Prague University of Economics and Business

Faculty of Informatics and Statistics

TIME SERIES OF STOCK PRICES

Semestral Work

Subject: Statistical Methods and Capital Markets 4ST441

Study programme: Economic Data Analysis

Specialization: Data Analysis and Modelling

Author: Artur Dragunov

Prague, May 2023

Contents

Introduction

The pharmaceutical industry has been one of the most critical sectors in the global economy, which faced a tremendous boost during COVID-19. However, after the pandemic severity started decreasing, so did global pharmaceutical company market caps. Investors and financial analysts are always interested in studying the performance of companies in this industry to make informed investment decisions. In our analysis, we study Merck & Co., Inc. (MRK), a global healthcare company that focuses on developing and delivering innovative solutions for the prevention and treatment of diseases. We present an analysis of MRK's stock price behavior using various time series techniques. Specifically, we examine the log returns of MRK's stock price, investigate the stationarity and autocorrelation of the data, and fit multiple prediction models. The analysis is based on daily price data of MRK's stock from January 1, 2019, to December 31, 2022, obtained from Yahoo Finance using the *pdfetch* package in R. This study aimed to provide insights into the behavior of MRK's stock price and to develop a reliable model for predicting its future returns. Analysis was mainly held in R but partially also in Eviews.

The empirical analysis was divided into three parts. First, we visualize our data, demonstrate descriptive statistics of close prices and log returns, and check time series for stationarity and autocorrelation. Next, we analyze close prices of MRK, fit different time series models, calculate and plot the forecast of the best-fitted model. The third part has a similar structure as the second, except that we analyze log returns of MRK stock. In addition, we add conditional heteroscedasticity modeling using Autoregressive Conditional Heteroskedasticity (ARCH) technique. We conclude our empirical analysis with a summary of the paper.

1 Empirical Analysis

1.1 Preliminary steps

Using *pdfetch* library, we downloaded daily time series of MRK stock directly to RStudio. Our analyzed period was 01.01.2019-31.12.2022. As the first step, we plotted MRK close prices with trade volumes below the line chart, see Figure 1.1. MRK stock witnessed a stagnation of around 80$, which should have been a strong resistance level. However, in 2022 stock prices accelerated. Interestingly, the volumes of this stock were very similar throughout time except for a few outliers. The most visible one happened on 01.10.2021 when 102.5 million stocks were traded during one daily session.

This active trading could be caused due to the news that came out during this day. Pharmaceutical company Merck has announced that its experimental oral antiviral medication, molnupiravir, has reduced the risk of hospitalization or death for patients with mild to moderate COVID-19 by 50%, compared to a placebo. The results have led the company to halt recruitment for its phase III study of the drug, and Merck plans to seek emergency-use authorization in the US as soon as possible, in addition to submitting applications to regulatory agencies worldwide. If approved, it could become the first oral antiviral medicine for COVID-19. Molnupiravir is being developed with Ridgeback Biotherapeutics (Merck & Co., 2021).

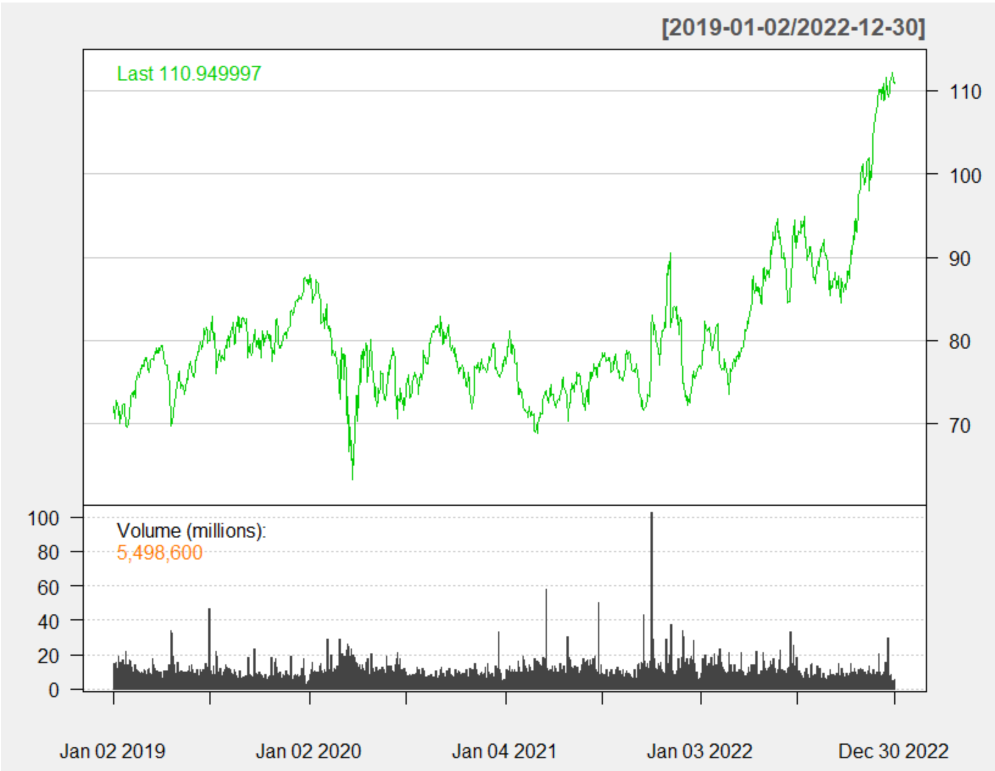


Figure 1.1 MRK close price and volume for period 01.01.2021-31.12.2022 (Source: Yahoo Finance, author)

For the next step, we calculated log returns as a log-transformed ratio of values in period t divided by the value of t-1. In other words, we used Formula 1.1

$$ln(Price_t/Price_{t-1}). \tag{1.1}$$

Plotted log returns can be found in Figure 1.2. It is visible that log returns oscillate around 0 with very similar fluctuations throughout time except for specific outliers, which happened during the beginning of COVID-19, and the second half of 2021. Contrary to the MRK close price, MRK log returns look stationary. However, as the stock return variance changes during the period, the time series seems to be heteroscedastic. We examined this question in the following subchapter.

In the following step, we visualized the distributions of our time series, calculated descriptive statistics, and tested the time series for normality. Figure 1.3 demonstrates that while log returns look symmetric but leptokurtic with some outliers to the right and left of the plot, the close prices of MRK stock seem to have higher skewness but lower kurtosis than log returns. Our finding can be confirmed by Table 1.1, where the mean and median are equal to 0 in case of log returns, while the mean for the close price is higher than the median, indicating a cohort of large values to the right of the plot. Kurtosis of log returns is two times higher than for the normal distribution confirming our words of being leptokurtic.

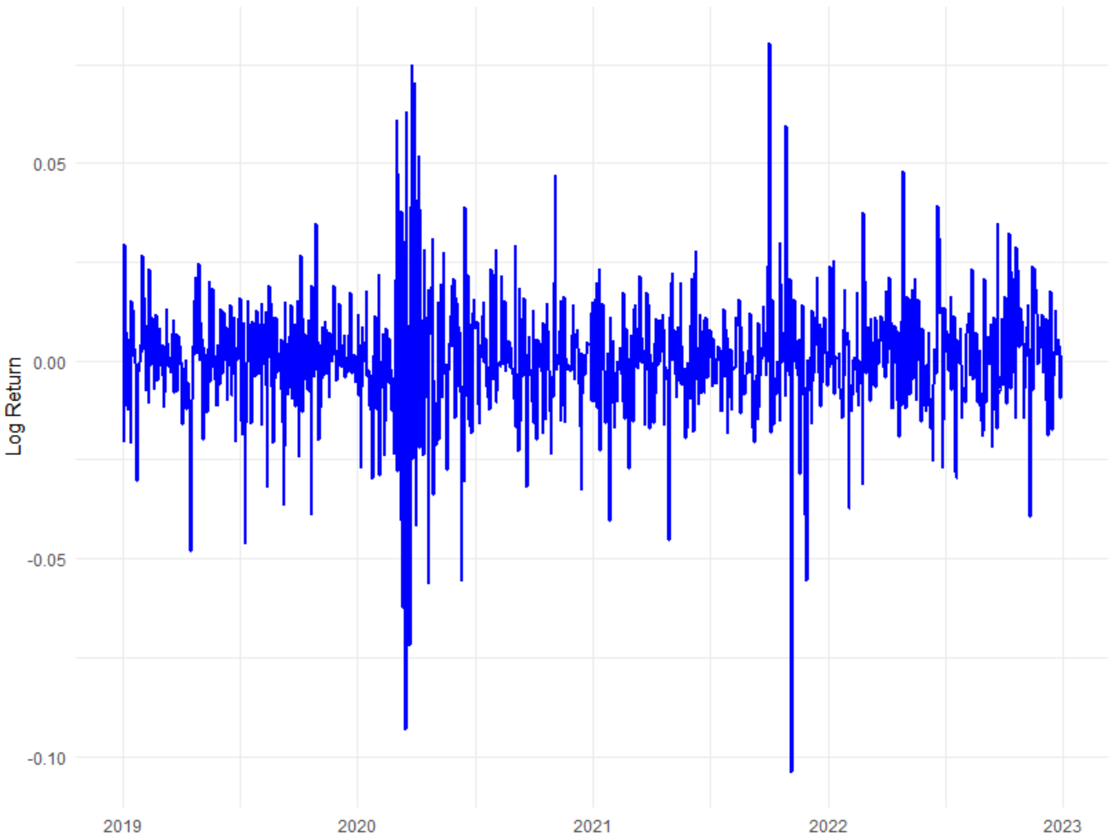


Figure 1.2 Log returns of MRK stock for period 01.01.2019-31.12.2022 (Source: Yahoo Finance, author's calculations)

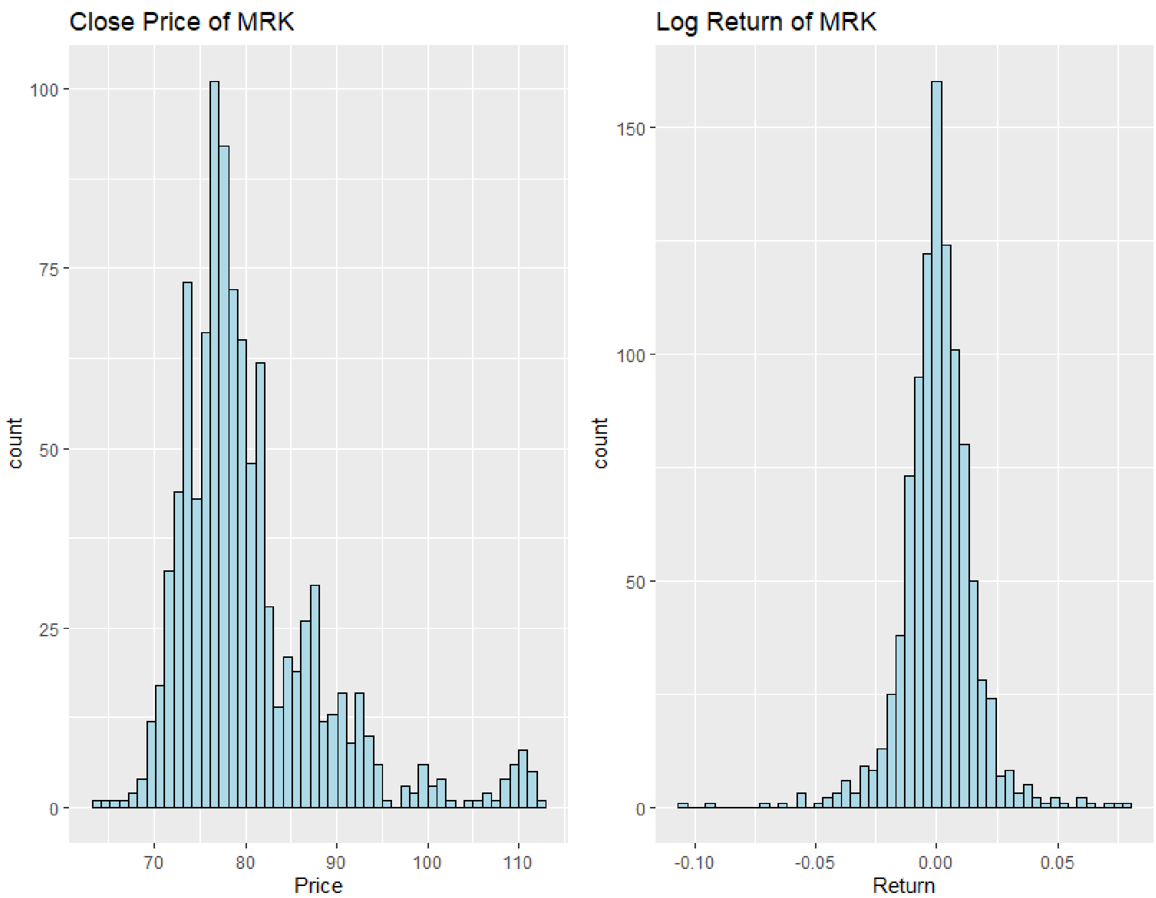


Figure 1.3 Histograms of close prices and log returns of MRK stock for studied period (Source: Yahoo Finance, author's calculations)

Table 1.1 Summary statistics for close price and log returns (Source: Yahoo Finance, author's calculations)

Estimation	**Close Price**	**Log Return**
Count	1008	1007
Mean	80.34	0.00
Standard Deviation	8.00	0.02
Median	78.31	0.00
Min	63.36	-0.10
Max	112.12	0.08
Range	48.76	0.18
Skewness	1.69	-0.36
Kurtosis	3.55	7.05

To formally test our conclusions that both time series were not normally distributed, we ran the Jarque-Bera test with the null hypothesis that distributions of the time series are normal. This goodness-of-fit test checks if a sample of data has skewness and kurtosis that is similar to the normal distribution. Based on the results of *jarque.bera.test()* function from *tseries* library, both time series were not normally distributed as p-values of the chi-squared distributions were below 0.05 significance level; see Appendix 1 for test outputs.

As mentioned before, the close price time series seems non-stationary, but log returns look stationary. We decided to check our assumptions with the Augmented Dickey-Fuller (ADF) test using *the CADFtest* library and same-name function. *CADFtest* has three main parameters: *type*– without constant and trend, with constant, with constant and trend; *criterion* – if a user wants to perform automatic model selection using the specified criterion – in our case, we used Bayesian information criterion (BIC), which penalizes complex models stronger than Akaike information criterion; and *max.lag* – the maximum lag of the differences of the dependent variable. For the *max.lag* parameter, we use a rule of thumb defined as the square root of the length of the dataset rounded to the nearest integer.

Contrary to the ADF test implemented in Eviews, where a user could examine each model's coefficients separately and validate a model with significant coefficients, it is not the case with *the CADFtest* function, whose primary output is the p-value of the model. As close prices had an uptrend, we first tested whether this trend was deterministic or stochastic. H0: The trend is stochastic; H1: The trend is deterministic. In our case, the trend was stochastic. Next, we tested models with and without constants, and both tests confirmed that the data was not stationary. H0: time series is not stationary; H1: time series is stationary. After using the first differences, close prices became stationary. In the case of log returns, the ADF test confirmed that the time series was stationary with the integration of order 0 – I(0). See test outputs in Appendix 2.

Next, we visualized correlograms and partial correlograms of two-time series, see Figure 1.4. Other names for these plots are autocorrelation function (ACF) and partial autocorrelation function (PACF) plots. Plots prove that even though our data is stationary, it is not white noise and possesses strong autocorrelation. We formally prove that autocorrelation exists using Ljung-Box statistics. The function *Box.test()* takes data and the maximum number of lags to test, which we kept the same as for the ADF test, and examines the following relationship:

$$H_0: \rho_1 = \rho_2 = \ldots = \rho_{k_{max}} = 0. \qquad 1\text{-}2$$

Test results prove that both time series had autocorrelation, see Appendix 3.

Both time series had persistent ACF and PACF even for eighth lags, suggesting we try complicated ARIMA model specifications. We decided to use *auto.arima()* function from *the forecast* library, which returns the best ARIMA model according to a specified information criteria value, to save time and reduce the space of the present paper. The function searches for possible models within the order constraints provided.

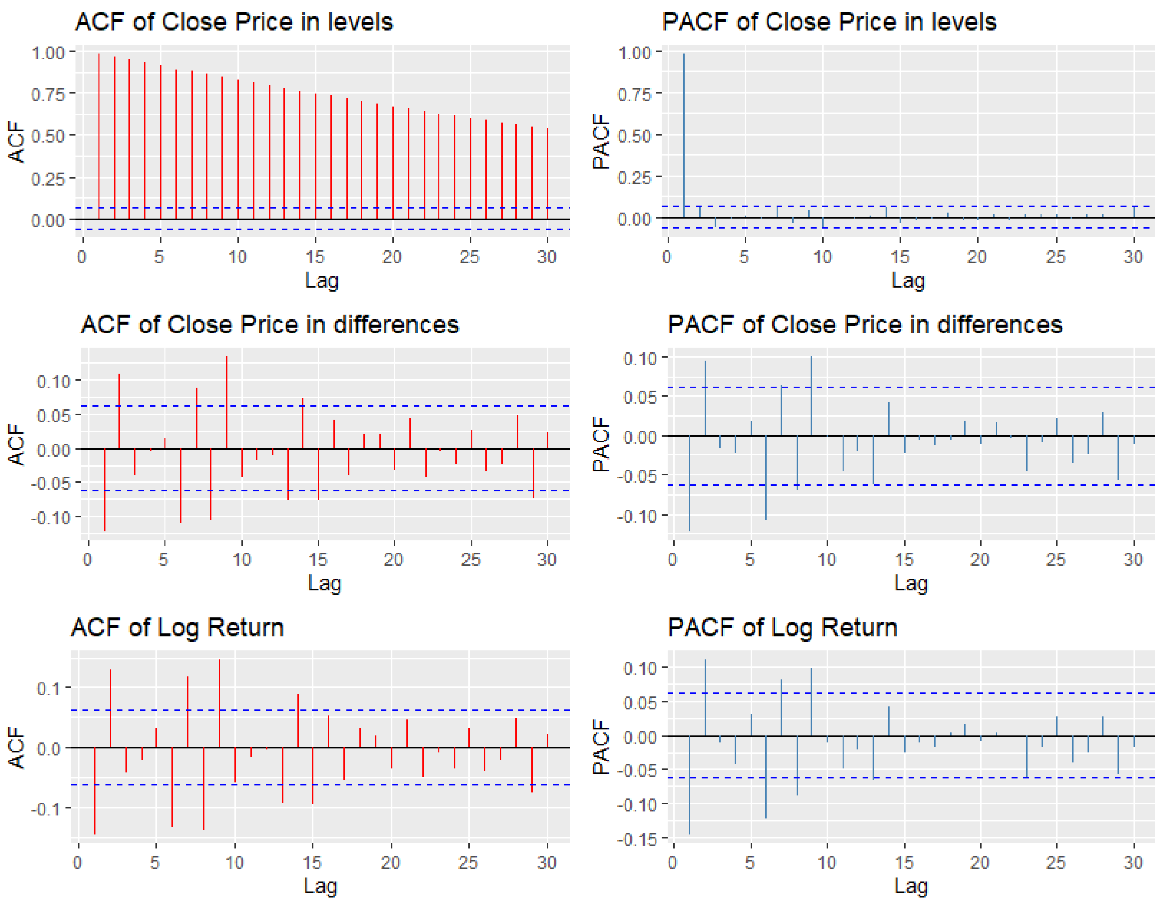


Figure 1.4 ACF and PACF plots of close prices and log returns (Source: Yahoo Finance, author's calculations)

1.2 Analysis of close prices

Auto.arima function output suggested an ARIMA(2,1,2) model with drift for our close prices time series. ARIMA model with drift means a model that includes an additional constant term in the model equation. This term is added to the mean of the time series and represents a long-term trend in the data. However, as the confidence interval (CI) of the drift term included 0[1]; we can conclude that there was not enough evidence to suggest that the drift term was statistically significant in the model. Both autoregressive (AR) and moving average (MA) terms were statistically significant, as they did not have 0 in their confidence intervals.

Auto.arima(), the same way as the usual *arima()* function, uses maximum likelihood estimation (MLE) as the method for estimating parameters. MLE estimates parameters by finding the parameter values that maximize the likelihood function so that the observed data is most probable based on the estimated parameters. Contrary to Eviews, which

[1] standard error (SE) = 0.0348, and drift coefficient is 0.0390. So, CI of drift is 0.0390 +- 2*0.0348 meaning that 0 is inside CI

suggests both least squares and MLE methods and provides information criteria for different models, *arima()* function in R provides information criteria only for the MLE method. So, to choose the best-specified model, we had to use MLE.

Auto.arima() correctly suggested that close prices were I(1) and that they had a complex autocorrelation structure. Next, we validated a model by examining residuals of the time series, see Figure 1.5. Based on the line chart, residuals had heteroscedasticity, as variance was not constant for specific periods. Residuals seemed to be not autocorrelated, which was a good sign that a model managed to explain the behavior of the time series well. The distribution of residuals seems symmetric with a bell curve but with several outliers to the right and left of the main curve. In addition, the distribution looks leptokurtic. Ljung-Box and Jarque-Bera tests confirmed that residuals were white noise but not normally distributed. In addition, we ran an *arch.test* function from the *tseries* library to test for heteroscedasticity. The null hypothesis states that data is homoscedastic. The alternative hypothesis assumes that data were heteroscedastic. Unfortunately, *arch.test* works only with objects created by *arima()* or *estimate()* functions. Thus, we had to recreate a model and fit it to the test manually. Based on both the Portmanteau-Q test and Lagrange-Multiplier multiple tests, residuals were heteroscedastic; see Appendix 5 for the test outputs and a plot of residuals from *arch.test(),* which confirms that residuals were indeed heteroscedastic.

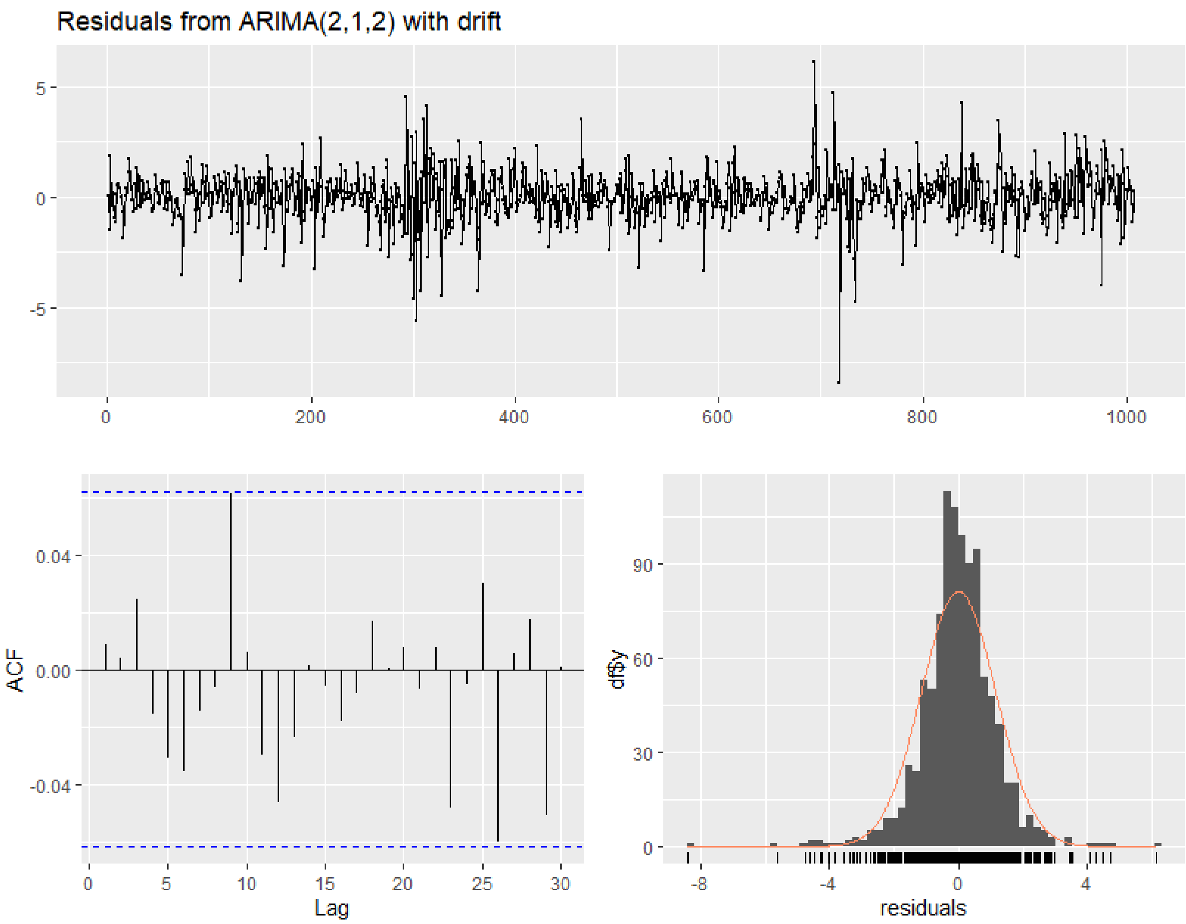


Figure 1.5 Residuals‘ examination for close prices ARIMA model (Source: Yahoo Finance, author’s calculations)

As the last step, we forecasted 20 observations using *the forecast()* function from the *forecast* library. Forecast was very steady, with minor fluctuations and with a minor uptrend. Of course, with the increased time horizon h, prediction intervals also became wider with increased uncertainty, see Figure 1.6.

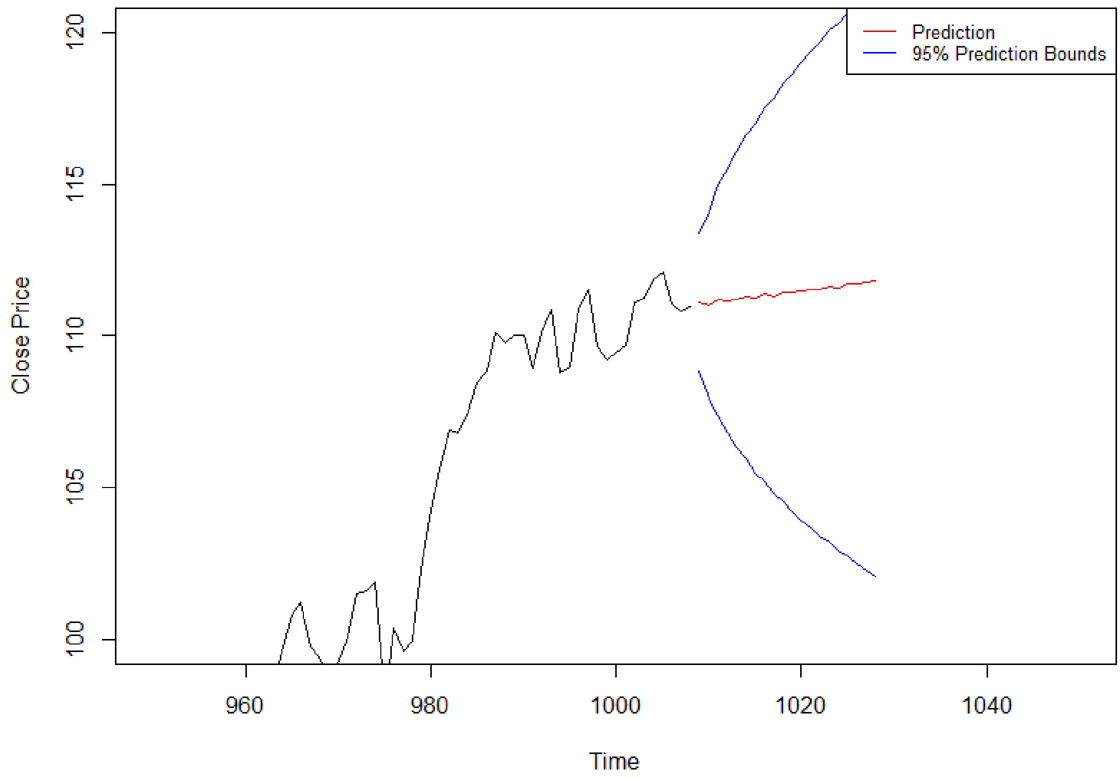


Figure 1.6 MRK Close Price Forecast (Source: Yahoo Finance, author's calculations)

1.3 Analysis of log returns

Analysis of log returns was similar to the one implemented on close prices. We first ran *auto.arima()* function and received a model with the lowest information criteria values. A chosen model was ARMA(4,4). Figure 1.7 represents a plot of residuals against time, ACF of residuals, and a histogram with a kernel function (orange line).

In the same way as for residuals from close prices, residuals of log returns seemed to have no autocorrelation, the distribution looked close to normal but with long tails to the right and left; variance of the residuals was not constant and possessed heteroscedasticity. We proved our words with statistical tests. The Ljung-Box test confirmed that data was white noise, the Jarque-Bera test showed that residuals were not normally distributed, and the ARCH test confirmed heteroscedasticity in the residuals; see Appendix 6 for tests' outputs.

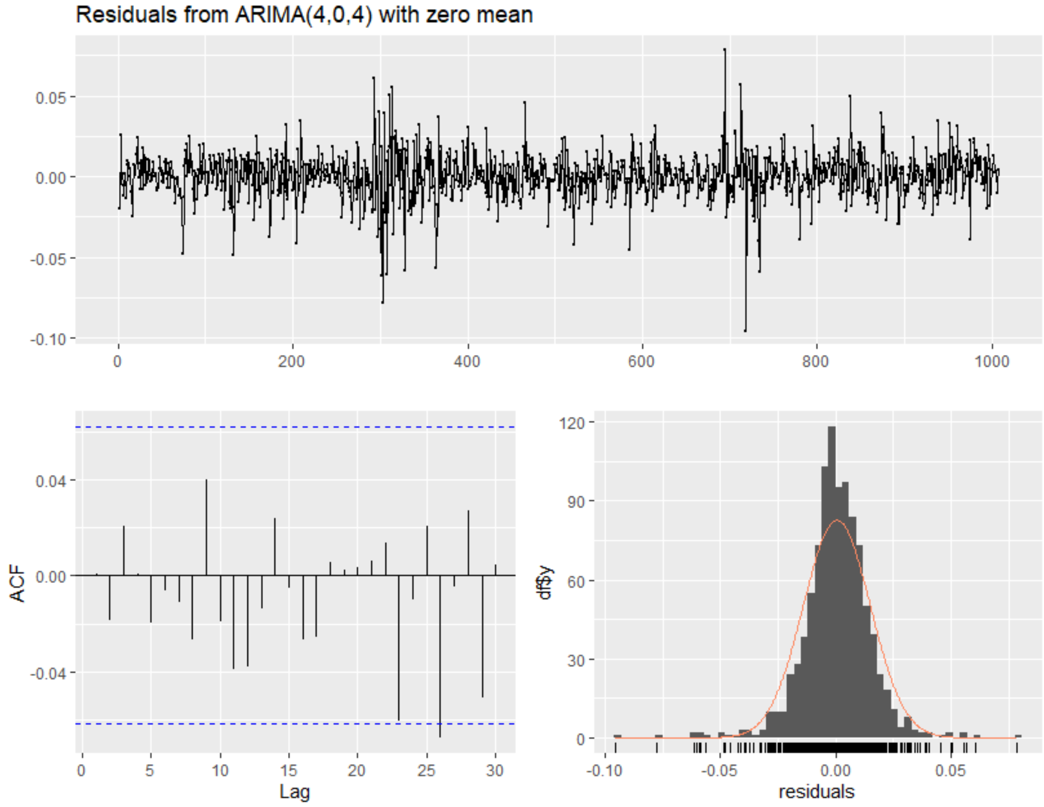


Figure 1.7 Residuals‘ examination of ARIMA model on log returns (Source: Yahoo Finance, author’s calculations)

Next, we forecasted 20 values and plotted them together with the log returns. The red line is our prediction, and the blue lines are 95% prediction intervals. Our prediction has fluctuated much less than the log returns, as all the values were near 0, see Figure 1.8. Our predictions were very possibly inaccurate and moved around a conditional mean.

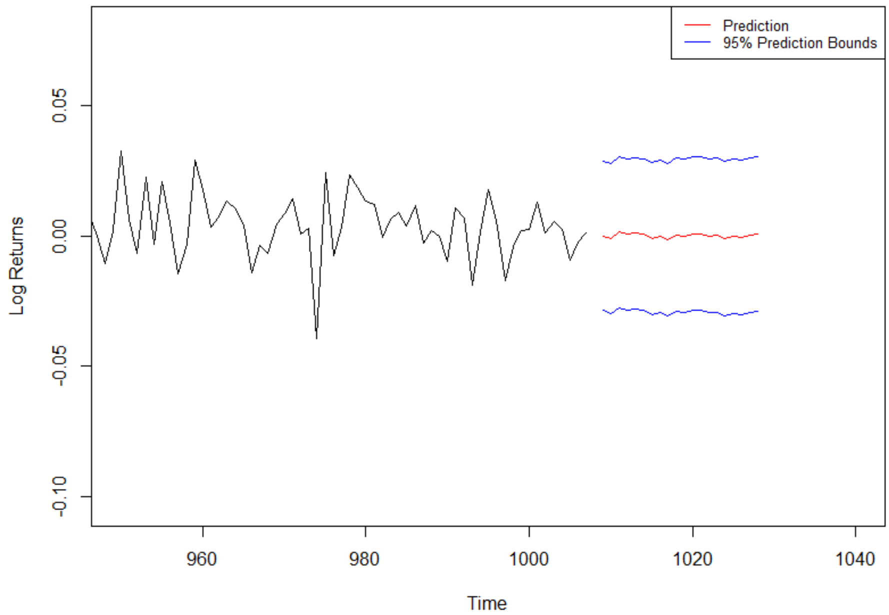


Figure 1.8 MRK Log Returns Forecast (Source: Yahoo Finance, author’s calculations)

We decided to create one more model using ARCH in Eviews. After testing several models with different lags, we decided to stop on a model with the following specifications:

$$r_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \alpha_2 r_{t-2}^2 + \alpha_3 r_{t-3}^2 + \varepsilon_t. \qquad 1\text{-}3$$

Based on the Correlogram – Q statistics from Residual diagnostics, we can confirm that the first four autocorrelations were insignificant. We also confirmed using the ARCH test for heteroscedasticity that residuals were homoscedastic. See model outputs, autocorrelations, and heteroscedasticity tests in Appendix 7. We added two figures to visualize the model's distribution of residuals and plot the model's diagnostics. Figure 1.9 demonstrates that residuals did not have a normal distribution but could possibly be log-normal, as there were many outliers to the right of the plot. Figure 1.10 demonstrates that the volatility of the residuals was constant except few periods. However, the ARCH test for homoscedasticity confirmed that volatility was constant in time.

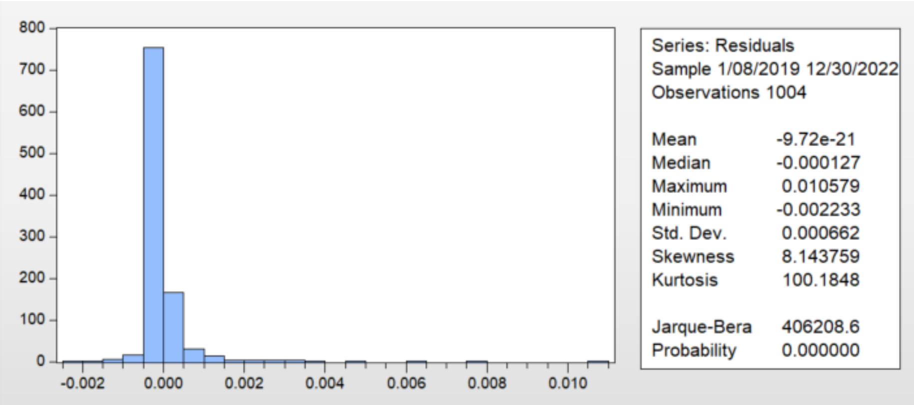


Figure 1.9 Distribution of residuals from ARCH model on log returns (Source: Yahoo Finance, author's calculations)

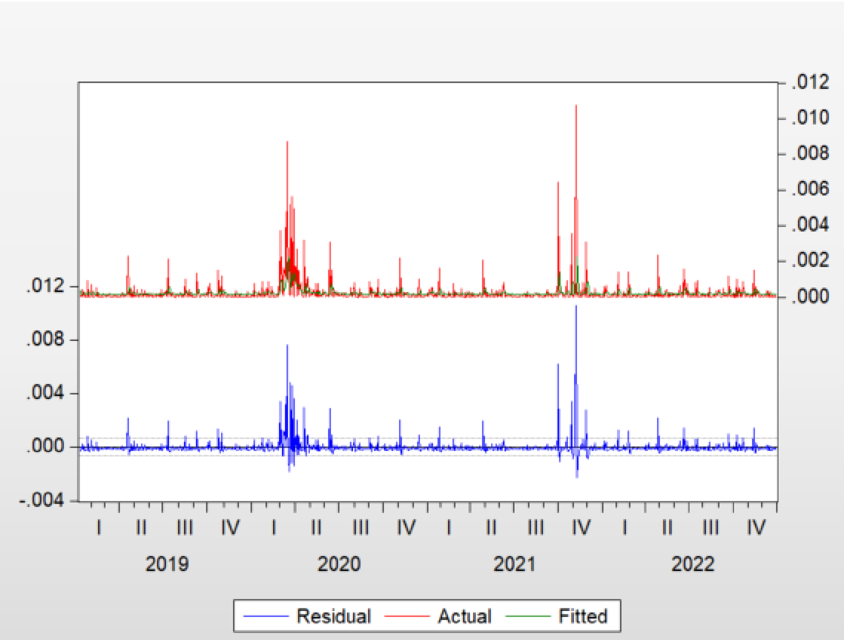


Figure 1.10 Residual, actual, and fitted value diagnostics (Source: Yahoo Finance, author's calculations)

Conclusions

In this paper, we analyzed the behavior of Merck & Co., Inc.'s (MRK) stock price using various time series techniques. We found that the log returns of MRK's stock price exhibit some degree of volatility clustering, which we modeled using the Autoregressive Conditional Heteroskedasticity (ARCH) technique. We found out that the ARCH model with three lags fitted the model very well, as residuals did not have strong autocorrelation and heteroscedasticity.

We also used ARIMA models for log returns and close prices and made 20 days forecasts for these time series. However, it could be confirmed that models had low memory powers, and forecasts showed very low fluctuations. To sum up, the tests and models in this paper can be deepened, and broader research can be done on a more extensive data set with more years of historical data.

List of references

Merck & Co. *Merck and Ridgeback's Investigational Oral Antiviral Molnupiravir Reduced the Risk of Hospitalization or Death by Approximately 50 Percent Compared to Placebo for Patients with Mild or Moderate COVID-19 in Positive Interim Analysis of Phase 3 Study - Merck.com*. (2023, March 22). Merck.com. https://www.merck.com/news/merck-and-ridgebacks-investigational-oral-antiviral-molnupiravir-reduced-the-risk-of-hospitalization-or-death-by-approximately-50-percent-compared-to-placebo-for-patients-with-mild-or-moderat/

Yahoo Finance. *Merck & Co., Inc. (MRK)*. (n.d.). https://finance.yahoo.com/quote/MRK/news?p=MRK

Appendices

Appendix 1: Jarque-bera test outputs constructed in R and examined on close price and log returns of MRK stock time series

```
> jarque.bera.test(na.omit(data$Price)) #Normality Test -> not normal

	Jarque Bera Test

data:  na.omit(data$Price)
X-squared = 1016.3, df = 2, p-value < 2.2e-16

> jarque.bera.test(na.omit(data$Return)) #Normality Test -> not normal

	Jarque Bera Test

data:  na.omit(data$Return)
X-squared = 2121.4, df = 2, p-value < 2.2e-16
```

Appendix 2: ADF test with different model specifications on close prices and log returns.

```
> (max.lag=round(sqrt(length(data$Price)))) # 32
[1] 32
> CADFtest(data$Price, type= "trend", criterion= "BIC", max.lag.y=max.lag) # trend is stochastic

	ADF test

data:  data$Price
ADF(2) = -1.5905, p-value = 0.7964
alternative hypothesis: true delta is less than 0
sample estimates:
       delta
-0.008996372

> CADFtest(data$Price, type= "drift", criterion= "BIC", max.lag.y=max.lag) # data not stationary

	ADF test

data:  data$Price
ADF(2) = -0.73977, p-value = 0.8345
alternative hypothesis: true delta is less than 0
sample estimates:
       delta
-0.003595426

> CADFtest(data$Price, type= "none", criterion= "BIC", max.lag.y=max.lag) # data not stationary

	ADF test

data:  data$Price
ADF(2) = 0.89271, p-value = 0.9006
alternative hypothesis: true delta is less than 0
sample estimates:
       delta
0.0004191467
```

```
> CADFtest(diff(data$Price), type= "drift", criterion= "BIC", max.lag.y=max.lag) # data stationary now

        ADF test

data:  diff(data$Price)
ADF(1) = -21.281, p-value < 2.2e-16
alternative hypothesis: true delta is less than 0
sample estimates:
    delta
-1.018275

> CADFtest(diff(data$Price), type= "none", criterion= "BIC", max.lag.y=max.lag) # data stationary now

        ADF test

data:  diff(data$Price)
ADF(1) = -21.259, p-value < 2.2e-16
alternative hypothesis: true delta is less than 0
sample estimates:
    delta
-1.016177

> CADFtest(data$Return, type= "drift", criterion= "BIC", max.lag.y=max.lag) # log returns are stationary

        ADF test

data:  data$Return
ADF(1) = -21.181, p-value < 2.2e-16
alternative hypothesis: true delta is less than 0
sample estimates:
    delta
-1.022436

> CADFtest(data$Return, type= "none", criterion= "BIC", max.lag.y=max.lag) # log returns are stationary

        ADF test

data:  data$Return
ADF(1) = -21.168, p-value < 2.2e-16
alternative hypothesis: true delta is less than 0
sample estimates:
    delta
-1.020859
```

Appendix 3: Ljung-Box test outputs for autocorrelation.

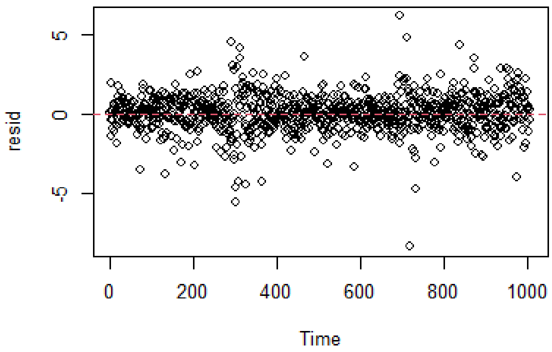
```
> Box.test(dprice$Price, lag = max.lag, type = "Ljung-Box") # Close prices have autocorrelation

        Box-Ljung test

data:  dprice$Price
X-squared = 119.26, df = 32, p-value = 5.52e-12

> Box.test(data$Return, lag = max.lag, type = "Ljung-Box") # have autocorrelation

        Box-Ljung test

data:  data$Return
X-squared = 170.86, df = 32, p-value < 2.2e-16
```

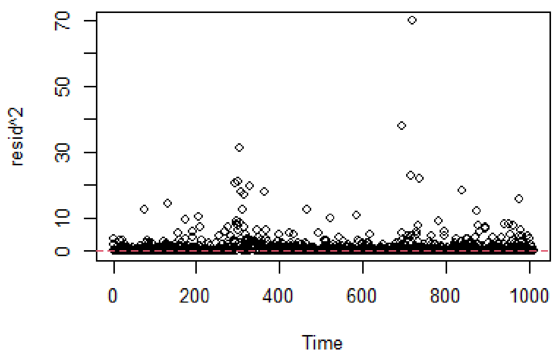
Appendix 4: Auto.arima function output.

```
> auto.arima(data$Price)
Series: data$Price
ARIMA(2,1,2) with drift

Coefficients:
          ar1      ar2     ma1     ma2   drift
      -1.7257  -0.9215  1.6352  0.8476  0.0390
s.e.   0.0309   0.0328  0.0434  0.0475  0.0348

sigma^2 = 1.347:  log likelihood = -1576.44
AIC=3164.88   AICc=3164.96   BIC=3194.37
```

Appendix 5: Ljung-Box and Jarque-Bera tests' outputs on residuals of ARIMA model on close prices.

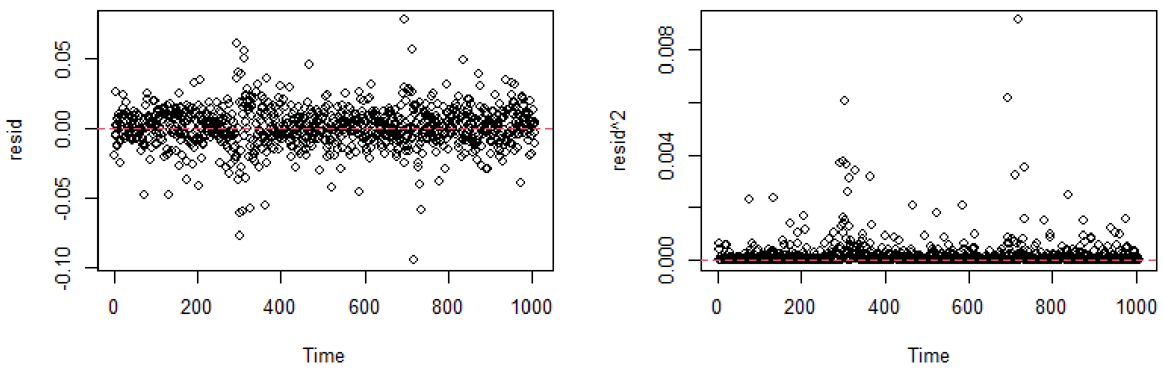
```
> Box.test(fit.close$residuals, lag=max.lag, type="Ljung-Box") # white noise

        Box-Ljung test

data:  fit.close$residuals
X-squared = 23.018, df = 32, p-value = 0.8777

> jarque.bera.test(fit.close$residuals) #Normality Test -> not normal

        Jarque Bera Test

data:  fit.close$residuals
X-squared = 1315.3, df = 2, p-value < 2.2e-16

> arch.test(fit_manual) # hetero
ARCH heteroscedasticity test for residuals
alternative: heteroscedastic

Portmanteau-Q test:
     order    PQ  p.value
[1,]     4  22.7 1.44e-04
[2,]     8  69.8 5.28e-12
[3,]    12  95.2 4.77e-15
[4,]    16 130.4 0.00e+00
[5,]    20 147.1 0.00e+00
[6,]    24 149.1 0.00e+00
Lagrange-Multiplier test:
     order   LM p.value
[1,]     4 1378       0
[2,]     8  561       0
[3,]    12  336       0
[4,]    16  219       0
[5,]    20  168       0
[6,]    24  138       0
```





Appendix 6: Ljung-Box and Jarque-Bera tests' outputs on residuals of ARIMA model on log returns.

```
> Box.test(fit.returns$residuals, lag=max.lag, type="Ljung-Box") # white noise

	Box-Ljung test

data:  fit.returns$residuals
X-squared = 23.019, df = 32, p-value = 0.8777

> jarque.bera.test(fit.returns$residuals) #Normality Test -> not normal

	Jarque Bera Test

data:  fit.returns$residuals
X-squared = 1186.9, df = 2, p-value < 2.2e-16

> arch.test(fit_manual_ret) # hetero
ARCH heteroscedasticity test for residuals
alternative: heteroscedastic

Portmanteau-Q test:
     order    PQ  p.value
[1,]     4  46.3 2.12e-09
[2,]     8 116.9 0.00e+00
[3,]    12 162.0 0.00e+00
[4,]    16 203.5 0.00e+00
[5,]    20 224.9 0.00e+00
[6,]    24 227.4 0.00e+00
Lagrange-Multiplier test:
     order     LM p.value
[1,]     4 -130.9       1
[2,]     8  -64.0       1
[3,]    12  -41.9       1
[4,]    16  -31.0       1
[5,]    20  -24.5       1
[6,]    24  -20.2       1
```



Appendix 7: Autocorrelations, ARCH model summary on log returns, and ARCH heteroscedasticity test.

Dependent Variable: LOG_RET^2
Method: Least Squares
Date: 04/10/23 Time: 13:56
Sample (adjusted): 1/08/2019 12/30/2022
Included observations: 1004 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000136	2.34E-05	5.815701	0.0000
LOG_RET(-1)^2	0.104553	0.031427	3.326896	0.0009
LOG_RET(-2)^2	0.188427	0.031021	6.074272	0.0000
LOG_RET(-3)^2	0.110934	0.031414	3.531301	0.0004

R-squared	0.076859	Mean dependent var	0.000228
Adjusted R-squared	0.074089	S.D. dependent var	0.000690
S.E. of regression	0.000663	Akaike info criterion	-11.79417
Sum squared resid	0.000440	Schwarz criterion	-11.77460
Log likelihood	5924.675	Hannan-Quinn criter.	-11.78674
F-statistic	27.75254	Durbin-Watson stat	2.010884
Prob(F-statistic)	0.000000		

Date: 04/10/23 Time: 13:58
Sample (adjusted): 1/08/2019 12/30/2022
Q-statistic probabilities adjusted for 3 dynamic regressors

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob*
		1	-0.006	-0.006	0.0313	0.860
		2	-0.032	-0.032	1.0782	0.583
		3	-0.055	-0.055	4.1382	0.247
		4	-0.002	-0.003	4.1405	0.387
		5	0.166	0.163	32.131	0.000
		6	0.053	0.054	35.014	0.000
		7	0.044	0.057	37.011	0.000
		8	0.108	0.135	48.912	0.000
		9	0.011	0.027	49.031	0.000

Heteroskedasticity Test: ARCH

F-statistic	0.532383	Prob. F(1,1001)	0.4658
Obs*R-squared	0.533163	Prob. Chi-Square(1)	0.4653

Test Equation:
Dependent Variable: RESID^2
Method: Least Squares
Date: 04/10/23 Time: 14:00
Sample (adjusted): 1/09/2019 12/30/2022
Included observations: 1003 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	4.29E-07	1.39E-07	3.089742	0.0021
RESID^2(-1)	0.023056	0.031599	0.729646	0.4658

R-squared	0.000532	Mean dependent var	4.39E-07
Adjusted R-squared	-0.000467	S.D. dependent var	4.37E-06
S.E. of regression	4.37E-06	Akaike info criterion	-21.84073
Sum squared resid	1.91E-08	Schwarz criterion	-21.83093
Log likelihood	10955.12	Hannan-Quinn criter.	-21.83701
F-statistic	0.532383	Durbin-Watson stat	2.003703
Prob(F-statistic)	0.465777		

Appendix 8: R script with all the steps implemented in the present seminar paper.

```
rm(list=ls())
library(quantmod)
library(dplyr)
library(PerformanceAnalytics)
library(ggplot2)
library(xts)
library("fable")
library("lubridate")
library("gridExtra")
library(tseries)
library(forecast)
library(rugarch)
library(CADFtest)
```

```
library(fGarch)
library(psych)
library(aTSA)
library(pdfetch)

mydata_xts = pdfetch_YAHOO(
  'MRK',
  fields = c("open", "high", "low", "close", "adjclose", "volume"),
  from = as.Date("2019-01-01"),
  to = as.Date("2022-12-31"),
  interval = "1d"
)

colnames(mydata_xts) <- c("MRK.Open"   ,  "MRK.High"   ,  "MRK.Low"   ,   "MRK.Close"  ,
"MRK.Adjusted","MRK.Volume")
lineChart(mydata_xts,theme = 'white', TA = c(addVo()), name = '', minor.ticks = FALSE)


#Data Frame
data <- cbind(
  Price = mydata_xts$MRK.Close,
  Return=CalculateReturns(mydata_xts$MRK.Close, method = 'log'))  #Calculating Returns and
transform into log values
colnames(data) <- c('Price','Return')
head(data)
ggplot(data, aes(x = index(data), y = Return)) +
  geom_line(color = "blue", size = 1) +
  labs(x="",y = "Log Return") +
  theme_minimal()

#Distributions and statistics
describe(data, skew=TRUE,omit=TRUE)
histprice = ggplot(aes(Price), data=data) + geom_histogram(col='black',fill='lightblue',
bins=50) + ggtitle('Close Price of MRK')
histreturn = ggplot(aes(Return), data=data) +
geom_histogram(col='black',fill='lightblue',bins=50) + ggtitle('Log Return of MRK')
grid.arrange(histprice,histreturn, ncol = 2, nrow = 1)
jarque.bera.test(na.omit(data$Price)) #Normality Test -> not normal
jarque.bera.test(na.omit(data$Return)) #Normality Test -> not normal

############################
#Stationarity and autocorrel#
############################
(max.lag=round(sqrt(length(data$Price)))) # 32
```

```
CADFtest(data$Price, type= "trend", criterion= "BIC", max.lag.y=max.lag) # trend is
stochastic
CADFtest(data$Price, type= "drift", criterion= "BIC", max.lag.y=max.lag) # data not
stationary
CADFtest(data$Price, type= "none", criterion= "BIC", max.lag.y=max.lag) # data not
stationary
CADFtest(diff(data$Price), type= "drift", criterion= "BIC", max.lag.y=max.lag) # data
stationary now
CADFtest(diff(data$Price), type= "none", criterion= "BIC", max.lag.y=max.lag) # data
stationary now

dprice = diff(data$Price)
plot(dprice) # looks stationary. what about white noise?

CADFtest(data$Return, type= "drift", criterion= "BIC", max.lag.y=max.lag) # log returns
are stationary
CADFtest(data$Return, type= "none", criterion= "BIC", max.lag.y=max.lag) # log returns are
stationary

#Charts
acfclose<- ggAcf(na.omit(data$Price), col='red',main='ACF of Close Price in levels')
pacfclose<- ggPacf(na.omit(data$Price),col='steelblue',main='PACF of Close Price in
levels')
acfdclose<- ggAcf(na.omit(diff(data$Price)), col='red',main='ACF of Close Price in
differences')
pacfdclose<- ggPacf(na.omit(diff(data$Price)),col='steelblue',main='PACF of Close Price in
differences')
acfreturn<- ggAcf(na.omit(data$Return), col='red',main='ACF of Log Return')
pacfreturn<- ggPacf(na.omit(data$Return),col='steelblue',main='PACF of Log Return')
grid.arrange(acfclose, pacfclose,acfdclose,pacfdclose,acfreturn,pacfreturn, ncol = 2, nrow
= 3)
Box.test(dprice$Price, lag = max.lag, type = "Ljung-Box") # Close prices have
autocorrelation
Box.test(data$Return, lag = max.lag, type = "Ljung-Box") # have autocorrelation

#############
#Close Price#
#############
# We know that it's not stationary. Hence, we took first diff
# ACF and PACF suggest the same model specifications as for log returns
# so, we try AR(3), MA(3), ARIMA(3)
fit.close = auto.arima(data$Price)
checkresiduals(fit.close)

# Let's now perform the formal test for white noise-> the Q-test or Ljung-Box test.
```

```
Box.test(fit.close$residuals, lag=max.lag, type="Ljung-Box") # white noise
jarque.bera.test(fit.close$residuals) #Normality Test -> not normal
# looks close to normal but failed to pass jarque bera test
# Fit ARIMA model manually
fit_manual <- arima(data$Price, order = c(2,1,2))
summary(fit_manual)
arch.test(fit_manual) # hetero
forecast.close <- forecast::forecast(fit.close,h=20)
close_ts <- ts(data$Price["2019-01-02/"])
plot(close_ts, main = "", ylab = "Close Price",xlim=c(950,1050),ylim=c(100,120)) # MRK
Close Price Forecast
lines(forecast.close$mean, col = "red")
lines(forecast.close$lower[,'95%'], col = "blue")
lines(forecast.close$upper[,'95%'], col = "blue")
legend("topright", legend=c("Prediction", "95% Prediction Bounds"),
       col=c("red", "blue"), lty=1, cex=0.8)

#############
#Log Returns#
#############

fit.returns = auto.arima(data$Return)
fit.returns
checkresiduals(fit.returns)

# Let's now perform the formal test for white noise-> the Q-test or Ljung-Box test.
Box.test(fit.returns$residuals, lag=max.lag, type="Ljung-Box") # white noise
jarque.bera.test(fit.returns$residuals) #Normality Test -> not normal
# looks close to normal but failed to pass jarque bera test
# Fit ARIMA model manually
fit_manual_ret <- arima(data$Return, order = c(4,0,4),include.mean = FALSE)
summary(fit_manual_ret)
arch.test(fit_manual_ret) # hetero
# Forecasts
forecast.returns <- forecast::forecast(fit.returns,h=20)
return_ts <- ts(data$Return["2019-01-03/"])
plot(return_ts, main = "", ylab = "Log Returns",xlim=c(950,1040)) # MRK Log Returns
Forecast
lines(forecast.returns$mean, col = "red")
lines(forecast.returns$lower[,'95%'], col = "blue")
lines(forecast.returns$upper[,'95%'], col = "blue")
legend("topright", legend=c("Prediction", "95% Prediction Bounds"),
       col=c("red", "blue"), lty=1, cex=0.8)
```