Department of Electrical and Electronic Engineering
Imperial College London
FSPML Report

Financial Signal Processing Coursework

Author: Fabio Deo
Course: EIE4
CID: 01338063
Lecturer: Prof. Danilo Mandic

Contents

1 Regression methods

In this section, regression methods are used to analyse the performance of some of the S&P 500 stock market listed companies.

1.1 Processing stock price data in Python

In a consistent manner throughout the whole project, stock market data has been pre-processed in order to remove any unexpected data (NaNs) in order to make it compatible with the further data analysis performed.

1.1.1 Logarithmic prices

The natural logarithm of the SPX price data time-series is computed in this section as the logarithm allows compacts prices in a smaller range, hence easing subsequent data manipulation. Further advantages of this technique will be discussed in greater detail later in this section. Figure 1.1.1 shows the SPX Index linear and logarithmic time evolution, respectively.

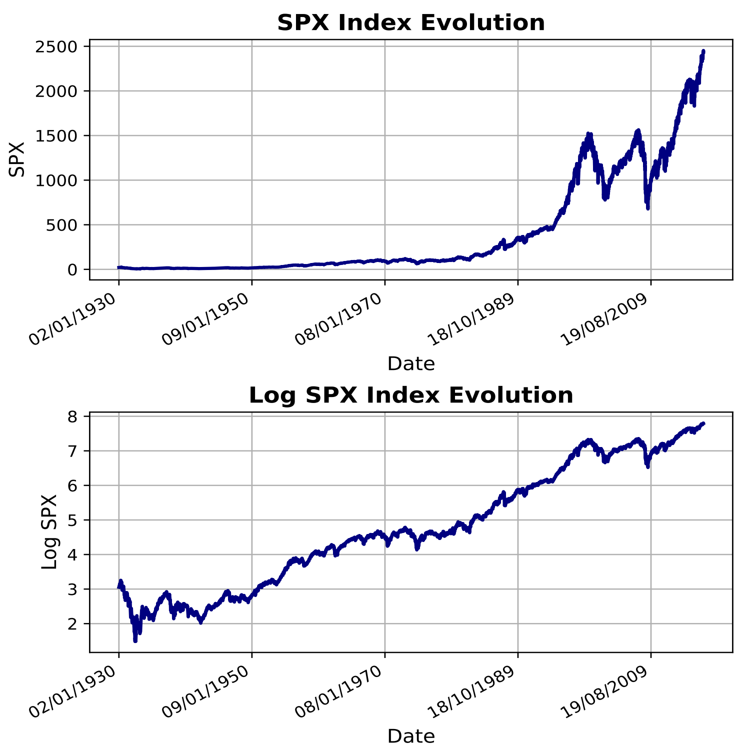

Figure 1.1.1: SPX log prices.

1.1.2 Stationarity analysis using rolling statistics

A stochastic signal is stationary if, when shifted in time, its unconditional joint probability distribution does not change and consequently its first moment is time-invariant. Figure 1.1.2 shows the mean and variance of the price and log-price for a sliding window of 252 days and 1-day increments.

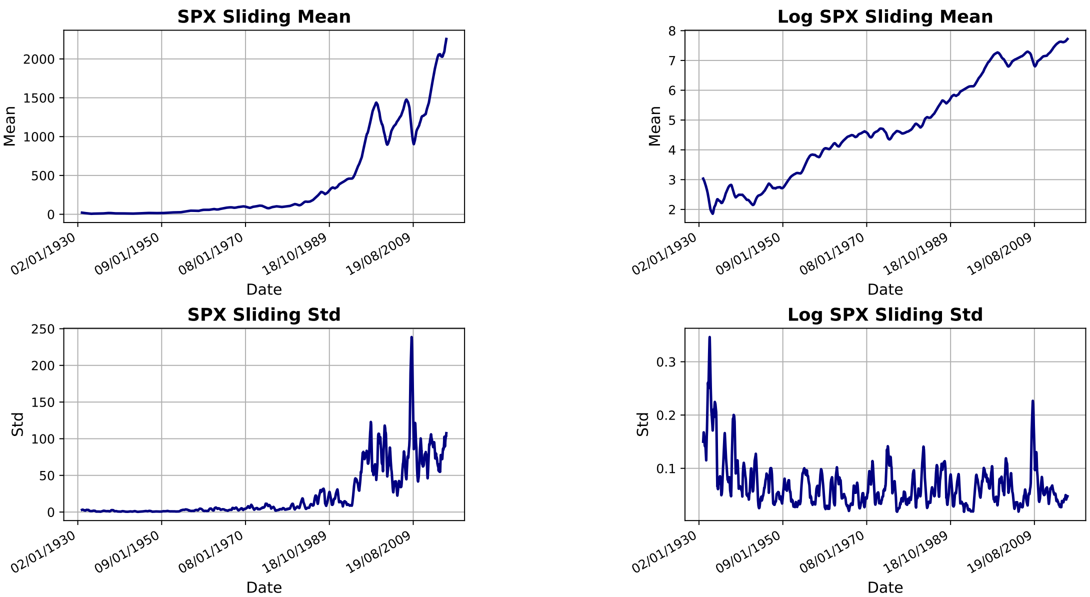

Figure 1.1.2: Mean and standard deviation analysis of the SPX prices and log-prices.

While the sliding mean performs a smoothing operation on the signal, the standard deviation gives information on the price volatility. The plots in Figure 1.1.2 imply non-stationarity as the moving average follows an upwards trend with respect to time. Therefore both the price and the log-price time-series do not have time-invariant statistics, i.e. they are non-stationary stochastic processes.

1.1.3 Compute simple and log-returns

The Simple Return is defined as:

$$R_t = \frac{p_t}{p_{t-1}} - 1 \tag{1.1.1}$$

where p_t is the price of a certain asset at time instant t and R_t is the simple return at the same time instant t. Similarly, the logarithmic return is defined as:

$$r_t = \log(p_t) - \log(p_{t-1}) \tag{1.1.2}$$

where p_t is the price of a certain asset at time instant t and r_t is the log-return at the same time instant t. Figure 1.1.3 shows the sliding statistics analysis on simple and log-return prices. Comparing the plots in Figure 1.1.3 to the ones discussed in Section 1.1.1, both the simple and log-return means show an oscillatory behaviour around zero that implies more stationarity than observed for price statistics.

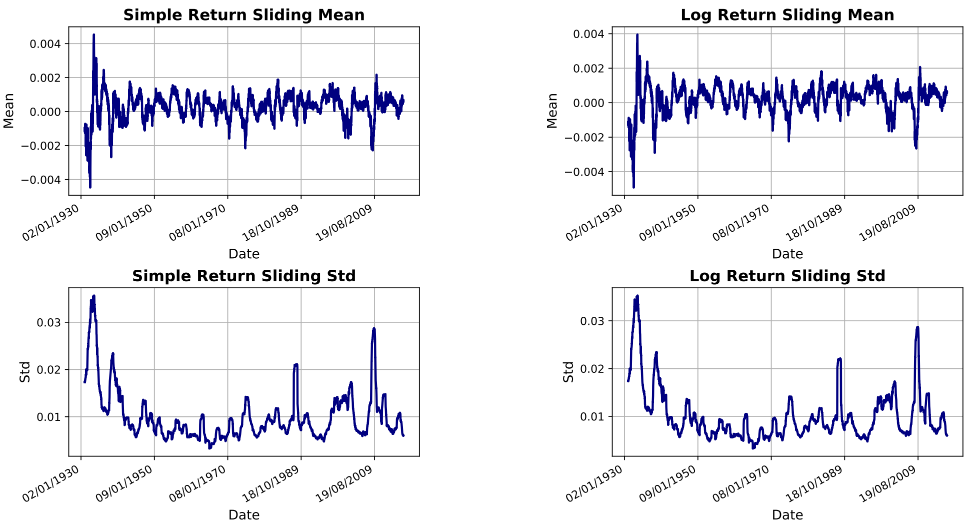

Figure 1.1.3: Mean and standard deviation analysis of simple return and log-return.

1.1.4 Advantages of log-returns and Jarque-Bera test

Considering the logarithmic return has several advantages over using the simple return. First, the logarithmic function is monotonic, which means that the relative order is preserved, i.e. for all a and b such that $a > b$ then $\log(a) > \log(b)$. Furthermore, prices are log-normally distributed over fairly short time periods, therefore the return at time t (r_t) is normally distributed. Gaussianity is a fundamental advantage as several signal processing and statistical techniques require it. Finally logarithms as a mathematical operator are more convenient when performing advanced calculus and provide better numerical stability. In this experiment, we can observe the gaussian-like behaviour of returns by plotting histograms showing both the simple return and log-return distribution, as shown in Figure 1.1.4.

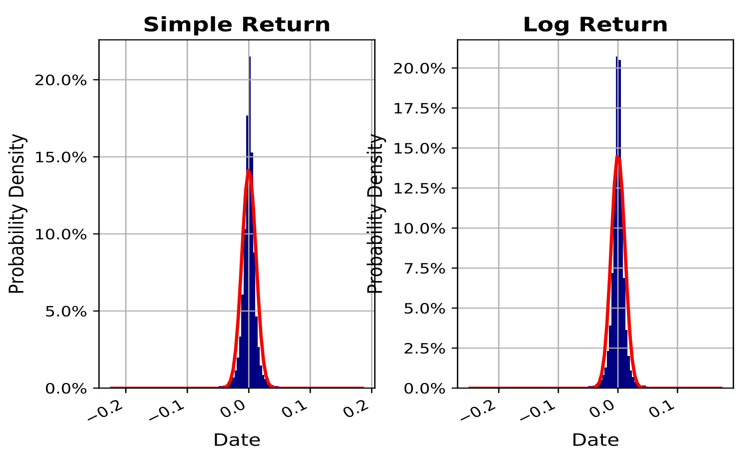

Figure 1.1.4: Distribution of simple and log-returns.

Gaussianity of data cannot only be appreciated through visual inspection, but also estimated using the Jarque-Bera goodness-of-fit test. This statistic quantifies how much the skewness and kurtosis of the sample data, i.e. symmetries and shape of the curve, match the ones of a normal distribution with zero skewness and kurtosis equal to three. Figure 1.1.5 shows the JB curve with respect to the number of data points considered, clearly showing that the difference among the two curves increases as the number of data points increases. Log-returns are proven to deviate more slowly from normality, hence making them more suitable for several statistical analysis.

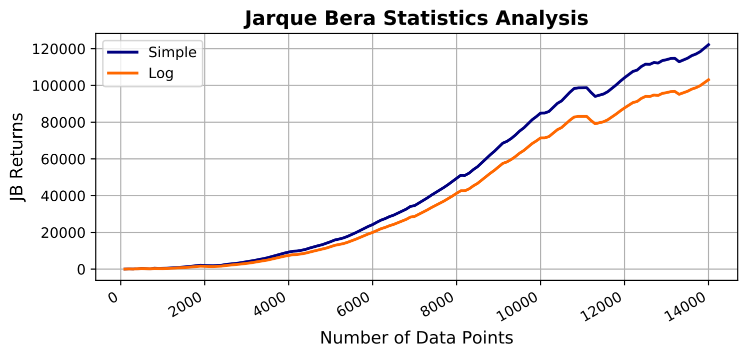

Figure 1.1.5: Jarque-Bera statistic for simple and log-returns as the number of datapoints increases.

1.1.5 Stock purchase example

You purchase a stock for £1. The next day its value goes up to £2 and the following day back to £1. What are the simple and logarithmic returns over this period and what can you conclude about logarithmic returns on the basis of this example?

The example above shows why log-returns are a better metric at describing the changes in value of an asset over time. Overall the value of the stock has not changed, given that eventually falls back to £1. However looking only at the simple returns ($[1, -0.5]$), their sum does not add up to zero. On the contrary, the log-returns ($[0.69, -0.69]$) cancel out, giving a better intuition that the asset did not gain or lose any value.

1.1.6 Advantages of simple returns

It should be mentioned that there are some properties of log-returns which make them less suitable than simple returns in certain situations. Firstly log-normality disappears on longer time spans, so it does not affect long-term analysis. Furthermore log-returns are not additive across assets while simple returns are, hence making the latter a more suitable option in portfolio formulation.

1.2 ARMA vs ARIMA Models for Financial Applications

The ARIMA (Auto Regressive Integrated Moving Average) model is a widely used time-series forecasting model. It is a more sophisticated version of the ARMA (Auto Regressive Moving Average) model. The functionality of these two models is summarised in by the two main components:

- **Auto Regressive**: The model attempts to predict future values based on past values in a stationary time series.
- **Moving Average**: The models attempts to predict future values based on past forecasting errors.

1.2.1 Advantages of ARMA and ARIMA models

In choosing which model between ARMA and ARIMA would suit better the given dataset, the S&P 500 logarithmic close-price curve was analysed as well as its first-order and second-order statistics. The two plots are visualised in Figure 1.2.1 and Figure 1.2.2 respectively.

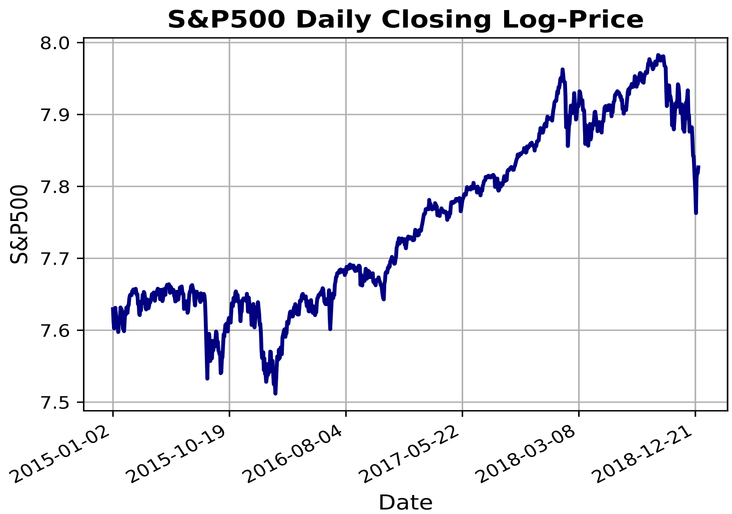

Figure 1.2.1: S&P 500 logarithmic close-price.

As discussed in Section 1.1.1, the log-price curve can be classified as non-stationary through visual inspection. This is because the mean follows a steady upward trend over time. The variance, on the other hand, alternates an initial positive trend to a steep fall. The lack of time-invariance in both the mean and variance statistics implies the non-stationarity of the stochastic process and, consequently, the unsuitability of ARMA modelling for this prediction task, as ARMA models assume stationarity. An ARIMA model, on the contrary, applies an extra integrating step which helps removing the elements of non-stationarity, which makes it a more appropriate choice in this specific occasion.

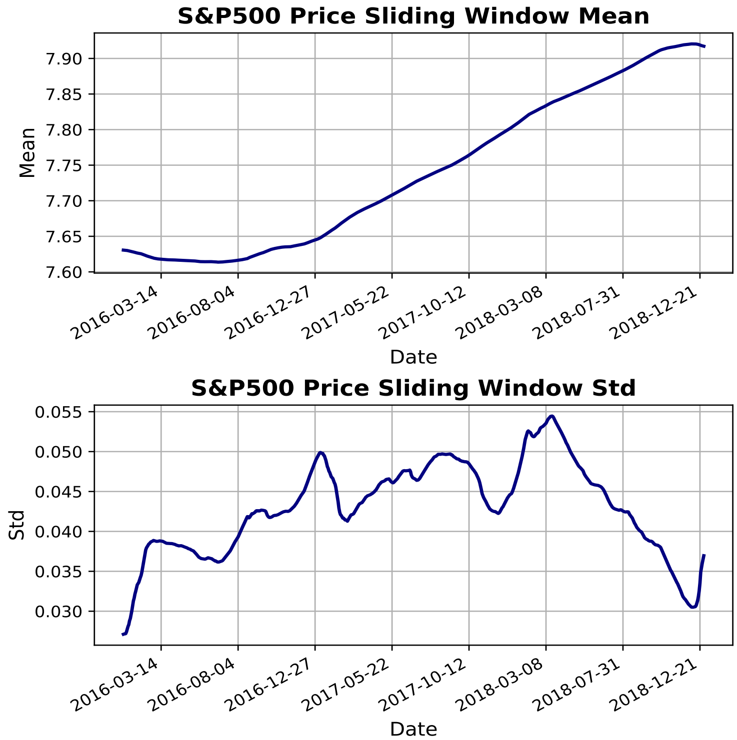

Figure 1.2.2: S&P 500 sliding mean and standard deviation analysis.

1.2.2 ARMA (1,0) modelling

In this subsection an ARIMA(1,0,0) model was used to fit the S&P 500 data. Such model linearly depends on the previous p terms as follows:

$$x_t = \alpha_1 x_{t-1} + \alpha_p x_{t-p} + \epsilon_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \epsilon_t \tag{1.2.1}$$

where α is defined as the autoregressive coefficient and ϵ denotes the noise. Figure 1.2.3 shows the actual and predicted log-price curves and the residuals on a time-span of 100 days (i.e. from day 150 to 250 of the time-series). The zoomed version of the full prediction curve on the left-hand side shows more clearly how well the model predicts the log price values, despite introducing a slight lag in the signal. It should be mentioned that while the mean absolute residual is approximately 0.006, the residual is often considerably smaller, with high volatility sections that cause oscillatory behaviour which forces the overall absolute mean to a higher value. The AR(1) model parameter α can be retrieved using the using the command:

```
ARIMA(snp_arma['Actual'].values, order=(1,0,0)).fit().arparams)
```

Listing 1: Retrieve AR(1) α parameter.

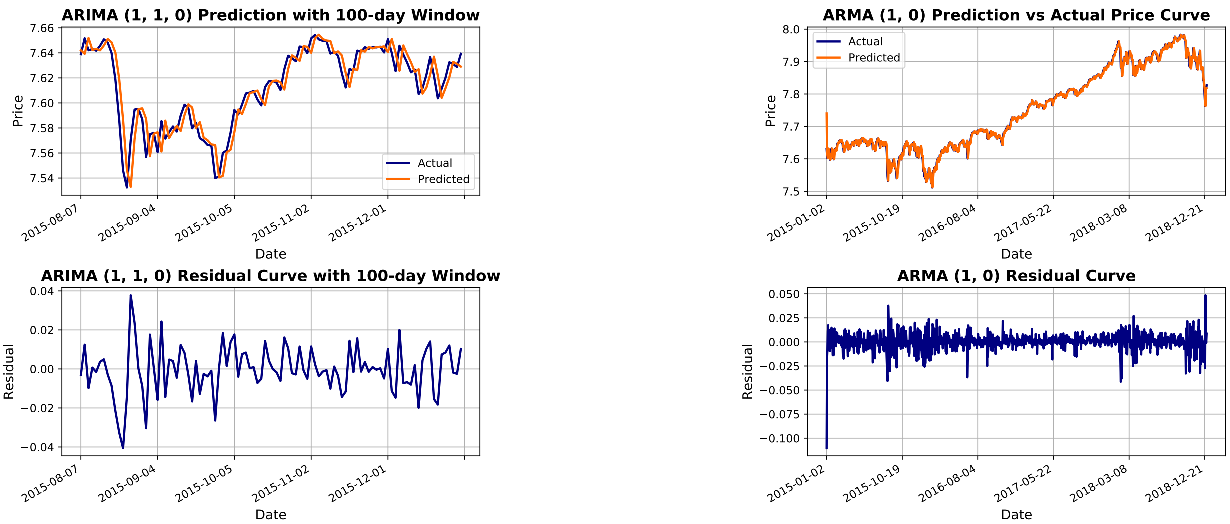

Figure 1.2.3: S&P 500 log-price predictions using an ARMA (1, 0) modelling (right) and zoomed version on a 100-day window (left).

The value of $\alpha = 0.99736$ is consistent with expectations as the fitted autoregressive model is essentially of order 1. When the autoregressive coefficient of an AR(1) model is equal to 1, then the model is modelling a random walk i.e. the value at time t is a function of the previous time-step $t-1$. The model can be mathematically formulated as follows:

$$x_t = 0.99736 \cdot x_{t-1} + \epsilon_t \tag{1.2.2}$$

which explains the modelling of the predicted curve as a lagged version of the true signal.

1.2.3 ARIMA (1, 1, 0) modelling

An ARIMA(p, d, q) model initially applies an extra differentiation step of order d on the data to remove sources of non-stationarity. The ARIMA(1, 1, 0) autoregressive coefficient $\alpha = 0.0088$ was retrieved analogously to Section 1.2.2 using the command in Listing 1. Similarly, the model equation for this section can formulated as:

$$x_t = -0.008752 \cdot x_{t-1} + \epsilon_t \tag{1.2.3}$$

The autoregressive coefficient value close to zero is consistent with the expectation of no correlations to the previous time-step and the prediction at time t, thanks to the differentiation step. The ARIMA model achieves better results as it focuses on maximising an objective function that describes log-returns i.e. the difference of the log prices. Maximising log-returns gives a physical meaning to the optimisation task as it can be considered equivalent to predicting the maximum profit. Figure 1.2.4 shows the full predictions and an 100-day window zoom.

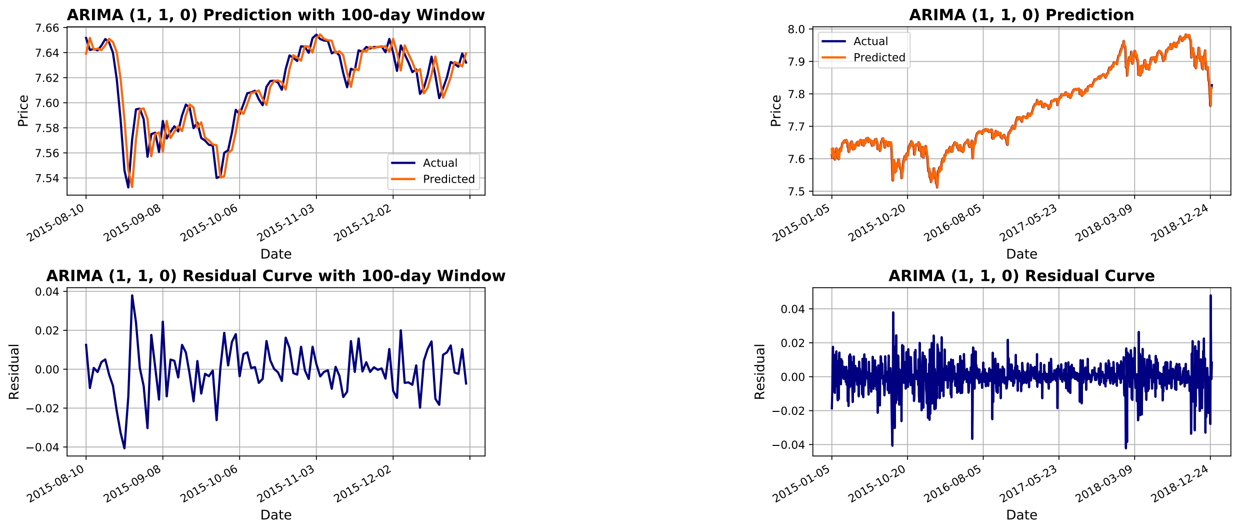

Figure 1.2.4: S&P 500 log-price predictions using an ARIMA (1, 1, 0) modelling (right) and zoomed version on a 100-day window (left).

1.2.4 ARIMA analysis with log-prices

As mentioned in Section 1.2.3, taking the natural logarithm of prices allows the model to maximise an objective function that physically describes log returns. Should the modelling be performed on non-logarithmic price changes instead, the operation could not be physically interpreted as modelling simple returns. Therefore, the results could not be interpreted meaningfully as a maximisation of the profit that can be achieved. Hence, taking the logarithm allows for better results and provides an easier interpretation of the modelling task. Figure 1.2.5 shows the logarithmic ARIMA analysis.

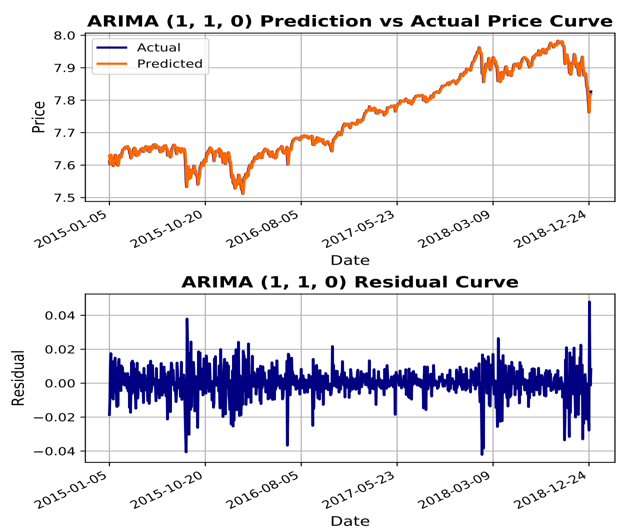

Figure 1.2.5: log-prices analysis.

1.3 Vector Autoregressive (VAR) Models

1.3.1 Concise matrix form

Vector Autoregressive Models are multivariate extensions of AR models. VAR(p) processes are modelled by the following equation:

$$\mathbf{y}_t = \mathbf{c} + \sum_{i=1}^{p} \mathbf{A}_i \mathbf{y}_{t-i} + \mathbf{e}_t \tag{1.3.1}$$

where p is the model order, $\mathbf{y}_i \in R^{k\times 1}$ $\forall i \in [t-1, t-p]$, $\mathbf{A}_i \in R^{k\times k}$ $\forall i \in [1,p]$, $\mathbf{c} \in R^{k\times 1}$ and $\mathbf{e}_t \in R^{k\times 1}$ $\forall t$. The equation above can be expanded in matrix form as:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \sum_{i=1}^{p} \begin{bmatrix} a^i_{1,1} & a^i_{1,2} & \dots & a^i_{1,k} \\ a^i_{2,1} & a^i_{2,2} & \dots & a^i_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^i_{k,1} & a^i_{2k,2} & \dots & a^i_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-i} \\ y_{2,t-i} \\ \vdots \\ y_{k,t-i} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix} , \tag{1.3.2}$$

which can be rewritten more concisely by making use of the matrices

$$\mathbf{B} = \begin{bmatrix} \mathbf{c} & \mathbf{A}_1 & \dots & \mathbf{A}_p \end{bmatrix} \tag{1.3.3}$$

$$\mathbf{Z} = \begin{bmatrix} 1 \\ \mathbf{y}_{t-1} \\ \vdots \\ \mathbf{y}_{t-p} \end{bmatrix} \tag{1.3.4}$$

$$\mathbf{Y} = \mathbf{y}_t \tag{1.3.5}$$

$$\mathbf{U} = \mathbf{e}_t \tag{1.3.6}$$

The resulting equation for the VAR model is:

$$\mathbf{Y} = \mathbf{BZ} + \mathbf{U} \tag{1.3.7}$$

1.3.2 VAR optimal coefficients

Minimising the cost function J yields the optimal solution $\mathbf{B}^*$. The least-squares method can be used to minimise the squared elements of the error term $\mathbf{U}$. Therefore, the cost function J can be defined as $J = \mathbf{U}^T\mathbf{U} = \mathbf{W}$ where $\mathbf{W}$ is the covariance matrix of errors and has dimensions R^T. Expanding the definition of J the following matrix manipulations can be performed on the cost function:

$$J = \mathbf{U}^T\mathbf{U} = (\mathbf{Y} - \mathbf{BZ})^T(\mathbf{Y} - \mathbf{BZ}) = \mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{BZ} - \mathbf{B}^T\mathbf{Z}^T\mathbf{Y} + \mathbf{B}^T\mathbf{Z}^T\mathbf{BZ} \tag{1.3.8}$$

where both $\mathbf{Y}$, $\mathbf{B}$ and $\mathbf{Z}$ are symmetric. This implies that their mutual products retain this property and that taking the transpose of any mutual product is equivalent to the product of the original matrices transposed. This means that $\mathbf{Y}^T\mathbf{B}\mathbf{Z} = \mathbf{B}^T\mathbf{Z}^T\mathbf{Y}$ and $\mathbf{B}^T\mathbf{Z}^T = \mathbf{B}\mathbf{Z}$, which can be used in Equation (1.3.9) to simplify the cost function further as follows:

$$J = \mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{B}\mathbf{Z} + (\mathbf{B}\mathbf{Z})^2 \tag{1.3.9}$$

To find the optimal solution $\mathbf{B}^*$ the derivative of the cost function is set to zero, leading to

$$\frac{\partial J}{\partial B} = -2\mathbf{Y}^T\mathbf{Z} + 2(\mathbf{B}\mathbf{Z})\mathbf{Z} = 0 \ , \tag{1.3.10}$$

that is further simplified to the final value of $\mathbf{B}^*$:

$$2\mathbf{Y}\mathbf{Z}^T = 2(\mathbf{B}\mathbf{Z})\mathbf{Z}^T \tag{1.3.11}$$

$$\mathbf{B}^* = \mathbf{Y}^T\mathbf{Z}(\mathbf{Z}\mathbf{Z}^T)^{-1} \tag{1.3.12}$$

1.3.3 VAR eigenvalues

Consider the following VAR(1) process:

$$\mathbf{y}_t = \mathbf{A}\mathbf{y}_{t-1} + \mathbf{e}_t \tag{1.3.13}$$

Using equation (1.3.13) recursively to substitute for earlier timesteps the equation for $\mathbf{y}_t$ can be rewritten as:

$$\mathbf{y}_t = \mathbf{A}_1^t\mathbf{y}_0 + \sum_{i=0}^{t-1}\mathbf{A}_1^i\mathbf{e}_{t-i} \tag{1.3.14}$$

This implies that the set of vectors $\mathbf{y}_i$ for $i \in \{1, 2, ..., t\}$ and their joint distribution are uniquely dependent on $\mathbf{y}_0$ and $\mathbf{e}_i$ for $i \in \{1, 2, ..., t\}$.

Therefore, taking into consideration an $n \times n$ matrix $\mathbf{A}$ with eigenvalues λ_i for $i \in \{1, 2, ..., m\}$, then it must exist a non-singular matrix $\mathbf{D}$ such that $\mathbf{A} = \boldsymbol{D\Lambda D^{-1}}$, where $\mathbf{D}$ and $\mathbf{\Lambda} \in R^{n\times n}$. $\mathbf{A} = \boldsymbol{D\Lambda D^{-1}}$ is mathematically defined as the singular value decomposition (SVD) of $\mathbf{A}$ where $\mathbf{\Lambda}$ is a diagonal matrix of $\mathbf{A}$'s eigenvalues as follows:

$$\mathbf{\Lambda}^i = \begin{bmatrix} \lambda_1^i & 0 & \dots & 0 \\ 0 & \lambda_2^i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n^i \end{bmatrix} \tag{1.3.15}$$

This definition implies that only if all the eigenvalues of matrix $\mathbf{A}_1$ have modulus smaller than 1, then the summation in Equation (1.3.14) does not diverge to $+\infty$.

1.3.4 Moving average VAR portfolio analysis

In this section VAR models have been applied to the data in the portfolio composed by the following list of tickers: CAG, MAR, LIN, HCP and MAT. First the time-series are detrended using a Moving Average filter of order 66, as shown in Figure 1.3.1, hence the data is fitted in a VAR(1) model. The eigenvalues obtained from the coefficient matrix can be analysed to determine the stability of the model as discussed in Section 1.3.3.

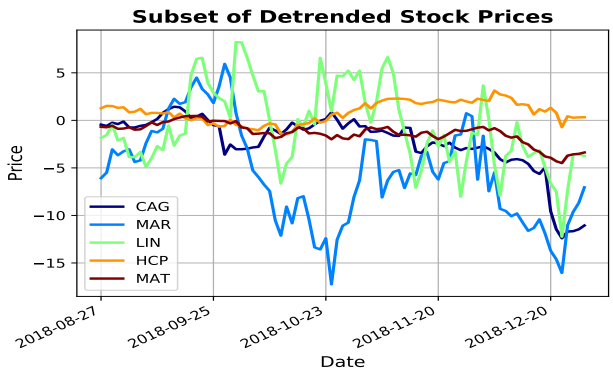

Figure 1.3.1: Detrended time-series tickers data.

Table 1 shows the parameters of the fitted VAR(1) model. Intuitively the values in the table show that the data is positively auto-correlated (high positive values on the diagonal), while showing little to no presence of correlation across assets. Finally, the eigenvalues of matrix A are:

	CAG	MAR	LIN	HCP	MAT
L1.CAG	0.872786	0.113179	-0.281265	0.011912	0.058776
L1.MAR	-0.063745	0.895820	-0.184820	-0.005004	0.022917
L1.LIN	0.000134	-0.111678	0.704023	0.004982	-0.025557
L1.HCP	-0.084776	-0.083831	-0.401417	0.931708	-0.046406
L1.MAT	0.643072	0.094931	2.033036	-0.012884	0.802974

Table 1: Parameters of the fitted VAR(1) model.

$$\left\{ \begin{array}{c} (0.714492880678538 + 0.12927612512612854j) \\ (0.714492880678538 - 0.12927612512612854j) \\ (1.0063596404610207 + 0j) \\ (0.8605189429713621 + 0j) \\ (0.9114451152082126 + 0j) \end{array} \right\} \tag{1.3.16}$$

Equation (1.3.16) shows the presence of an eigenvalue with magnitude greater than 1. As discussed in section 1.3.3 this is symptom of instability in the VAR(1) model.

1.3.5 Sector-based VAR portfolio analysis

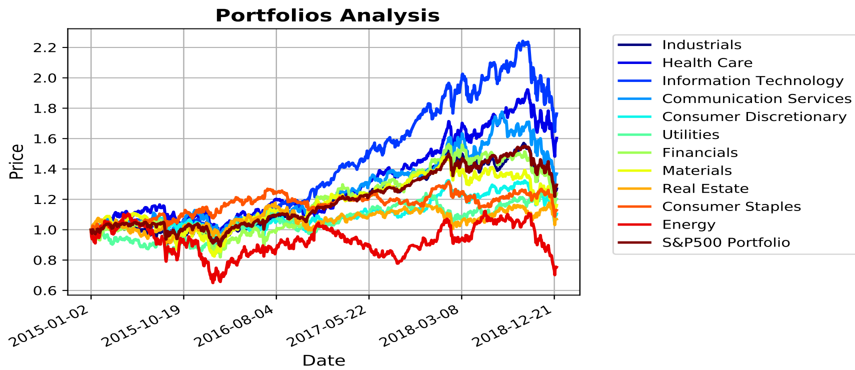

Figure 1.3.2: Time-series data for different market sectors.

Figure 1.3.2 shows the price analysis of various portfolios grouping stocks in the same market area. The maximum eigenvalue of each sector provides information on the stability of the VAR model used to predict prices, as previously discussed in Section 1.3.3 and 1.3.4. According to Table 2, all sectors except for the Financials sector are stable given that the maximum eigenvalue does not exceed 1, therefore they can be modelled by a stable VAR(1) model. Despite grouping stocks by sector is shown as beneficial in reducing the chance of instability when modelling the portfolio using a VAR(1) model, it causes a lack in investment diversification. Constructing a portfolio of tightly clustered assets is generally a bad investment practice, because it does not spread the risk onto multiple market sectors. Diversifying across multiple areas, on the contrary, provides makes the portfolio more robust against bad performance of specific market areas, with positive performances that may come from the other sectors.

	min eigenvalue	max eigenvalue
Industrials	0.371246	0.991721
Health Care	0.092157	0.994153
Information Technology	0.374081	0.992738
Communication Services	0.752488	0.982263
Consumer Discretionary	0.447563	0.99065
Utilities	0.042115	0.985648
Financials	0.152575	1.00434
Materials	0.137838	0.991744
Real Estate	0.763563	0.982785
Consumer Staples	0.546458	0.991508
Energy	0.825707	0.985577

Table 2: Eigenvalues statistics for different stock sectors.

2 Bond Pricing

2.1 Examples of Bond Pricing

2.1.1 Percentage return per annum

An investor received USD 1,100 in one year in return for an investment of USD 1,000 now. Compounding is the operation of adding interest to the initial investment on a rigorous time schedule. Common compounding time-frames that will be overlooked in this example are: annual, semiannual, monthly and continuous. Compounding is mathematically computed as follows:

$$R = C\left(1 + \frac{r}{n}\right)^{nt} \tag{2.1.1}$$

where R denotes the final sum returned, C is the initial investment, r is the interest rate, n is the number of times that the interest is applied (compounded) per time period. Finally t denotes the number of time periods elapsed since the investment was made. Given n, the compound return can be inferred as follows:

$$r_n = n\left[\left(\frac{R}{C}\right)^{\frac{1}{n}} - 1\right] \tag{2.1.2}$$

Using Equation (2.1.2) the various compound returns for the initial example can be computed, i.e. for $R = 1100$ and $C = 1000$. Plugging in the given data in equation (2.1.2) gives:

$$r_n = n\left[\left(\frac{1100}{1000}\right)^{\frac{1}{n}} - 1\right] \tag{2.1.3}$$

- **Annual compounding**:
  Plugging $n = 1$ in Equation (2.1.3) yields to:
  $$r_1 = 10\% \tag{2.1.4}$$
- **Semi-annual compounding**:
  Plugging $n = 2$ in Equation (2.1.3) yields to:
  $$r_2 = 9.76\% \tag{2.1.5}$$
- **Monthly compounding**:
  Plugging $n = 12$ in Equation (2.1.3) yields:
  $$r_{12} = 9.57\% \tag{2.1.6}$$
- **Continuous compounding**:
  For $n = +\infty$ and recalling the definition of the Euler Number e as :
  $$e = \lim_{i\to\infty}\left(1 + \frac{1}{i}\right)^{i} \tag{2.1.7}$$

Equation (2.1.1) can be simplified into a more compact exponential form as follows:

$$R_\infty = \lim_{i \to \infty} C \left(1 + \frac{r}{n}\right)^{nt} \tag{2.1.8}$$

Let $i = \frac{n}{r}$ and substitute into the original equation:

$$R_\infty = \lim_{i \to \infty} C \left(1 + \frac{1}{i}\right)^{irt} = \lim_{i \to \infty} \left[C \left(1 + \frac{1}{i}\right)^{i}\right]^{rt} \tag{2.1.9}$$

Simplify the exponential term to e leveraging the definition in Equation (2.1.7):

$$R_\infty = Ce^{rt} \tag{2.1.10}$$

Finally, the equation above can be rearranged to find the continuous compounding interest rate:

$$r_\infty = \ln\left(\frac{R}{C}\right) \tag{2.1.11}$$

which ultimately can be used to find the continuous compounding interest rate for the specific example taken into consideration in this section (for $R = 1100$ and $C = 1000$):

$$r_\infty = \ln\left(\frac{1100}{1000}\right) = 9.53\% \tag{2.1.12}$$

2.1.2 Monthly compounding to continuous

This exercise aims to compute the rate of interest with continuous compounding equivalent to 15% per annum with monthly compounding. The equivalence formula between continuous interest rate and a discrete compounding interest rate of frequency n is:

$$r_\infty = n \cdot \ln\left(1 + \frac{r_n}{n}\right) \tag{2.1.13}$$

Solving Equation (2.1.15) for a monthly compounding with interest rate 15 %, i.e. $n = 12$ and $r_{12} = 0.15$ gives:

$$r_\infty = 12 \cdot \ln\left(1 + \frac{0.15}{12}\right) = 14.91\% \tag{2.1.14}$$

2.1.3 Quarterly compounding to continuous

This exercise replicates the calculation undertaken in Section 2.1.2 to find the continuous compounding equivalent to an interest rate of 12% per annum quarterly compounded. Therefore, solving again Equation (2.1.15) for the new use-case of quarterly compounding with interest rate 12 %, i.e. $n = 4$ and $r_4 = 0.12$ gives:

$$r_\infty = 4 \cdot \ln\left(1 + \frac{0.12}{4}\right) = 12.182\% \tag{2.1.15}$$

Given an initial deposit of 10,000 USD, the interest received in year will sum up to $r_\infty \cdot 10{,}000$, i.e. 1,218.18 USD. In terms of quarterly cashflow, the interest would be paid in 4 occasions, once every three months, as shown in Table 3.

End of Month	3	6	9	12
	304.5	313.8	323.4	333.2

Table 3: Quarterly cashflow on 10,000 USD investment.

2.2 Forward Rates

2.2.1 Evaluate and compare forward rates

Suppose that the one–year interest rate, r_1 is 5%, and the two–year interest rate, r_2 is 7%. If you invest USD 100 for one year, your investment grows to 1001.05 = *USD 105; if you invest for two years, it grows to* 1001.072 =*USD 114.49. The extra return that you earn for that second year is* 1.072/1.051 = 0.090*, or* 9.0%*.*

2.2.1.a Longer investment's return
The extra return made in the second year is nothing more than the market expectation of what the sport rate will be one year from now. Therefore, assuming that there will be no unexpected developments in the market (i.e. the spot rate will evolve according to predictions), there is no difference in committing to a 2-year investment at the beginning or repeating two-one year investments. However it should be mentioned that average investors do not have an infinite availability of cash, therefore there might be the need for the investor to access said cash sometime during a 2 year time-frame. In this case investing for just one year would give the investor more flexibility into accessing the cash in case of emergencies.

2.2.1.b Investment strategies analysis
For the market to be arbitrage-free, it must mean that the rates offered for different investment periods are consistent with market expectations and therefore should not impact investors' decisions because they essentially do not constitute any tangible advantage. On the contrary, the need for liquidity already mentioned in Section 2.2.1.a and the chance of varying investment strategies are important factors to be taken into consideration. In this sense, shorter investments give more opportunities to the investor as they can be cashed out earlier and new investment strategies can be attempted if deemed appropriate.

2.2.1.c Advantages and disadvantages of forward rates
It is clear that taking longer investments is riskier as they span a longer time-frame in which anything could negatively impact the investment. Therefore it is expected that the forward rate $f_{1,2}$ is higher than r_1 and r_2. Hence assuming that the market conditions stay the same, the forward rate results in higher gains than a two 1-year investment strategy. However, fixing the rate can turn into an advantage or disadvantage based on the market fluctuation during the investment time-frame.

2.2.1.d Changing the investment period
Using the current spot rate and the implied forward rate defined in Equation (2.2.1), the change from a 1-year investment strategy to a 2-year one can be computed.

$$(1 + r_j)^j = (1 + r_i)^i (1 + f_{i,j})^{ji} \tag{2.2.1}$$

2.3 Duration of a coupon-bearing bond

2.3.1 Duration

Duration is the weighted average of the times to each of the cash payments. The times are the future years 1, 2, 3, etc., extending to the final maturity date, which we call T. The weight for each year is the present value of the cash flow received at that time divided by the total present value of the bond.

$$Duration = \sum_{i=1}^{n} t_i \frac{PV_i}{V_{tot}} \tag{2.3.1}$$

where n is the number of payments, PV_i is the present value of the ith cash payment, t_i is the years until the ith payment is due and V_{tot} is the total present value of the cashflow.

2.3.1.a. Macaulay duration

For the given data in Table 4 the bold entries have been computed. The Macaulay duration of the 1% bond is computed as the sum of the last row's elements which amounts to 6.76 years.

Year	1	2	3	4	5	6	7	Total
Payment	\$10	\$10	\$10	\$10	\$10	\$10	\$1010	\$1070.00
$PV(C_t)$ at 5%	\$9.52	\$9.07	\$8.64	\$8.23	\$7.84	\$7.46	\$717.79	$PV = \$768.55$
Fraction of PV: $\frac{PV(C_t)}{PV}$	0.0124	0.0118	0.0112	0.0107	0.0102	0.0097	0.934	**1**
Year × Fraction of PV: $t * \frac{PV(C_t)}{PV}$	0.0124	0.0236	0.0337	0.0428	0.051	0.0583	6.5377	**6.76 years**

Table 4: Calculating the duration of the 1% 7-year bonds. The yield to maturity is 5% a year.

2.3.1.b. Modified duration

D_m is the modified duration, another statistic of bonds. It is defined as the derivative of the Macaulay duration with respect to the yield λ:

$$D_m = \frac{1}{P(\lambda_0)} \frac{dP(\lambda)}{d\lambda} \; for \; \lambda = \lambda_0 \tag{2.3.2}$$

Modified duration is a measure of risk, as it quantifies how sensitive the duration is to changes in the yield, i.e. how much a variation in the interest rates causes a change of price of the bond. It can be mathematically proved that the two durations are linked by the relationship $D_m = \frac{D}{1+\frac{\lambda}{m}}$, where m is the amount of yearly compounding intervals, so $m = 1$ for yearly compounding. For continuous compounding instead, $m = \infty$ which yields that the two durations are exactly equivalent $D_m = D$. In the specific example taken into consideration in this section the given yield is 5%, and $m = 1$. Therefore $D_m = 6.438\%$, as shown in Table 5.

2.3.1.c. Advantages of duration against volatility

A long-term investment is characterised by a larger duration than a bond with shorter time to maturity. A larger duration implies a greater impact of changes in interest rate, as more coupon payments are due in the future in the scenario of a pension plan. As discussed in Section 2.3.1.b, the modified duration gives a measure of the riskiness of the bond. Both measures give the investor

Year	1	2	3	4	5	6	7	Total
Payment	\$10	\$10	\$10	\$10	\$10	\$10	\$1010	\$1070.00
$\frac{\partial P(\lambda)}{\partial \lambda}$ at 5%	\$9.07	\$17.28	\$24.68	\$31.34	\$37.31	\$42.64	\$47855	$PV = \$4947.58$
$\frac{1}{P(\lambda_0)} \frac{\partial P(\lambda)}{\partial \lambda}$	0.0118	0.0225	0.0321	0.0408	0.0485	0.0555	6.3364	**6.438**

Table 5: Calculating the modified duration of the 1% 7-year bonds. The yield to maturity is 5% a year.

information on changes on interest rates and therefore are useful in assessing the market status and predicting sudden changes to protect the pension plan. The technique that leverages statistics such as duration and convexity to design a robust portfolio against volatility is called immunisation and consists in matching the aforementioned statistics which mathematically map Taylor's series terms.

2.4 Capital Asset Pricing Model (CAPM) and Arbitrage Pricing Theory (APT)

This section will focus on discussing the Capital Asset Pricing Model (CAPM) and Arbitrage Pricing Theory (APT). The CAP model is mathematically formulated in Equation (2.4.1):

$$E[R_i] = R_f + \beta_i(E[R_m] = R_f) \tag{2.4.1}$$

where $E[R_i]$ is the expected return of the given asset, R_f is the risk-free return, such as government bonds or bank's fixed interest rate plans. $E[R_m]$ is the expected return of the market and β_i is a measure of volatility (systematic risk) of the asset i compared to the whole market. The systematic risk coefficient β is defined as:

$$\beta_i = \frac{cov(r_i, r_m)}{var(r_m)} \tag{2.4.2}$$

2.4.1 Equally-weighted market return estimation

To estimate the market return the average of each individual company's return is taken on a daily basis. However, some companies must be filtered out of the data, as they are presented with incomplete information. Figure 2.4.1 shows the returns of all companies taken into consideration as well as the equally-weighted market return. It can be visually appreciated how most companies have small returns oscillating across the x-axis, with some more volatile assets spiking in both the negative and positive direction. Therefore, this oscillating trend can be similarly observed in the market return curve as well. Overall the equally-weighted market return averages 0.00472% per day with standard deviation 0.00665.

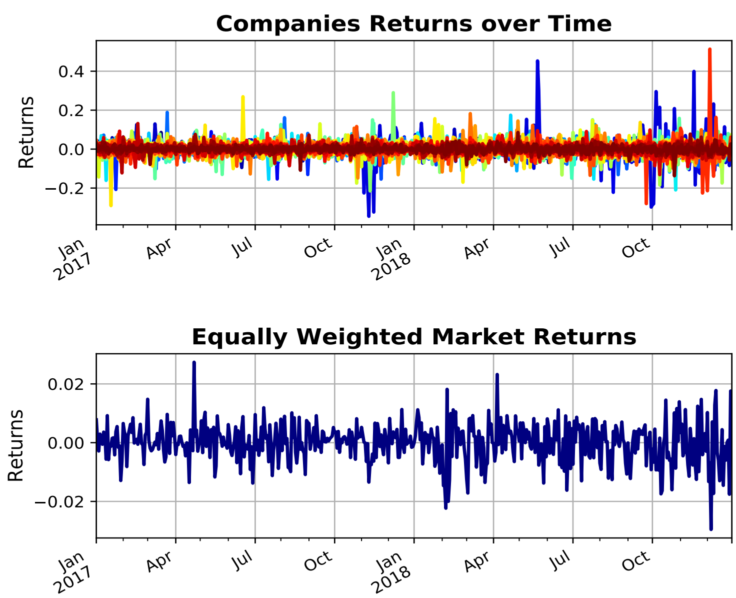

Figure 2.4.1: Equally-weighted returns over time for all companies in the market.

2.4.2 Equally-weighted rolling β estimation

The systematic risk coefficient β, as previously defined in Equation (2.4.2) can be computed for all companies i taken into consideration. For this exercise the rolling window has been set to 22 days. Figure 2.4.2 shows the rolling beta estimation for each company as well as its normal-like distribution.

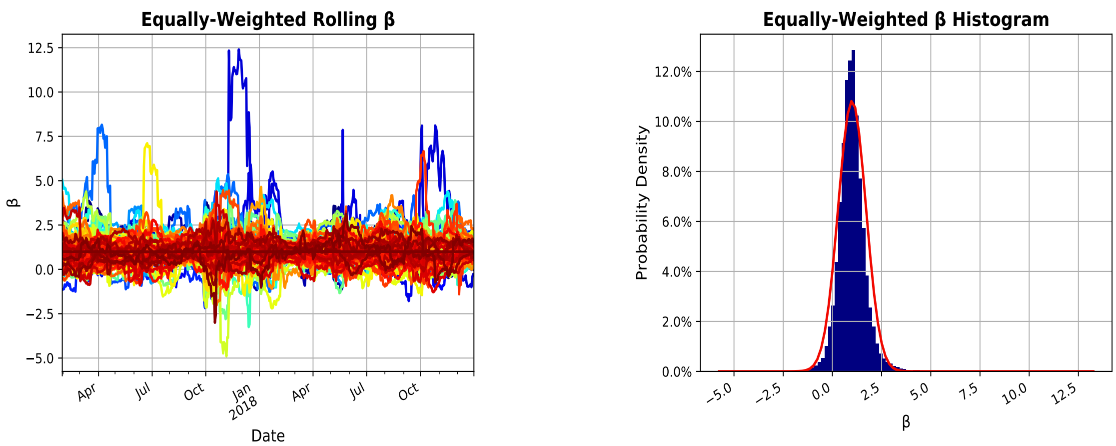

Figure 2.4.2: Equally-weighted rolling beta of all Companies (left) and its probability distribution (right).

As mentioned in the introduction to Section 2.4, the β coefficient provides the investors with insights on the volatility of a stock in correlation to the market movement. In order to analyse the riskiness of an asset and ultimately decide whether or not include it in a portfolio, investors can interpret β's magnitude as follow:

- $\beta_i < 1$: The stock i is less volatile than the market. Adding such assets to a market portfolio would reduce the average risk associated with it.
- $\beta_i = 1$: The stock i is perfectly correlated to the market. Adding such assets to a market portfolio would make no difference in terms of risk.
- $\beta_i > 1$: The stock i is more volatile than the market. Adding such assets to a market portfolio would increase the average risk, but riskier assets can also carry higher returns.

For the example taken into consideration in this section the average systematic risk is estimated to be $\bar{\beta} = 1.0$, which is expected as the portfolio is composed by all assets in the market equally weighted. The statistic on the standard deviation, which is found to be equal to $std(\beta) = 0.708$ characterises a market composed by stocks that are quite diverse in terms of riskiness. This observation is further confirmed by looking at the distribution of the systematic risk coefficients in the histogram in Figure 2.4.2. The spread in β's magnitude values implies a market composed by both robust stocks and highly volatile ones.

2.4.3 Market-cap-weighted market return estimation

Instead of considering an equally weighted portfolio, this section focuses on the market portfolio formed using the weighted market return, i.e. a portfolio of all assets in the market weighted with respect to their market capitalisation (cap). The weight market return R_m is defined in Equation (2.4.3):

$$R_m = \sum_i \frac{mcap_i}{mcap_{tot}} \tag{2.4.3}$$

where $mcap_i$ denotes the market cap of the asset i and $mcap_{tot}$ the sum of the market caps of all assets in the market portfolio.

The resulting market portfolio, can be used form an optimal mean-variance portfolio (One-fund theorem) if combined with a risk-free asset. Figure 2.4.3 shows how the weighting positively impact the portfolio in terms of risk-aversion. This is intuitively confirmed by recognising the intrinsically riskier nature of low market capitalisation assets. Therefore a market portfolio constructed using the weighted market return is expected to be less volatile as it includes riskier assets in smaller proportions. The statistics recorded for this market portfolio further show this reduction in risk, with a market return mean of 0.000188 and an std of 0.006599.

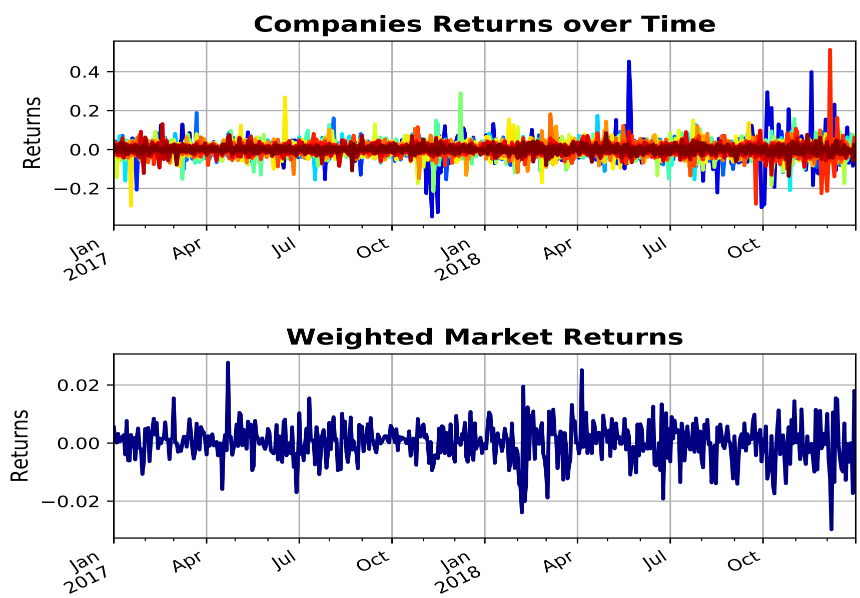

Figure 2.4.3: Market-cap-weighted returns over time for all companies in the market.

2.4.4 Market-cap-weighted rolling β estimation

A systematic risk statistic analysis for the new portfolio can be performed, using the same procedure followed in Section 2.4.2. The average systematic risk $\bar{\beta}$ is 0.9616, which is lower than the 1.0 observed earlier. The standard deviation has been also slightly reduced ($std(\beta) = 0.694$), which implies less volatility of the weighted portfolio.

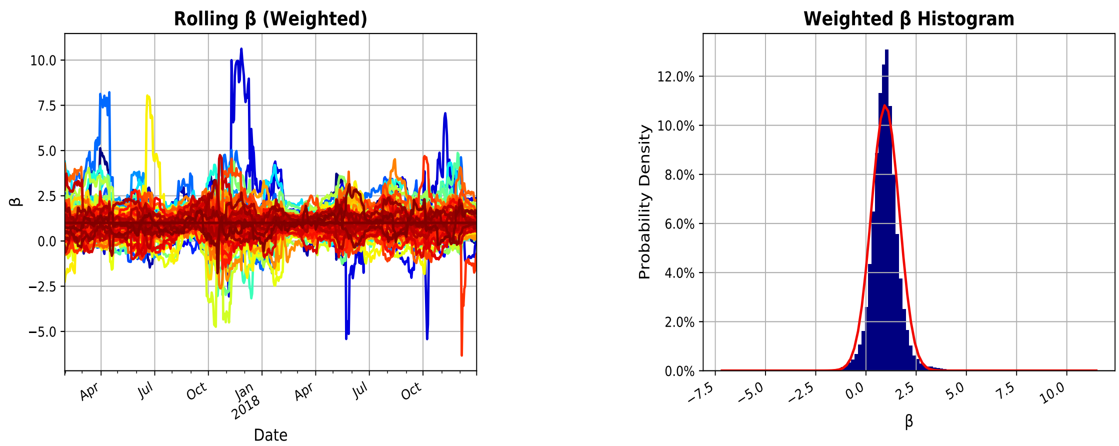

Figure 2.4.4: Market-cap-weighted rolling Beta of all companies (left) and its probability distribution (right).

2.4.5 Arbitrage pricing theory

This section is based on the assumption that the arbitrage pricing theory (APT) holds for a two-factor model. The return of company i, can be formulated in this scenario as:

$$r_i = \alpha + \beta_{m,i} R_m + \beta_{s,i} R_s + \epsilon_i \tag{2.4.4}$$

where ϵ denotes the residual of this company's specific return and α the excess return. $\beta_{s,i}$ are the excess market returns. These can be simply computed by taking the logarithm of the market-cap-weighted market returns.

2.4.5.a Estimate factor returns

Furthermore, Ordinary Least Squares can be applied to estimate parameters α, R_m and R_s for every day of the time-series. By defining the matrices

$$\boldsymbol{\beta}_m = \begin{bmatrix} \beta_{m,1} \\ \beta_{m,2} \\ \vdots \\ \beta_{m,i} \end{bmatrix}, \quad \boldsymbol{\beta}_s = \begin{bmatrix} \beta_{s,1} \\ \beta_{s,2} \\ \vdots \\ \beta_{s,i} \end{bmatrix}, \tag{2.4.5}$$

let:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{\beta}_m & \boldsymbol{\beta}_s \end{bmatrix}, \quad \boldsymbol{p} = \begin{bmatrix} \alpha \\ R_m \\ R_s \end{bmatrix} \text{ and } \boldsymbol{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_i \end{bmatrix} \tag{2.4.6}$$

Hence, minimising the OLS is mathematically equivalent to minimising the following objective function:

$$\min_{r} ||\boldsymbol{X}\boldsymbol{p} - \boldsymbol{r}||_2^2 \tag{2.4.7}$$

2.4.5.b Variance and magnitude of parameters

Figure 2.4.5 shows the parameters magnitude over time as well as the density distribution. The density distribution plots for all parameters show that R_m has a slightly larger magnitude on average and a higher variance. This implies that R_m is more volatile than the other parameters analysed in this section. Furthermore, R_s has variance very close to 0, with close to no occurrences of magnitude values larger than 0.01 in magnitude.

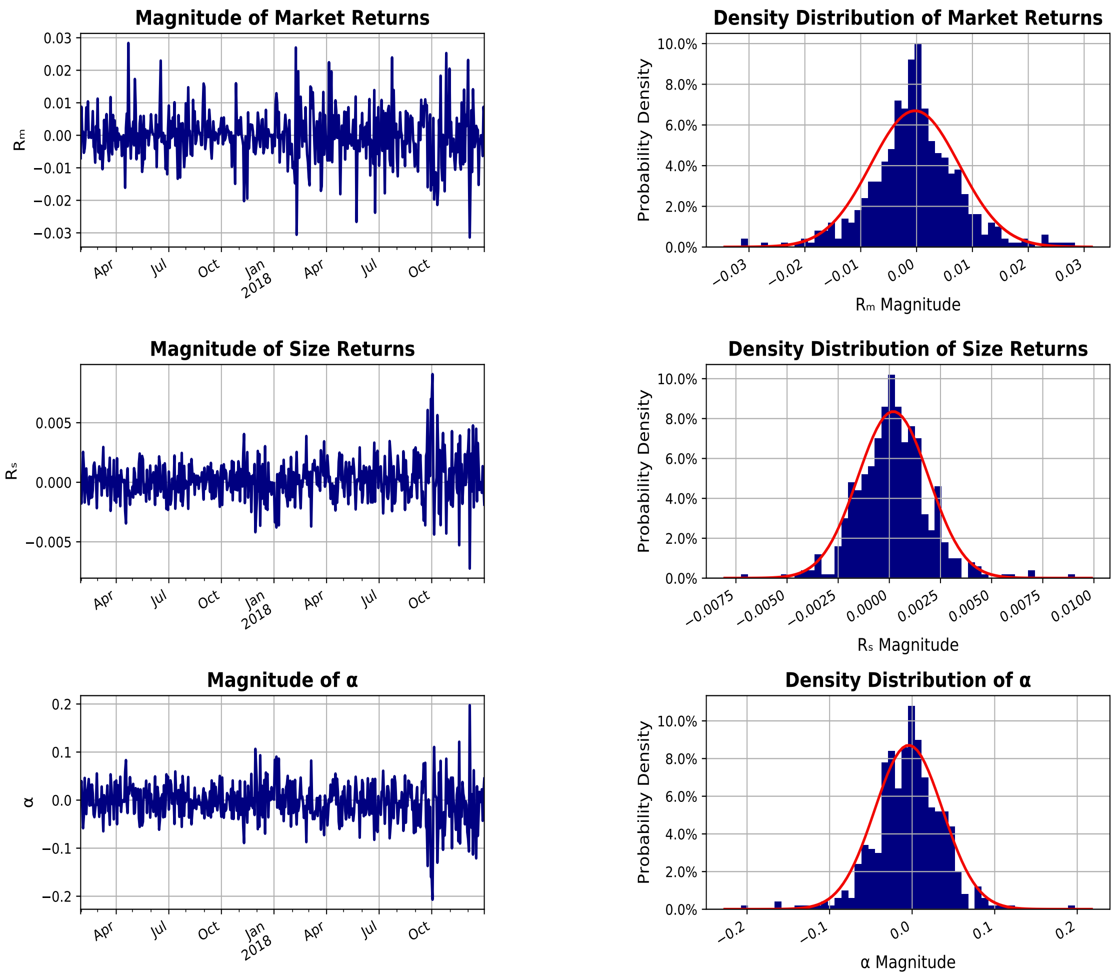

Figure 2.4.5: Parameters analysis for magnitude and variance.

2.4.5.c Specific return

Figure 2.4.6 shows the correlation between each company's return (r_i) and specific return (ϵ_i) $< \epsilon_i, r_i >$. The specific return is the difference between the actual return of the company and the one explainable through the analysis of the factors taken into consideration, i.e. the market and size factors discussed in Section 2.4.5.b. The correlation values are mostly greater than 0.7, which implies a positive and significantly high average correlation. High correlations together with the definition of specific returns imply that the factors analysed are not enough to explain the actual returns observed, thus more are needed for the Arbitrage Pricing Theory model to better predict the companies' actual returns.

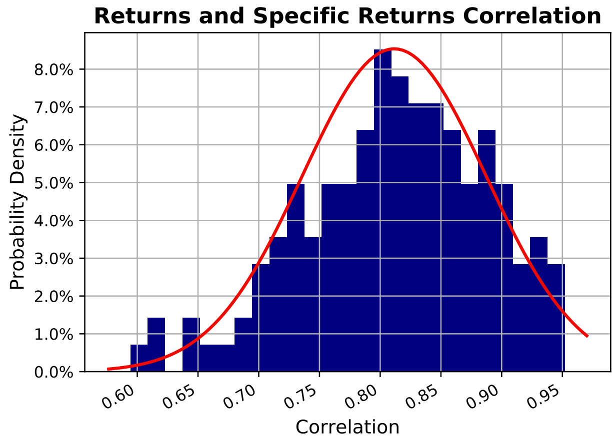

Figure 2.4.6: Correlation for every company through time.

.

2.4.5.d Covariance matrix stability analysis

Let the covariance matrix R be the concatenation of R_m and R_s as defined in equation (2.4.8) below:

$$\boldsymbol{R} = \begin{bmatrix} R_{m,1} & R_{s,1} \\ R_{m,2} & R_{s,2} \\ \vdots & \vdots \\ R_{m,500} & R_{s,500} \end{bmatrix} \tag{2.4.8}$$

The $\mathbf{cov}(\mathbf{R})$ matrix is calculated using a rolling window of 22 days and shown in Figure 2.4.7. The highly oscillatory behaviour shown in the plot, together with the high variance shown in the histogram imply a significant instability. Furthermore $\mathbf{cov}(\mathbf{R})$ is very small in magnitude for all the time period taken into consideration, which means there is very little correlation between market returns.

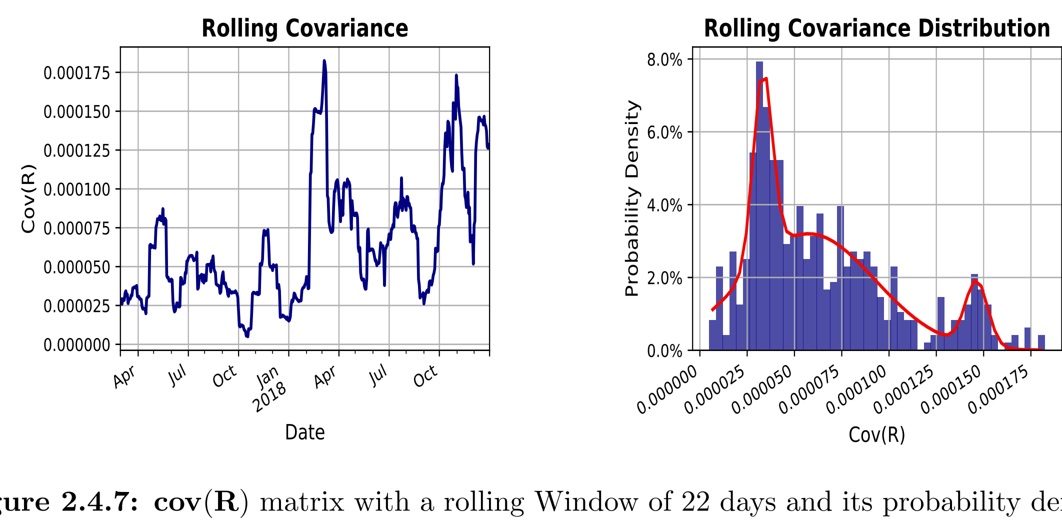

Figure 2.4.7: $\mathbf{cov}(\mathbf{R})$ matrix with a rolling Window of 22 days and its probability density.

2.4.5.e Principal component analysis (PCA)

The specific return of company i at time t is defined as $\epsilon_{i,t}$. The matrix form $\boldsymbol{\epsilon}$ of all specific returns is formulated in Equation (2.4.9) given I companies and T time-steps:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{1,0} & \epsilon_{2,0}, & \cdots & \epsilon_{I,0} \\ \epsilon_{1,1} & \epsilon_{2,1}, & \cdots & \epsilon_{I,1} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{1,T} & \epsilon_{2,T}, & \cdots & \epsilon_{I,T} \end{bmatrix} \tag{2.4.9}$$

In this section the data taken under consideration encompasses 141 companies' stock prices in 500 different time-steps, as some have been discarded due to incompleteness in the data. Therefore, the covariance of matrix $\boldsymbol{\epsilon}$ will have dimensions: 157×157, i.e. it will generate 141 eigenvalues. Through Principal Component Analysis such eigenvalues can be inferred from $\mathbf{cov}(\boldsymbol{\epsilon})$. Figure 2.4.8 shows the Scree plot for the eigenvalues of $\mathbf{cov}(\boldsymbol{\epsilon})$ alongside their cumulative variance, which denotes the amount of information retained by that subset of principal components. The plots further show that 95% of the information in the original data can be retained while reducing the dimensionality of the data to 94 by taking into consideration only the first 94 principal components. The significant magnitude of numerous principal components shows that modelling prices accurately needs high dimensionality, which implies immensely computationally demanding and sometimes even unfeasible operations.

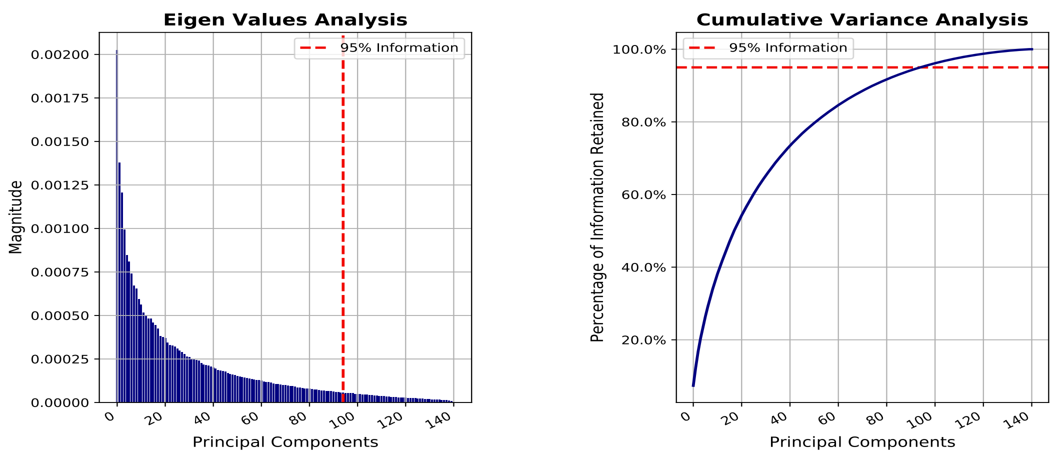

Figure 2.4.8: PCA analysis of $\mathbf{cov}(\mathbf{R})$'s eigenvalues and their cumulative variance.

3 Portfolio Optimisation

3.1 Adaptive Minimum-Variance Portfolio Optimisation

This section will discuss how to optimise the portfolio for minimum-variance, i.e. find the best rate of return for the minimum risk, as mathematically formulated in the Equation (3.1.1)

$$\min_{\boldsymbol{w}} J(\boldsymbol{w}, \boldsymbol{C}) = \frac{1}{2}\boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w} \tag{3.1.1}$$

$$\text{subject to: } \boldsymbol{w}^T \mathbf{1} = 1$$

where $\boldsymbol{w}$ is the weights vector, $\boldsymbol{C}$ is the covariance matrix of all assets and the expected return and variance of the portfolio are defined as:

$$\bar{\mu} = \boldsymbol{w}^T \boldsymbol{\mu} \quad \text{and} \quad \bar{\sigma}^2 = \boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w} \tag{3.1.2}$$

3.1.1 Optimal weights

The Lagrangian optimisation problem for the portfolio defined in Section 3.1 is:

$$\min_{\boldsymbol{w}, \lambda} L(\boldsymbol{w}, \lambda, \boldsymbol{C}) = \frac{1}{2}\boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w} - \lambda(\boldsymbol{w}^T \mathbf{1} - 1) \tag{3.1.3}$$

To solve the minimisation in Equation (3.1.3) the Langrangian is differentiated with respect to λ and with respect to the weight vector $\boldsymbol{w}$, then both derivatives are set to zero to compute the parameters value at optimality. It should be mentioned that $\boldsymbol{C}$ is symmetric, hence there is no difference between the covariance matrix and its transpose. It is also assumed to be invertible, and $\boldsymbol{C}^{\mathbf{-1}}$ will also be equivalent to its transpose. Finally, the weights $\boldsymbol{w}$ are not assumed to be non-negative, which physically means that shorting is allowed. However, the average investor often does not have the option to short assets, so in order to accommodate this extra constraint should be added in the optimisation problem. The derivative of the Lagrangian L with respect to the weights is:

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{C}\boldsymbol{w}^* - \lambda \mathbf{1} \tag{3.1.4}$$

where $\boldsymbol{w}^*$ denotes the optimal set of weights. Setting the derivative to zero and rearranging it yields:

$$\boldsymbol{w}^* = \lambda \boldsymbol{C}^{-1} \mathbf{1} \tag{3.1.5}$$

Differentiating with respect to the Lagrangian parameter λ gives:

$$\frac{\partial L}{\partial \lambda} = (\boldsymbol{w}^{*T} \mathbf{1} - 1) \tag{3.1.6}$$

Which is then set to zero and rearranged to:

$$\boldsymbol{w}^{*T} \mathbf{1} = 1 \tag{3.1.7}$$

Equation 3.1.7 can be substituted in Equation 3.1.5 by performing some matrix manipulations on the latter as follows:

$$\boldsymbol{w}^{*T}\mathbf{1} = \lambda\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1} \tag{3.1.8}$$

and finally substituting Equation 3.1.7 into the Equation above to obtain:

$$\lambda\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1} = 1 \tag{3.1.9}$$

From Equation 3.1.9, the value of λ can be retrieved and it is defined by:

$$\lambda = \frac{1}{\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1}} \tag{3.1.10}$$

which can be substituted back in the optimal weights definition in Equation 3.1.5 to get:

$$\boldsymbol{w}^* = \frac{\boldsymbol{C}^{-1}\mathbf{1}}{\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1}} \tag{3.1.11}$$

Having computed the optimal set of weights, they can simply be plugged into the equations for the expected mean return and variance of the portfolio defined in Section 3.1 to obtain the theoretical return and variance of the minimum-variance portfolio, which are the following:

$$\bar{\mu}^* = \boldsymbol{w}^{*T}\boldsymbol{\mu} = \frac{\boldsymbol{C}^{-1}\mathbf{1}}{\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1}}\boldsymbol{\mu} \tag{3.1.12}$$

$$(\bar{\sigma}^2)^* = \boldsymbol{w}^{*T}\boldsymbol{C}\boldsymbol{w}^* = \frac{1}{\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1}} \tag{3.1.13}$$

3.1.2 Minimum-variance portfolio analysis

This section takes into consideration a dataset composed by 10 stocks. Their daily returns, which will be used to formulate the equally weighted portfolio and minimum variance portfolio further in this section, are shown in Figure 3.1.1.

The data is split using a 50:50 ratio into training and testing set, which represent respectively the years 2017-18 and 2018-19. The results obtained on both sets for both the equally weighted portfolio and the minimum-variance one are compared in Figure 3.1.2. The figure shows the comparison for returns and cumulative returns, which are defined as:

$$\text{Cumulative Return:} \quad R_T = \sum_{t=1}^{T} \bar{r}_t \tag{3.1.14}$$

where denotes the time period in days.

The minimum variance-portfolio is performing slightly better than the equally weighted one. However, while the cumulative return on the train set is steadily increasing for both portfolios, the

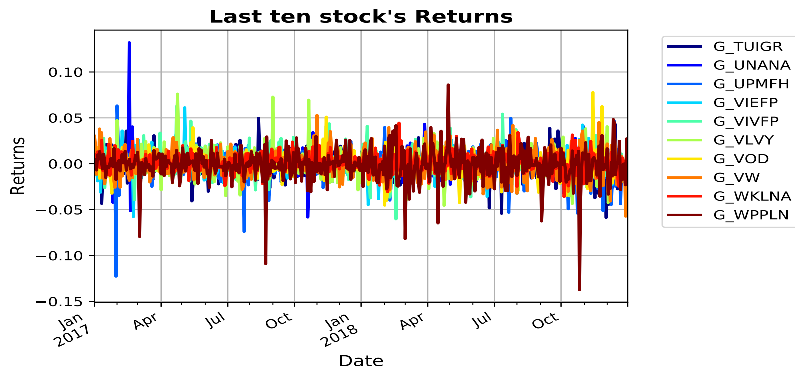

Figure 3.1.1: Daily returns for the 10 stocks in the dataset.

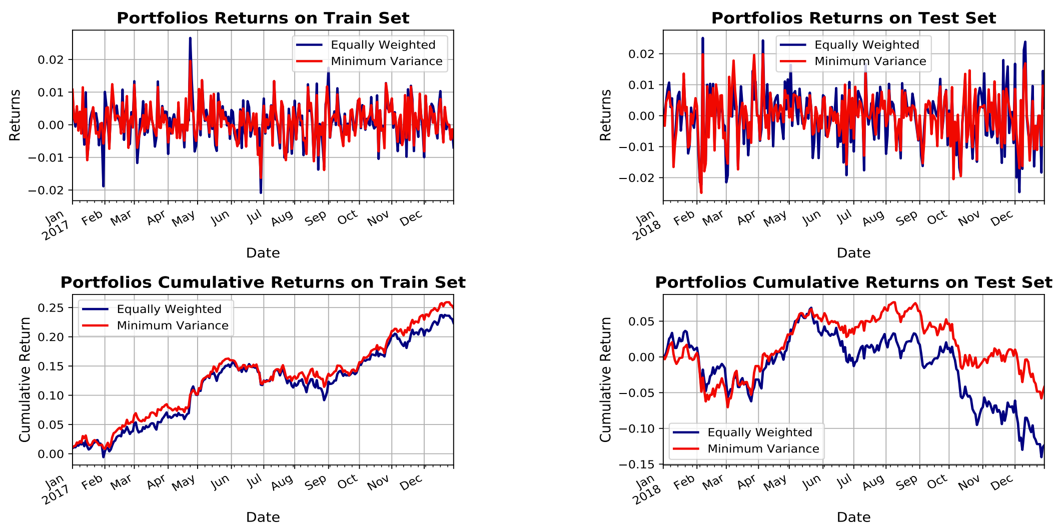

Figure 3.1.2: Portfolio returns and cumulative returns on both training and testing set.

test set shows poor performances, with the minimum variance portfolio performing better than the equally-weighted one but still unable to achieve a positive cumulative return. One possible improvement can be achieved by modifying the train-to-test ratio. It is common practice to have a larger training set, in order to generalise better. Splits in the range of (70:30 - 85:15) are generally optimal.

	Mean Return	Cumulative Return	Variance	Theoretical Variance
Equally Weighted Train	0.0008595546	0.2234842	3.7495e-05	-
Minimum Variance Train	0.0009579316	0.2490622063	2.86163e-05	2.86163e-05
Equally Weighted Test	-0.0004731904	-0.1235027	7.93339e-05	-
Minimum Variance Test	-0.0001602424	-0.0418232764	5.84487e-05	5.84487e-05

Table 6: Analysis of the equally-weighted and minimum-variance portfolio statistics.

For the minimum-variance portfolio, the mathematical formulation of the theoretical variance was computed in Equation (3.1.13). Using this formula we can prove that the variance of the minimum-variance portfolio computed in the practical example matches exactly the theoretical result achieved in Section 3.1.1. The portfolio's statistics are summarised in Table 6.

3.1.3 Adaptive minimum-variance portfolio analysis

The adaptive time-varying minimum-variance portfolio, which uses the empirical estimate of the mean vector and covariance matrix and a rolling window of length M days, is given by:

$$\hat{\boldsymbol{\mu}} - t = \frac{1}{M} \sum_{\tau=t-M+1}^{t} \boldsymbol{r}_\tau \tag{3.1.15}$$

$$\hat{\boldsymbol{C}}_t = \frac{1}{M} \sum_{\tau=t-M+1}^{t} (\boldsymbol{r}_\tau - \hat{\boldsymbol{\mu}}_t)(\boldsymbol{r}_\tau - \hat{\boldsymbol{\mu}}_t)^T \tag{3.1.16}$$

$$\hat{\boldsymbol{w}}_t = \underset{\boldsymbol{w},\lambda}{\arg\min}\ J(\boldsymbol{w}, \lambda, \hat{\boldsymbol{C}}_t) \tag{3.1.17}$$

The cumulative returns for various rolling window sizes are analysed in Figure 3.1.3. The larger the window, the larger is the timeframe which is taken into consideration during the covariance matrix estimation. As observed in the comparison plot, shorter window sizes perform better as they are able to detect short-period patterns in the stock data. However, spikes in the price are symptom of high volatility and hence high riskiness of the asset. Furthermore, in a practical setting, it must be mentioned that leveraging short-period oscillations in prices through investing can become very expensive due to transaction fees.

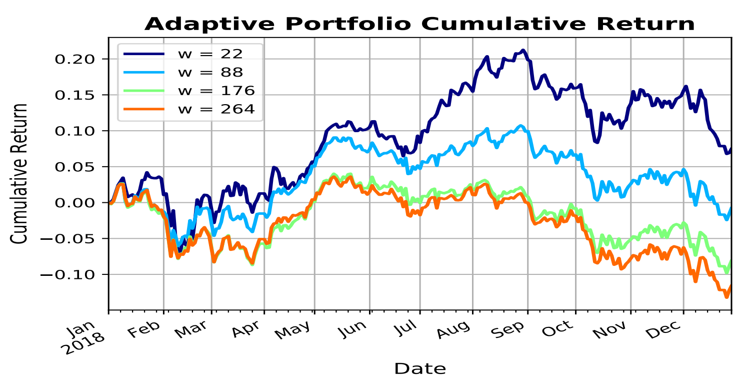

Figure 3.1.3: Evolution of the cumulative returns for adaptive minimum-variance portfolios with different window sizes.

The window size of 22 was the best performing among the ones analysed, achieving a positive cumulative return of 0.075: this is significantly better than the negative cumulative returns observed for the equally weighted and minimum variance model for the test set in Section 3.1.2. Recursively updating the weights allows the model to predict the covariance across assets more accurately, because it effectively trains on shorter time periods, with the related advantages and disadvantages discussed earlier in this section. In summary, the adaptive minimum variance portfolio performs better than the aforementioned alternatives for most window sizes, which was expected. However, additional optimisation processes could be still beneficial to further improve performance. In the covariance matrix estimation the returns of the past M days are equally weighted towards making the prediction. Other weighting methods, such as exponential decay, give more relevance to recent price movements rather than outdated ones.

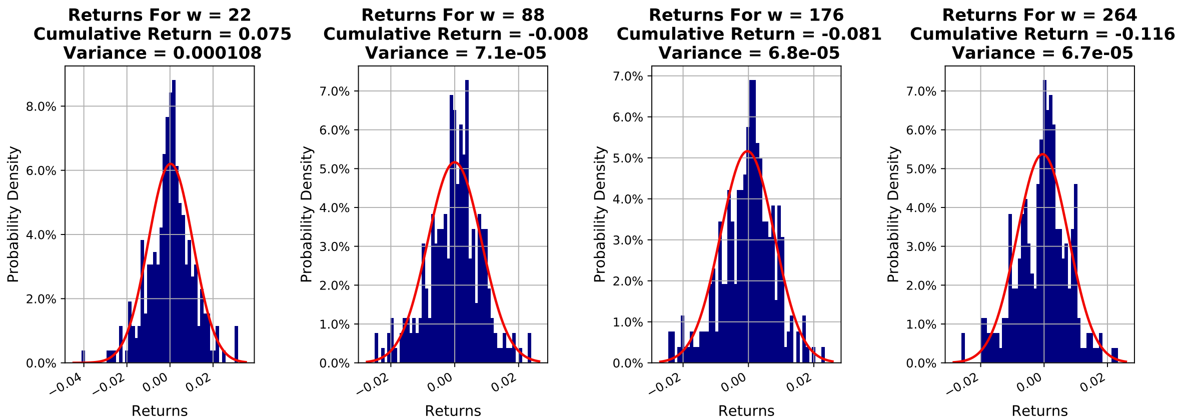

Figure 3.1.4: Returns probability distribution and statistics for adaptive minimum-variance portfolios with different window sizes.

4 Robust Statistics and Non Linear Methods

4.1 Data Import and Exploratory Data Analysis

4.1.1 Key descriptive statistics

This section will apply the several robust statistics techniques on 3 stocks (AAPL, IBM, JPM) and 1 index (DJI). In earlier sections some of these statistics have been already applied to summarise some key characteristics of data, like tendency (mean) and dispersion (variance). The descriptive statistics discussed in this section are the following:

- **Mean** - This gives an intuition of data's tendency.
- **Median** - This gives an intuition of data's tendency.
- **Inter-Quartile Range (IQR)** - This quantifies data dispersion.
- **Standard Deviation** - This quantifies data dispersion.
- **Median Absolute Deviation (MAD)** - This quantifies data dispersion.
- **Skew** - This quantifies asymmetry in the data.
- **Kurtosis** - This quantifies how much the tails differ from the tails of a Normal distribution.

Table 7 summarises the descriptive statistics for the aforementioned list of stocks and the DJI index.

4.1.2 Probability density functions (PDF)

Figure 4.1.1 shows the histogram and the corresponding fitted probability density function (PDF) of the adjusted closing price and 1-day returns for the set of stocks taken into consideration in this experiment. The PDF was estimated using a multigaussian fitting model, estimating initial values for the mean, variance and amplitude of the single gaussian components via inspection. The same analysis is shown in Figure 4.1.2 for the DJI index data.

Figure 4.1.1 demonstrates that AAPL is the most volatile stock, which requires a three-gaussian model to accurately fit the price evolution in a multi-gaussian PDF. IBM is second, with two very distinct peaks around the values of 129 and 145. Finally JPM shows the least volatile behaviour, with a price distribution that is the most similar to a normal gaussian. Given that the technology sector is characterised by an higher volatility than the financial sector, which is extremely stable, the results meet expectations. The returns plotted on the right support this reasoning further, as JPM is shown as a shorter gaussian, with a slighter smaller mean return, while IBM and AAPL are taller curves with higher mean and variance. For the DJI Index, the price is observed to follow a quite gaussian-like behaviour in Figure 4.1.2. It should be mentioned that although the mean

AAPL

	Open	High	Low	Close	Adj Close	Volume
Mean	187.686694	189.561753	185.823705	187.711953	186.174273	32704750.199203
Median	186.289993	187.399994	184.940002	186.119995	184.351776	29184000.0
StdDev	22.145621	22.281577	22.008797	22.160721	21.904664	14179721.592994
MAD	15.889999	15.610001	15.919998	15.940002	15.476044	7573900.0
IQR	36.0	36.339996	36.059998	36.755004	35.68544	16311700.0
Skew	0.259917	0.300385	0.220489	0.263849	0.29077	1.743317
Kurtosis	-0.912594	-0.924602	-0.917632	-0.932425	-0.928017	4.353182

IBM

	Open	High	Low	Close	Adj Close	Volume
Mean	138.454382	139.492072	137.329243	138.363108	134.902751	5198937.450199
Median	142.809998	143.990005	142.059998	142.710007	138.566391	4237900.0
StdDev	12.114308	11.913079	12.204633	12.028123	10.671648	3328955.530426
MAD	5.270004	5.309998	5.190002	5.230011	4.493515	920700.0
IQR	15.379998	14.720001	16.340004	15.504997	14.103938	1952950.0
Skew	-0.676024	-0.622707	-0.713446	-0.682246	-0.811222	3.192896
Kurtosis	-0.585272	-0.623607	-0.561975	-0.584037	-0.420852	11.796897

JPM

	Open	High	Low	Close	Adj Close	Volume
Mean	108.707689	109.652072	107.682988	108.606574	107.26259	14700689.243028
Median	109.18	110.529999	107.790001	109.019997	107.219269	13633000.0
StdDev	5.359081	5.20287	5.432537	5.30048	4.833316	5349770.564456
MAD	4.470001	4.309997	4.239998	4.350006	3.450157	3035400.0
IQR	8.810006	8.845002	8.845001	8.834999	7.222442	6233600.0
Skew	-0.420811	-0.376221	-0.377517	-0.374853	-0.344491	1.693457
Kurtosis	-0.322536	-0.544163	-0.2657	-0.396579	-0.105437	4.430197

DJI

	Open	High	Low	Close	Adj Close	Volume
Mean	25001.257268	25142.041965	24846.002226	24999.153581	24999.153581	332889442.231076
Median	25025.580078	25124.099609	24883.039063	25044.289063	25044.289063	313790000.0
StdDev	858.834708	815.203959	903.302186	859.132105	859.132105	94078038.14115
MAD	543.541015	537.61914	601.56836	590.720703	590.720703	50460000.0
IQR	1109.43457	1077.816406	1204.418945	1158.155273	1158.155273	108930000.0
Skew	-0.372127	-0.239367	-0.456447	-0.380147	-0.380147	1.73956
Kurtosis	0.485736	0.118153	0.557592	0.400668	0.400668	5.857581

Table 7: Descriptive Statistics for the Stocks: AAPL, IBM, JPM and the index DJI

return is positive the curve is asymmetric, namely it has a taller tail in the negative direction, which means that very high returns are less likely than very low ones.

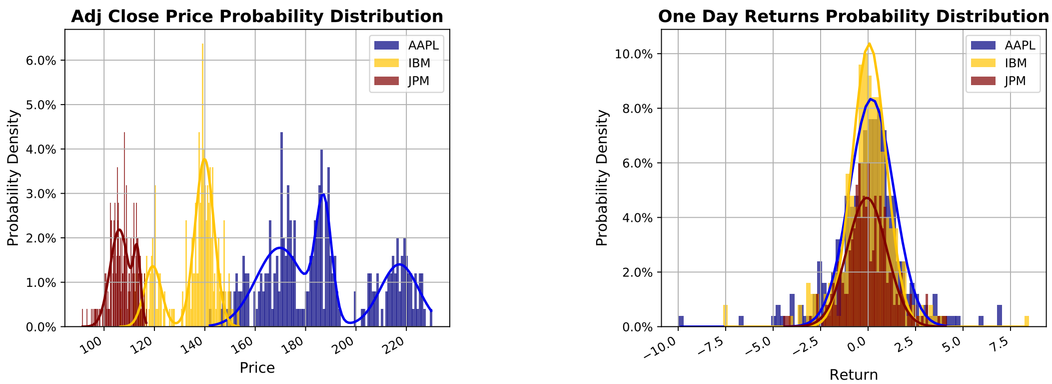

Figure 4.1.1: Adj-close prices and returns probability density function for the three stocks.

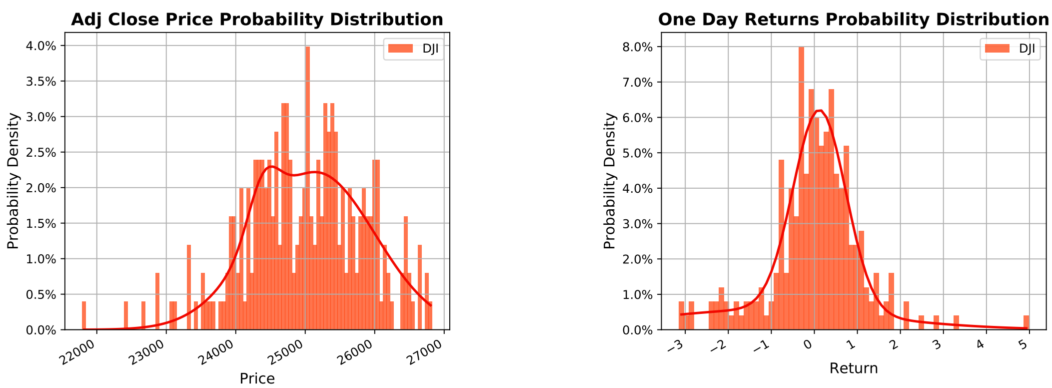

Figure 4.1.2: Adj-close prices and returns probability density function for the DJI Index.

4.1.3 Mean and median estimators analysis

This section analyses the performance of two Z-score based methods to detect outliers. The first method uses rolling price mean (μ) and standard deviation σ to classify as outliers prices that fall out of the range $[\mu - 1.5\sigma,\ \mu + 1.5\sigma]$. Figure 4.1.3 shows the region bounded by the z-score constraint, as well as the price curve and the rolling average with window $w = 5$ for the list of tickers taken into consideration throughout this section. The second method leverages the median (med) and median absolute deviation (mad) to compute the range $[med - 1.5\ mad,\ med + 1.5\ mad]$, which is used to detect outliers in the same fashion. The results yielded by the moving median are shown in Figure 4.1.4.

The faded blue surface in the plots represents the range in which prices are considered inliers. It is evident that the mean method defines is less strict, in the sense that the range is wider and therefore less points have been detected as outliers. This is coherent with the descriptive statistics used, as the mean is less robust against sudden changes in the data compared to the median, and therefore its correspondent measure of deviation will be larger, hence resulting in wider z-bounds. The final outliers count for the two discussed methods on all four assets are summarised in the first two columns of Table 8 and compared to the results obtained in Section 4.1.4.

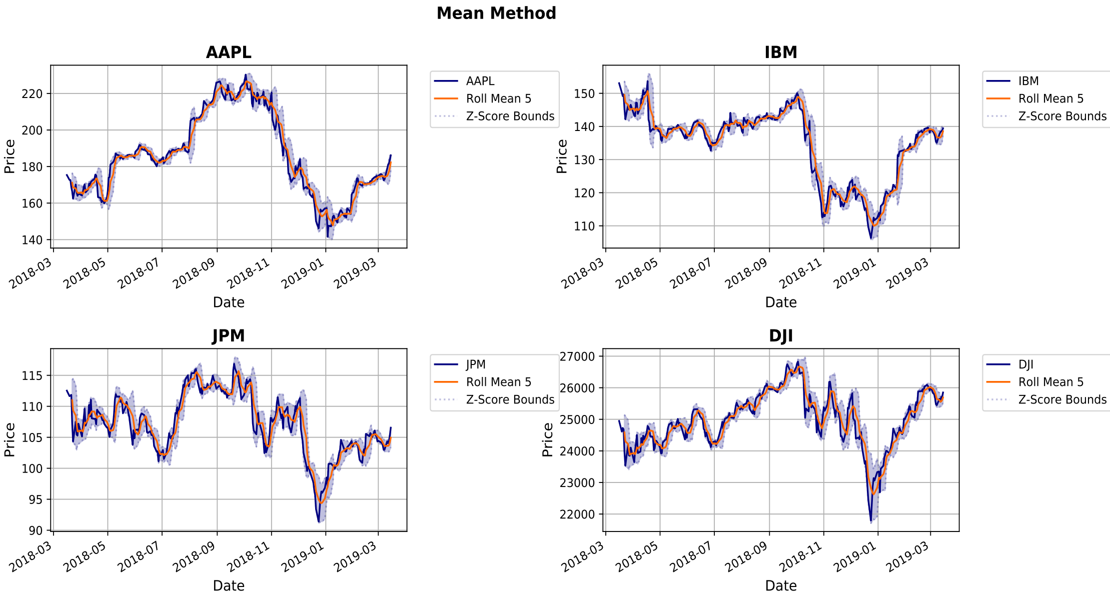

Figure 4.1.3: Moving-Average outlier detection with rolling window of 5.

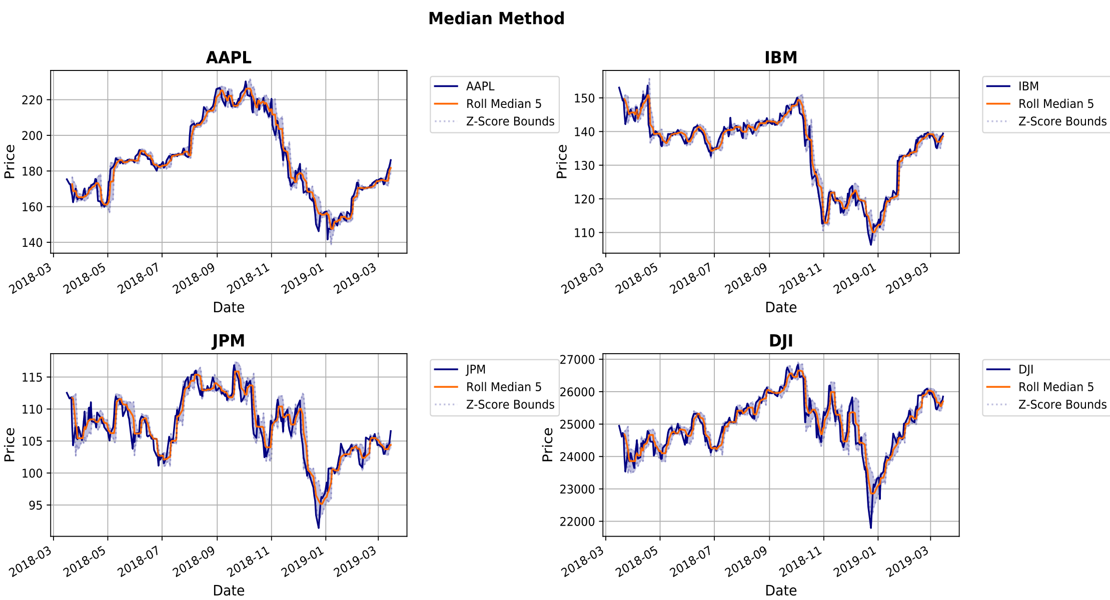

Figure 4.1.4: Moving-Median outlier detection with rolling window of 5.

4.1.4 Estimators' robustness against outliers

The same analysis discussed in Section 4.1.3 is repeated after having artificially produced outliers in the data by multiplying the asset's prices by 1.2 in the following dates: [2018-05-14, 2018-09-14, 2018-12-14, 2019-01-14]. Figures 4.1.5 and 4.1.6 show the results for the mean and median methods replicated on the artificially tweaked dataset.

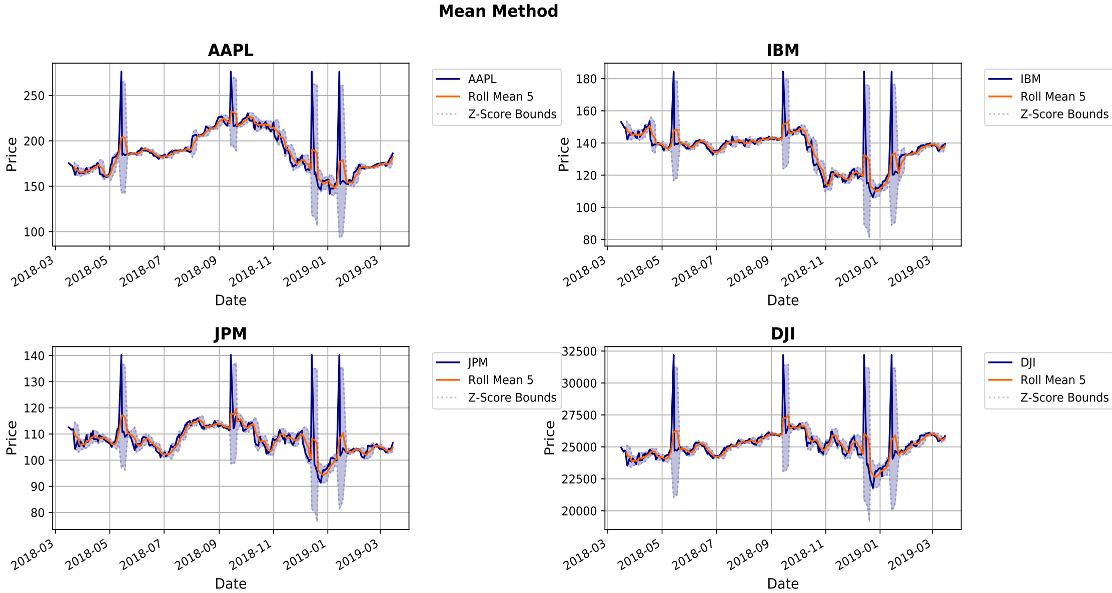

Figure 4.1.5: Moving-Average outlier detection with rolling window of 5 on data with artificial outliers.

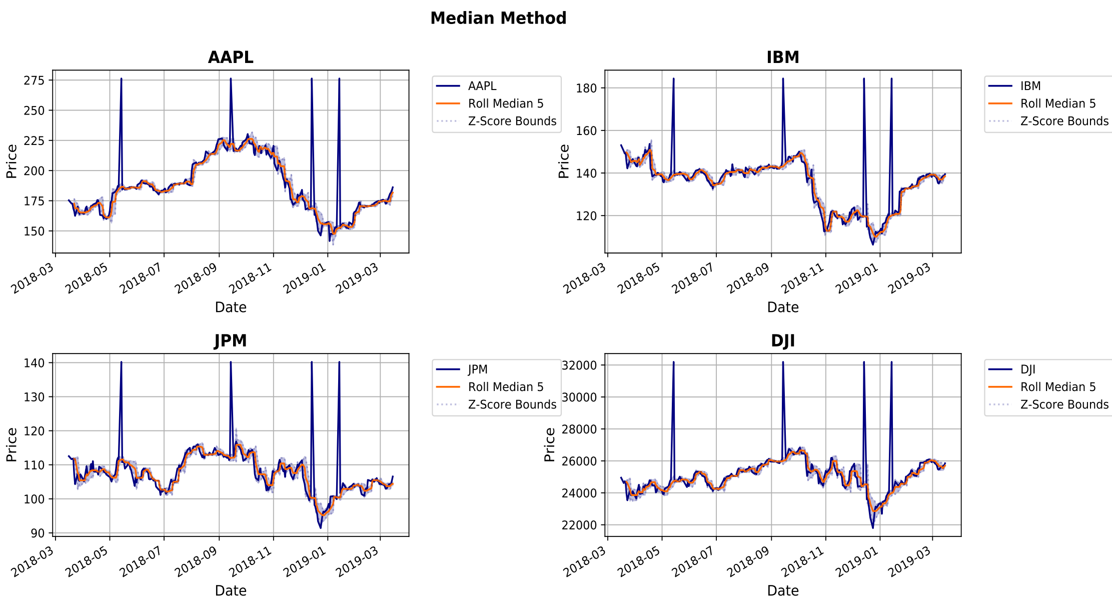

Figure 4.1.6: Moving-Median outlier detection with rolling window of 5 on data with artificial outliers.

The higher robustness of the moving median descriptive statistic over the moving average is confirmed by these results, which clearly show that the range is highly affected for the first method (mean), while the median z-score bounds are not significantly affected by the presence of large outliers. To summarise, the median together with the median absolute deviation outperformed the mean with standard deviation method on highly corrupted data. The final outlier count for the two methods on all four assets are summarised in the third and fourth columns of Table 8.

	Mean - STD	Median - MAD	Mean - STD With Outliers	Median - MAD With Outliers
AAPL	30	103	32	102
IBM	31	94	31	93
JPM	33	105	33	101
DJI	30	97	29	96

Table 8: Outliers found for both the mean and median methods with and without artificial outliers.

4.1.5 Box plot analysis

Box plots provide an illustration of the median (2nd quartile), maximum value, minimum value, and the inter-quartile range statistics.

The five vertical lines represent the minimum, the first three quartiles and the maximum, in this order. The box goes from the first to the third quartile respectively, with these being the boundaries of the box (the IQR) and the median plotted in blue. Finally the dots outside of the plot denote outliers. Figure 4.1.7 shows the box plots for AAPL, IBM, JPM and the DJ index.

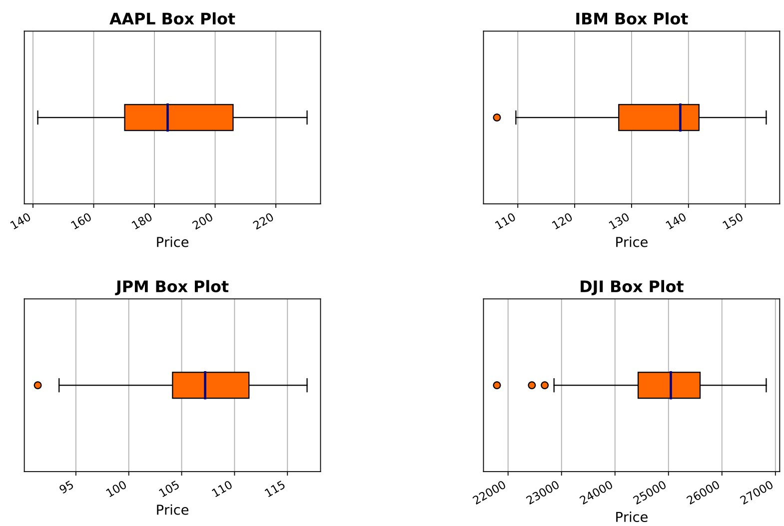

Figure 4.1.7: Box Plots.

The box plots visually confirm the descriptive statistics summarised in the Adj Close column in Table 7. It can be observed that IBM has the largest skew magnitude, which is denoted by the asymmetricity in the two tails. The high and negative skew of IBM prices implies much lower prices than the median have been observed. Furthermore AAPL has the biggest IQR by a large amount, which is denoted by the width of the orange box. Large inter-quartile ranges imply an high volatility of the asset.

4.2 Robust Estimators

This section involves the implementation, analysis and assessment of the following estimators:

- Robust location estimator: **Median**.
- Robust scale estimator: **IQR**.
- Median Absolute Deviation: **MAD**.

4.2.1 Robust estimators' definition

The Python code in Listing 2 defines the custom function that calculate the aforementioned estimators:

```
def my_median(data):
    data = data.sort_values()
    # if length of sequence is odd, simply take the middle value.
    # else return the average of the two middle values.
    if len(data) % 2 == 0:
        return data[int(len(data) / 2)]
    else:
        median_low  = data[int(len(data) / 2) - 1]
        median_high = data[int(len(data) / 2)]
        return (median_low + median_high) / 2

def my_iqr(data):
    data = data.sort_values()

    # if length of sequence is odd, simply take the first and third
       quartiles.
    # else return the average of the two values around the quartiles
       .
    if len(data) % 2 == 0:
        quart1 = data[int(len(data) / 4)]
        quart3 = data[int(len(data) * 3 / 4)]
        return quart3 - quart1
    else:
        quart1_low = data[int(len(data) / 4) - 1]
        quart1_high = data[int(len(data) / 4)]
        quart3_low = data[int(len(data) * 3 / 4) - 1]
        quart3_high = data[int(len(data) * 3 / 4)]

        return (quart3_high + quart3_low) / 2 - \
               (quart1_high + quart1_low) / 2
```

```python
def my_mad(data):
    # compute the magnitude of the data deviation from the mean.
    dev = abs(data - my_median(data))
    # return the median of the deviations.
    return my_median(dev)
```

Listing 2: Custom-made functions for computing the median, IQR and MAD estimators.

The Median is computed by sorting the input values and then taking the number which is in the middle of the sorted sequence. If the sequence has even length though, the mathematical median is defined as the arithmetic average of the two middle elements of the sorted sequence. This logic is replicated in the code provided in Listings 1. Similarly for the quartile computations in the inter-quartile range function, the sorted values are sampled such that the value on the first and third quartiles are computed accordingly to the the sequence's length parity. The absolute difference of the two values just computed gives the interquartile range. Finally the median absolute deviation is computed by the function `my_mad`. This is achieved by firstly calculating the absolute deviation of each data point from the median of the data and then by taking the median of the deviations.

4.2.2 Complexity of custom estimators

Sorting the `pandas.Series` is the first step in computing both the median and the inter-quartile range. Pandas documentation defines the syntax of the built-in function `Series.sort_values` as shown in Listing 3:

```python
Series.sort_values(axis=0, ascending=True, inplace=False,
                   kind='quicksort', na_position='last')
```

Listing 3: Built-in `Series.sort_values` function declaration.

Arguments that are equated to a value in the function declaration in Python are parsed so that the value becomes the default of that argument, in case the function call does not explicitly specify otherwise. Therefore the merging used in the custom function applies the "quicksort" algorithm, that has complexity $O(n \cdot \log(n))$, where n is the length of the `pandas.Series`. Then, the operations of complexity $O(1)$ are: the arithmetic and modulus operations, the logic comparison with zero, the accesses to a series element through the subscript operator, the calls to built-in functions such as `len` and the dynamic castings `int`. This total of seven operations (including modulus), one logic comparison, three indexing operations, three `len` function calls and three dynamic casting operations have complexity $17 \cdot O(1)$ and can be neglected in the overall complexity computation. A similar analysis can be carried on the function `my_iqr`. The sorting operation has complexity of $O(n \cdot \log(n))$ as before, while the number of $O(1)$ operations performed doubles (total of 34). Finally the MAD computation applies the `abs` built-in function to take the absolute value of the deviations, which are computed by subtracting the data from the median, which is calculated using the custom `my_median` function that is $O(n \cdot \log(n))$ complex. The `abs` and subtraction operations have complexity $O(1)$ as before, but they are now applied leveraging Python's vectorisation capabilities to each element of the input `pandas.Series`. Therefore, applying the two operations to the whole series has complexity $n \cdot O(1) = O(n)$. Finally, the function `my_median` is called again on the input sequence `dev`, which has been defined as the series of deviations and is by definition of the same length of the original data which is n. Therefore, the median operation on the deviations has complexity $O(n \cdot \log(n))$. Studying the overall computational cost of the custom functions discussed in this section it should be mentioned that for reasonably small input sequences the $O(1)$ operations are not neglegible, thus making the median estimator the fastest, closely followed by the IQR, which performs just a few more $O(1)$ operations and finally the MAD. However, as $n \to \infty$ the Big O notation allows to neglect the computationally less expensive operations and scaling factors, hence resulting in the complexities summarised in Table 9.

Function	Complexity	Complexity as $n \to \infty$
`my_median`	$O(n \cdot \log(n)) + 17 \cdot O(1)$	$O(n \cdot \log(n))$
`my_iqr`	$O(n \cdot \log(n)) + 34(1)$	$O(n \cdot \log(n))$
`my_median`	$2 \cdot O(n \cdot \log(n)) + 2 \cdot O(n)$	$O(n \cdot \log(n))$

Table 9: Complexity analysis for the custom-defined functions of estimators.

4.2.3 Breakdown points analysis

Breakdown point analysis measures the robustness of an estimator, i.e. the maximum rate of outliers that an estimator can tolerate. It is defined as a decimal variable in the range $[0, 0.5]$ where high values denote high robustness of the estimators. The upper limit of 0.5 is related to the logical assumption that if more than half of the data constitutes outliers, then the outliers would become the inliers and viceversa. The median estimator computation is based on the middle value of the sorted series, which means that up to half of the sequences' length minus one can be contaminated without affecting the estimation. Mathematically, the breakdown point (BP) for the Median estimator can be defined as:

$$BP_{median} = \frac{\left(\frac{n-1}{2}\right)}{n} = \frac{1}{2} - \frac{1}{2n} \tag{4.2.1}$$

As the input series length gets larger ($n \to \infty$), the asymptotic breakdown point is:

$$\lim_{n\to\infty}[BP_{median}] = \lim_{n\to\infty}\left[\frac{1}{2} - \frac{1}{2n}\right] = \frac{1}{2} \tag{4.2.2}$$

Hence the median has the highest possible robustness against outliers, with a breakdown point of 0.5. The IQR instead takes into account the values of the first and third quartile. This is equivalent to splitting the sorted data into two sequences of equal length and considering the median of each sequence. Therefore the breakdown point of IQR is half the one of the median, so 0.25. Finally, the MAD takes into account the median of the data to calculate the deviations and then the median of the absolute deviations. Therefore, the breakdown point will be the same as the median, which is 0.5.

4.3 Robust and OLS Regression

This section aims to compare the performances of the following two regression methods:

- **OLS** (Ordinary Least Squares).
- **Huber** (Robust Regression).

Table 10 summarises the estimated values for both parameters α and β, obtained in the practical experiments further discussed in Section 4.3.1 and 4.3.2 by both OLS and Huber regression methods, respectively.

	AAPL	IBM	JPM	DJI
OLS α	0.000165	-0.000441	-0.000316	-0.0
OLS β	1.32558	0.960092	0.931408	1.0
Huber α	-0.00013	-0.000509	-0.000801	0.0
Huber β	1.270212	0.973562	0.919662	1.0

Table 10: Parameters α and β for both OLS and Huber Regression Methods.

4.3.1 Ordinary least squares (OLS) regression

The regression problem solved by OLS is defined by the following mathematical equation:

$$\boldsymbol{r} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e} \quad \rightarrow \quad \boldsymbol{e} = \boldsymbol{r} - \boldsymbol{X}\boldsymbol{b} \tag{4.3.1}$$

where $\boldsymbol{r}$ is the vector of returns for the stock, $\boldsymbol{X}$ is a two column matrix where the first column is populated with ones, while the second column with the returns of the DJI and $\boldsymbol{e}$ is the error vector. OLS regression solves a minimisation problem of the magnitude of the squared error as follows:

$$\min \ ||\boldsymbol{e}||^2 = ||\boldsymbol{r} - \boldsymbol{X}\boldsymbol{b}||^2 \tag{4.3.2}$$

which is optimal when $\boldsymbol{b} = \boldsymbol{b}^*$, defined as:

$$\boldsymbol{b}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{r} \tag{4.3.3}$$

Minimising the squared errors is a methodology significantly sensitive to data corruption as the presence of outliers provokes a spike in the squared error metric. Figure 4.3.1 shows the returns predictions and the regression linear approximation achieved via the OLS method.

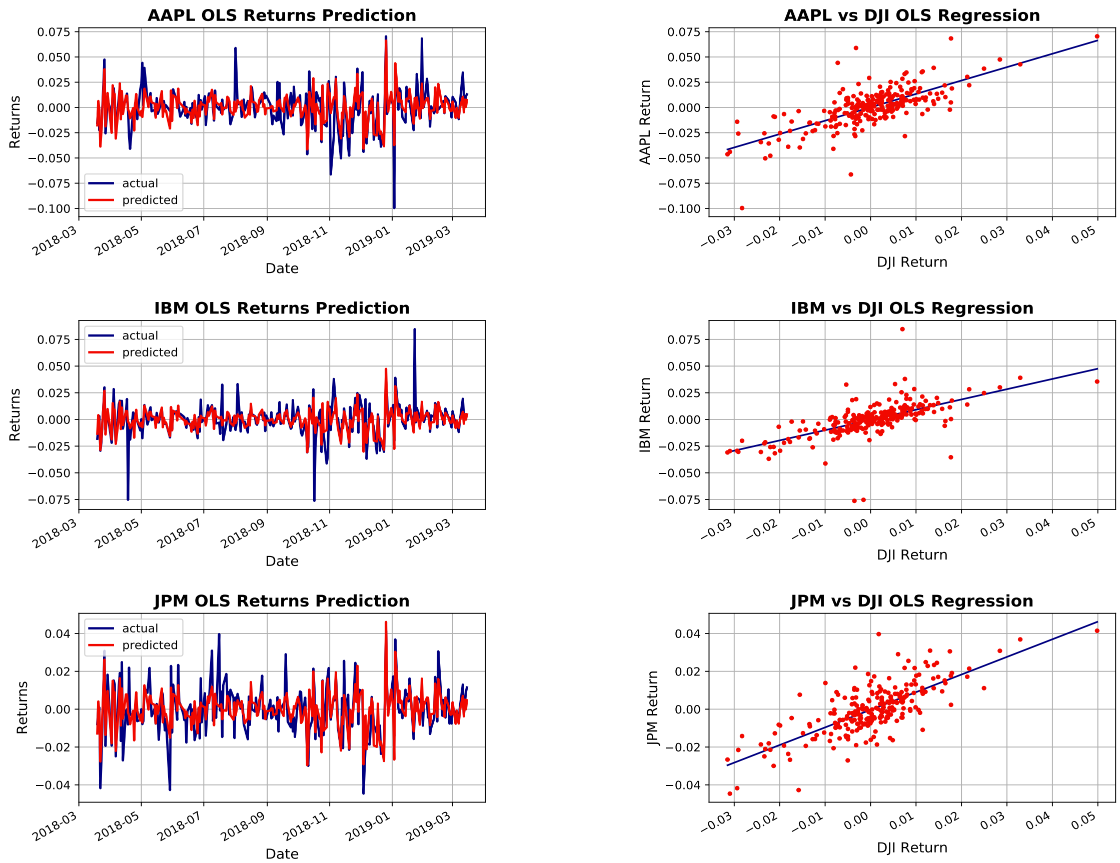

Figure 4.3.1: Returns predictions (left) and stock versus DJI index regression (right) for the OLS method.

4.3.2 Huber (robust) regression

The Huber regression is more robust than the OLS method as it limits the overshoot of high values transforming the objective function to linear for errors that exceed a certain threshold. The joint minimisation problem is defined as:

$$\min \begin{cases} ||\boldsymbol{e}||^2 = ||\boldsymbol{r} - \boldsymbol{X}\boldsymbol{b}||^2, & \text{for} \frac{||\boldsymbol{r} - \boldsymbol{X}\boldsymbol{b}||}{\sigma} < \epsilon \\ ||\boldsymbol{e}|| = ||\boldsymbol{r} - \boldsymbol{X}\boldsymbol{b}||, & \text{for} \frac{||\boldsymbol{r} - \boldsymbol{X}\boldsymbol{b}||}{\sigma} \geq \epsilon \end{cases} \tag{4.3.4}$$

Tuning the parameter ϵ sets the threshold after which a point is considered by the regression as an outlier. Figure 4.3.3 shows the returns predictions and the linear regression approximation achieved through the Huber method.

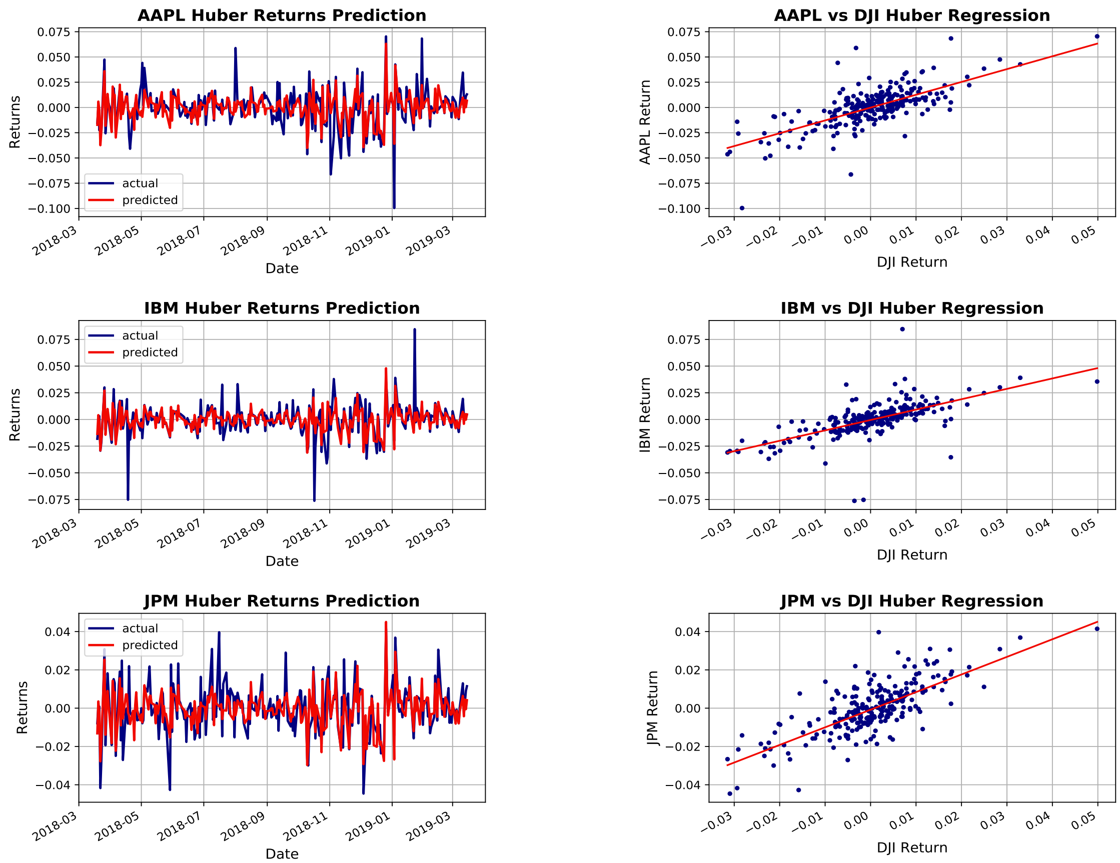

Figure 4.3.2: Returns predictions (left) and stock versus DJI index regression (right) for the Huber method.

4.3.3 Robustness of regression methods against outliers

The purpose of this section is to repeat the calculations performed in previous sections on the same dataset after the addition of some outliers. The outliers are generated by multiplying the returns data by a factor of 1.2 in the same set of dates used in Section 4.1.4. Table 11 summarises the estimated values for both parameters α and β obtained by both OLS and Huber regression methods on the dataset with artificial outliers. Comparing the values obtained for the regression parameters,

	AAPL	IBM	JPM	DJI
OLS with Outliers α	0.00035	-0.000063	-0.000467	0.0
OLS with Outliers β	1.350465	1.22204	0.906468	1.0
Huber with Outliers α	0.000122	-0.000339	-0.000933	0.0
Huber with Outliers β	1.330773	1.066643	0.890359	1.0

Table 11: Value of the parameters α and β for both OLS and Huber Regression methods on data with artificial outliers.

it can be observed that Huber reacts more robustly to the abrupt changes in returns caused by the artificial outliers, varying its estimated parameters of approximately 55%, while OLS' regression parameters change of more than 80% in value.

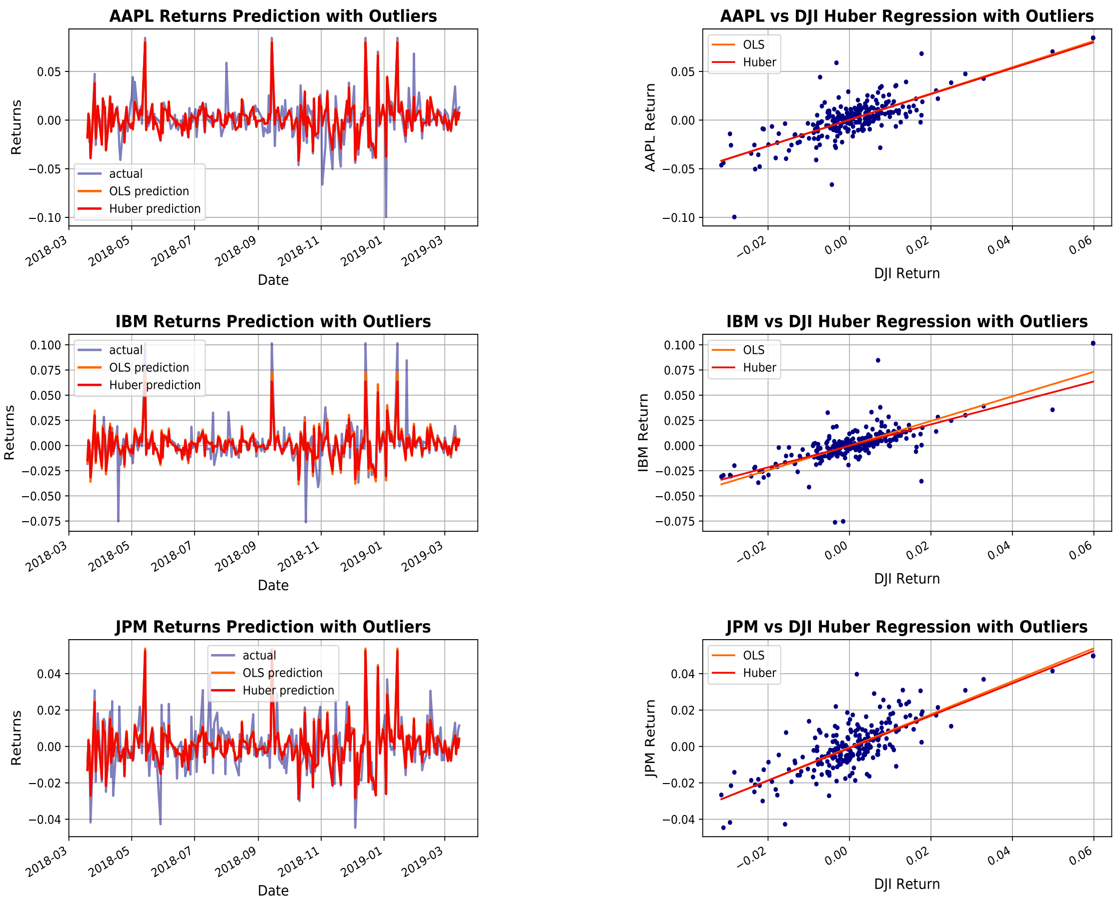

Figure 4.3.3: Returns predictions (left) and stock vs DJI index regression (right) for the both OLS and Huber methods on data with artificial outliers.

4.4 Robust Trading Strategies

The Moving Average Crossover strategy is a robust trading strategy that computes the buying and selling opportunities based on the rolling mean on both a 20-day and 50-day window according to the following rules:

- Buy X shares of a stock when its 20-day MA > 50-day MA
- Sell X shares of the stock when its 20-day MA < 50-day MA

This section will evaluate the robustness of the Moving Average strategy against outliers, as well as the Moving Median alternative.

4.4.1 Moving average crossover

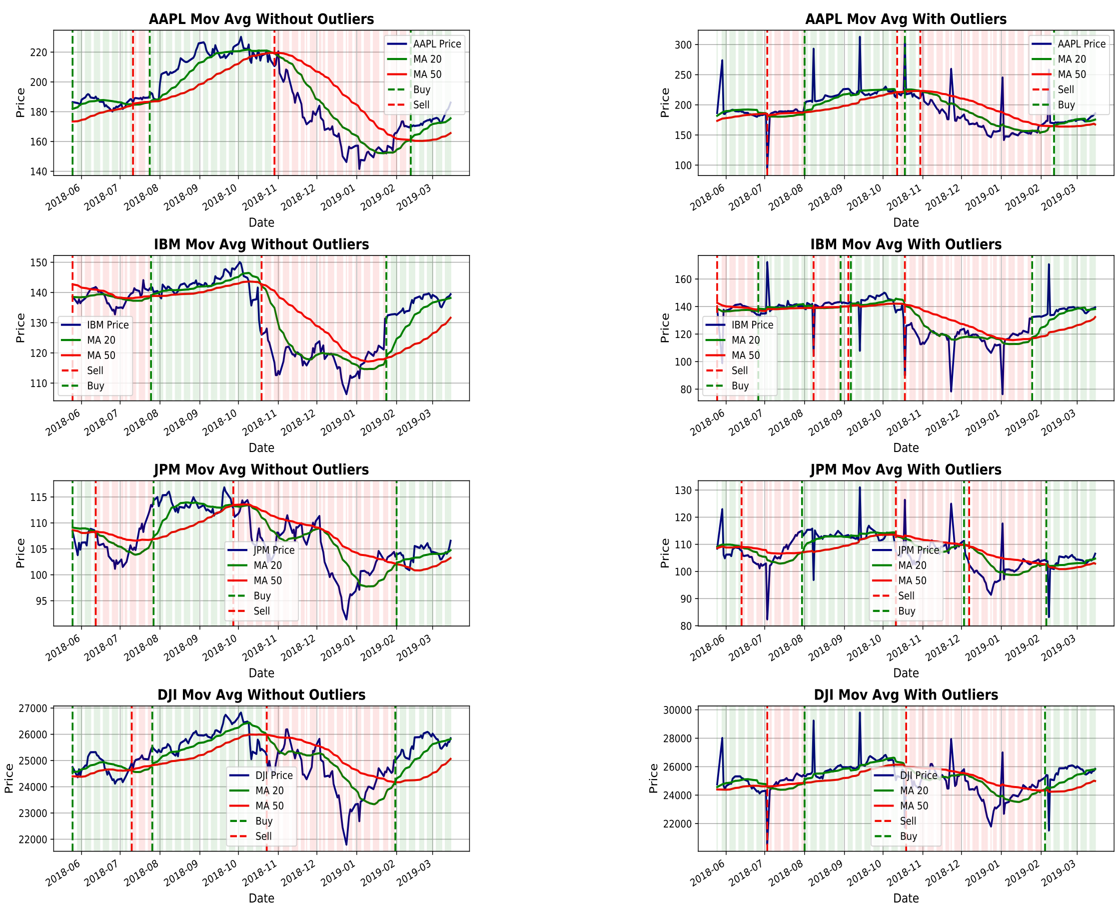

Figure 4.4.1: Moving Average Crossover strategy for the Adjusted-close values of AAPL, IBM, JPM and the DJI.

Figure 4.4.1 shows the Moving Average Cross-over strategy for the stocks and index taken into consideration in this section. Visually inspecting the buying and selling instants highlighted in the plots, it is evident that the final strategy is significantly influenced by the presence of corrupted data. This behaviour is expected as the mean is not a robust descriptive statistic against outliers. The first column in Table 12 summarises the similarity percentage of the two strategies identified for the real prices and the artificially corrupted ones. According to the table IBM prices are the sequence that is most effected by the presence of outliers, which can be appreciated comparing the two plots in the second row of images in Figure 4.4.1.

4.4.2 Moving median crossover

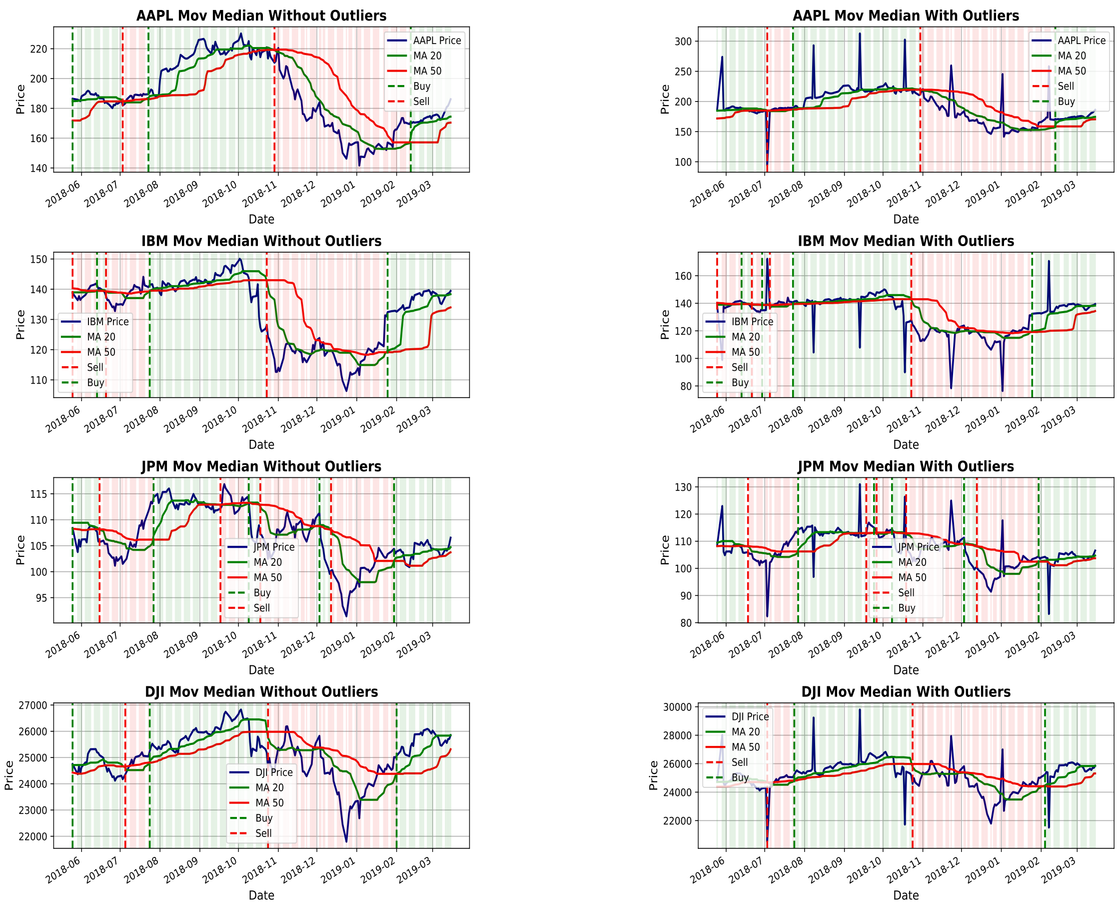

Figure 4.4.2: Moving Median Crossover strategy for the Adjusted-close values of AAPL, IBM, JPM and the DJI.

The analysis performed in Section 4.4.1 is repeated using the rolling median as descriptive statistic. Figure 4.4.2 shows the Moving Median Crossover strategy for the stocks and index taken into consideration in this section. Comparing the results obtained by the median to the ones discussed in the previous section for the mean, it is evident that the former method is significantly more robust. The second column in Table 12 summarises the similarity percentage of the moving median strategy on real and corrupted data. The similarity can also be visually assessed by comparing the plots in Figure 4.4.2. In this case there is almost no difference in the two strategies, which denotes the high robustness of the trading stategy.

	MA Strategy Similarity w/out Outliers	MM Strategy Similarity w/out Outliers
AAPL	91.584158	99.50495
IBM	80.693069	97.524752
JPM	91.584158	96.039604
DJI	94.059406	99.009901

Table 12: Robustness analysis on the percentage change in strategies caused by outliers for both the MA and MM Crossover methods.

5 Graphs in Finance

5.1 SP stocks subset

In this section, graphs are used to provide a visual intuition of a subset of stocks within the SP 500 index. For the purpose of the exercise, a list of companies listed in the GICS Sector of "Information Technology" and based in either San Jose, California or Santa Clara, California have been selected. The choice of restricting the analysis to a single sector and a particular geographic location has been made to limit the complexity of the relationship that link companies in the real world setting. Variables like State laws, local population and other location-specific factors are therefore removed from the equation, allowing the graphical study to focus on a less complex system. Despite these constraints, the choice of the "Information Technology" sector allows plenty of variety on the sub-industry domain level, presenting companies that provide services in the semiconductors field, data processing, communication and software. Table 13 shows the information relative to the set of companies taken into consideration in this section.

Symbol	Security	GICS Sector	GICS Sub Industry	Headquarters Location
ADBE	Adobe Systems Inc	Information Technology	Application Software	San Jose, California
CDNS	Cadence Design Systems	Information Technology	Application Software	San Jose, California
ANET	Arista Networks	Information Technology	Communications Equipment	Santa Clara, California
CSCO	Cisco Systems	Information Technology	Communications Equipment	San Jose, California
PYPL	PayPal	Information Technology	Data Processing & Outsourced Services	San Jose, California
AMAT	Applied Materials Inc.	Information Technology	Semiconductor Equipment	Santa Clara, California
INTC	Intel Corp.	Information Technology	Semiconductors	Santa Clara, California
MXIM	Maxim Integrated Products Inc	Information Technology	Semiconductors	San Jose, California
NVDA	Nvidia Corporation	Information Technology	Semiconductors	Santa Clara, California
XLNX	Xilinx	Information Technology	Semiconductors	San Jose, California

Table 13: Set of the S&P 500 companies in the Information Technology field analysed.

5.2 Correlation matrix and network

In this section the correlation between the companies discussed in Section 5.1 is computed and visualised through a correlation matrix and graphs. Figure 5.2.1 shows the correlation matrix for the ten assets. Trivially, self-correlation is shown on the diagonal with the maximum value of 1 and coloured in dark blue, while cross-asset correlations are labelled and colour-coded according to the colour map shown on the right hand side. The mean correlation for each company is shown next to the correlation matrix.

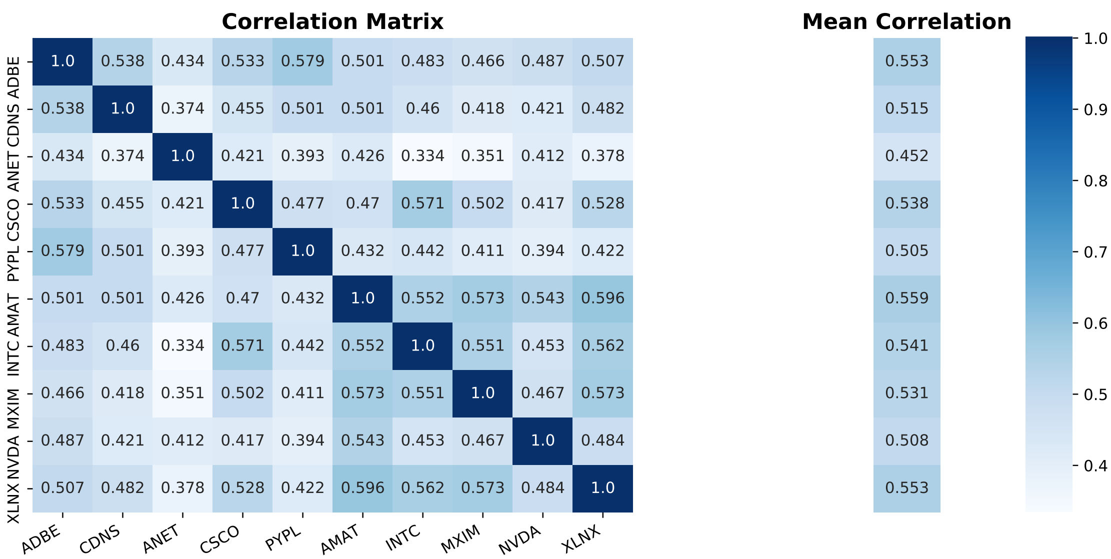

Figure 5.2.1: Correlation matrix and mean correlation of the ten assets.

By visual inspection, the minimum correlation is 0.334 (INTC and ANET), while all other values lie in the range $[0.35, 0.6]$. Therefore, the graphical analysis of correlations is shown in Figure 5.2.2 for minimum correlation thresholds of $[0.35, 0.4, 0.45, 0.5]$, in order to convey a graphical interpretation of the stronger degree of correlation existing amongst a smaller subset of companies. For the same purpose, the colour of the edges is colour-coded, with darker shades of blue indicating stronger connections. Finally, the size of the nodes is also dynamic, with larger nodes indicating companies with a higher mean correlation.

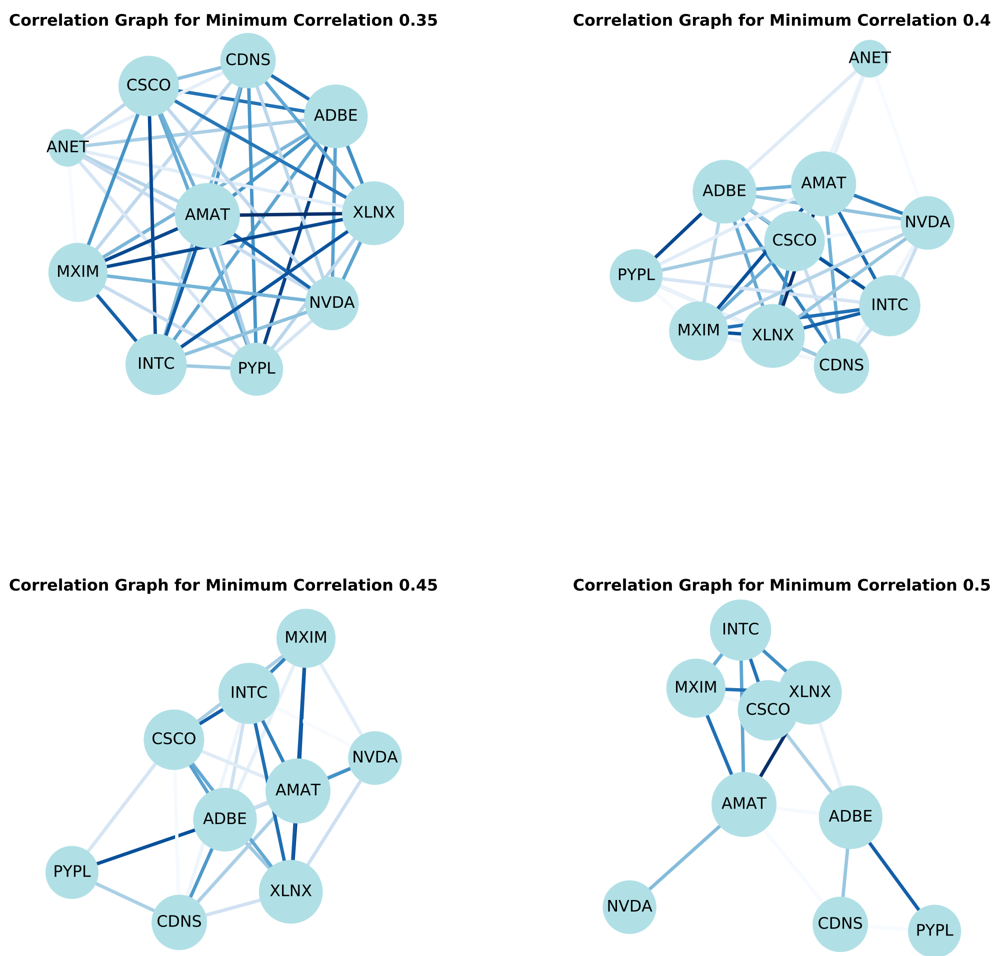

Figure 5.2.2: Correlation network for different correlation thresholds.

5.3 Correlation network analysis

The features discussed in Section 5.2 and illustrated in the graphs of Figure 5.2.2 facilitate the identification of the biggest players and the amount of interactions among companies in the Information Technology market sector, as well as the recognition of patterns and groups linked more strongly than others. As expected, companies like Arista Networks (ANET) are less interlinked with the other companies on the list, since they operate in sub-industries that are less prone to strong interaction with numerous other companies of the sector. Moreover, their presence in the market is not as solid as other considered companies. These weaker nodes are visually smaller and appear in a corner of the graph, with very light connections to few other companies. On the contrary, companies like Xilinx (XLNX) and Adobe Systems Inc. (ADBE), that respectively provide widely used software and hardware, take a central position in the plots and display numerous strong connections with the other companies on the list. Moreover, as the minimum correlation threshold increases, clusters of companies start to appear, representing groups that closely cooperate with one another. In the last plot in Figure 5.2.2, two main groups are identified, with Adobe, Paypal and Cadence Design Systems forming separate clusters.

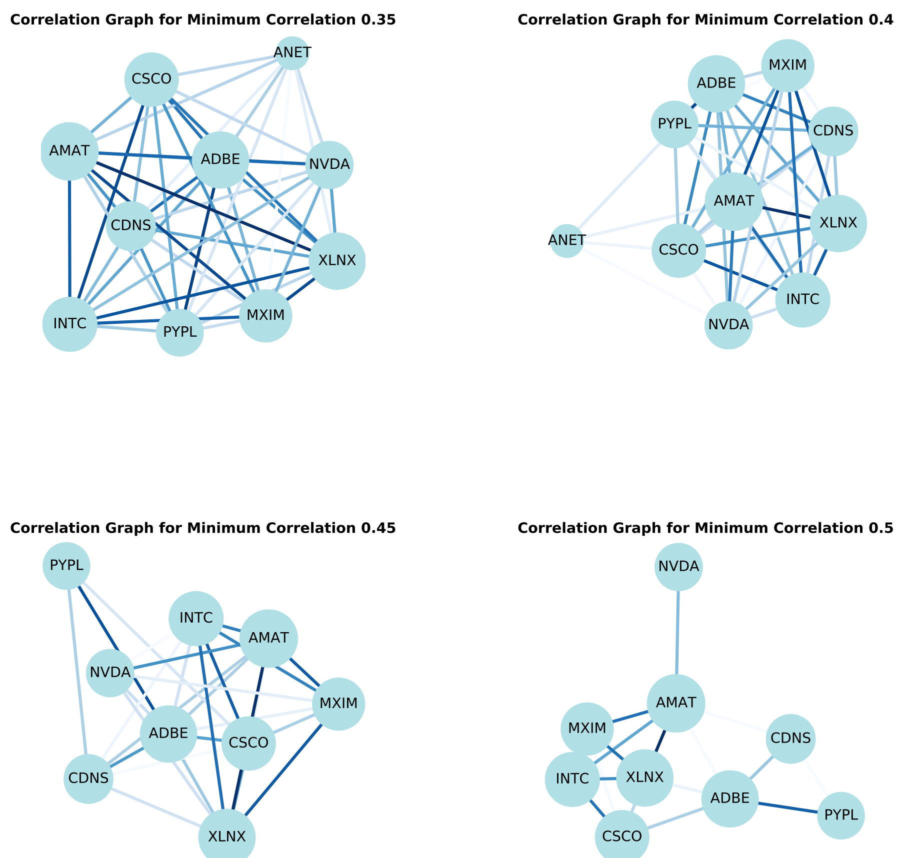

Figure 5.3.1: Correlation network for different correlation thresholds after time-series data reshuffling.

Correlation is independent of data ordering as it measures the coefficient of similarity between two stocks. Due to the internal implementation of the `networkx` package the exact visualisation of data is non-deterministic in terms of the position of the nodes across the figure space. However, the weights of the connections and nodes is deterministic, since they are computed from the confusion matrix, which is deterministic itself. Therefore, shuffling data both on the rows and columns axes will result in plots that are equivalent in all information presented, with the exception of the arrangement of the nodes. Figure 5.3.1 shows the same graphs of Figure 5.2.2 recomputed after reordering the input data.

5.4 Dynamic time warping distance matrix and network

This section explores an alternative similarity measure and compares its results to those obtained in the previous sections when using correlation. For the purpose of this exercise, Dynamic Time Warping (DTW) has been chosen. DTW is a method that computes an optimal match between two given sequences of arbitrary length, as the mapping from the one data sequence to the other is not necessarily one-to-one. The higher complexity of this method allows it to recognise patterns in the given data that are not highlighted by a simple correlation analysis. For this reason, DTW is widely used in literature as a similarity measure in stock returns analysis and prediction. Figure 5.4.2 shows the DTW distance of each asset from one another.

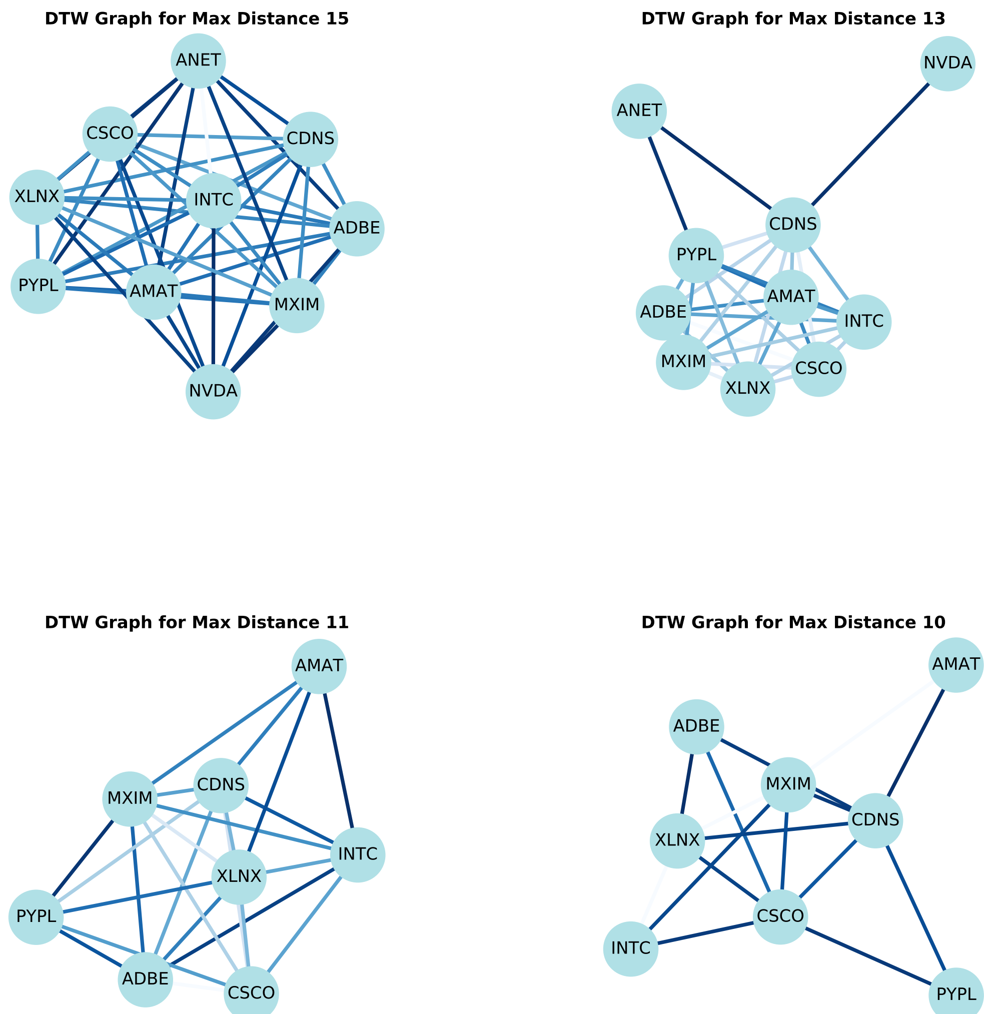

Figure 5.4.1: Dynamic Time Warping network for different DTW distance thresholds.

The main diagonal is populated with zeros because the dynamic time warping distance of an asset's returns data to itself is zero. This means that the higher the distance the smaller the similarity among the two companies. In order to visually capture this concept in the graphs shown in Figure

5.4.1, the darkness of the connections and the size of the nodes are inversely proportional to the DTW distance and mean DTW distance respectively, i.e. darker connections denote more similar assets and larger nodes indicate assets with an high average of similarities. Indeed, Dynamic Time Warping identifies different patterns compared to the ones found in Section 5.2. The final plot in Figure 5.4.1 shows a clear cluster of seven companies tightly connected together, which does not match either of those identified in Figure 5.2.2.

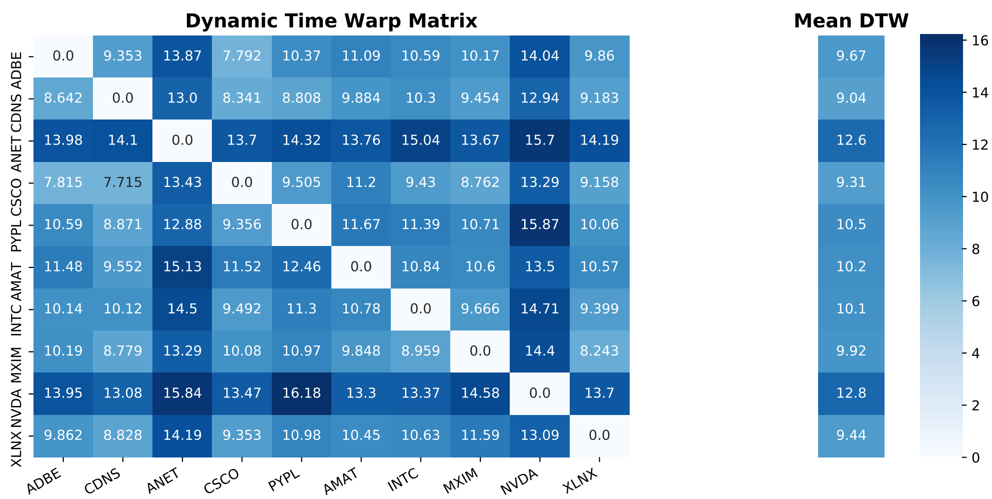

Figure 5.4.2: Dynamic Time Warping distance matrix for the ten companies.

As opposed to the correlation, which is time-invariant, the Dynamic Time Warping is dependent on the time-series order, therefore reshuffling the returns data would completely change the outcome of the analysis. Figures 5.4.3 and 5.4.4 show the Dynamic Time Warping matrix and graph on reshuffled returns data.

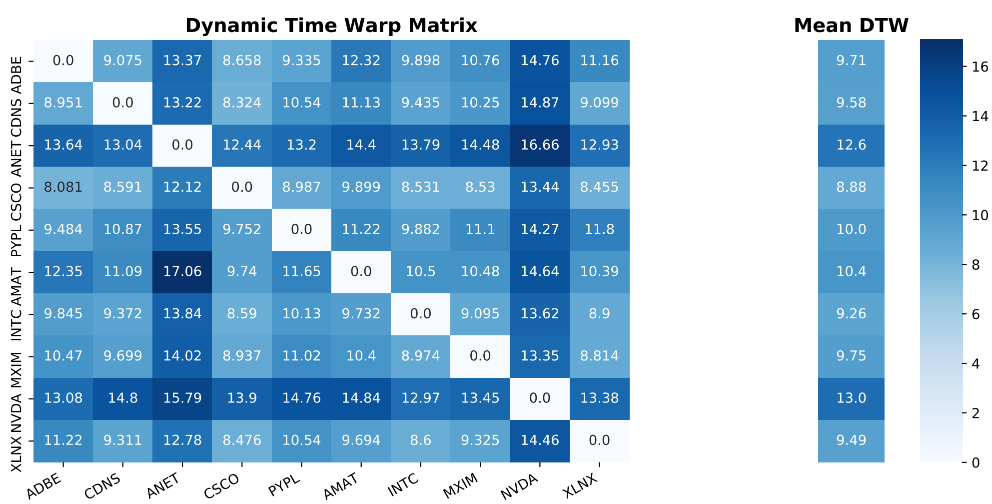

Figure 5.4.3: Dynamic Time Warping distance matrix for the ten companies after time-series data reshuffling.

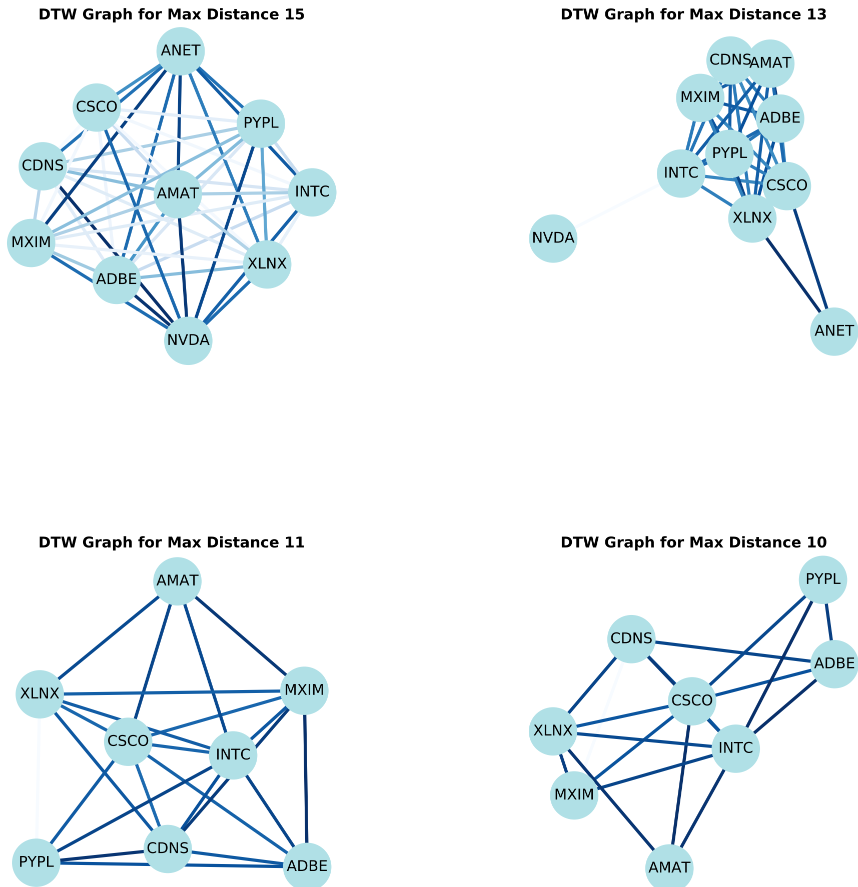

Figure 5.4.4: Dynamic Time Warping network for the ten companies after time-series data reshuffling for different DTW distance thresholds.

5.5 Disadvantages of considering raw prices

This final section aims to discuss the usage of raw prices instead of the logarithmic returns in the analysis carried in the previous sections. The main issue with using raw prices is that, as opposed to returns, prices are not independent from one another. Given a time-series of raw prices $\boldsymbol{p}$ of length t where $\boldsymbol{p}_i$ denotes the price of the asset at time i the asset returns are defined as follows:

$$\boldsymbol{r} = \frac{p_i - p_{i-1}}{p_{i-1}} \quad for \;\; i = 1, ..., t \tag{5.5.1}$$

Using the definition of returns given in Equation (5.5.1), the price at each time-step can be rewritten in terms of the returns, giving the following equation for the final price of the asset at time t (p_t):

$$p_t = p_0 + p_0 r_1 + p_1 r_2 + \cdots + p_{t-1} r_t \tag{5.5.2}$$

Equation (5.5.2) can be used recursively to rewrite all prices at each time-step in terms of a linear combination of previous prices times returns. Therefore, each price can be substituted by a linear combination of only the initial price and the returns. In this substitution earlier returns will retain higher importance, as they appear more often in the equation, whereas r_t will only be included once, and r_1 will contribute t times. Physically this means that early price movements will have a higher significance in the computation of the correlation, compared to more recent price changes.

References

[1] Danilo, Mandic (2021) ”Signal Processing and Machine Learning for Finance Lecture Notes”