients (25,26). But we do not get the expected result since the number of simulated values is too low (**check code comments**)

\*Then compute the **Value at Risk (VaR)** only for the simulated 4 months simulation. Here are the results:

**\*VaR\_emperical\_sim = -0.04814** (4.814% profit)

**\*Var\_theoretical\_sim =** -**0.04652** (4.652% profit)

Since we consider the VaR to be not as reliable, we also compute the **Conditional Value at Risk (CVaR)**, also known as the **Expected Shortfall (ES).** The ES is defined as a risk assessment measure that quantifies the amount of tail risk (risk of extreme events) an investment portfolio has. Has a value of: **Expected Shortfall (ES\*)** = **-0.06241 (6.241%),** this number shows us that the possibilities of rare events (**black swan** **events**) are quite high, therefore we should really think about gambling with the Varta stock.

# **Exercise 3**

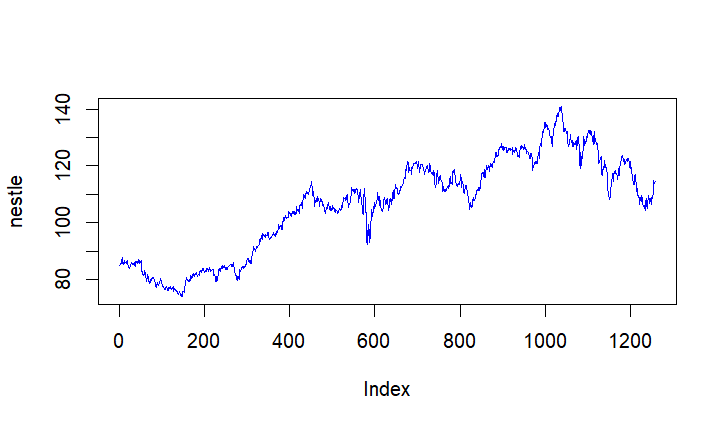
Since we **considered Varta to be riskier**, **we choose** to gamble a bit and expect increasing prices on the safer stock alternative – **Nestle**.

\* **Load** again the raw price data from the Nestle stock

\* We **use the close prices** again

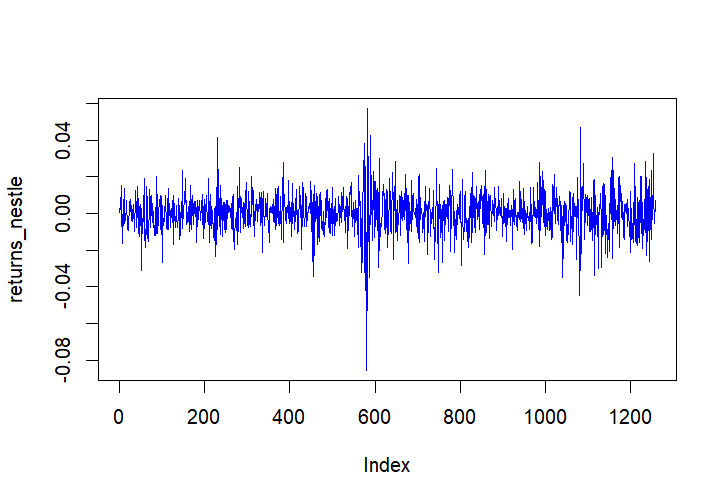
\* Already we have **checked for NAs**, but it’s a good practice to check again -> **No NAs**

\* **Plot** the actual Nestle stock prices **(Figure 3.1**):



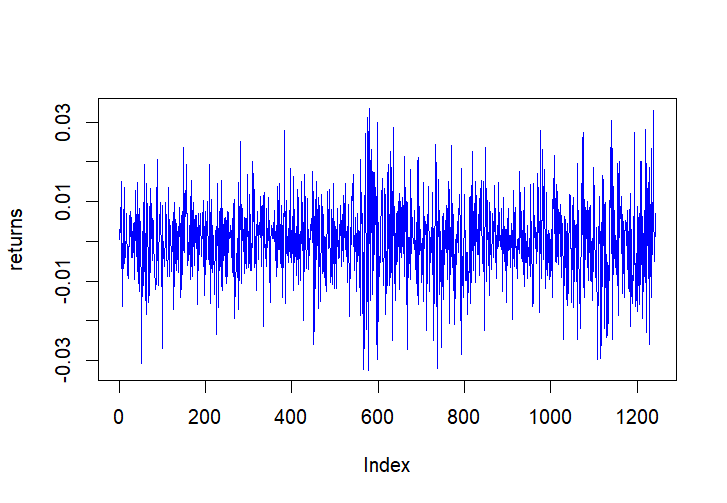
**Figure 3.1**

\* Once again, we **detrend the prices data** using the log-differences (**Figure 3.2**)



**Figure 3.2**

\* As we know we already know **there are outliers that have to filter**, so we do that the same way as we did it on **Exercise 2** (here is a plot of the filtered Nestle returns data, **Figure 3.3**):

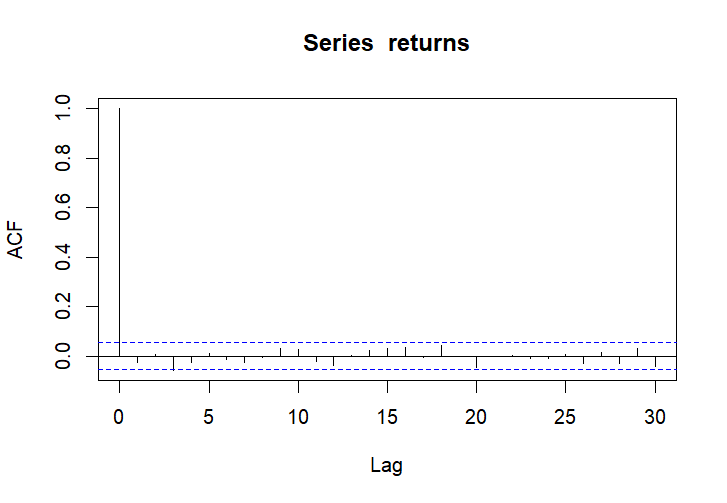


**Figure 3.3**

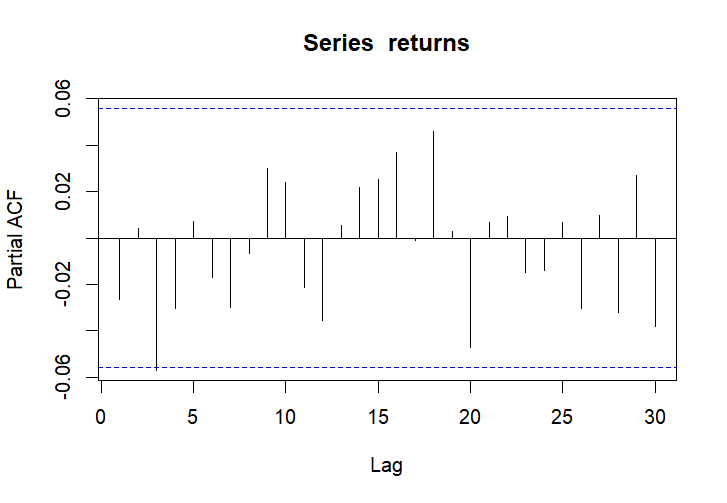
\*We already know that the **filtered Nestle return values** are Student T distributed. Then we get the **df(degrees of freedom value)**  the same way as we did on **Exercise 2**



\*Then we plot the ACF and PACF **(Figure 3.4 and 3.5):**



**Figure 3.4**



**Figure 3.5**

**(Figure 3.4) ACF** -> as expected we have **no significant lag values**

**(Figure 3.5) PACF** -> we have only **one single significant lag value (3)**, therefore we are going to use an **AR(1) model** for further simulations

\*We set **k** to be equal to **60**

\*Again we **get the** **third phi value from the PACF variable** and we **do the AR(1) process**

\*Then a total of **(k + 3)** **values simulation**(+3 days lag) is generated by using the AR(1) process

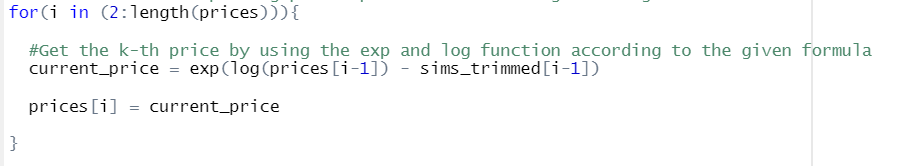
\*We cut off the **3 days lag** (caused by the AR(1) process)

\*We generate a total of **k** prices simulation by:

-**setting the first price\*** to be equal to the **last price from the Nestle prices**



-**computing and setting each next price\*** according to the given equation, as we have simulations for the returns, we reconstruct simulations for the absolute prices and store them into the price vector.



Now that we have generated **k (60)** total Nestle price simulations, we have to compute the European call option value.

\*We set the **strike price S** to be equal to **first starting price + 10 percent increase** (multiply by 1.1)



\*Then we set the option value to be equal to the **difference** between the the **last simulated Nestle price** and the **strike price S;** we also use the max function to filter out when we have a downward trend (negative value)



Now let’s **run a few simulations** to see what the option value would be in different scenarios:



**Figure 3.6**

**Option = 0**



**Figure 3.7**

**Option = 0**

(**Figure 3.6, Figure 3.7**) As we notice a down-trend (**loss or profit < 10%**) the **option value is 0**



**Figure 3.8**

**Option = 5.6242**

(**Figure 3.8**) We see that there is an up-trend (profit >= 10%) the **option value is positive > 0**

**Conclusion:** As we continuously generate new price simulations, we observe that the **option value often has the value of 0,** because an up-trend in which we reach or exceed the **strike price S** is rare to occur for this short, forecasted period of **k = 60**. If we want to see more positive values of the option we have to either i**ncrease the value of k** or l**ower the strike price S** (e.g. make it equal to 5% above the current price).