COURSEWORK

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Financial Signal Processing

Author:

Adam Horsler (CID: 01738592)

Date: April 5, 2023

Contents

List of Figures

Section 1: Regression Methods

1.1: Processing stock price data in python

1.1.1

Figure 1 shows the log plot of the SPX index. Log plots provide several benefits over normal ones for future analysis, for example:

- Better representation of the proportional change in stock prices, rather than the absolute change.
- Reduce effects of volatility and extreme values.

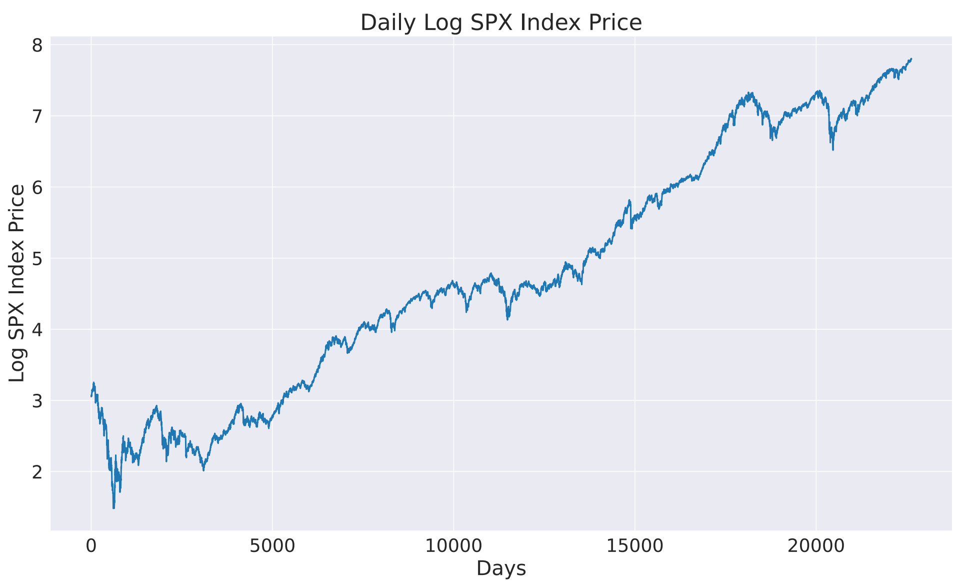


Figure 1: Daily Log SPX Index Prices

1.1.2

A series is considered stationary when it's statistical properties (e.g. mean and standard deviation (std)) are independent of time. The rolling mean and standard deviation of log SPX Index is shown in Figure 2. The series is not stationary because the mean shows a clear upward trend, from 3 at the start to 7 towards the end.

In addition, the standard deviation has decreased from 0.15 at the start to 0.05 towards the end.

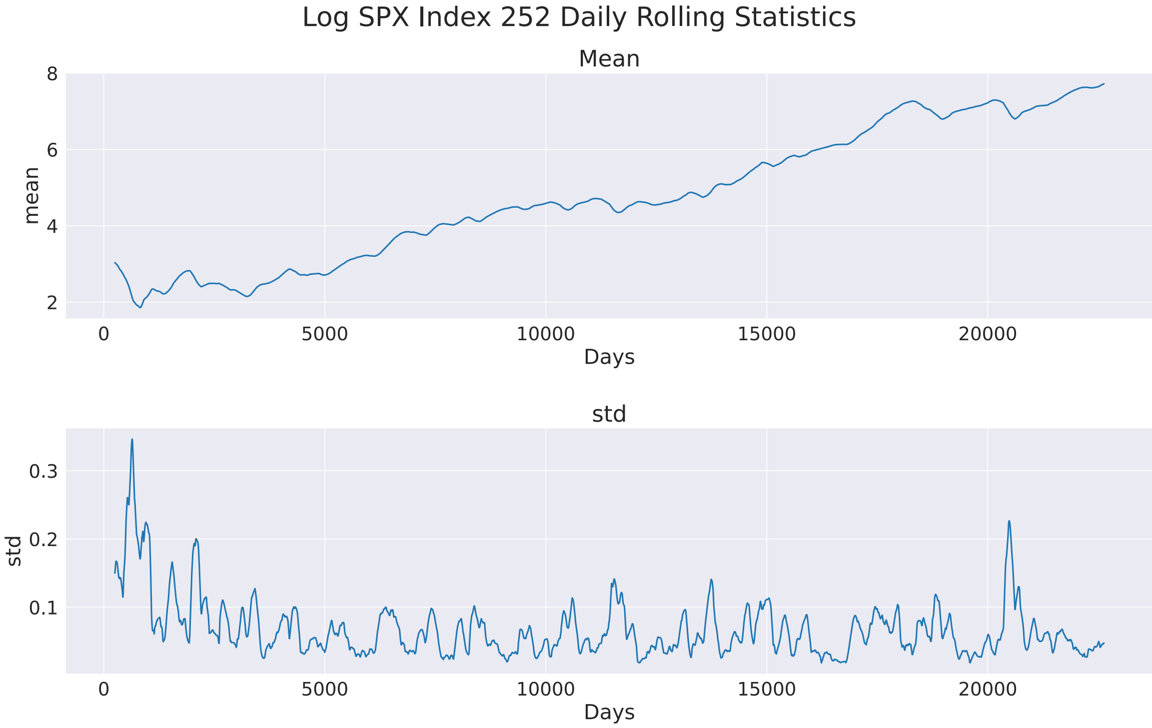


Figure 2: Daily Log SPX Index Rolling Statistics

1.1.3

Figure 3 shows the simple and log returns of the SPX Index. It can be seen that they are very similar.

Figure 4 and Figure 5 show the rolling statistics for the log and simple returns, respectively. Their statistics are also very similar, and using the definition of stationarity from section 1.1.2, they both exhibit some degree of stationarity as their mean and std are constant (independent of time). It is noted that the std of both the log and simple returns show some level of decreasing trend through time, indicating slight non-stationarity.

Figure 3: Daily SPX Index Log and Simple Returns

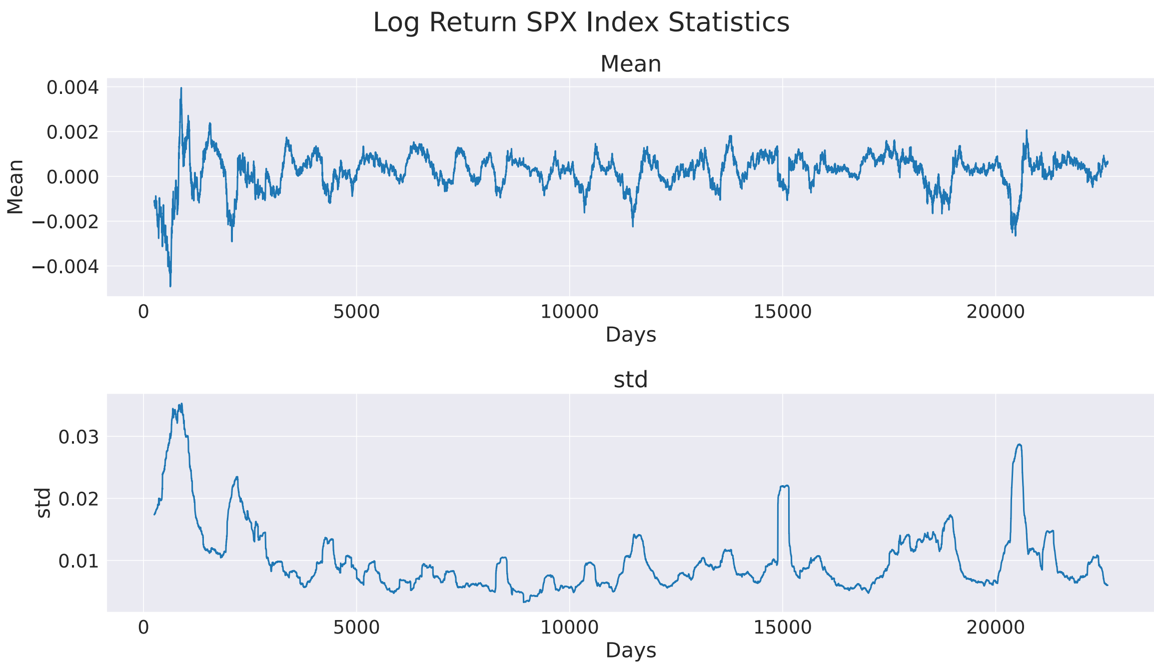


Figure 4: Log Returns Rolling Statistics

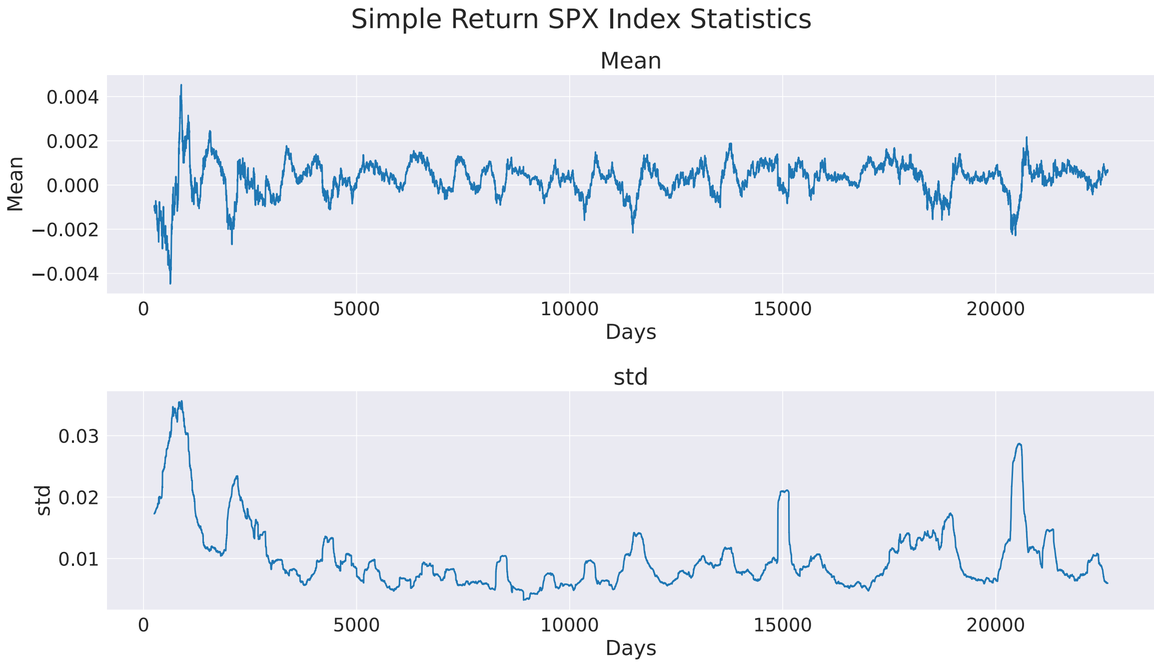


Figure 5: Simple Returns Rolling Statistic

1.1.4

There are several benefits for using log returns over simple returns, such as:

- Normally (Gaussian) distributed over short time periods: Prices can be approximated to be log-normal as the log function suppresses extreme values. Thus, if prices are log-normal, then log-returns can be considered normally distributed.
- Time additivity: The logarithmic return of an asset between a period of t to T is the sum of the log returns between t and T.
- Mathematical tractability and numerical stability: log-returns are more stable and tractable at very low values, which can occur for small price changes over short periods.

The Jarque-Bera test is a statistical test for Gaussianity of the data that compares the sample skewness and kurtosis to the skewness and kurtosis of a normal distribution. Skewness measures the degree of asymmetry of the data, while kurtosis measures the degree of peakedness or flatness of the data relative to a normal distribution.

A p-value of $0 < p < 0.5$ indicates that the data is likely non-Gaussian and so reject the null hypothesis. As expected, the p-value returned for the simple and log returns is 0.0, indicating that neither series has Gaussian properties. This is because log returns are only normally distributed over short time periods, however this series has over 20,000 entries and so does not satisfy this condition.

1.1.5

Log returns: $\mathrm{r[t]} = \log(\frac{p[t]}{p[t-1]}) + \log(\frac{p[t-1]}{p[t-2]}) = \log(\frac{1}{2}) + \log(\frac{2}{1}) = 0$
Simple returns: $\mathrm{R[t]} = \frac{p[t]-p[t-1]}{p[t-1]} + \frac{p[t-1]-p[t-2]}{p[t-2]} = \frac{1-2}{2} + \frac{2-1}{1} = 0.5$
Therefore, log returns have time additivity whereas simple returns do not because the log return over n periods is the difference in log between initial and final periods. This is not the case for simple returns as shown.

1.1.6

Log returns should not be used when:

- Using long time scales. A positive skew is assumed in log-normal distributions, but most financial data is negatively skewed in long time-scales and so the log-normality assumption is often violated.
- Doing addition along assets as log returns are not asset-additive, whereas simple returns are. The weighted average of log returns of individual stocks is not equal to the portfolio return because log returns are not a linear function of asset weights.

1.2 ARMA vs ARIMA Models for Financial Applications

1.2.1

Figure 6 shows the Log S&P 500 Daily Closing Price.
To analyse the appropriateness of an ARMA or ARIMA model, the stationarity of the time series is analysed. This is because an ARMA model requires the data to be stationary, whereas an ARIMA model can non-stationary data into the model using the Integrated (I) term. Figure 7 shows the rolling mean and std. It can be seen that the series is not stationary due to the clear upward trend in the mean.
Therefore, an ARIMA model is more appropriate.

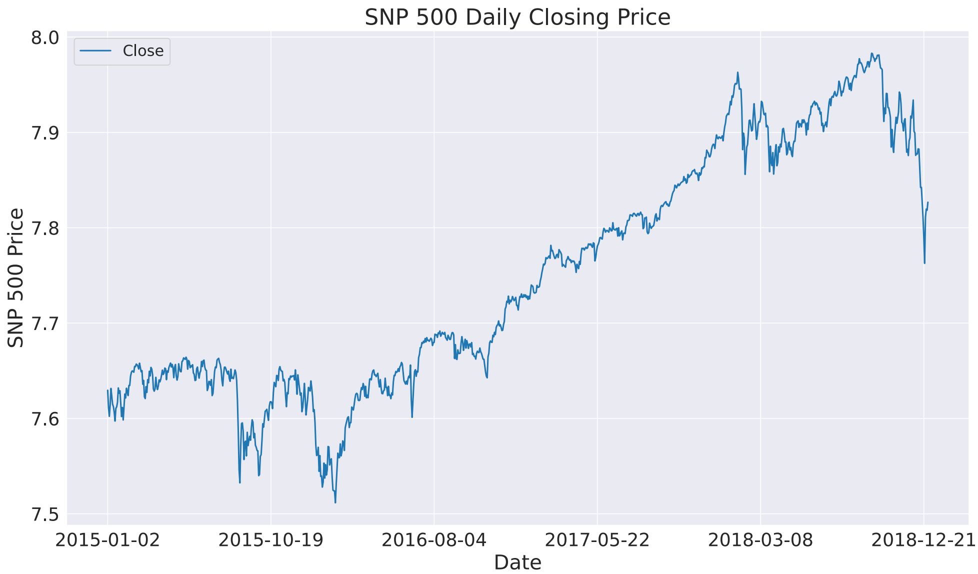


Figure 6: Log S&P 500 time-series

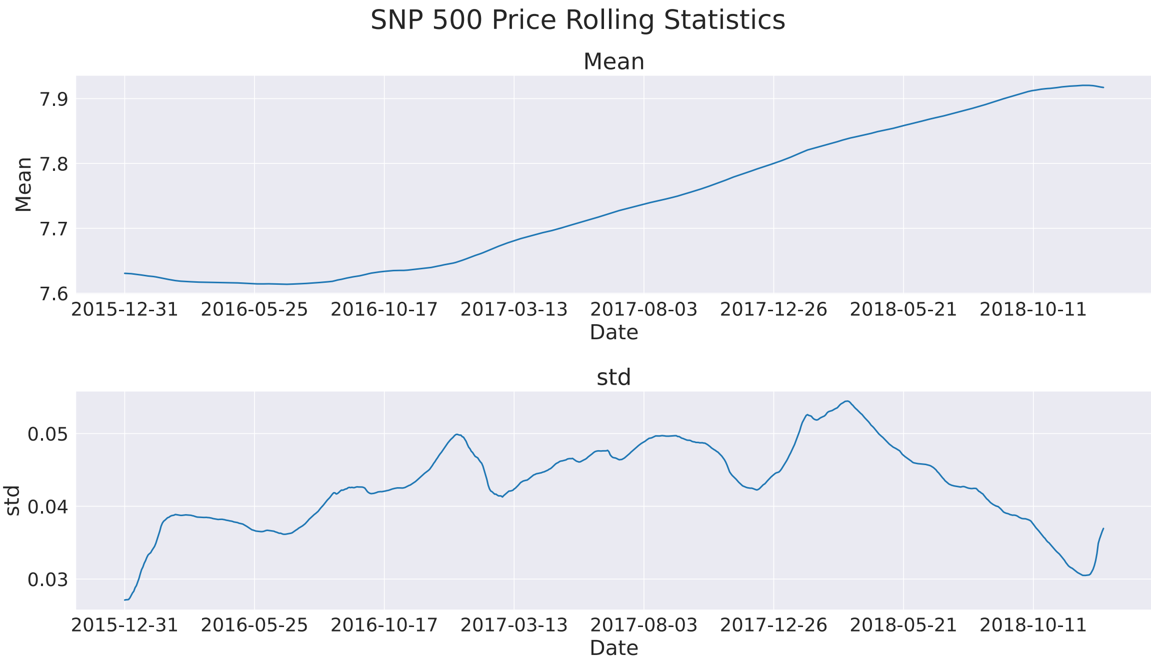


Figure 7: Log S&P 500 Rolling Statistics

1.2.2

Figure 8 shows the prediction of an ARMA(1,0) model against the true values. The ARMA(1,0) model can be mathematically formulated as:

$$x[t] = a_1 x[t-1] + \mu[t]$$

Analysing the model parameters on the data gives:

$$a_1 = 0.997$$

$$\mu[t] = 7.74$$

The parameters show that the ARMA(1,0) model has learnt the mean of the SNP500 index as the constant (7.8). However, this is not useful as the data is non-stationary, so this mean is not a good fit.

Secondly, the 0.997 ar_1 value shows that the model places a high positive weight on the value of the time series at time lag = 1. This is because the data is non-stationary, and so the model has essentially learnt the overall upwards trend that can be seen in the data. Therefore, this is also not very useful in practice.

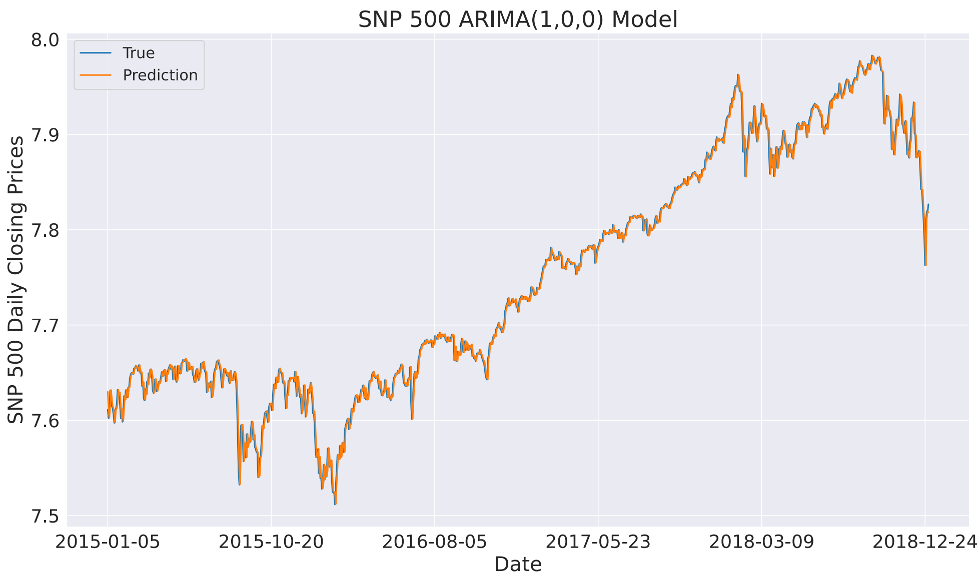


Figure 8: S&P500 Daily Closing ARMA(1,0,0) Model

1.2.3

Figure 9 shows the prediction of an ARIMA(1,1,0) model against the true values. Analysing the model parameters using the same method as in section 1.2.2 gives:

$$a_1 = -0.0088$$

$$\mu = 0.000196$$

These model parameter values are very different to the ARIMA(1,0,0) model. This result is more meaningful as the differencing of 1 (intrgrated = 1 in ARIMA) has made the data approximately stationary. Therefore, the analysis of the model coefficients now shows that there is < 0.01 weighting on the first time lag. This shows that there is little correlations between time steps. This is expected and shows that the S&P 500 Daily Closing Price is likely influenced by other factors, other than the previous time lag.
In addition, the data has effectively been zero-meaned as the constant parameter 0.

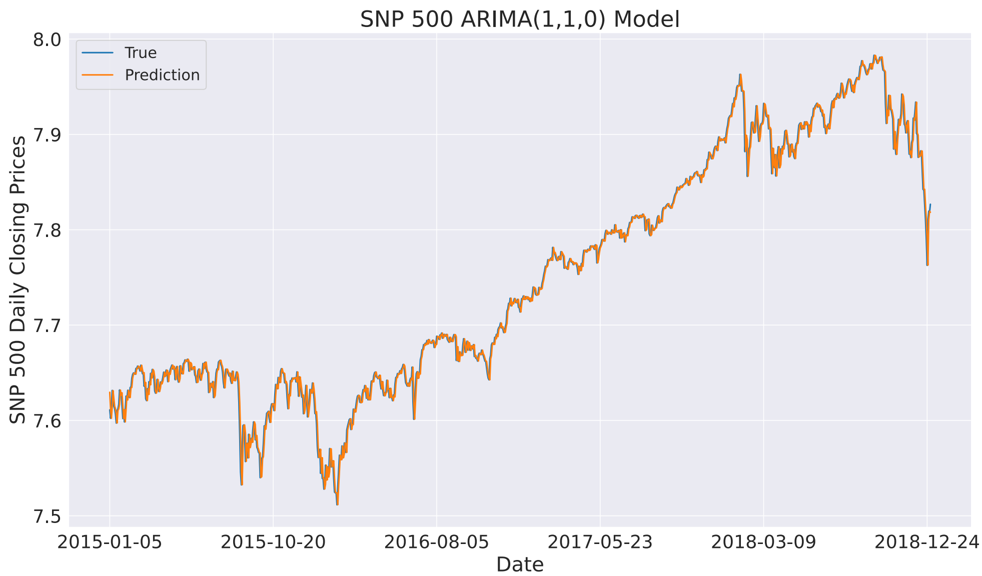


Figure 9: S&P500 Daily Closing ARMA(1,1,0) Model

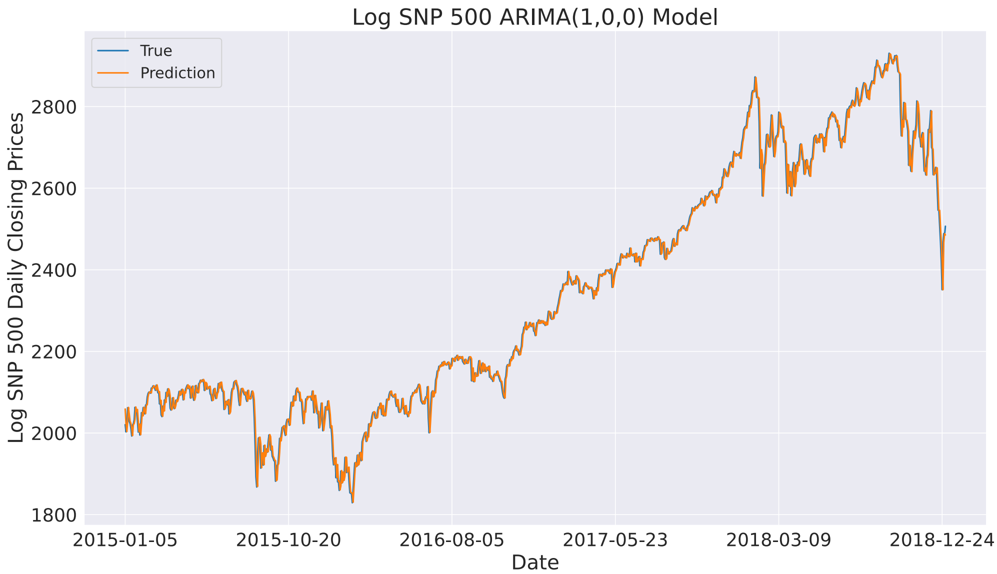
1.2.4

Log-pricing with a differencing term of 1 (integrated = 1 in ARIMA) will create log-returns. Thus, the time-series will now have the properties associated with log-returned data, for example time additivity and log-normality. This may make it easier to model the time-series.

	Log ARIMA(1,0,0)	Log ARIMA(1,1,0)
a_1	0.997	-0.0068
μ	2312	0.447

Table 1: Log-pricing ARIMA model parameters

To evaluate the necessity of taking a log of the prices for ARIMA analysis, the ARIMA(1,0,0) and ARIMA(1,1,0) models were applied to the log-transformed series. Figure 10 shows the prediction and true values. It is evident that they are very similar to the original series.
The model parameters for the ARIMA models are summarised in Table 1.
These parameter values are very similar to those from the non-log transformed data. Thus, it is not necessary to take log of prices for ARIMA analysis.



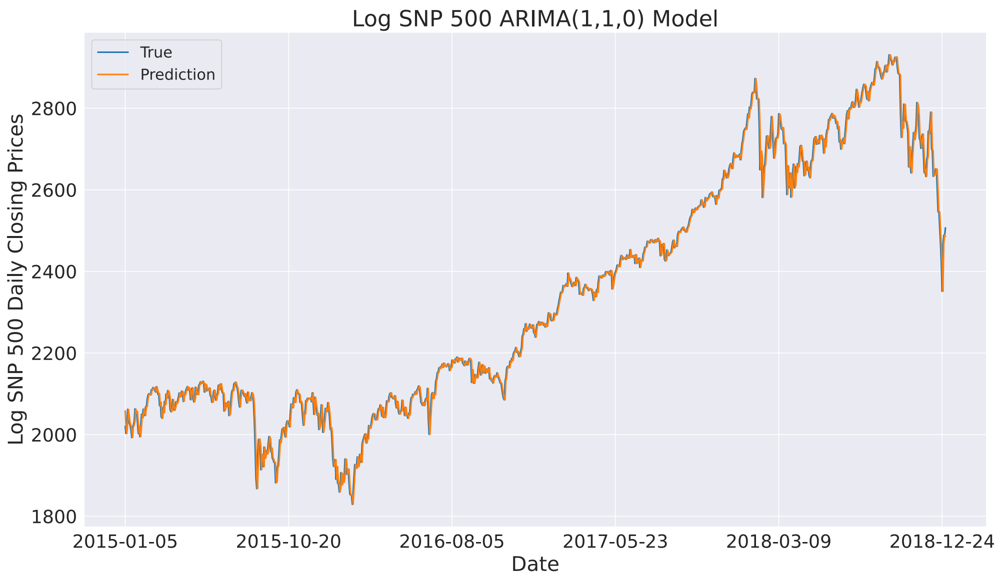


Figure 10: Log S&P500 Daily Closing ARMA Models

1.3 Vector Autoregressive (VAR) Models

1.3.1

Let $\boldsymbol{Y} \in \mathbb{R}^{KxT}$ be a KxT matrix where each column represents the vector of K variables at time t, therefore:

$$Y = \begin{bmatrix} y_{1,t} & \cdots & y_{1,T} \\ \vdots & \ddots & \vdots \\ y_{K,t} & \cdots & y_{K,T} \end{bmatrix}$$

Let $\boldsymbol{B} \in \mathbb{R}^{Kx(KP+1)}$ be a $Kx(KP+1)$ matrix where the first column contains the intercept $\boldsymbol{c}$ and the remaining columns contain the coefficients of the lagged values of $\boldsymbol{Y}$, therefore:

$$B = \begin{bmatrix} c_1 & A_1^1 & \dots & A_1^p \\ \vdots & \vdots & \ddots & \vdots \\ c_k & A_k^1 & \dots & A_k^p \end{bmatrix}$$

Let $\boldsymbol{Z} \in \mathbb{R}^{(KP+1)xT}$ be a $(KP+1)xT$ matrix where the first row contains 1's (for the intercept term c) and the next K rows contain the lagged values of $\boldsymbol{y}$ up to p lags. Therefore,

$$Z = \begin{bmatrix} 1 & 1\dots & 1 \\ y_{1,t-1} & \cdots & y_{1,t-p} \\ \vdots & \ddots & \vdots \\ y_{k,t-1} & \cdots & y_{k,t-p} \end{bmatrix}$$

Let $\boldsymbol{U} \in \mathbb{R}^{KxT}$ be a KxT matrix for the error terms $\boldsymbol{e_t}$, therefore:

$$U = \begin{bmatrix} e_{1,t} & \dots & e_{1,T} \\ \vdots & \ddots & \vdots \\ e_{K,t} & \dots & e_{K,T} \end{bmatrix}$$

Then in concise matrix form:

$$\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{Z} + \boldsymbol{U}$$

1.3.2

This is a Least Squares solution with squared loss function:

$$\boldsymbol{L}(\boldsymbol{B}) = \frac{1}{2}||\boldsymbol{Y} - \boldsymbol{B}\boldsymbol{Z}||_2^2$$

Differentiating w.r.t $\boldsymbol{B}$:

$$(\boldsymbol{Y} - \boldsymbol{B}\boldsymbol{Z}) - \boldsymbol{Z}^{\boldsymbol{T}}$$

Setting $=0$ and solving for $\boldsymbol{B_{opt}}$:

$$\boldsymbol{Y}\boldsymbol{Z}^{\boldsymbol{T}} = \boldsymbol{B}\boldsymbol{Z}\boldsymbol{Z}^{\boldsymbol{T}}$$

$$\boldsymbol{B_{opt}} = \boldsymbol{Z}^{\boldsymbol{T}}\boldsymbol{Y}(\boldsymbol{Z}^{\boldsymbol{T}}\boldsymbol{Z})^{-1}$$

1.3.3

It is possible to rewrite the equations:

$$\boldsymbol{y_t} = \boldsymbol{A}\boldsymbol{y_{t-1}} + \boldsymbol{e_t} \tag{1}$$

and

$$\boldsymbol{y_{t-1}} = \boldsymbol{A}\boldsymbol{y}_{t-2} + \boldsymbol{e_{t-1}} \tag{2}$$

into the following equation:

$$\boldsymbol{y_t} = \boldsymbol{A}^{\boldsymbol{t}}\boldsymbol{y_0} + \boldsymbol{A}\Sigma_{i=0}^{t-1}\boldsymbol{e_i} + \boldsymbol{e_t} \tag{3}$$

Therefore, it can be seen from this form that the matrix A determines how much past values of the time series will affect the current value. This can be shown using eigen decomposition:

$$\boldsymbol{A}^{\boldsymbol{t}}\boldsymbol{v} = \boldsymbol{\lambda}^{\boldsymbol{t}}\boldsymbol{v} \tag{4}$$

From this, if an eigenvalue is less than 1 in absolute value, it means that the corresponding eigenvector changes at a decreasing rate over time, which leads to a stable and stationary time series. On the other hand, if an eigenvalue is greater than 1 in absolute value, it means that the corresponding eigenvector changes at an increasing rate over time, leading to an unstable and non-stationary time series which will eventually diverge to $+\infty$ as $t \to \infty$.

Mathematically,

$$\text{as t} \to \infty, \boldsymbol{y_t} \to \begin{cases} \infty, & \text{if } |\boldsymbol{\lambda}| \geq \mathbf{1} \\ \boldsymbol{A}\Sigma_{i=0}^{t-1}\boldsymbol{e_i} + \boldsymbol{e_t}, & \text{otherwise} \end{cases}$$

1.3.4

The appropriateness of using these stocks can be analysed in a variety of ways. When analysing the eigenvalues of the matrix $\boldsymbol{A}$ as defined in section 1.3.3, it is required that the eigenvalues all be less than 1 in absolute value. However, one of the eigenvalues is greater than 1 (1.006), thus it is not appropriate to use these stocks when looking at the eigenvalues due to the instability of the VAR model over longer time periods.

1.3.5

A similar analysis using the same methods as in section 1.3.4 can be done. When analysing the eigenvalues of the matrix $\boldsymbol{A}$, all are less than 1 in absolute terms. This means the VAR(1) process will be stable. Thus, performance factors can be analysed more critically.

In general, sector-grouping stocks in VAR models can help identify the relationships between different sectors and provide insights into how they may perform in different economic conditions. This can also provide diversification, a method of reducing risk, as different sectors may perform differently in different economic conditions

Section 2: Bond Pricing

2.1 Examples of Bond Pricing

2.1.1

$$r_1 = 1100$$

$$r_0 = 1000$$

Annually:

$$1 + r_f = \frac{1100}{1000}$$

$$r_f = \frac{1100}{1000} - 1 = 0.1 = 10\%$$

Semi-annually:

$$1100 = 1000 * (1 + \frac{r_f}{2})^2$$

$$\sqrt{1.1} = 1 + \frac{r_f}{2}$$

$$r_f = (\sqrt{1.1} - 1) * 2 = 0.0976 = 9.76\%$$

Monthly:

$$1100 = 1000 * (1 + \frac{r_f}{12})^{12}$$

$$1.1^{1/12} = 1 + \frac{r_f}{12}$$

$$r_f = (1.1^{1/12} - 1) * 12 = 0.0957 = 9.57\%$$

Continuous:

$$1100 = 1000 * e^{r_f}$$

$$r_f = ln(1.1) = 0.0953 = 9.53\%$$

2.1.2

Let $r_0 = 1000$

$$r_1 = 1000 * (1 + \frac{0.15}{12})^{12} = 1160.75$$

Therefore,

$$r_f = ln(\frac{r_1}{1000}) = 0.149 = 14.9\%$$

2.1.3

Let $r_0 = 10000$

$$r_f = ln((1 + \frac{0.12}{4})^4) = 0.122 = 12.2\%$$

. Therefore, year interest to be paid = $0.122 * 10,000 = 1,220$.
ANSWER HERE.

2.2 Forward Rates

2.2.1

a)
Forward rates are calculated to determine future values.
If investing for one year, $r_0 = 100$

$$r_1 = 100 * 1.05 = 105$$

$$r_2 = 105 * 1.05 = 110.25$$

Where the investor has re-invested his one year return, assuming the same one-year interest rate. However, $110.25 < 114.49$, so it is better to invest for 2 years rather than one.

On the other hand, the opportunity cost of keeping the $100 locked in may not be suitable for an investor, in case the economic conditions change between year 1 and year 2, e.g. a more favourable spot rate after year 1.

b)
The 5% investment strategy is the most flexible as the investor has the choice to re-invest his earnings after one year, or choose a different asset.

The 7% strategy is more optimal given that the investor does not require the invested $100 until 2 years, e.g. through arbitrate opportunities, i.e. the market is efficient. Arbitrage should be exploited if found and so the investor would want all the cash available to profit maximally.

The extra return strategy of 9% relies on the seller's prediction of market interest rates in one years time to be accurate. However, economic shocks or new innovations (assuming Adaptive Market Hypothesis) may make these predictions in-accurate, and so the extra return strategy may lead to even better gains, or worse returns than the 5% strategy.

c)

Advantages: - If the forward rate rate of interest of 9% is accurate, this will lead to a more optimal allocation of resources (cash) by the investor now. - The investor (and seller) can be more confident in their returns, as the agreement ensures a fixed return after 2 years regardless of market conditions later. This reduces risk (variance).

Disadvantages: - It is almost impossible to predict forward rates of interest with absolute certainty, and so there will be a non-negligible probability (following a probability distribution) that the forward rate of 9% will be sub-optimal after the 2nd year. For example, economic shocks or new innovations may create arbitrage opportunities after year 1, which the investor would want all the resources (i.e. the invested $100 with one-year interest rate) to invest with.

d) Let r_i be the future rate of interest, r_j be the current rate of interest, $f_{i,j}$ be the forward rate of interest from time i to time j. Then,

$$(1 + r_i) = (1 + r_j)(1 + f_{j,i}) \tag{5}$$

In this example, $i = 2$, $j = 1$, $r_i = 0.07$, $r_j = 0.05$, $f_{i,j} = 0.09$.

2.3 Duration of a coupon-bearing bond

2.3.1

a)

Duration =

$$\frac{1*9.52}{768.55}+\frac{2*9.07}{768.55}+\frac{3*8.64}{768.55}+\frac{4*8.23}{768.55}+\frac{5*7.84}{768.55}+\frac{6*7.46}{768.55}+\frac{7*717.79}{768.55}$$

$$= 6.76 \text{ years}$$

b)

$$\text{Modified duration} = \frac{duration}{1+yield} = \frac{6.76}{1.05} = 6.44$$

The modified duration will always be smaller than duration for a positive yield.

The modified duration is an approximation to $-\frac{dP}{dy}$, whereas duration is the weighted average of times to each of the cash payments.

c)

Duration and modified duration is a measure of the sensitivity of a bond's price to changes in interest rates. It is therefore a measure of risk (through measures of variance/volatility) when looking at bonds.

By using duration or modified duration to assess the potential impact of changes in interest rates on its investments, a pension plan can better understand the risks associated with those investments and make informed decisions about how to manage those risks. For example, a pension plan might choose to invest in bonds with shorter duration's or modified duration's in order to reduce its exposure to changes in interest rates, or it might choose to invest in a diversified portfolio of bonds with a range of duration's or modified duration's in order to spread out the risk of changes in interest rates.

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

2.4.1

The market returns per day of 157 European companies are shown in Figure 11. The figure shows seemingly random fluctuations around 0. The returns are all within ± 0.03 and so are small. These characteristics suggests an overall return (at t = T) close to 0, which is confirmed by summing the daily market returns of 0.032.

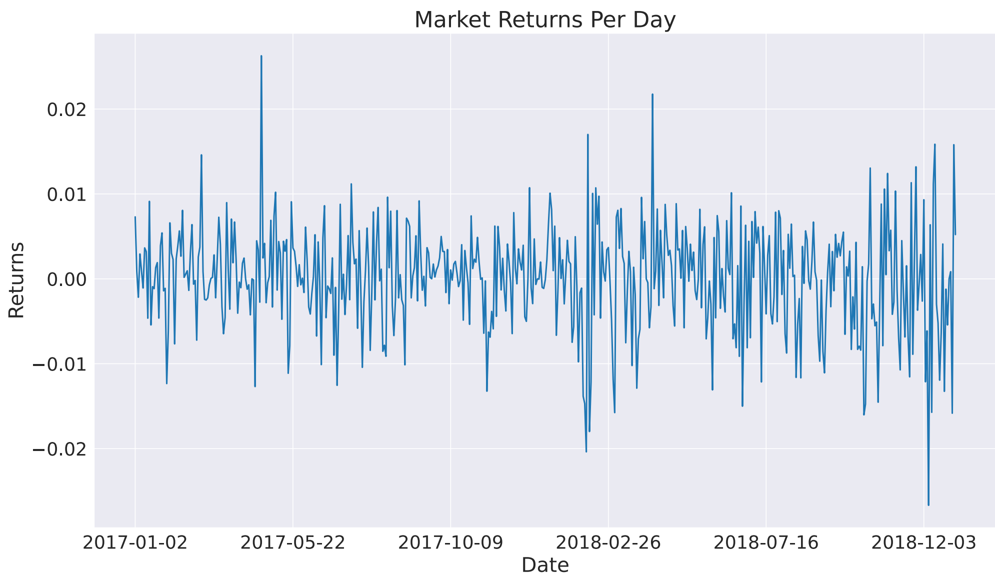


Figure 11: Market Returns Per Day

2.4.2

The beta's, $\beta_{i,t}$ in a CAPM model is the sensitivity of the expected excess asset returns to the expected excess market returns. In other words,

$$\beta_{i,t} = \frac{Cov(R_{i,t}R_{m,t})}{Var(R_{m,t})} \tag{6}$$

The mean beta value is 1.0. Thus, a beta value higher than 1.0 indicates that that beta is more volatile and therefore risky when compared to the market, and vice versa. Beta values can therefore be used to inform portfolio decisions. Figure 12 shows the 22 day rolling windows of 157 European companies. Most companies have rolling betas of ±2.5 and variances less than 2.5.

Across the date range, only a few companies have rolling betas higher than 5. These companies represent the most volatile, or risky, companies in the dataset.

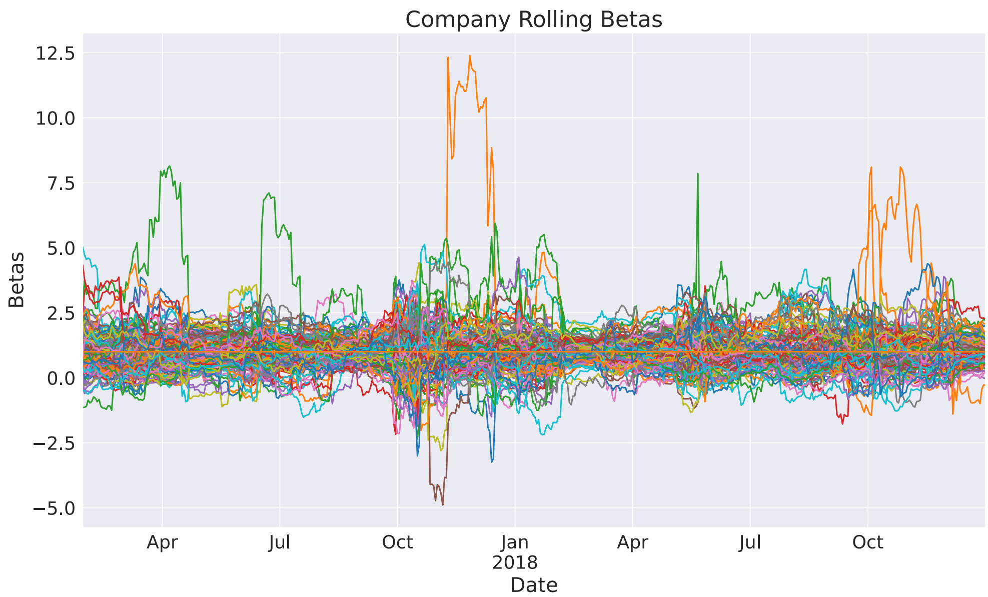


Figure 12: Company Rolling betas

2.4.3

The cap-weighted market return is based on the market cap of the 157 European companies. Larger companies' (by market cap) returns would therefore represent a greater portion of the market returns.

Figure 13 shows the cap-weighted market returns. The total cap-weighted market returns is 0.098, which is greater than the equally-weighted market returns of 0.03. This indicates that the larger companies have higher overall returns than smaller ones.

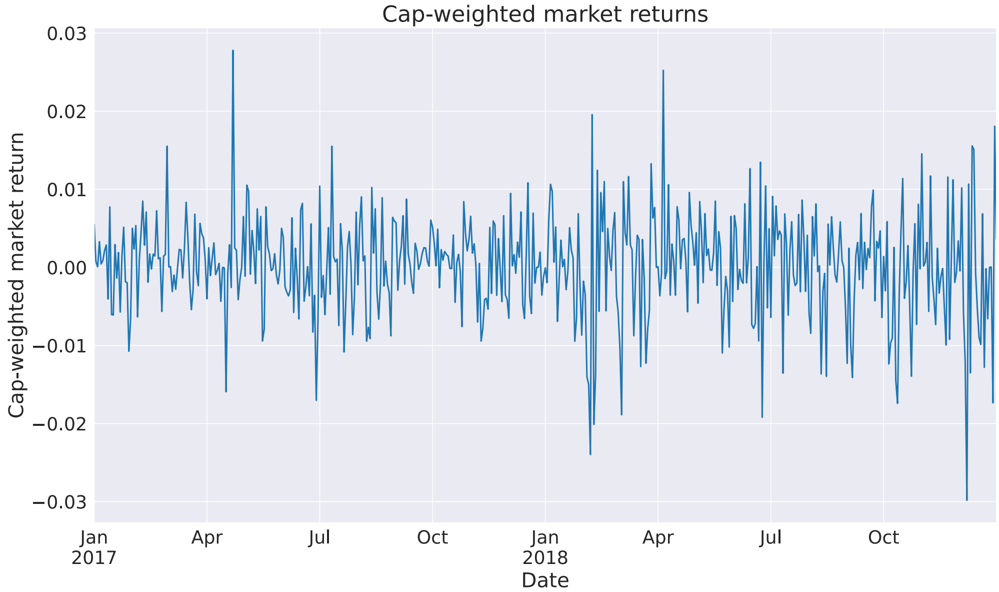


Figure 13: Cap-weighted market returns

2.4.4

Figure 14 shows the cap-weighted 22 day rolling betas. When compared to the equally weighted betas, there are some similarities and differences.

The difference between the two betas lies in the weighting scheme used. In the equally-weighted beta, each stock is given an equal weight, regardless of its market capitalization. This means that the beta of a smaller stock would have the same impact on the overall beta of the portfolio as a larger stock. On the other hand, in the cap-weighted beta, each stock is weighted by its market capitalization, which means that the beta of a larger stock would have a greater impact on the overall beta of the portfolio than a smaller stock. This has impacted the results as shown in the plots. For example, the highest beta seen in the equally-weighted rolling betas was 12.5, but in the cap-weighted betas this was 10. Thus, this company's beta has been weighted down corresponding to its proportion of market cap compared to the other companies.

However, the overall patterns of the beta's are relatively similar, as both beta's are affected by the same macroeconomic factors that affect the overall stock market and are both still measures of the systematic risk of a portfolio of stocks.

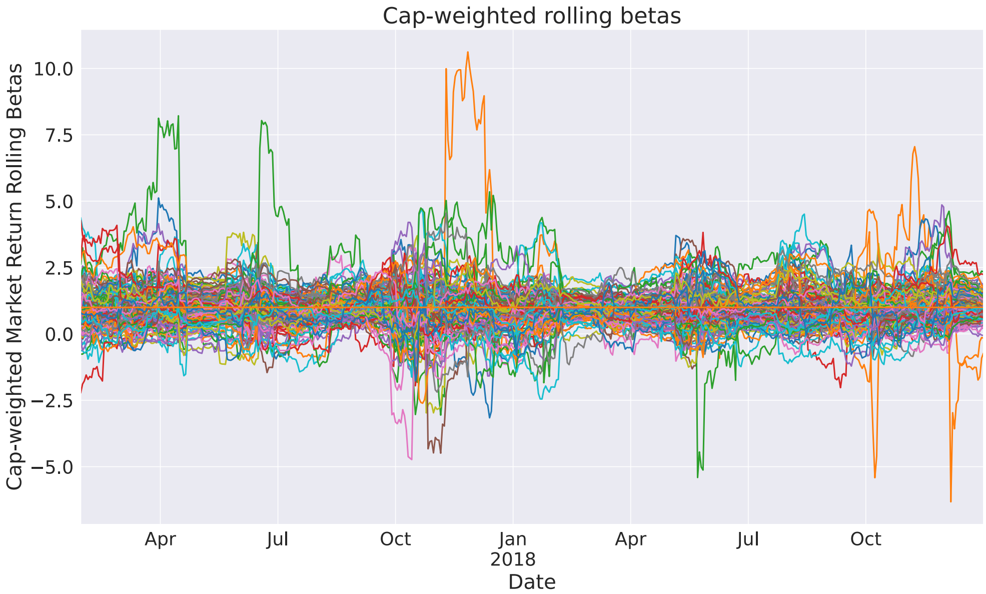


Figure 14: Cap-Weighted market return rolling betas

2.4.5a

Equation 7 shows the re-formulated APT model for regression analysis. $\boldsymbol{\theta}^*$ consists of the optimal a, R_m and R_s for regression. ϵ_i is also known as the residual of the model as it is the random noise not captured in the regression analysis.

$$\begin{gathered} r_i = a + b_m R_m + b_{s_i} R_s + \epsilon_i \\ \text{Let } \boldsymbol{\theta} = [a, R_m, R_s] \\ \text{Let } \boldsymbol{A} = [1, b_{m_i}, b_{s_i}] \\ \text{Then } \boldsymbol{\theta}^* = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{r_i} \\ \text{And } \boldsymbol{\epsilon_i} = r_i - \boldsymbol{\theta}^{*T} \boldsymbol{A} \end{gathered} \tag{7}$$

Figures 15, 16 and 17 show the resulting plots and PDF's from the regression.

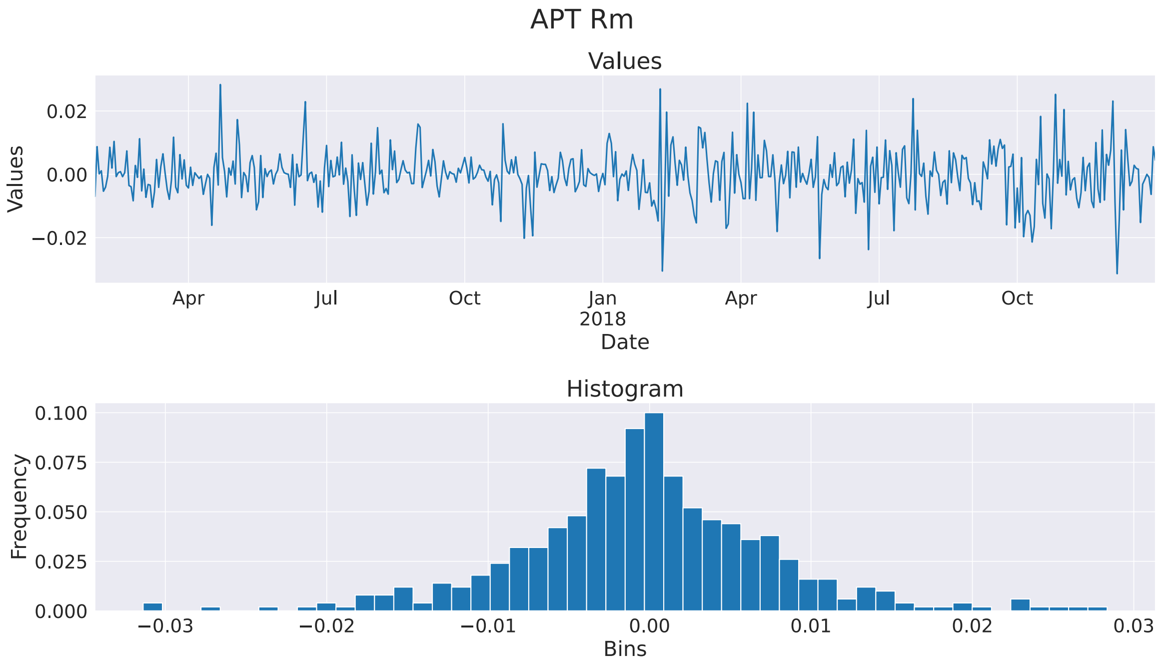


Figure 15: APT R_m Estimate Graph and Histogram Plots

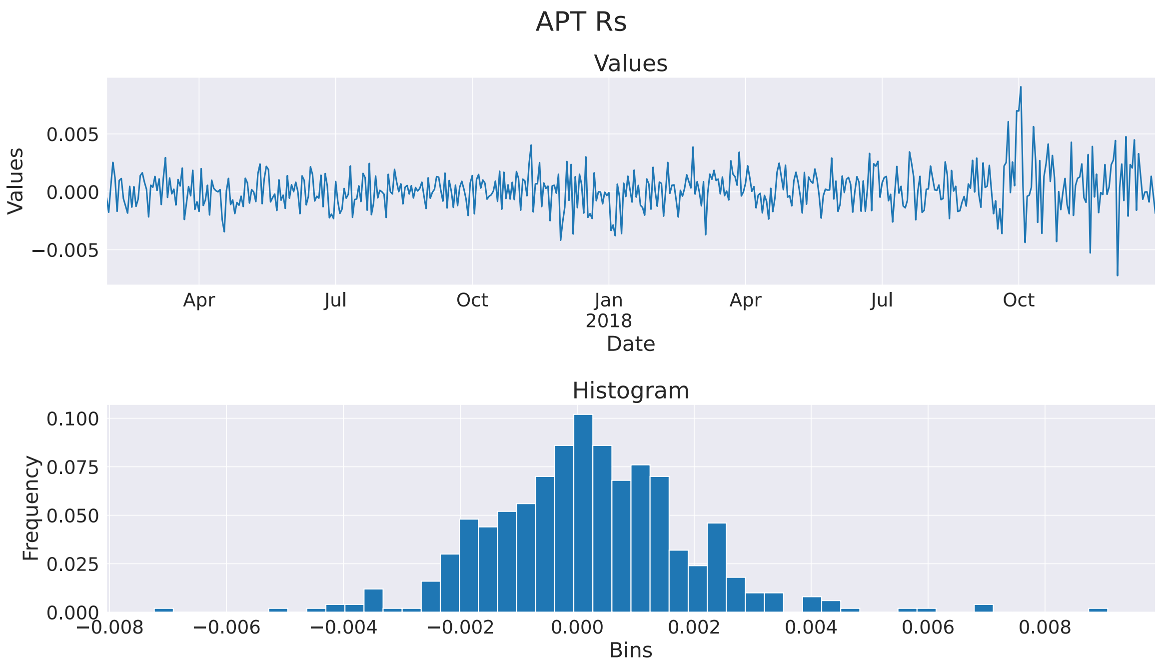


Figure 16: APT R_s Estimate Graph and Histogram Plots

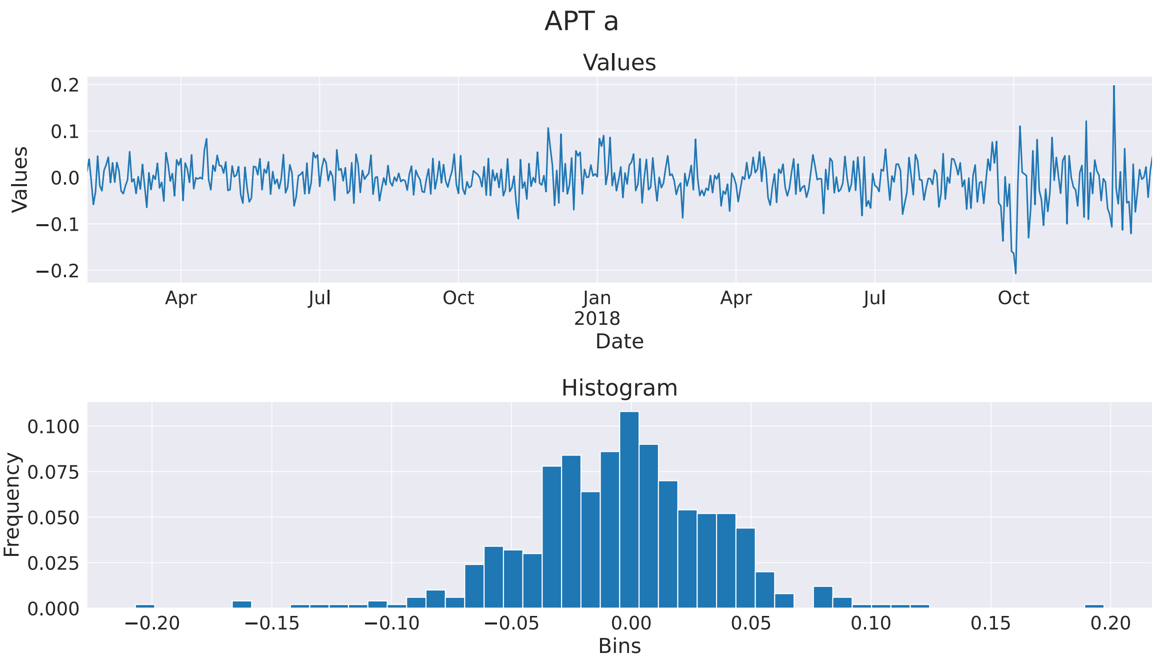


Figure 17: APT a Estimate Graph and Histogram Plots

2.4.5b

The magnitude and variance of $a, R_m, and R_s$ can be analysed from the Figures in Section and is summarised in Table 2. In general, the magnitude and variances of the three APT model parameters are close to 0, all being < $|0.005|$. R_s has the lowest magnitude and variance while a has the largest magnitude by approximately a factor of 10x. a also has the highest variance as a result.

Parameter	Magnitude	Variance
a	-0.004	0.0017
R_m	-0.00029	$6.26e^{-05}$
R_s	0.00019	$3e^{-06}$

Table 2: Parameters Magnitude and Variance

2.4.5c

The specific return, ϵ_i is calculated using the formula from Equation 7 in Section 2.4.5.a. Figure 18 shows the specific return for each company on each day. At any given day, there are less than 3 companies with a specific return greater than —0.1—. The highest specific return was around 0.43.

The correlation between the returns per company and specific return can then be calculated using the pearson coefficient. It should be noted that this method can only capture linear correlations. Figure 19 shows this correlation. The mean and variance are 0.8 and 0.1 respectively, indicating a high correlation through time.

A high correlation between returns and specific returns indicates that the 2 factors b_m and b_s are not able to explain the variance in returns per company using regression.

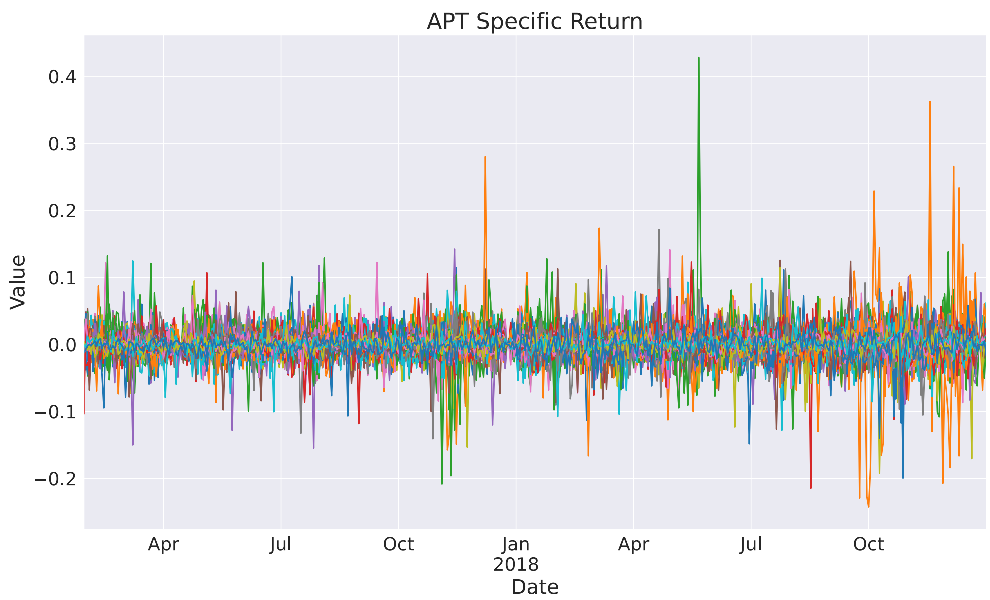


Figure 18: APT Specific Return ϵ_i

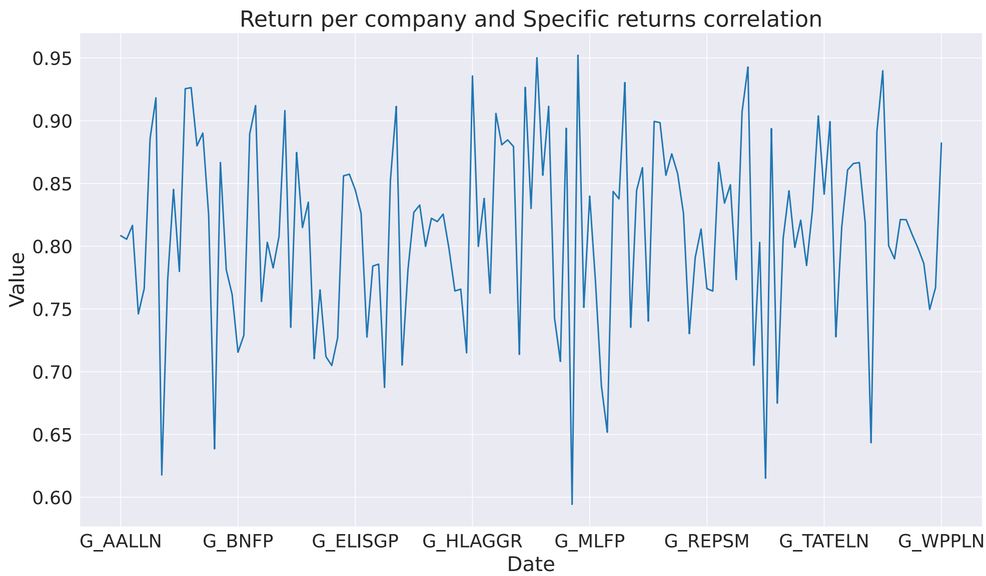


Figure 19: APT Return and Specific Return Correlation

2.4.5d

The covariance matrix of $\boldsymbol{R} = [\boldsymbol{R}_m^T, \boldsymbol{R}_s^T]$ is calculated using a rolling window of 22 days. The magnitudes of the covariance matrix are small in magnitude, with a covariance value of $-3e^{-6}$.

The covariance matrix is stable if it does not change significantly over time. Figure 20 shows the rolling co-variances and thus can be used to assess the stability. R_m fluctuates between 0 and 1.5e-4, while R_s is close to 0 at all times. This suggests the covariance matrix is stable.

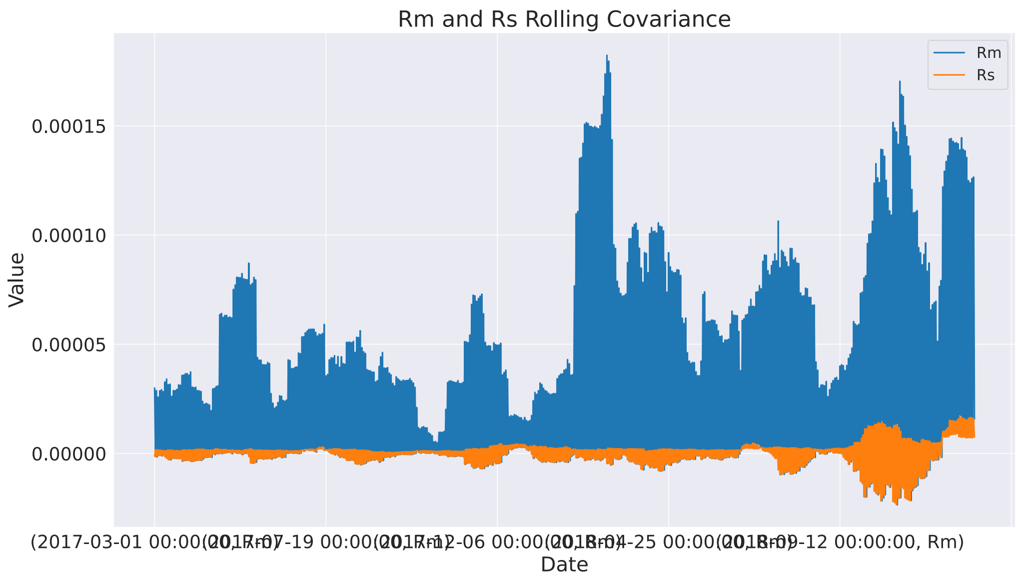


Figure 20: APT R_m and R_s Rolling Covariance

2.4.5e

Figure 21 shows the first 3 principal components from performing PCA on the specific returns covariance matrix. As shown, the first principal components accounts for 17% of the explained variance.

A higher percentage of variance explained by the first principal component suggests that there may be a systematic factor or set of factors that is not captured by the APT model. However, 17% is not a high percentage, suggesting that the unexplained variation or noise in the model may not have a strong systematic factor or set of factors that is not captured by the APT model.

Another possibility is that the specific returns covariance matrix follows a non-linear pattern that cannot be captured with PCA, as PCA is only a linear model.

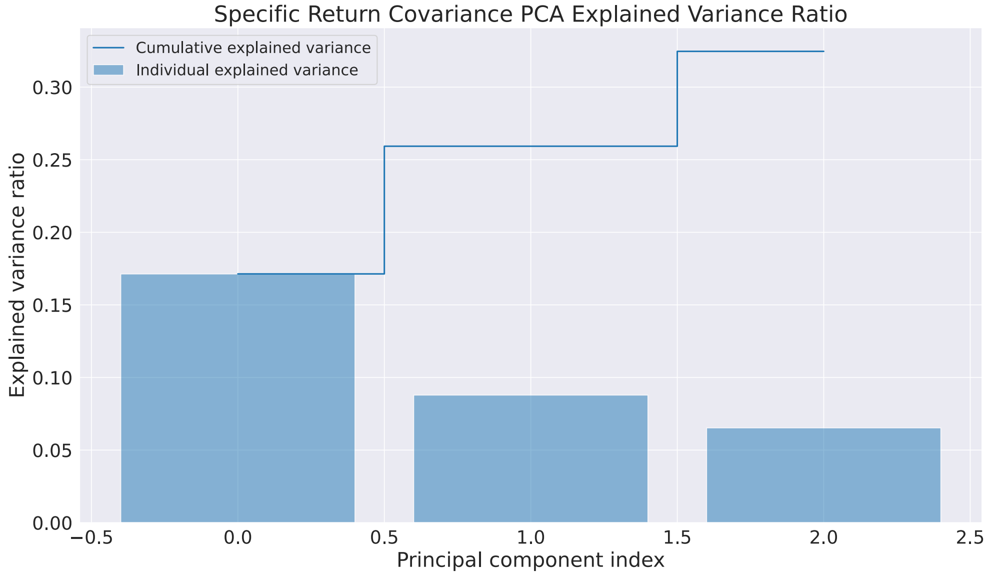


Figure 21: APT specific returns PCA Covariance

Section 3: Portfolio Optimization

3.1 Adaptive minimum-variance portfolio optimization

3.1.1

The Lagrangian optimisation formulation:

$$min_{w,\lambda} J'(w, \lambda, \boldsymbol{C}) = \frac{1}{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{C}w - \lambda(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{1} - 1)$$

To solve, we calculate partial derivatives and set them $= 0$ to find optimal solutions.

$$\frac{\partial J'}{\partial w} = \boldsymbol{w}^*\boldsymbol{C} - \lambda\boldsymbol{1} = 0$$

Therefore,

$$\boldsymbol{w}^*\boldsymbol{C} = \lambda\boldsymbol{1}$$

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

We can also calculate the optimal λ:

$$\frac{\partial J'}{\partial \lambda} = -\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{1} + 1 = 0$$

$$\boldsymbol{w}^{\boldsymbol{T}}\mathbf{1} = 1$$

Substituting optimal $\boldsymbol{w}$:

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

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

We can then substitute this result into the theoretical variance of the portfolio:

$$\hat{\sigma}^2 = \boldsymbol{w}^{*\boldsymbol{T}}\boldsymbol{C}\boldsymbol{w}^* = \lambda\boldsymbol{C}^{-\mathbf{1}}\boldsymbol{C}\lambda\boldsymbol{C}^{-\boldsymbol{T}} = \lambda^2\boldsymbol{C}^{-\boldsymbol{T}} = \frac{1}{\mathbf{1}^{\boldsymbol{T}}\boldsymbol{C}^{-\mathbf{1}}\mathbf{1}}$$

3.1.2

The comparison between the minimum-variance portfolio and equally weighted portfolio is shown in Tables 3 for the train set and Table 4 for the test set. As expected, the train set variance is lower for the minimum variance portfolio compared to the equally weighted, as this was the solved optimisation problem. In addition, the test set performance is worse than the train set. This is due to the portfolio optimisation being done on the train set and so this will have the best performance.

However, both stocks had negative cumulative returns by the end of the test set, with the equal weighted portfolio doing worse by 66%. Figure 22 shows the train and test set cumulative returns of the portfolio's.

The theoretical variances calculated using the formula from Section 3.1.1 were $2.86e^{-05}$ and $1.4e^{-05}$ for the train and test set respectively. The train set's theoretical and empirical variances are the same, while the test set has a higher empirical variance than the theoretical one. This indicates that the optimal portfolio has not been successful in creating the minimum variance portfolio when analysing test set performance.

Portfolio	Train Set Variance	Train Set Cumulative Returns
Minimum Variance	$2.85e^{-05}$	0.25
Equally Weighted	$3.7e^{-05}$	0.22

Table 3: Minimum Variance and Equal Weighted Portfolio Train Set Statistics

Portfolio	Test set variance	Test Set Cumulative Returns
Minimum Variance	$8.16e^{-05}$	-0.118
Equally Weighted	$7.903e^{-05}$	-0.125

Table 4: Minimum Variance and Equal Weighted Portfolio Test Set Statistics

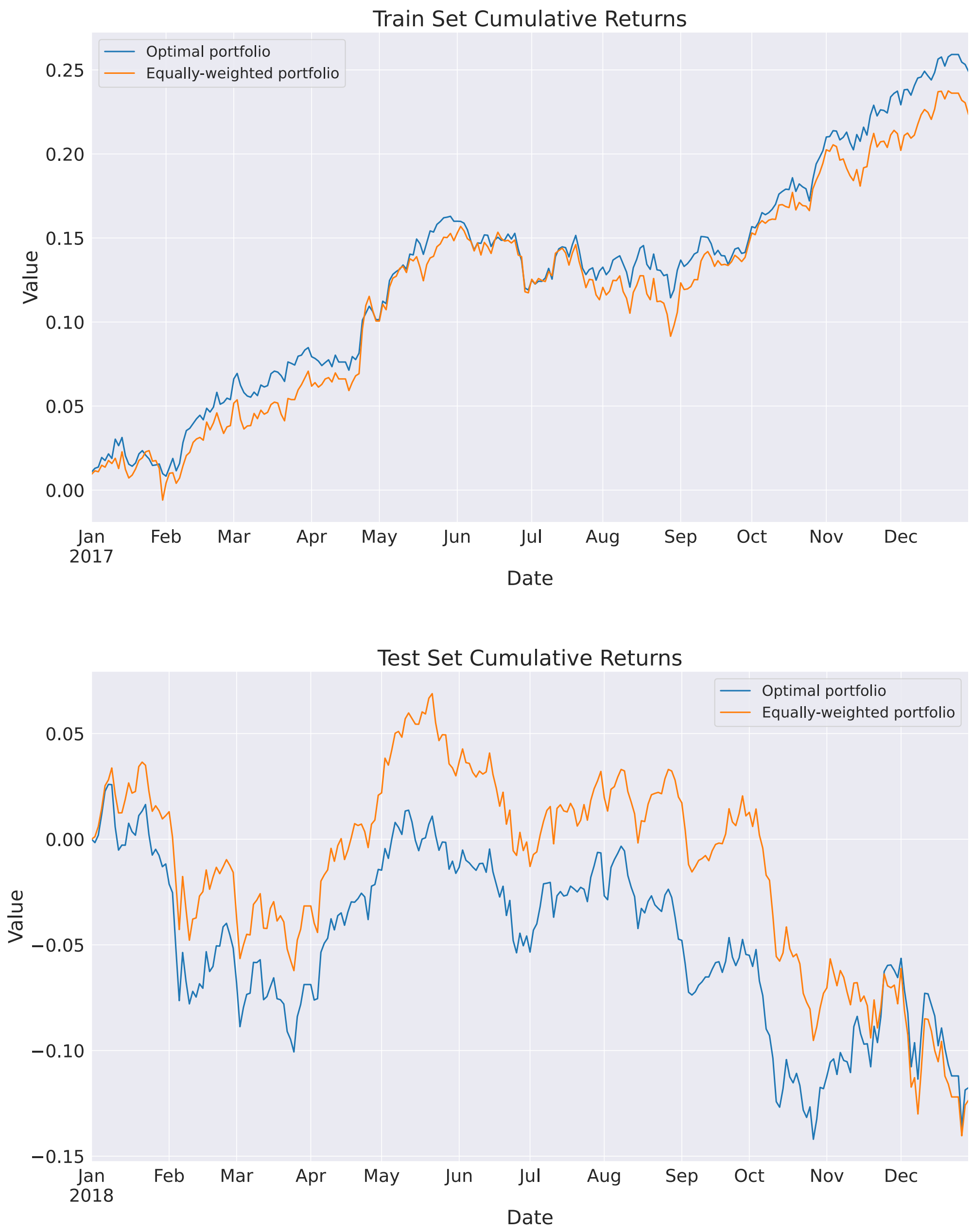


Figure 22: Minimum Variance and Equal Weighted Portfolio Cumulative Returns

3.1.3

The adaptive minimum variance scheme uses a rolling window of returns and covariance to calculate optimal portfolio weights, while the solution from Part 1 uses the full historical returns to calculate the minimum variance portfolio. The equally-weighted strategy assigns equal weights to all assets.

The effect of the recursive update of the variables involved in the adaptive minimum variance scheme is that the portfolio weights and covariance matrix are updated at each time step based on the most recent data, allowing the portfolio to adapt to changing market conditions. When assuming the Adaptive Market Hypothesis, this can result in a portfolio that is better suited to current market conditions compared to the minimum variance portfolio calculated using historical data as it can quickly adapt to new opportunities.

However, the adaptive minimum variance method may also be more sensitive to short-term fluctuations in the data, which could lead to more frequent portfolio re-balancing, which may not be desired (for example due to the cost of re-balancing).

When comparing test set performance as shown in Figure 23, the adaptive portfolio outperforms the method in Section 3.1.2. The final adaptive cumulative return was 0.07, compared to -0.13 for the equally weighted portfolio and -0.11 for the non-adaptive optimal portfolio. This indicates that updating based off the most recent data is beneficial to capturing the patterns in the return movements.

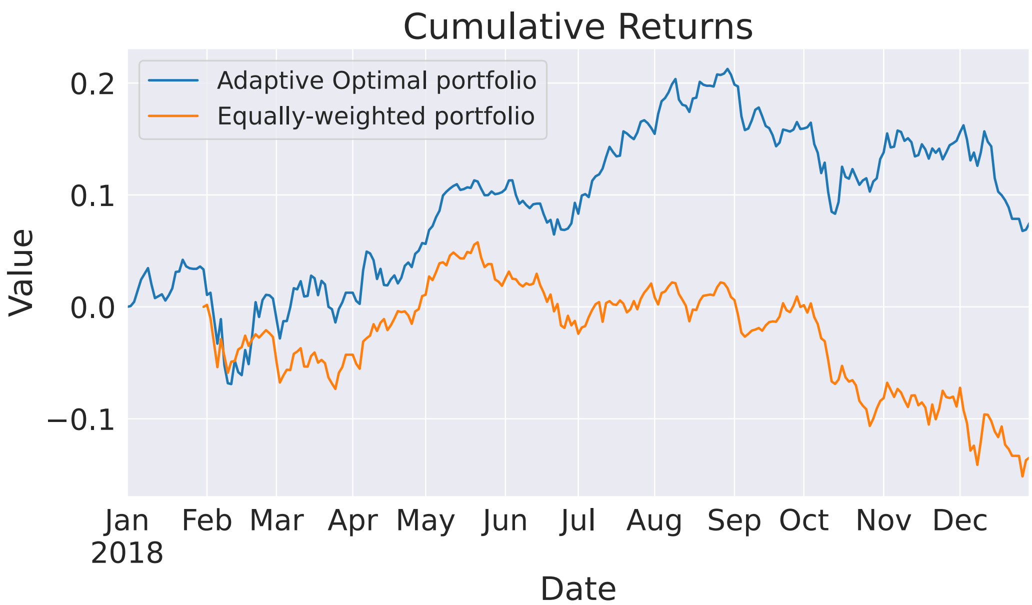


Figure 23: Adaptive Minimum Variance vs Equally Weighted Cumulative Returns

Section 4: Robust Statistics and Non Linear Methods

4.1: Data Import and Exploratory Data Analysis

4.1.1

The key descriptive statistics for the Apple, JPM, IBM and DJI stocks are shown in Figures 24, 25, 26 and 27 respectively. The descriptive statistics also include a ”ret” column, which consists of the 1-day returns for the respective stock.

	Open	High	Low	Close	Adj Close	Volume	ret
count	250.000000	250.000000	250.000000	250.000000	250.000000	2.500000e+02	250.000000
mean	187.722841	189.603520	185.856520	187.750720	186.217571	3.267795e+07	0.000426
std	22.182625	22.316426	22.046793	22.196646	21.937841	1.420180e+07	0.019323
min	143.979996	145.720001	142.000000	142.190002	141.582779	1.251390e+07	-0.099607
25%	171.285000	173.017502	169.712498	170.977497	170.212494	2.288638e+07	-0.008675
50%	186.319999	187.534996	184.965003	186.214996	184.381439	2.910085e+07	0.001611
75%	207.340001	209.437500	205.937496	207.875003	205.912826	3.896078e+07	0.009429
max	230.779999	233.470001	229.779999	232.070007	230.275482	9.624670e+07	0.070422

Figure 24: Apple Key Descriptive Statistics

	Open	High	Low	Close	Adj Close	Volume	ret
count	250.000000	250.000000	250.000000	250.000000	250.000000	2.500000e+02	250.000000
mean	108.681160	109.624280	107.652360	108.579240	107.241526	1.467994e+07	-0.000133
std	5.353291	5.194606	5.421677	5.293358	4.831455	5.350369e+06	0.013088
min	92.690002	94.220001	91.110001	92.139999	91.397758	6.488400e+06	-0.044636
25%	104.537497	105.399997	103.699997	104.469999	104.125000	1.080685e+07	-0.007409
50%	109.139999	110.509998	107.785000	109.000000	107.207867	1.361360e+07	-0.000603
75%	113.320000	114.239998	112.462498	113.274997	111.290037	1.699078e+07	0.007561
max	119.129997	119.239998	118.080002	118.629997	116.856049	4.131390e+07	0.041459

Figure 25: JPM Key Descriptive Statistics

	Open	High	Low	Close	Adj Close	Volume	ret
count	250.000000	250.000000	250.000000	250.000000	250.000000	2.500000e+02	250.000000
mean	138.367960	139.402120	137.240000	138.275520	134.830157	5.182682e+06	-0.000252
std	12.060831	11.851259	12.146777	11.971778	10.630776	3.325637e+06	0.015562
min	108.000000	111.000000	105.940002	107.570000	106.331108	1.963200e+06	-0.076282
25%	131.219997	132.227497	128.817501	130.715004	127.584597	3.428025e+06	-0.006501
50%	142.714996	143.905006	142.029999	142.700004	138.563195	4.216950e+06	0.000409
75%	146.642494	147.237499	145.554997	146.372505	141.834999	5.376600e+06	0.006723
max	159.710007	162.000000	158.509995	160.910004	153.671936	2.206370e+07	0.084639

Figure 26: IBM Key Descriptive Statistics

	Open	High	Low	Close	Adj Close	Volume	ret
count	250.000000	250.000000	250.000000	250.000000	250.000000	2.500000e+02	250.000000
mean	25001.752938	25142.486133	24845.957875	24999.364156	24999.364156	3.316040e+08	0.000197
std	860.521573	816.808842	905.113955	860.849051	860.849051	9.203173e+07	0.010476
min	21857.730469	22339.869141	21712.529297	21792.199219	21792.199219	1.559400e+08	-0.031472
25%	24484.380371	24611.017090	24272.650879	24428.174805	24428.174805	2.699000e+08	-0.004096
50%	25029.750000	25127.459961	24883.049805	25048.559571	25048.559571	3.134000e+08	0.000375
75%	25597.672851	25692.405762	25480.434570	25594.992676	25594.992676	3.777150e+08	0.005794
max	26833.470703	26951.810547	26789.080078	26828.390625	26828.390625	9.005100e+08	0.049846

Figure 27: DJI Key Descriptive Statistics

4.1.2

The PDF of the adjacent close and 1-day returns for the Apple, JPM, IBM and DJI stocks are shown in Figures 28, 29, 30 and 31 respectively.

The PDF of the returns is centered around zero and exhibits a bell shape, with most of the probabilities concentrated around small positive or negative returns. This indicates a normal distribution with mean 0 and small variance.

The PDF of the Adj. Closes have a much higher variance as their values take on a wider range. In addition, there is no obvious PDF compared to the returns' plots, although within sub-groups of the bins in the Adj. Close there are some indications of a bell curve, for example between bins 180 and 195 of the Apple stock.

This is likely because stock prices tend to follow a random walk, where the daily price changes are relatively small and follow a normal distribution, while the overall trend of the stock price may be increasing or decreasing over time.

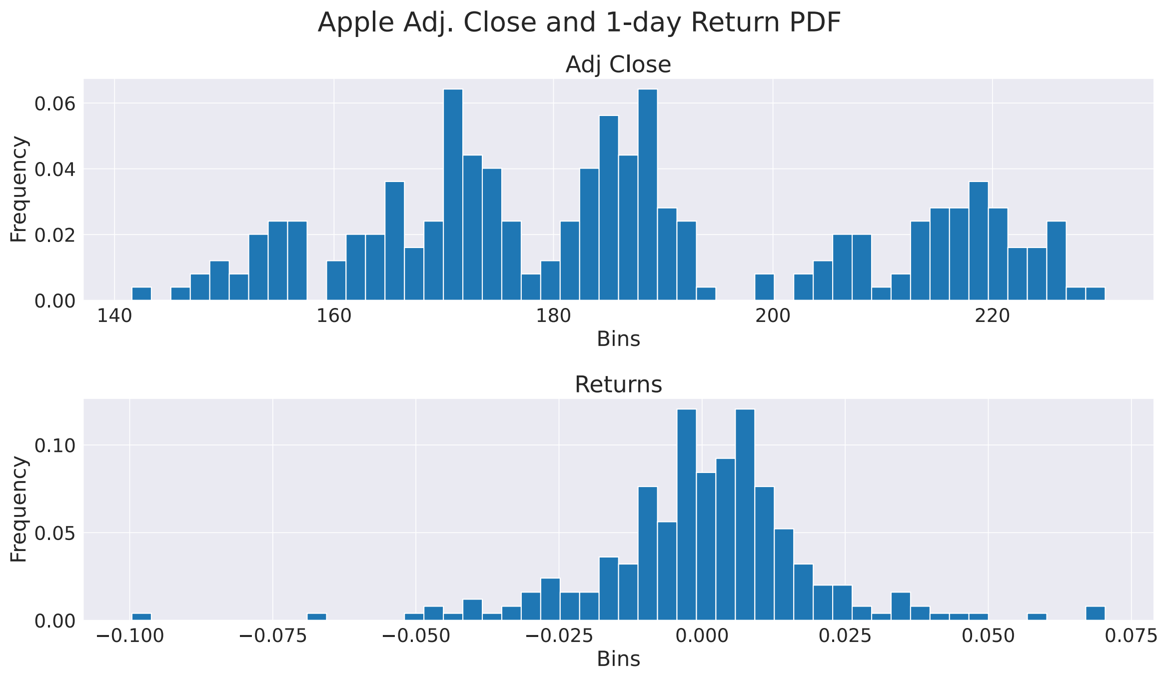


Figure 28: Apple PDF Adj. Close and 1-Day Returns

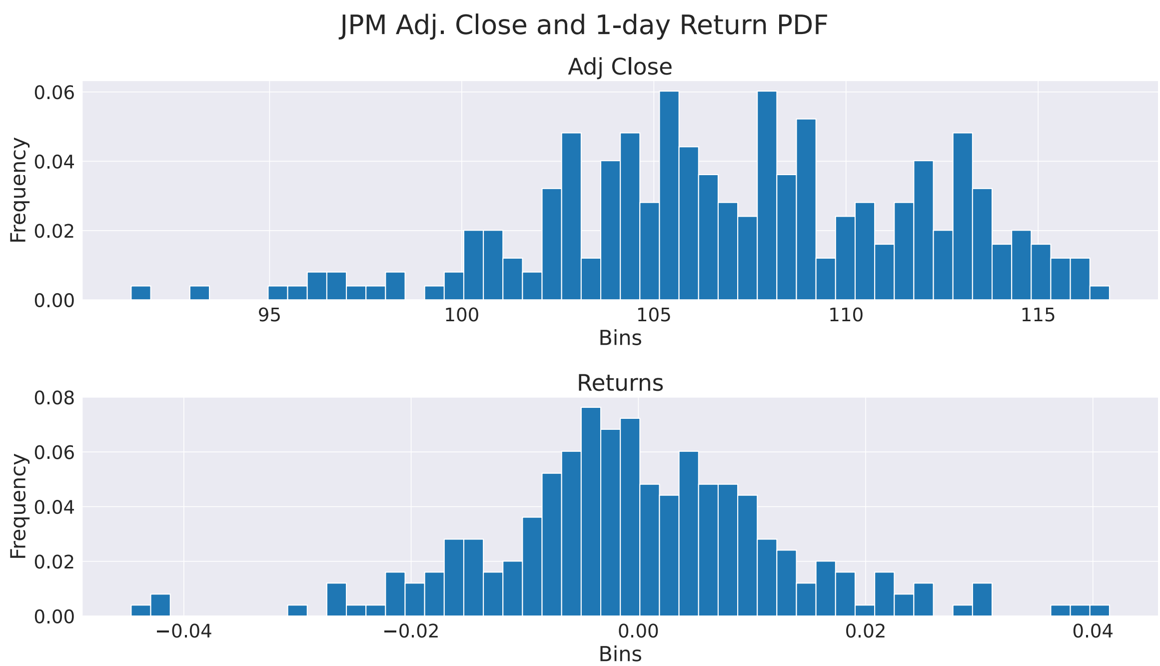


Figure 29: JPM PDF Adj. Close and 1-Day Returns

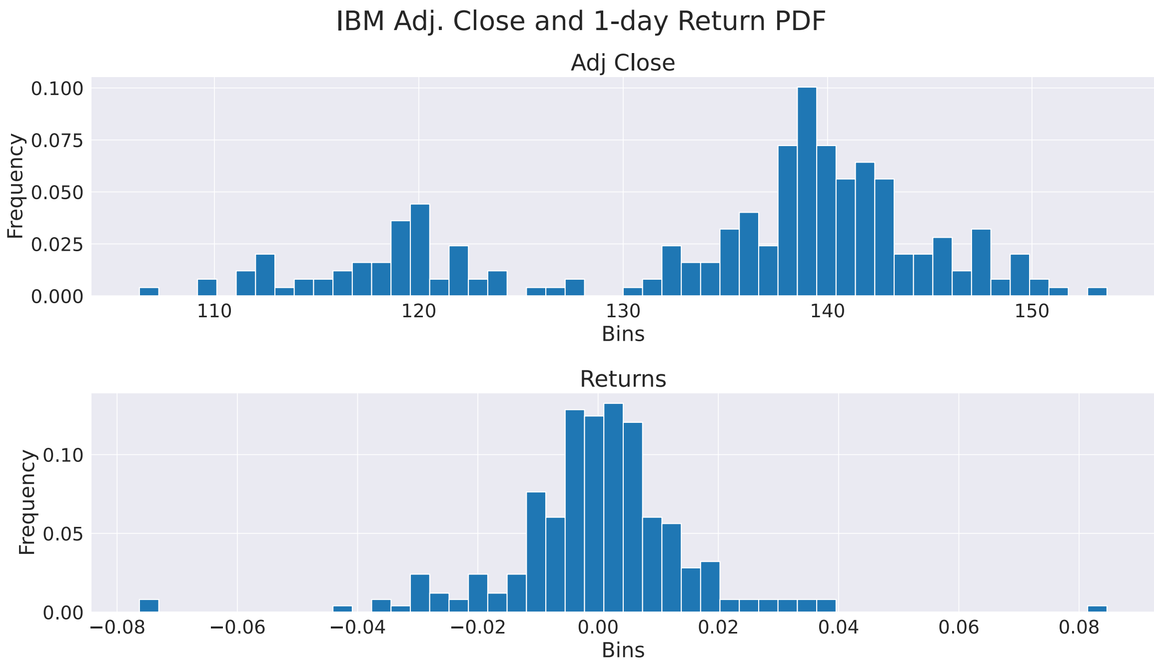


Figure 30: IBM PDF Adj. Close and 1-Day Returns

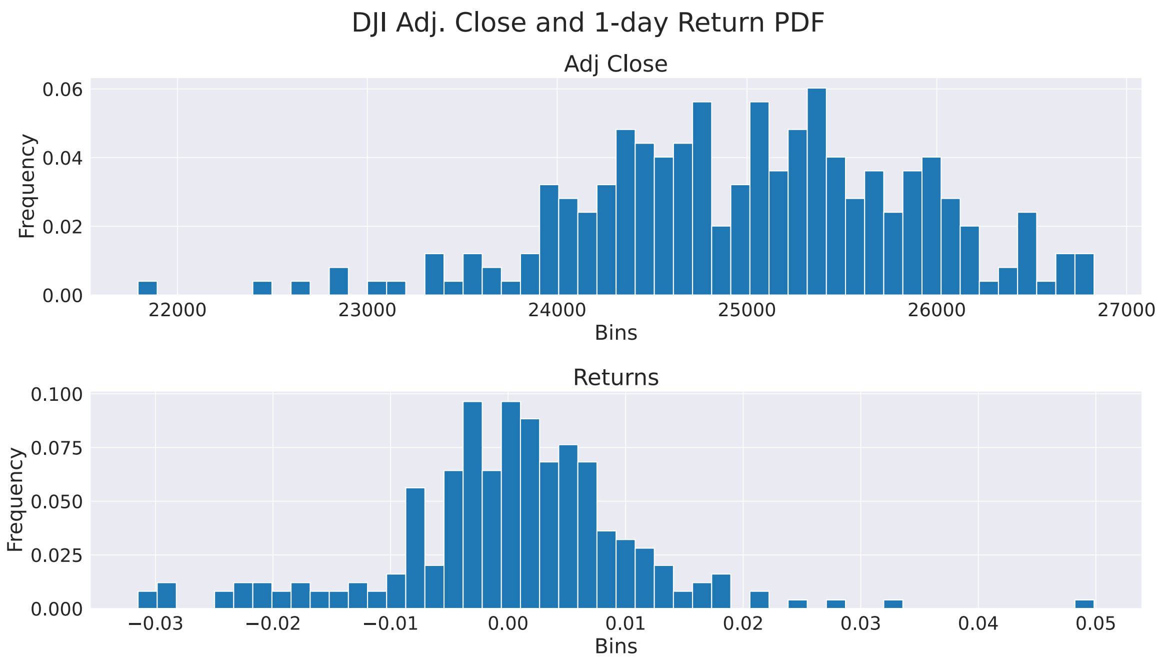


Figure 31: DJI PDF Adj. Close and 1-Day Returns

4.1.3

The Adj Close. 5-day Rolling Mean/Std and Median/Mad of the Apple, JPM, IBM and DJI stocks are shown in Figures 32, 33, 34 and 35 respectively.

Overall, the rolling Mean/Std and Median/Mad are similar with only slight differences. Firstly, the Mean/Std figures are smoother than Median/Mad and so the latter figure can be seen as noisier. Secondly, the Mean/Std figures are more sensitive to quick changes in the Adj. Close, increasing or decreasing with the Adj. Close more than the Median/Mad figures.

The rolling mean and 1.5+/-std method assumes that the data follows a normal distribution, and outliers are detected for data points that fall outside of the range of 1.5 standard deviations from the rolling mean. However, this method may not be appropriate for data that do not follow a normal distribution, as extreme values in a non-normal distribution may not necessarily be outliers. Additionally, this method may not be robust to data with many outliers, as the presence of outliers in the data can inflate the estimate of the standard deviation and lead to false positives.

The rolling MAD and 1.5*MAD method, on the other hand, is a robust method for detecting outliers that does not assume a specific distribution. MAD is a measure of variability that is less sensitive to outliers than the standard deviation

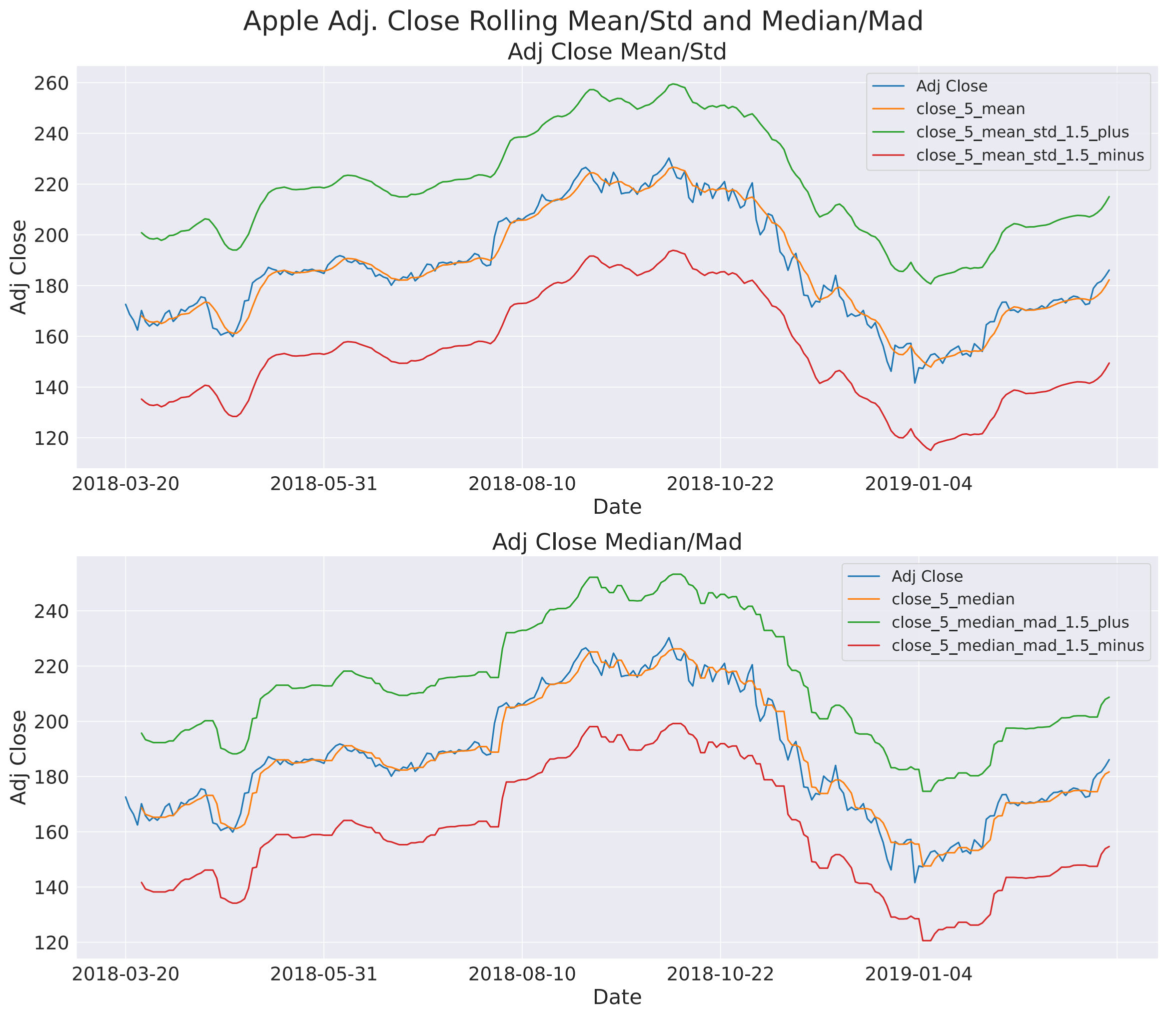


Figure 32: Apple Adj. Close Rolling Mean/Std and Median/Mad

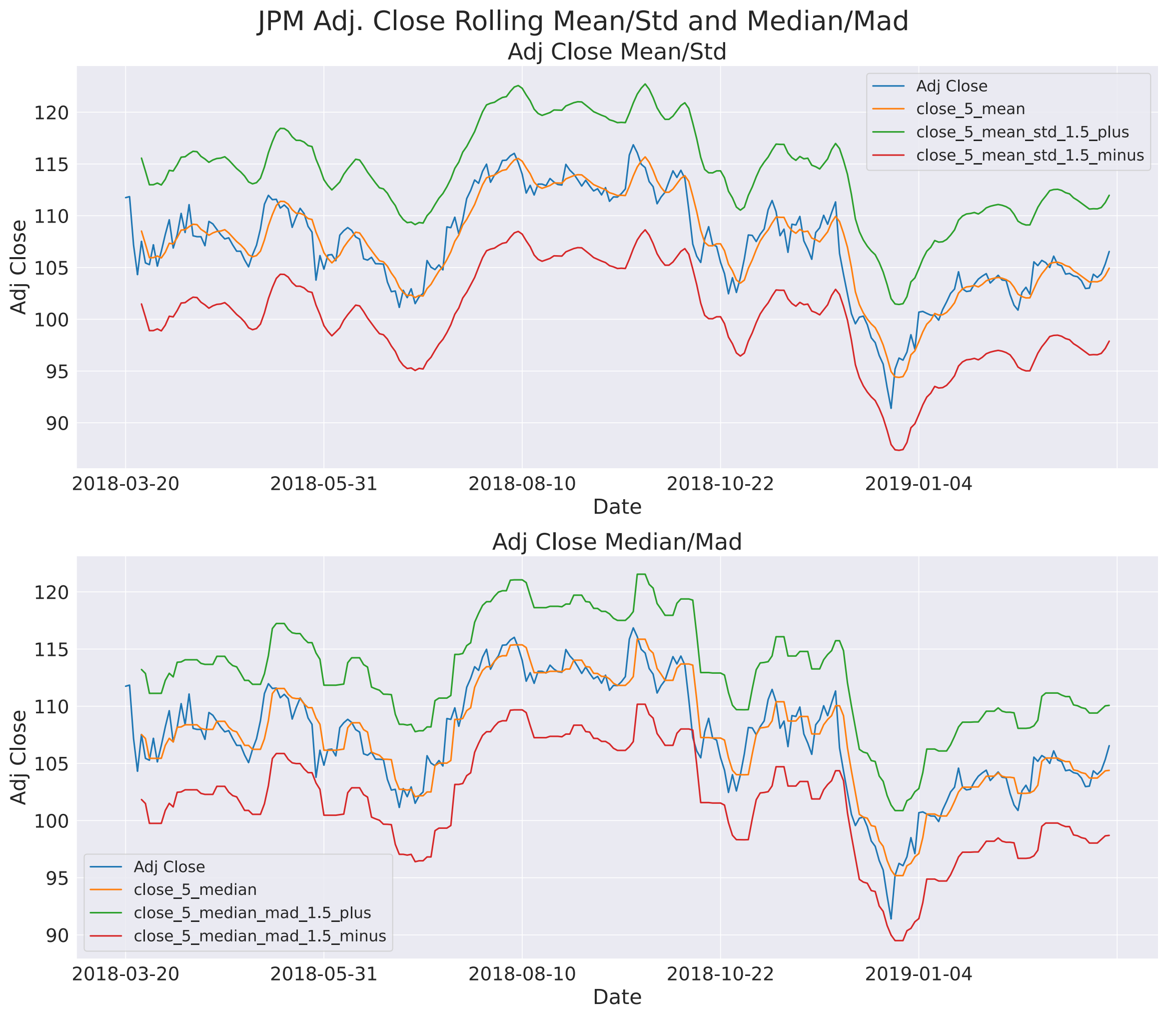


Figure 33: JPM Adj. Close Rolling Mean/Std and Median/Mad

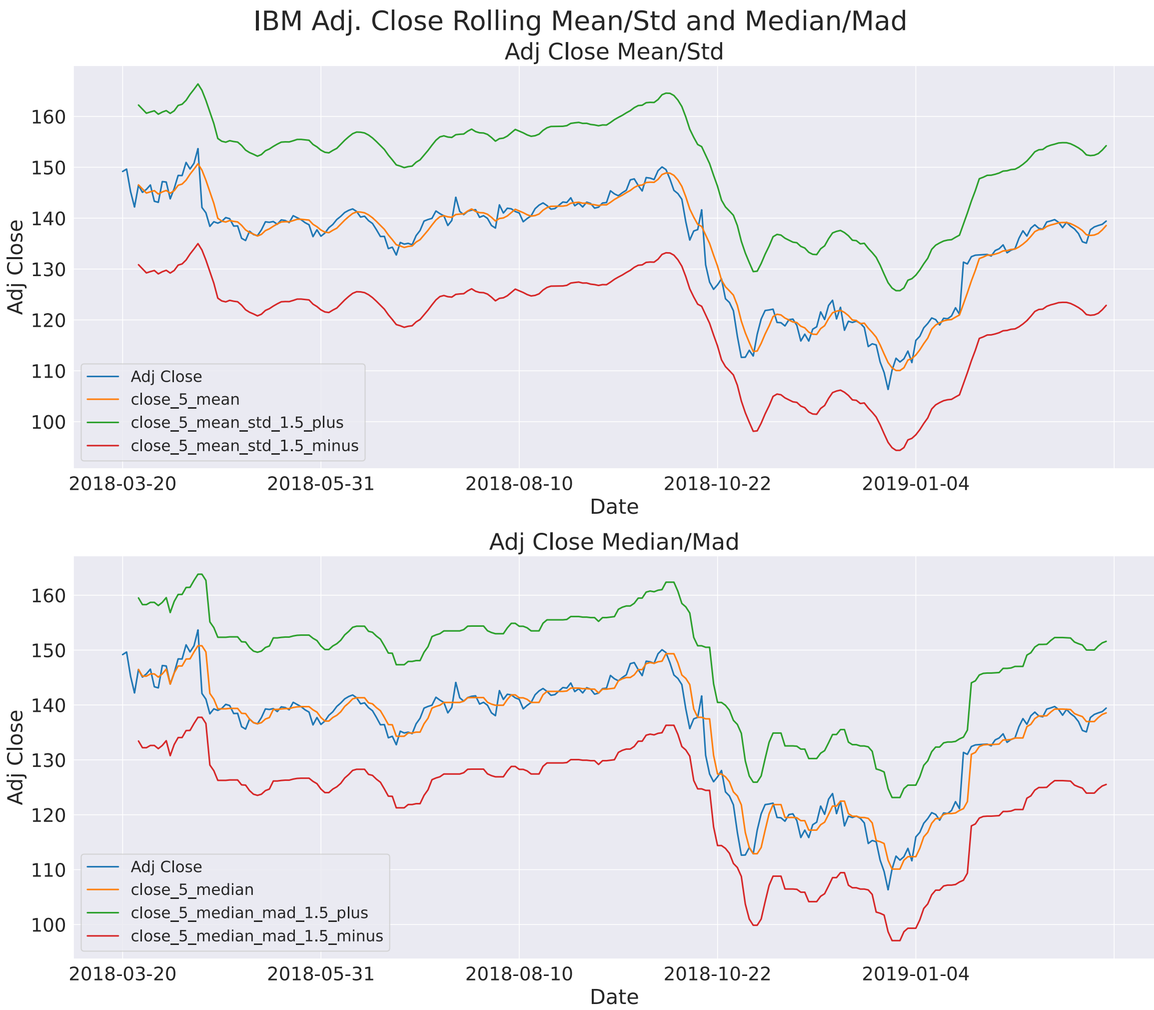


Figure 34: IBM Adj. Close Rolling Mean/Std and Median/Mad

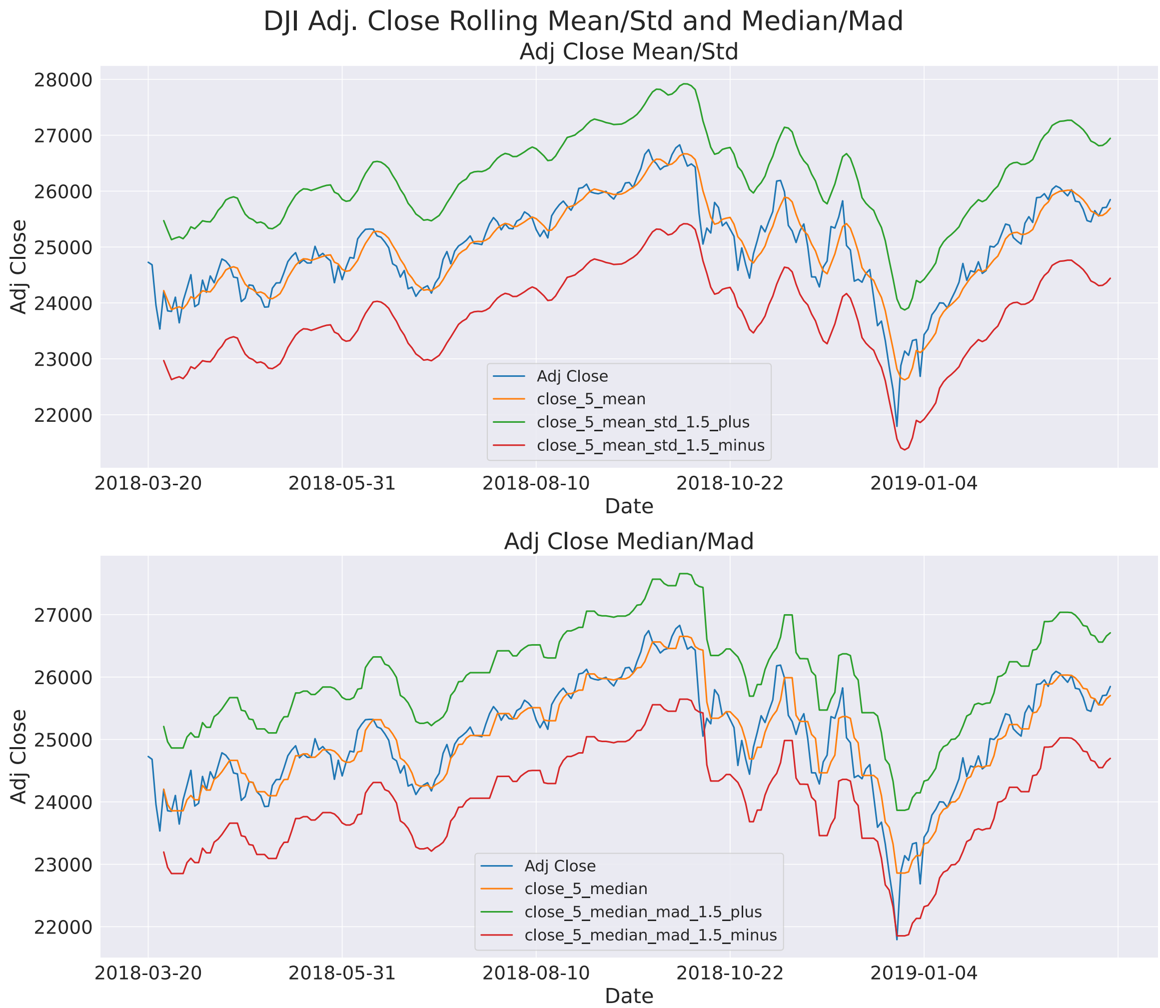


Figure 35: DJI Adj. Close Rolling Mean/Std and Median/Mad

4.1.4

The same Figures as in Section 4.1.3 but with introduced outlier points are shown in Figures 36, 37, 38 and 39.

These outlier points cause larger spikes in the Mean and Std lines compared to the Median and Mad lines. This is due to the fact that mean and std are more sensitive to fluctuations, as discussed in Section 4.1.3.

However, the rolling window calculation for both methods results in a much smaller increase in magnitude than the outlier magnitude itself, as the effect is averaged out and therefore weighted downwards by the other points in the window.

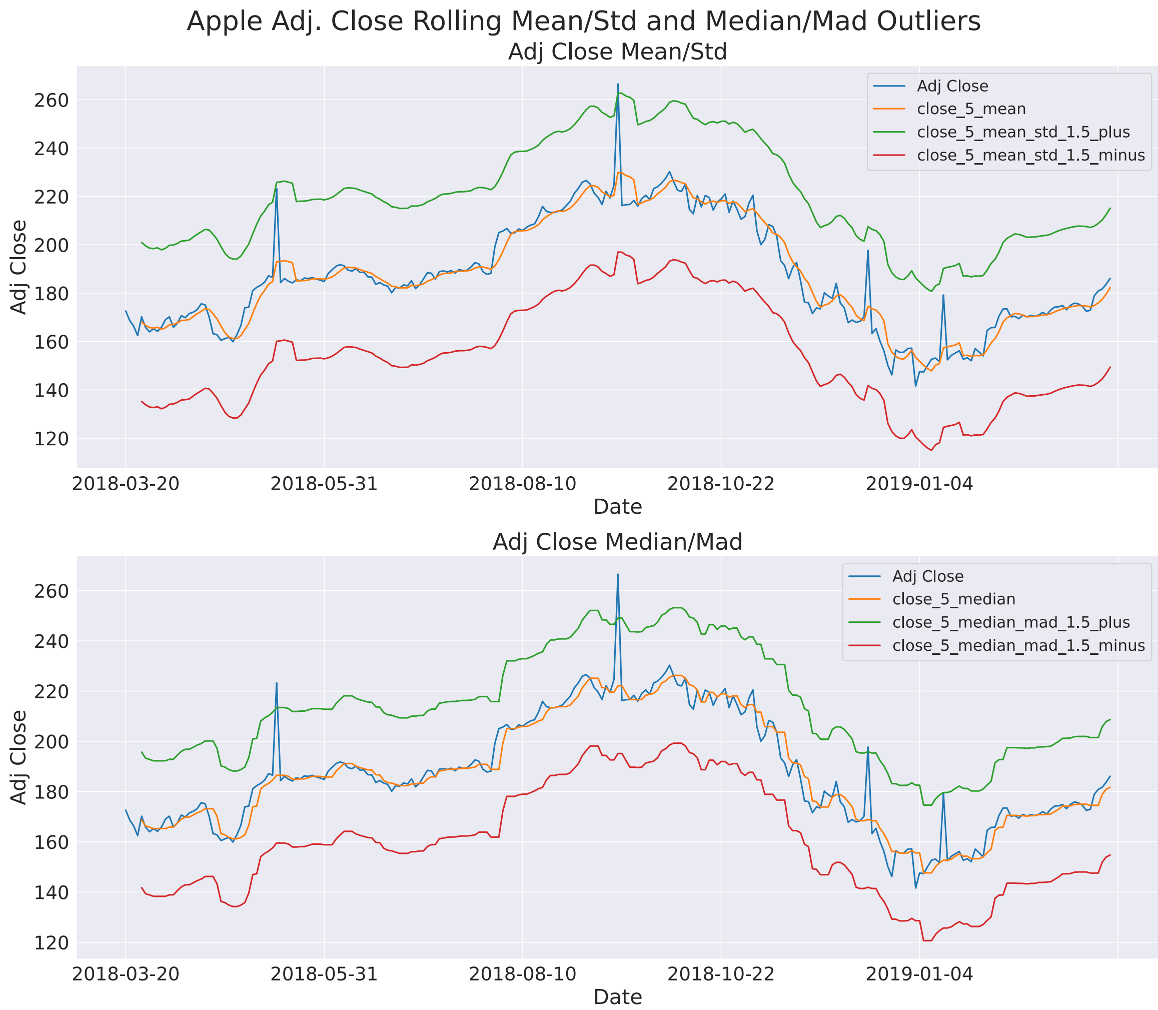


Figure 36: Apple Adj. Close Rolling Mean/Std and Median/Mad with Outliers

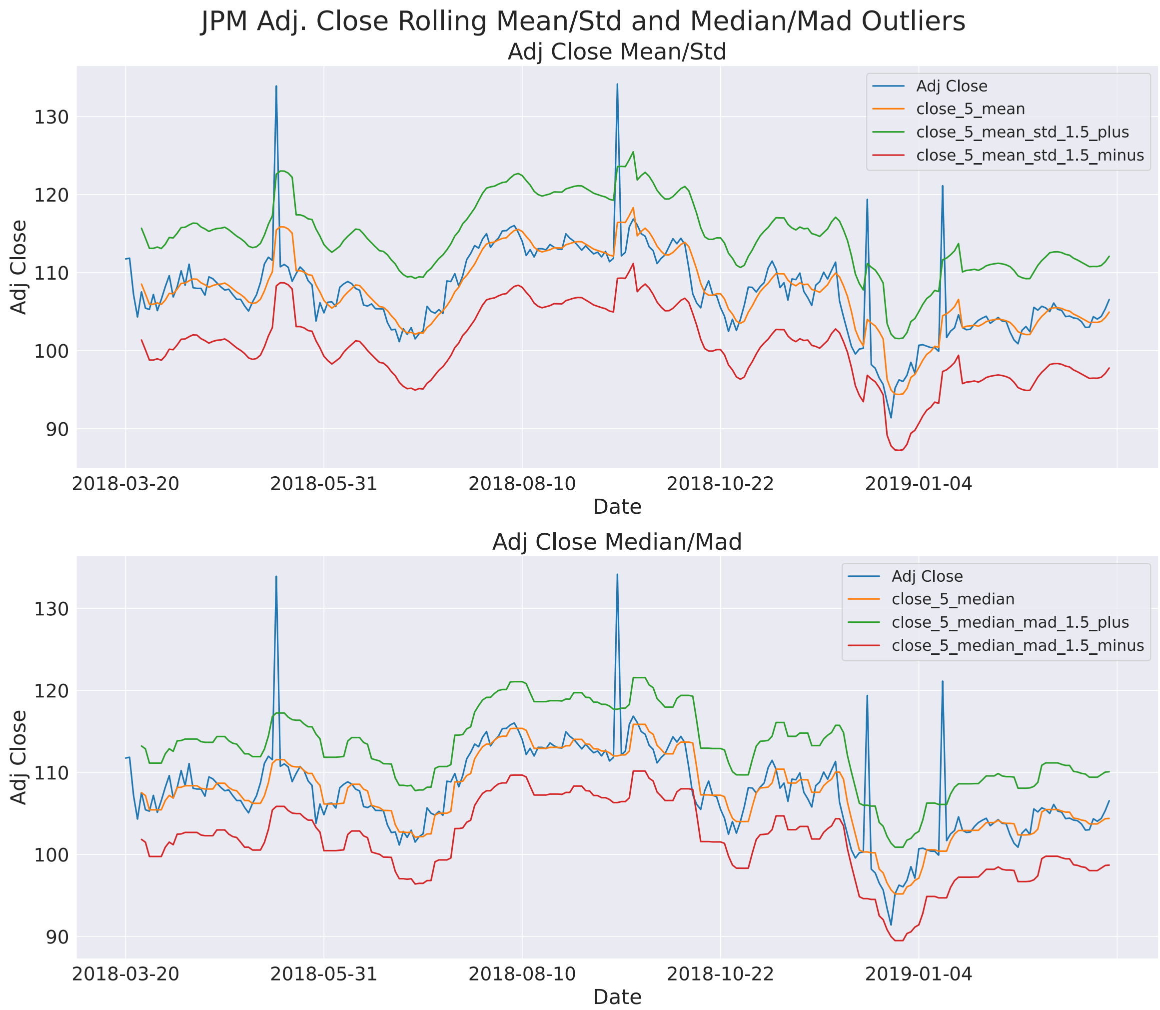


Figure 37: JPM Adj. Close Rolling Mean/Std and Median/Mad with Outliers

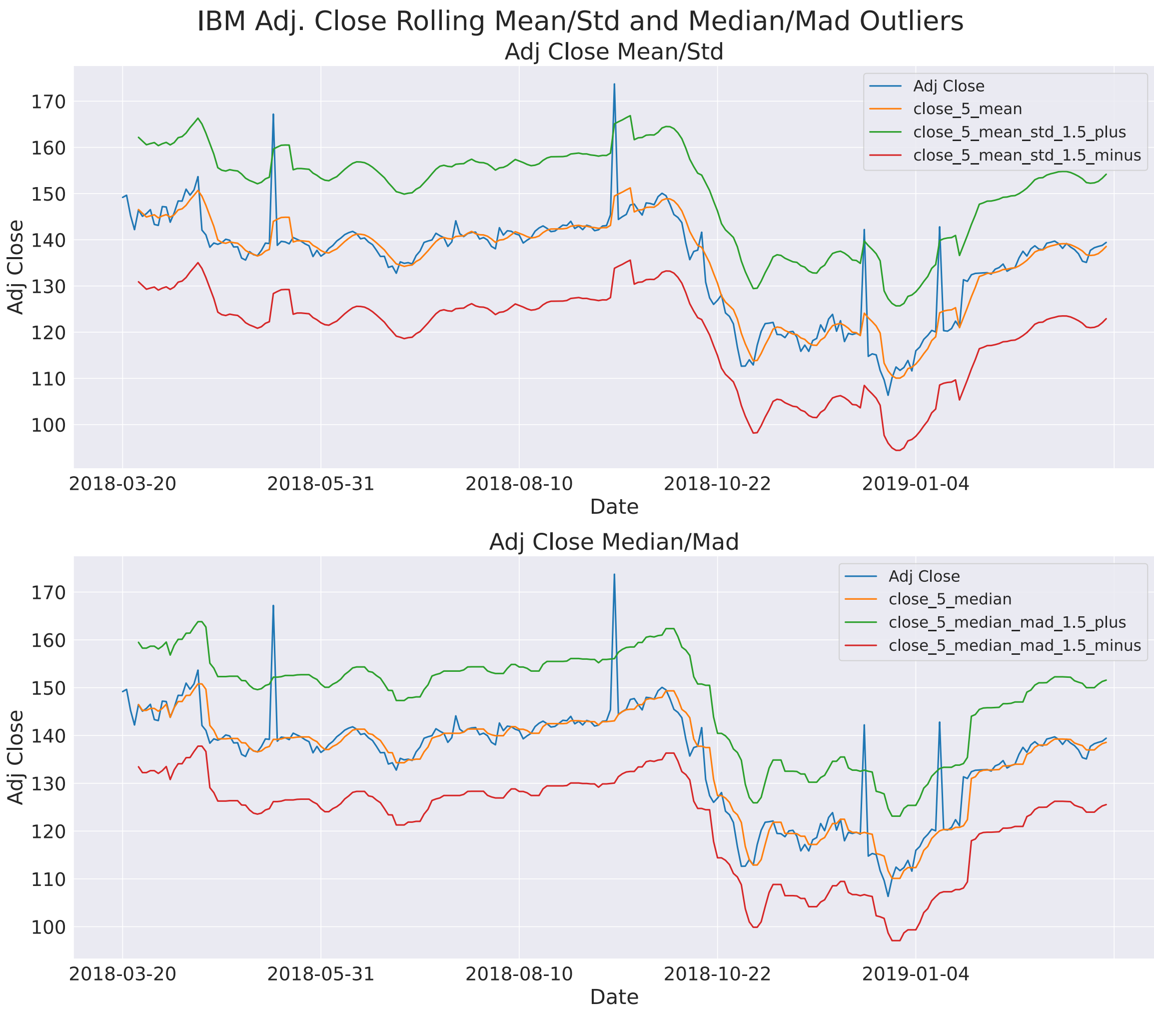


Figure 38: IBM Adj. Close Rolling Mean/Std and Median/Mad with Outliers

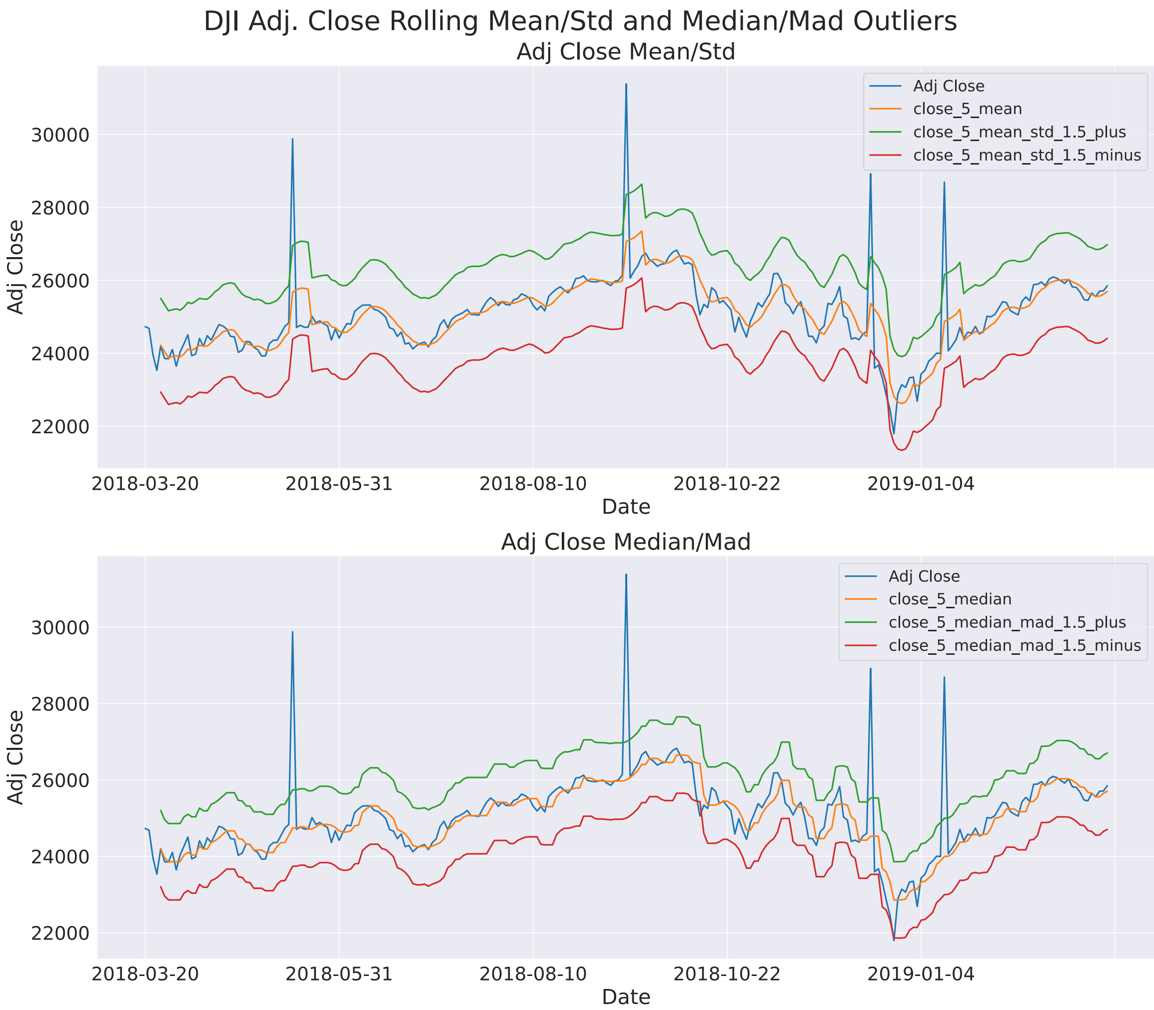


Figure 39: DJI Adj. Close Rolling Mean/Std and Median/Mad with Outliers

4.1.5

Figure 40 shows the Stock Adj. Close box plots.

The box plot consists of a box, which represents the interquartile range (IQR) of the data, and a line inside the box, which represents the median of the data. The lines that extend from the box represent the minimum and maximum values of the data that are not outliers. Outliers are represented by the circles outside of the lines.

The DJI stock has the smallest box, indicating smallest IQR which can be perceived as a robust measure of variability around the median. This is expected as the DJI stock is an index and so generally has lower variability.

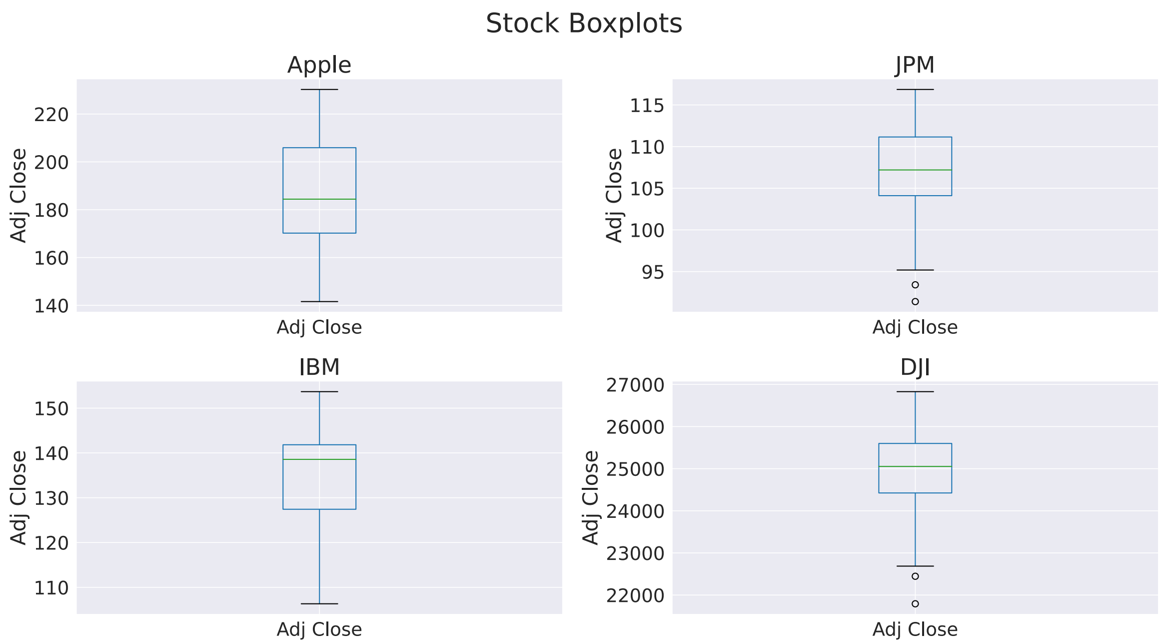


Figure 40: Stock Adj. Close Box Plots

4.2: Robust Estimators

4.2.1

The code for the Python functions of the estimator types: median, IQR and MAD is shown in Appendix A. For comparisons, Figure 41 show the values of the Python pandas built-in functions and the custom created functions. The median and IQR are almost identical (to 3 significant figures), however the MAD values are different.

The pandas MAD function uses a normalization factor of 1.4826, while the custom function does not. This normalization factor is used to make the median absolute deviation (MAD) estimate consistent with the standard deviation of a normally distributed random variable.

	Median	IQR	MAD	custom_median	custom_iqr	custom_mad
Apple	184.411102	35.715225	17.971016	184.381439	35.691315	15.577713
JPM	107.196465	7.042633	3.921722	107.189042	7.056469	3.403046
IBM	138.559998	14.396965	8.665439	138.558357	14.626244	4.433212
DJI	25052.830078	1175.480468	689.96902	25048.559571	1171.695312	595.97461

Figure 41: Public and Custom Estimator Function Values for each stock

4.2.2

To compare computational efficiency, the time to run each custom function and the Python pandas built-in functions were calculated. The results are shown in Table 5.

Comparing between Pandas times and Custom times, they are similar (i.e. within the same factor). Comparing between functions, the median and IQR calculations are within the same factor. whereas the MAD function is a factor of 10 slower.

Robust Estimator	Pandas	Custom
Median	0.0003	0.0003
IQR	0.0004	0.0004
MAD	0.001	0.002

Table 5: Robust Estimators Execution Time

4.2.3

The breakdown point of an estimator is the maximum percentage of outliers that an estimator can handle without producing an unreliable result.

The breakdown point of the median is 50%, which means that if up to 50% of the data are outliers, the median will still provide a reliable estimate of the center of the distribution.

The breakdown point of the IQR is also 50%, but it estimates the scale of the data instead of the center. Therefore, if up to 50% of the data are outliers, the IQR will still provide a reliable estimate of the spread of the distribution.

The breakdown point of the MAD is 50% for normally distributed data but decreases for heavy-tailed distributions.

4.3: Robust and OLS Regression

4.3.1

Figures 42, 43 and 44 show the Apple, JPM and IBM 1-day DJI OLS regression respectively. For each figure, the top graph shows the true vs predictions for each day, while the bottom graph shows the return values as a scatter plot with a line of best fit.

In general, all three stocks show a general positive trend, shown by the positive gradient from the line of best fit. This shows that the OLS regression is able to capture some information. To further check the performance, the R^2 score is computed for each stock. The results are shown in Table 6. The R^2 value is a measure of how well the model fits the data and usually takes values between 0 and 1.

In general, a commonly used guideline is that an R^2 value of 0.7 or higher is considered a strong fit, while a value between 0.3 and 0.7 is considered a moderate fit, and a value below 0.3 is considered a weak fit. Thus, all three stocks exhibit a moderate fit, with the IBM stock showing the least well fit at 0.14 less than Apple and JPM.

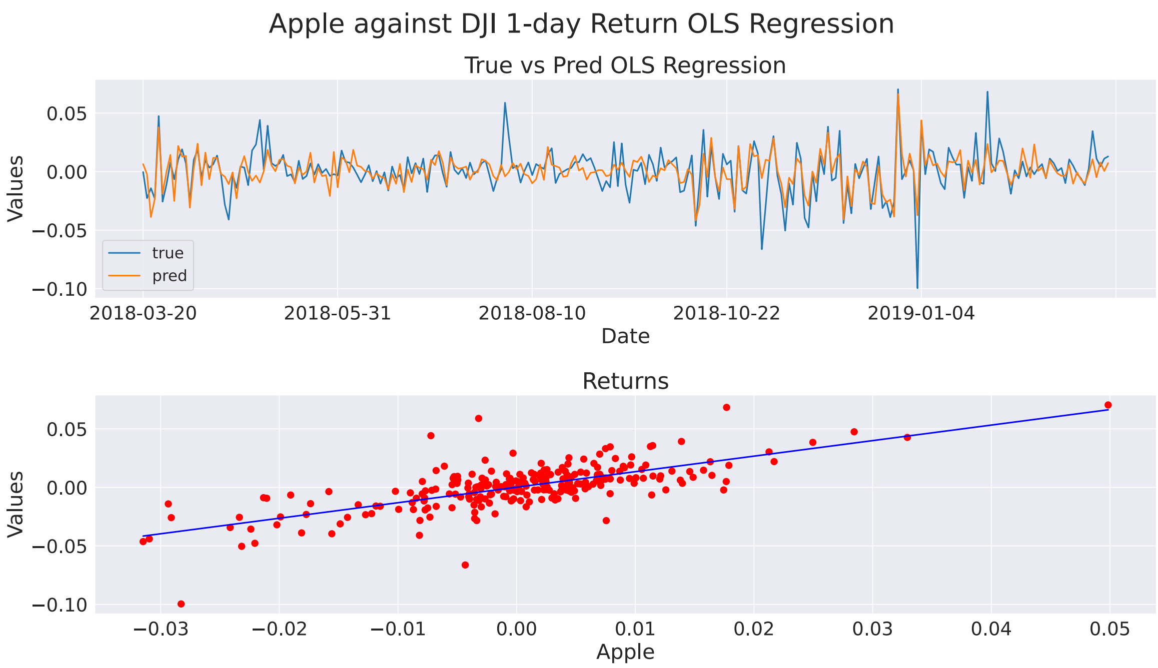


Figure 42: Apple 1-day DJI returns OLS regression

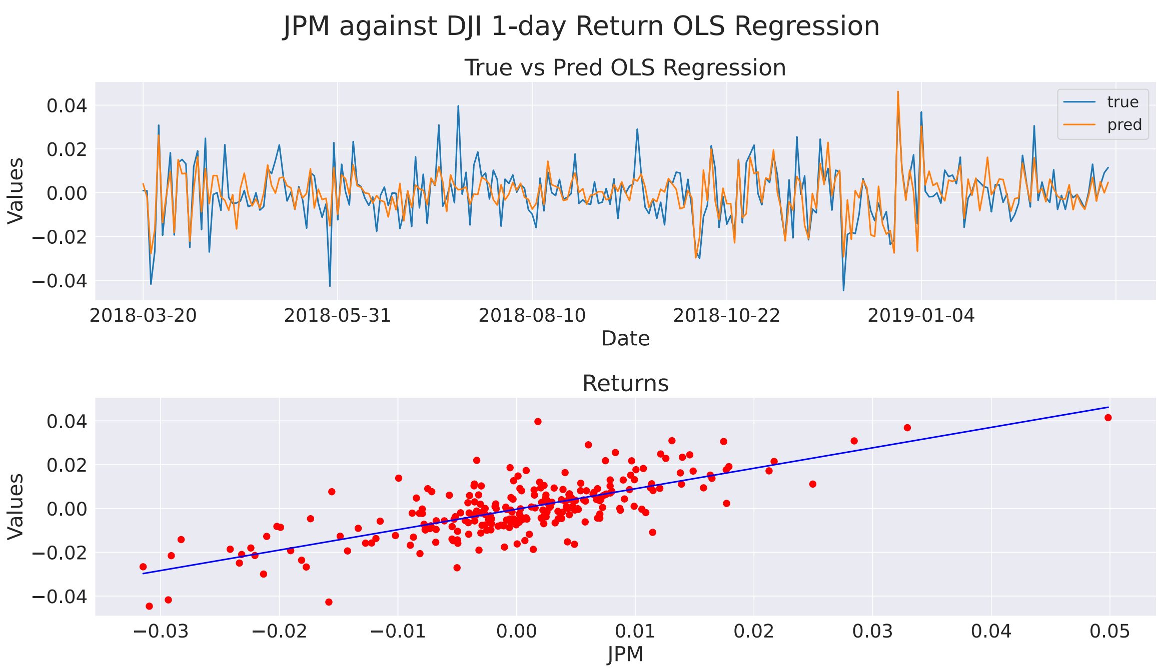


Figure 43: JPM 1-day DJI returns OLS regression

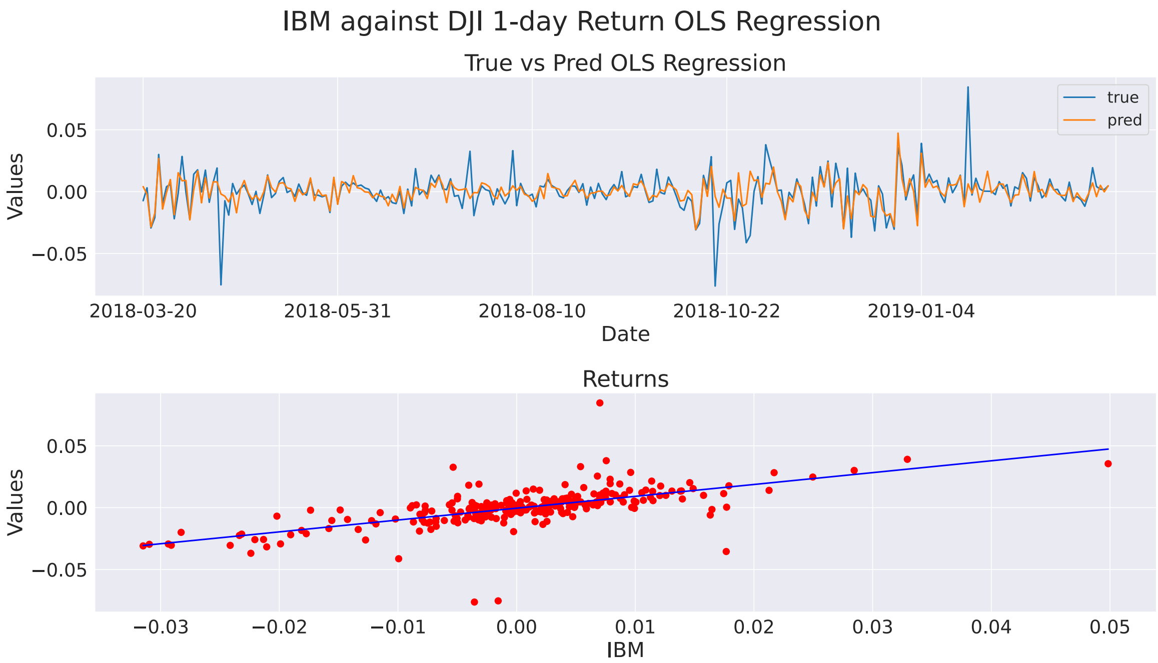


Figure 44: IBM 1-day DJI returns OLS regression

Stock	R^2 Score
Apple	0.52
IBM	0.42
JPM	0.56

Table 6: R^2 of Stocks OLS Regression

4.3.2

Figures 45, 46 and 47 show the Apple, JPM and IBM 1-day DJI Huber regression respectively. For each figure, the top graph shows the true vs predictions for each day, while the bottom graph shows the return values as a scatter plot with a line of best fit.

Huber regression is less sensitive to outliers compared to OLS regression. In OLS regression, the model tries to minimize the sum of the squared residuals, which can be heavily influenced by outliers. As a result, if there are outliers in the data, the OLS estimator can produce biased or inefficient results.

On the other hand, Huber regression minimizes the sum of the squared residuals for small residuals, and using a linear function of the residual for larger residuals. This allows Huber regression to effectively balance between being influenced by outliers and not being influenced by them. In general, if there are a significant number of outliers in the data or if the data is suspected to have heavy-tailed errors, Huber regression may be preferred over OLS regression.

Table **??** shows the R^2 score. In general, the scores are similar to OLS regression, being between 0.4 and 0.6. However, for each stock, the Huber regression produces a slightly lower score than OLS.

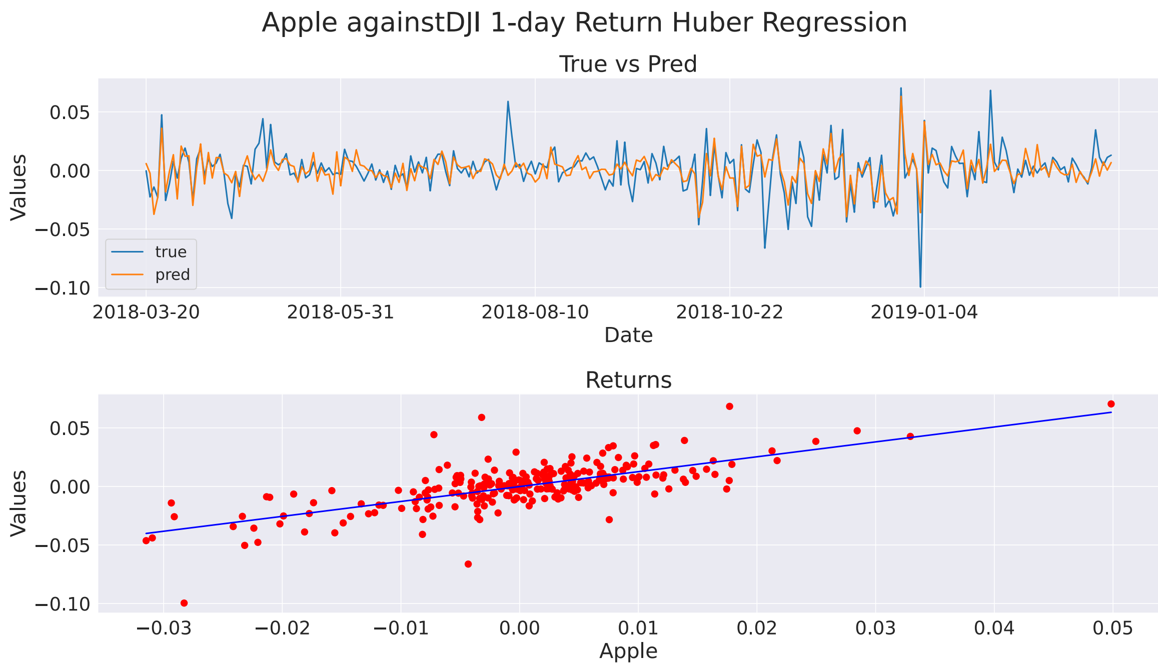


Figure 45: Apple 1-day DJI returns Huber regression

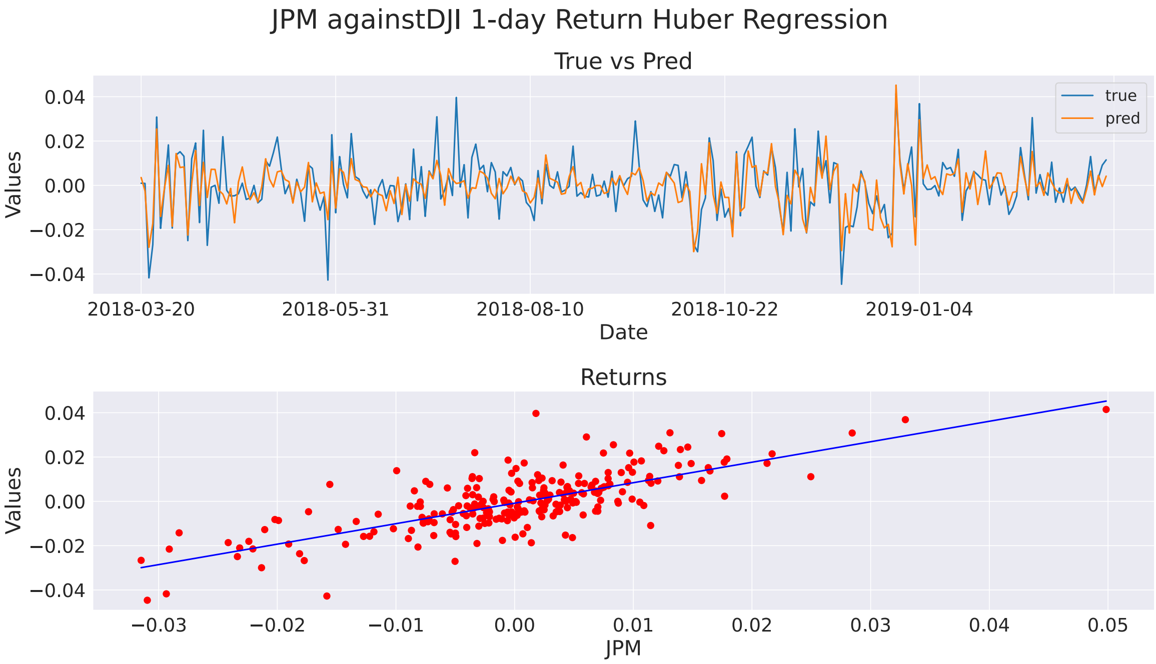


Figure 46: JPM 1-day DJI returns Huber regression

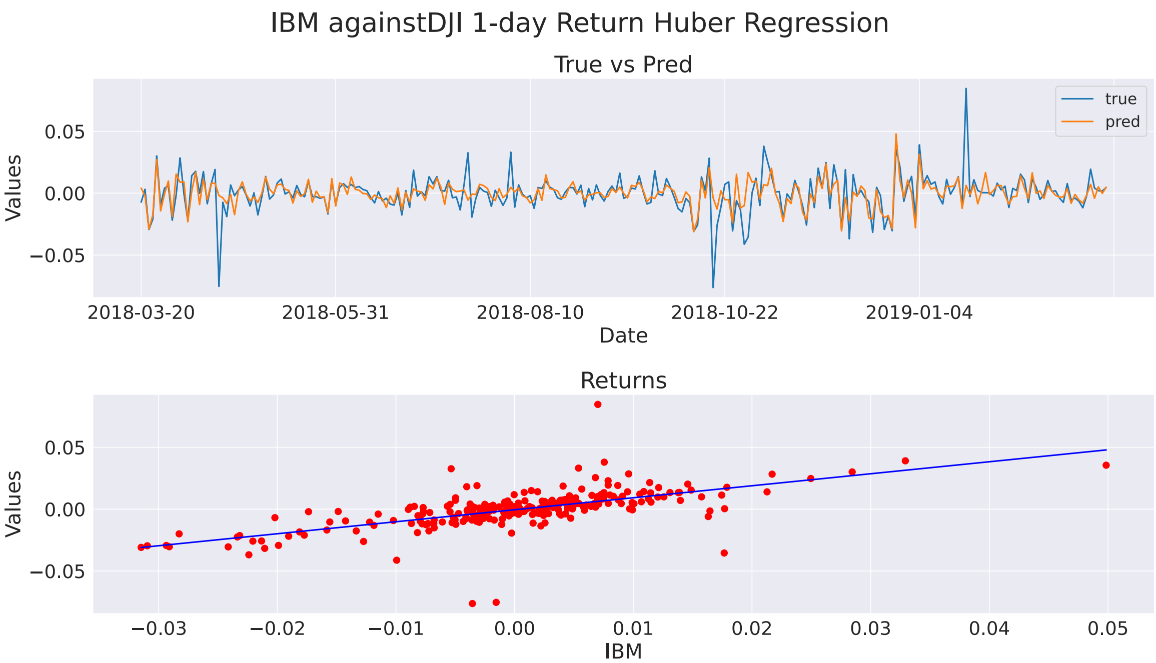


Figure 47: IBM 1-day DJI returns Huber regression

Stock	R^2 Score
Apple	0.51
IBM	0.41
JPM	0.55

Table 7: R^2 of Stocks Huber Regression

4.3.3

Figures 48, 49 and 50 show the Apple, JPM and IBM 1-day DJI Huber regressions with outliers. For each figure, the top graphs show OLS regression, while the bottom graphs show the Huber regression.

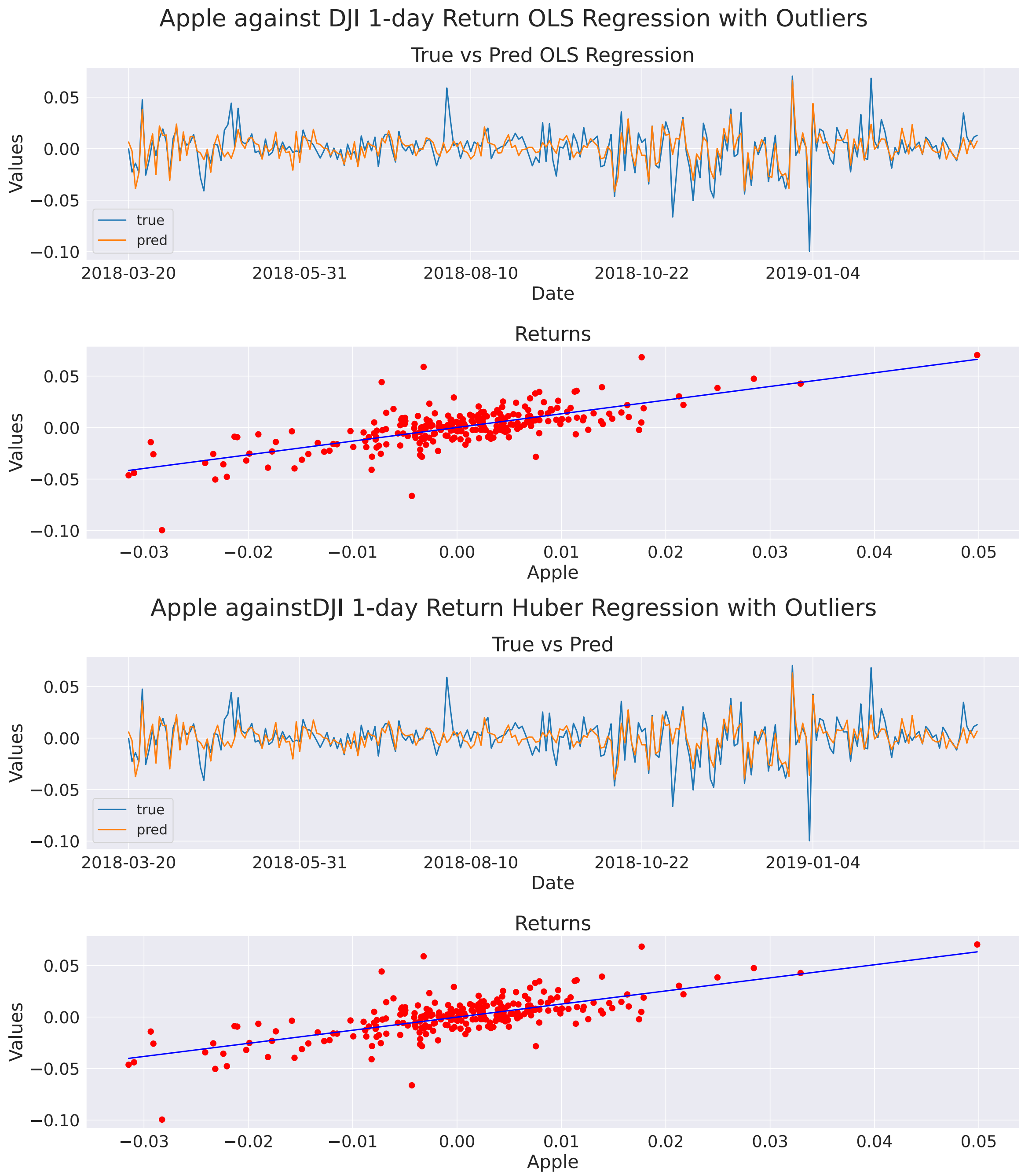


Figure 48: Apple 1-day DJI returns regression with Outliers

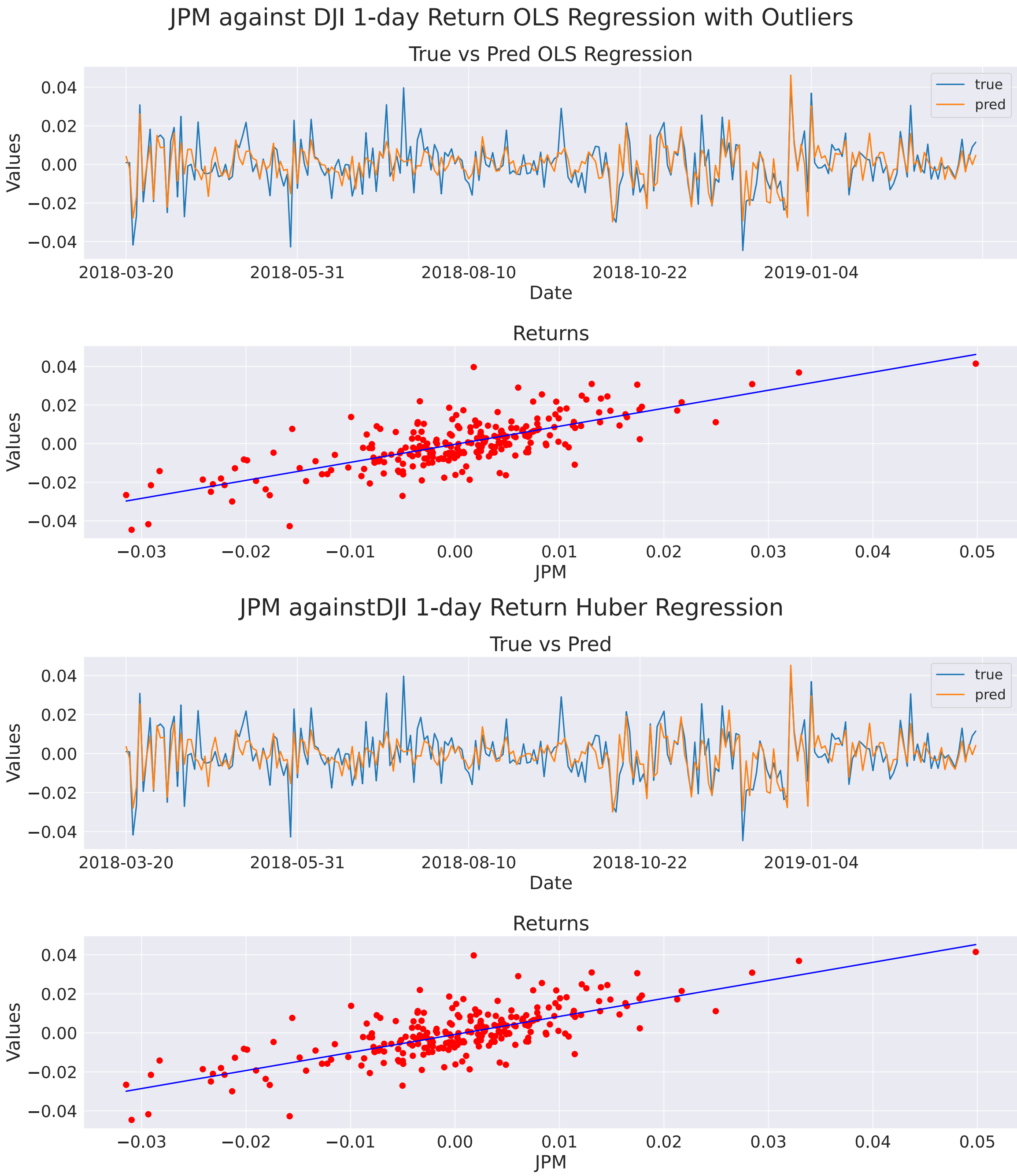


Figure 49: JPM 1-day DJI returns regression with Outliers

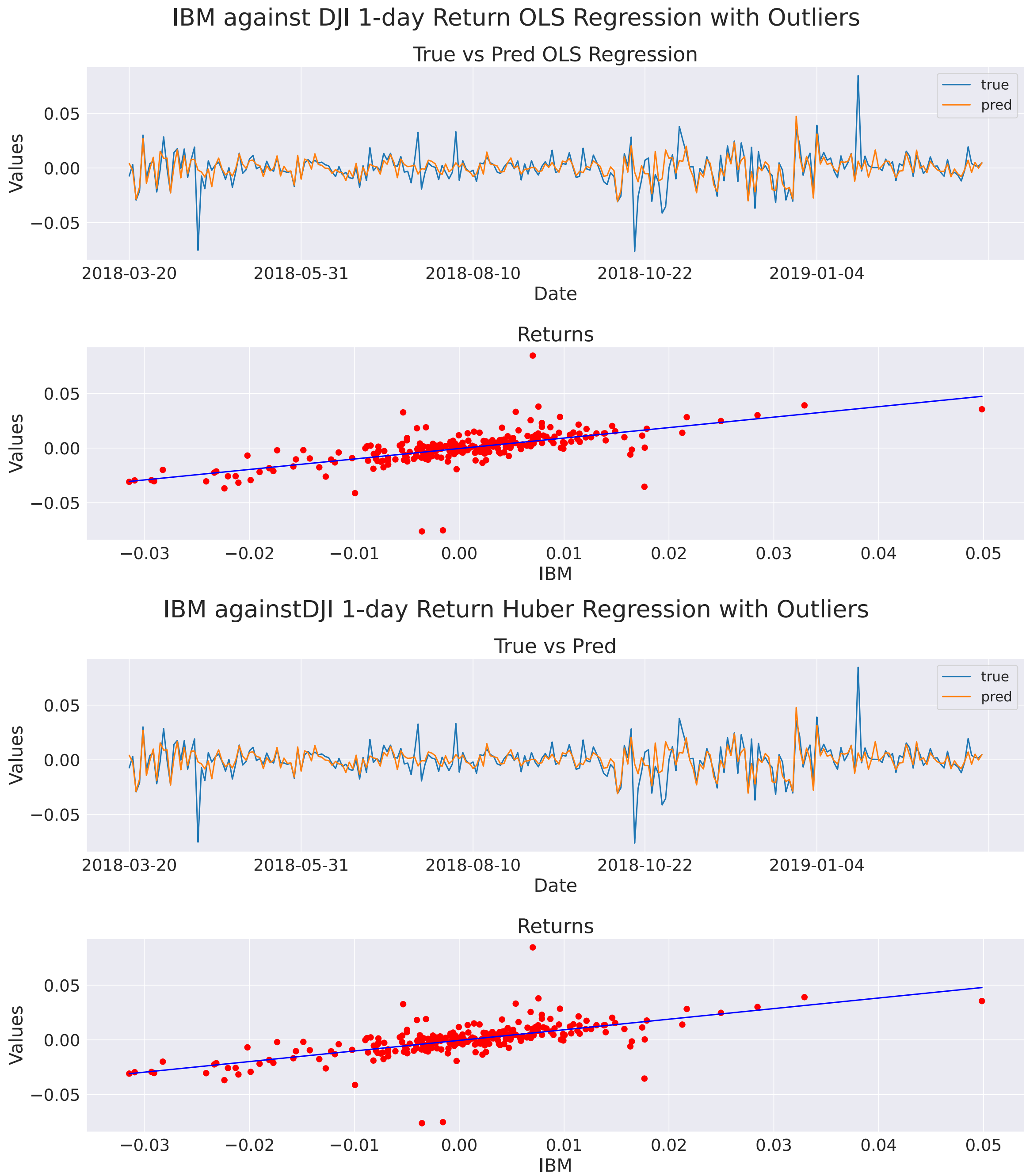


Figure 50: IBM 1-day DJI returns regression with Outliers

Stock	R^2 Score OLS	R^2 Score Huber
Apple	0.51	fudfhwi
IBM	0.41	weufwu
JPM	0.55	wdfgwuf

Table 8: R^2 Scores of Stocks corrupted with Outliers

4.4: Robust Trading Strategies

4.4.1

Figures 51, 52, 53 and 54 show the Adj. Close, 20-day and 50-day moving averages. For each figure, the top graph shows the non-corrupted time series, while the bottom graph shows the time-series corrupted by outliers.

Outlier corruption occurs on the following dates:

- 2018-05-14
- 2018-09-14
- 2018-12-14
- 2019-01-14

When corrupting with outliers, for each outlier date and for each stock, the Adj. Close will randomly increase or decrease by 20%. The random seed is included in the notebook for reproducibility.

Table's 9 and 10 shows the Buy and Sell Dates with and without outliers (called Not Corrupted and Corrupted), using the Moving Average Crossover trading strategy.

The results show that corrupting the data produces more buy and sell dates from the moving average crossover strategy. When synthetic outliers are present in the data, they can cause significant fluctuations in the price, which can result in more frequent crossovers of the moving averages. This is because the outliers can cause the shorter-term moving average to cross above or below the longer-term moving average more frequently, leading to more trading signals being generated.

Stock	Series	Buy dates
Apple	Not Corrupted	2018-07-24, 2019-02-12
	Corrupted	2018-07-25, 2019-02-12
JPM	Not Corrupted	2018-07-27, 2019-02-01
	Corrupted	2018-05-31, 2018-06-12, 2018-07-27, 2019-02-01
IBM	Not Corrupted	2018-07-25, 2019-01-24
	Corrupted	2018-07-25, 2019-01-23
DJI	Not Corrupted	2018-07-26, 2019-01-31
	Corrupted	2018-07-24, 2018-07-26, 2019-01-31

Table 9: Stock Moving Mean Crossover Buy Dates

Stock	Series	Sell dates
Apple	Not Corrupted	2018-07-11, 2018-10-29
	Corrupted	2018-07-10, 2018-10-26
JPM	Not Corrupted	2018-06-12, 2018-09-27
	Corrupted	2018-06-07, 2018-06-14, 2018-10-12
IBM	Not Corrupted	2018-10-19
	Corrupted	2018-10-18
DJI	Not Corrupted	2018-07-10, 2018-10-23
	Corrupted	2018-07-12, 2018-07-25, 2018-10-22

Table 10: Stock Moving Mean Crossover Sell Dates

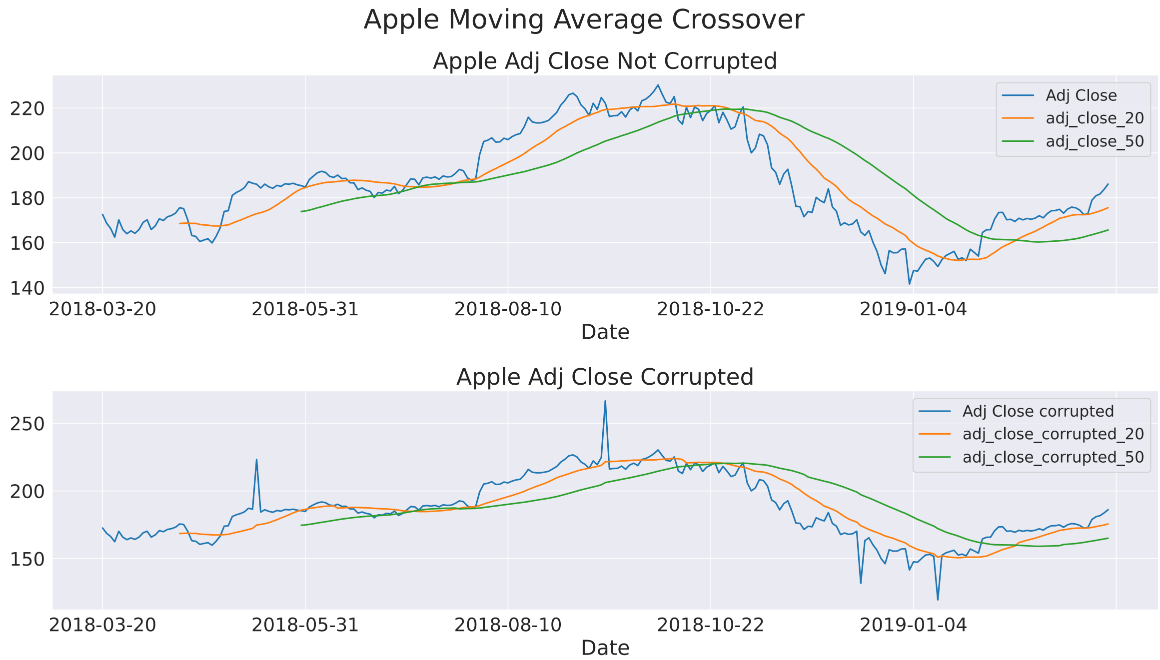


Figure 51: Apple Moving Mean Crossover

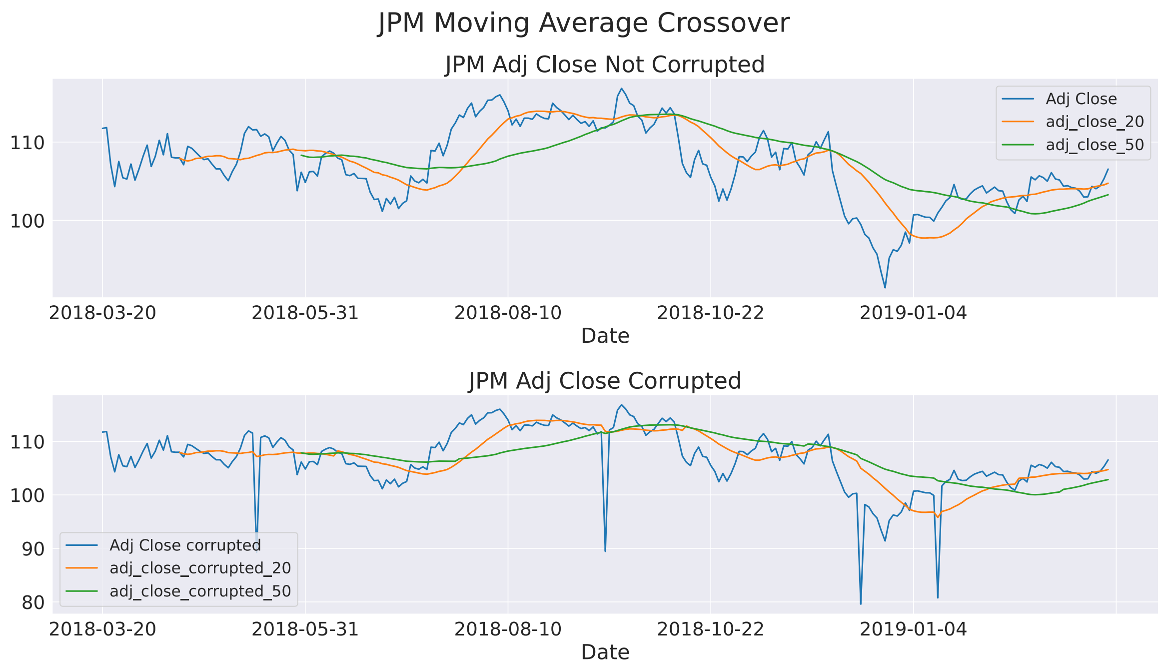


Figure 52: JPM Moving Mean Crossover

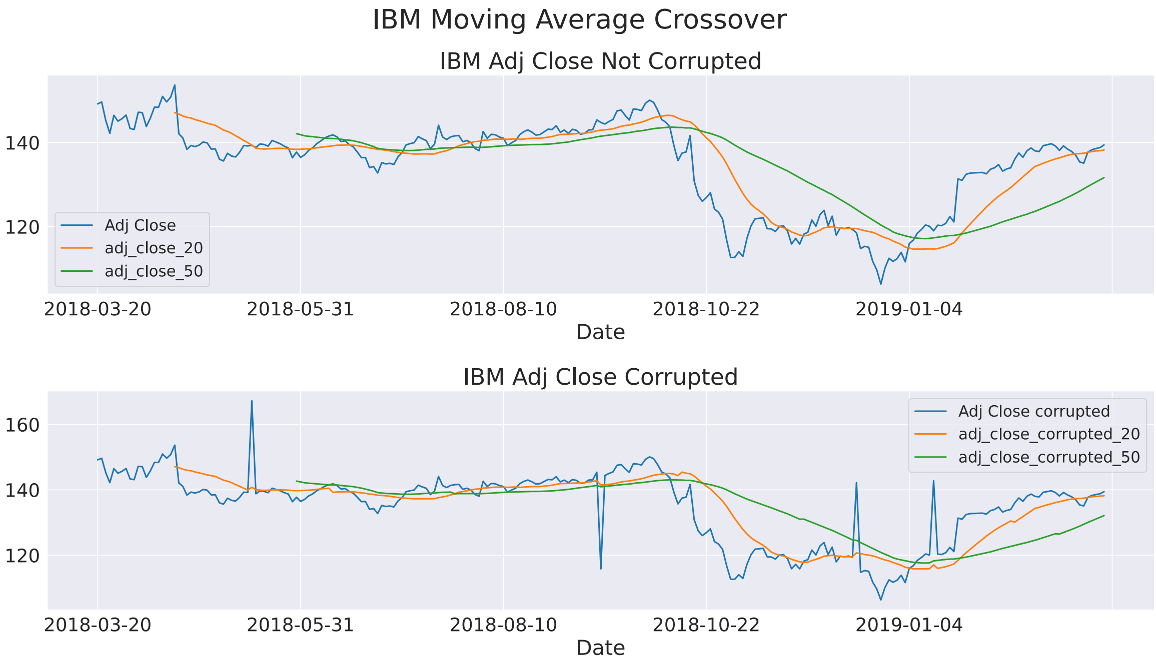


Figure 53: IBM Moving Mean Crossover

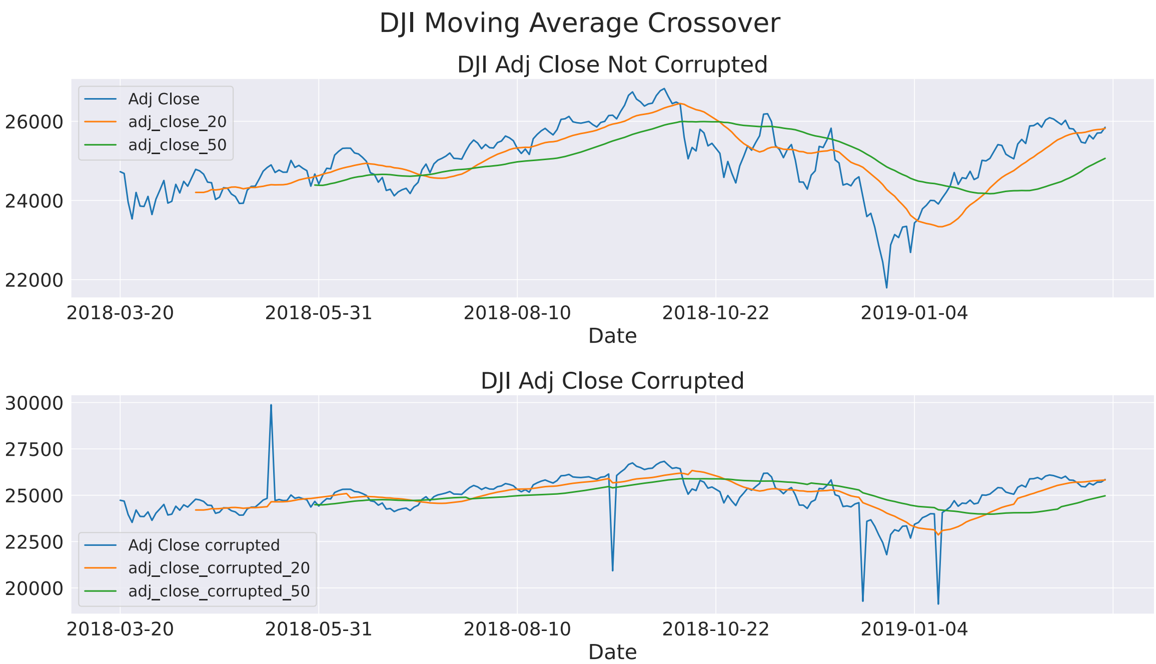


Figure 54: DJI Moving Mean Crossover

4.4.2

Figures 55, 56, 57 and 58 and Tables 11 and 12 show the same information as in Section 4.4.1, but using the rolling median instead.

If the mean is used to generate trading signals, the presence of synthetic outliers can lead to false trading signals. For example, if the mean is skewed upwards due to an outlier, it may suggest a buy signal even if the actual trend is downwards. In contrast, using the median can provide a more accurate representation of the trend and generate more reliable trading signals.

The Median results show that these buy and sell dates do not change much when outliers are introduced, for example Apple, IBM and DJI still have 2 crossover points each, with and without outliers. However, when using the mean, IBM and DJI have more buy and sell dates for the corrupted data than the not corrupted. This shows the sensitivity of the mean to outliers.

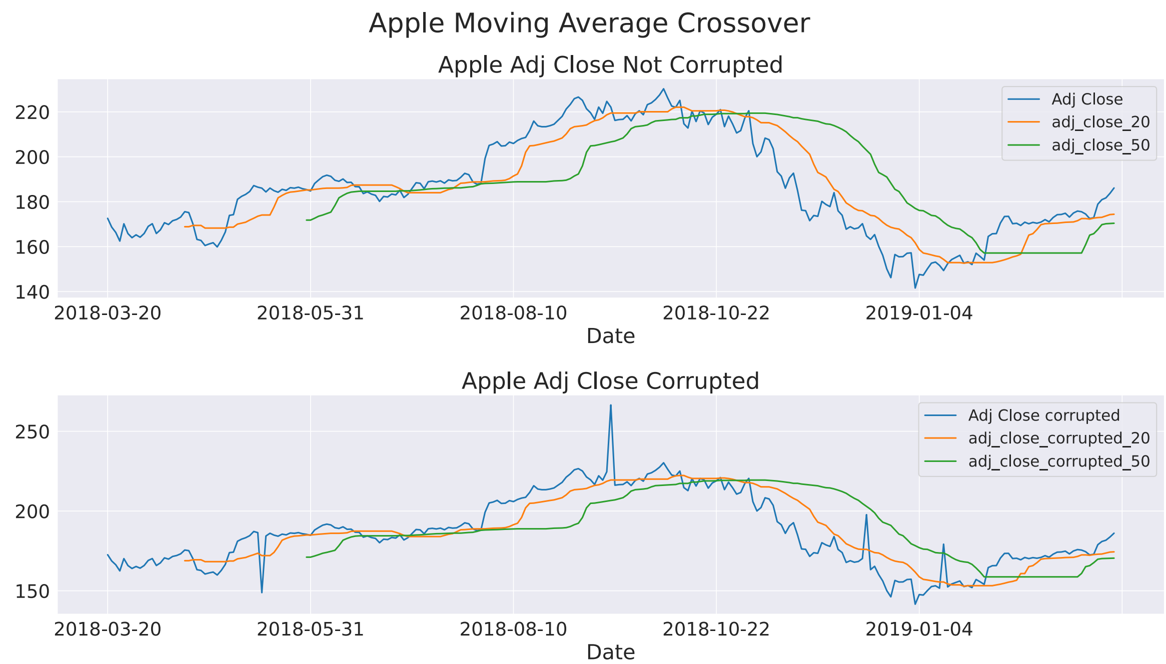


Figure 55: Apple Moving Median Crossover

Stock	Series	Buy dates
Apple	Not Corrupted	2018-07-24, 2019-02-12
	Corrupted	2018-07-25, 2019-02-12
JPM	Not Corrupted	2018-07-27, 2019-02-01
	Corrupted	2018-05-31, 2018-06-12, 2018-07-27, 2019-02-01
IBM	Not Corrupted	2018-07-25, 2019-01-24
	Corrupted	2018-07-25, 2019-01-23
DJI	Not Corrupted	2018-07-26, 2019-01-31
	Corrupted	2018-07-24, 2018-07-26, 2019-01-31

Table 11: Stock Moving Median Crossover Buy Dates

Stock	Series	Sell dates
Apple	Not Corrupted	2018-07-11, 2018-10-29
	Corrupted	2018-07-10, 2018-10-26
JPM	Not Corrupted	2018-06-12, 2018-09-27
	Corrupted	2018-06-07, 2018-06-14, 2018-10-12
IBM	Not Corrupted	2018-10-19
	Corrupted	2018-10-18
DJI	Not Corrupted	2018-07-10, 2018-10-23
	Corrupted	2018-07-12, 2018-07-25, 2018-10-22

Table 12: Stock Moving Median Crossover Sell Dates

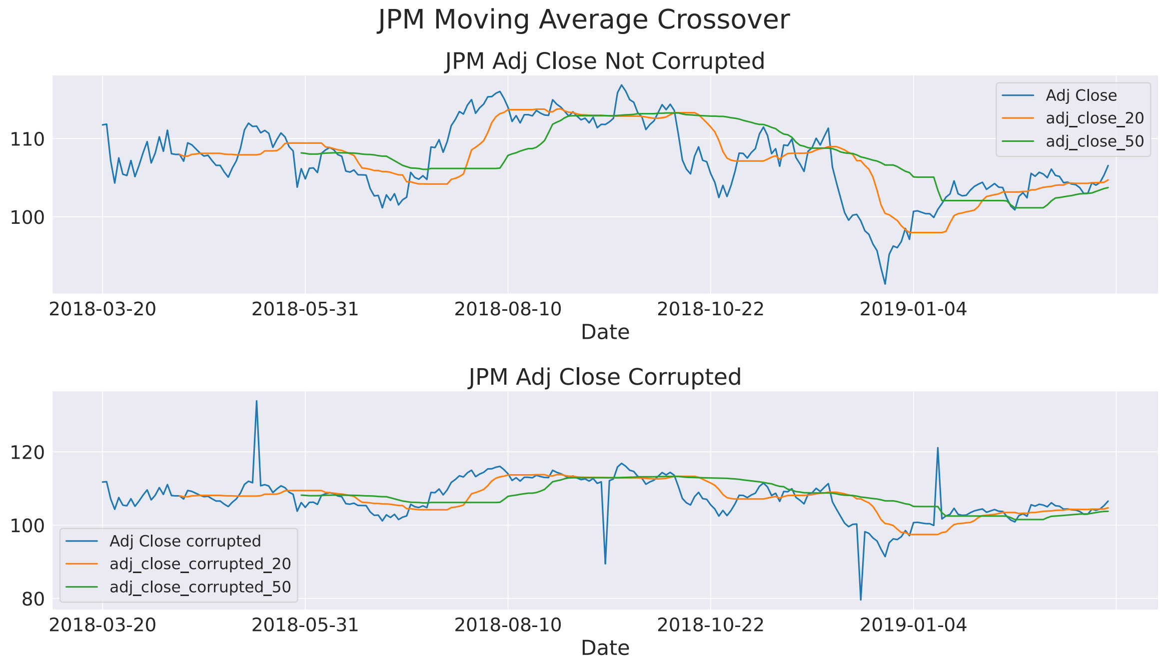


Figure 56: JPM Moving Median Crossover

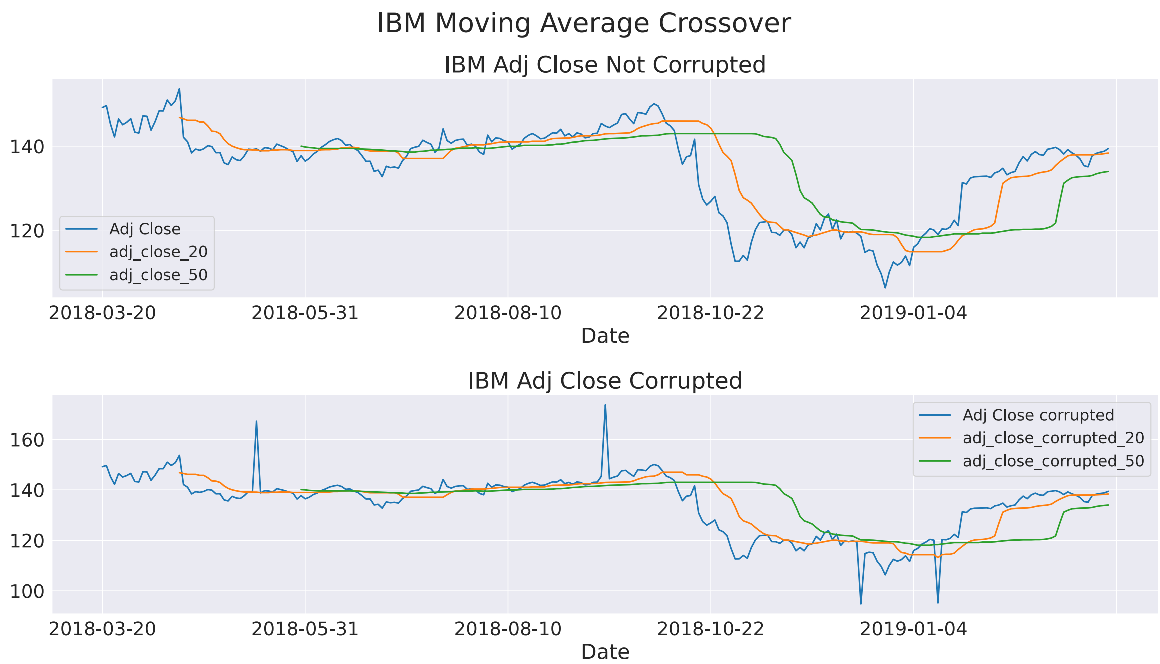


Figure 57: IBM Moving Median Crossover

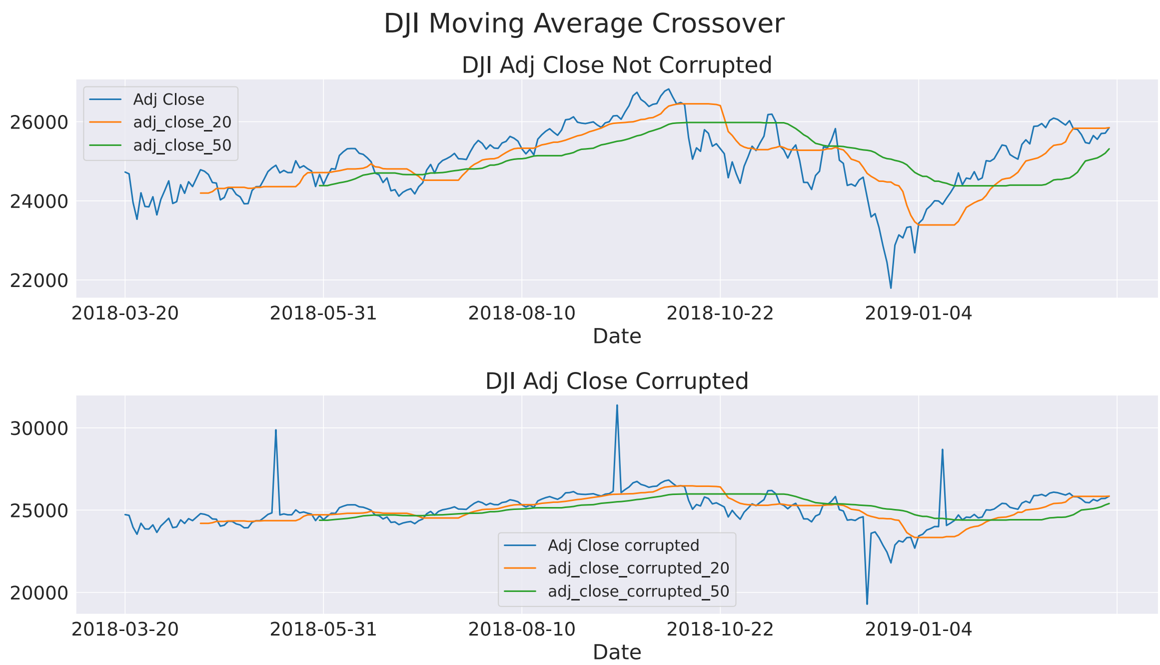


Figure 58: DJI Moving Median Crossover

Section 5: Graphs in Finance

5.1

The 10 chosen assets are stocks that have experienced the largest percentage change in market cap over the time period and on a per-GICS Sector basis. Mathematically:

$$\boldsymbol{x} = \max_{1 \leq i \leq 10} (f(x_1), ..., f(x_n))$$

$$f(x_i) = abs(x_{i,T} - x_{i,0})/x_{i,0}$$

subject to $x_i \in \boldsymbol{x}$ being from a unique GICS Sector

where $x_i$is stock i and T is the final stock value at the time period.

This criteria was chosen for several reasons.

Firstly, it allows for a comparison of growth rates among companies of different sizes. By dividing the change in market cap by the initial value, we can calculate the percentage change in market cap over the time period.

Secondly, it can help identify companies that have experienced the most significant growth over a given time period. For example, a company with a small initial market cap that experiences a large percentage increase in market cap may be a more attractive investment opportunity than a larger company with a smaller percentage increase.

Thirdly, this metric can help identify companies that have outperformed the broader market over a given time period. By comparing the percentage change in market cap to the percentage change in a market benchmark, such as the S&P 500, we can identify companies that have delivered stronger returns than the market as a whole.

The motivation for selecting on a per GICS Sector basis is to potentially unveil unknown or hard to find patterns between sectors as opposed to within sectors. In addition, the top growth rate stocks over the time period are from a technology perspective, so limiting this to just one may be more useful.

Figure 59 shows the descriptive statistics of the log returns of the chosen stocks using the mentioned criteria.

	BHF	ABMD	AMD	NFLX	AMZN	CPRT	MKC	NEM	AWK	KMI
count	487.000000	487.000000	487.000000	487.000000	487.000000	487.000000	487.000000	487.000000	487.000000	487.000000
mean	0.019778	0.004445	0.003970	0.003451	0.003250	0.001972	0.001299	0.001197	0.001069	-0.002102
std	0.477977	0.039731	0.077280	0.030323	0.027408	0.027718	0.013352	0.037552	0.016343	0.024573
min	-1.609438	-0.213226	-0.190518	-0.110780	-0.081424	-0.143712	-0.053604	-0.268332	-0.055000	-0.194278
25%	-0.010273	-0.008490	-0.017251	-0.011537	-0.006643	-0.006197	-0.005734	-0.010209	-0.005480	-0.010218
50%	0.000000	0.003016	0.001724	0.001415	0.001904	0.001775	0.001028	0.000000	0.001096	-0.000722
75%	0.007481	0.017042	0.021935	0.017899	0.012248	0.008614	0.007825	0.009657	0.007639	0.007719
max	9.385973	0.427730	1.480847	0.214442	0.336310	0.488318	0.103238	0.594726	0.251259	0.245928

Figure 59: Chosen Stocks Log Returns Description

5.2

The correlation matrix and correlation graph of the log returns of the stocks are shown in Figures 60 and 61 respectively.

The role of the correlation matrix is to show how the returns of each stock are related to the returns of every other stock in the portfolio. The topology of the graph would show the relationships between the stocks, with each stock represented as a node and the strength of the correlation between two stocks represented as an edge between the corresponding nodes.

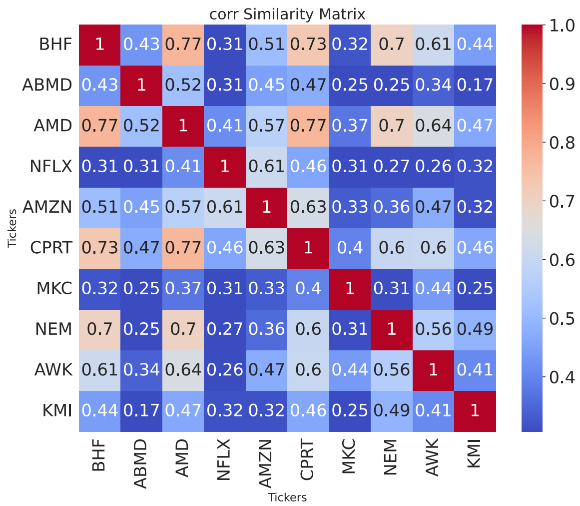


Figure 60: Chosen Stocks Log Returns Correlation Heatmap

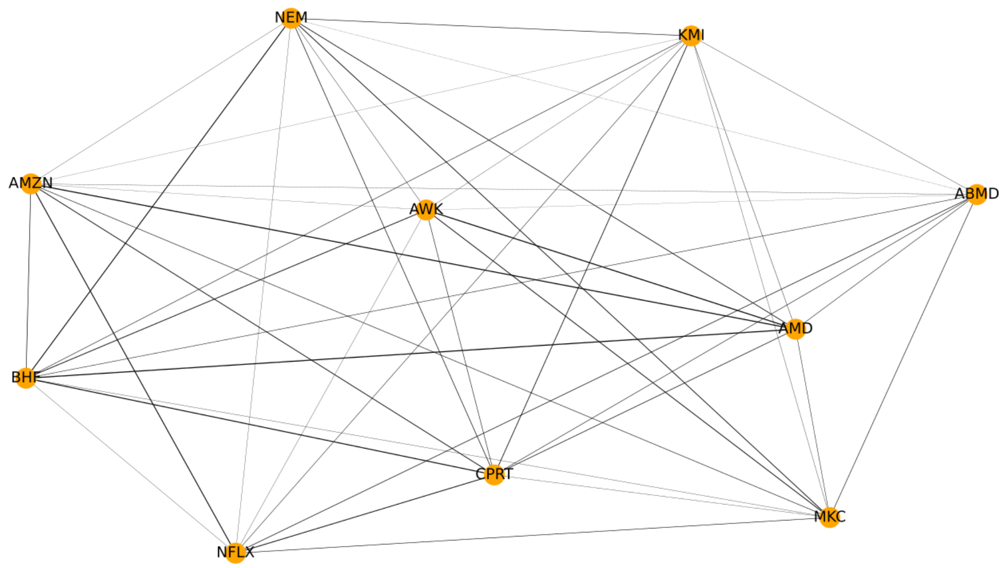


Figure 61: Chosen Stocks Log Returns Correlation Graph

5.3

The results show some expected patterns between the chosen stocks. The minimum correlations seen are 0.17, while the maximum is 0.77. Generally, there is relatively significant positive correlations throughout the chosen stocks.

Notable correlations are between AMD (Information Technology) and BHF (Financials) at 0.77. This is interesting as there is no obvious reason for a correlation between these companies. AMD and CPRT also show a high correlation of 0.77, however this could be due to AMD being a microchip company and CPRT being in the industrials sector, where microchips are used heavily. Another explainable correlation is between NFLX and AMZN. NFLX has low relative correlations with the other stocks except AMZN, at 0.61. This is explainable as NFLX and AMZN have large online software presences as part of their business.

The topology of the graph is dictated by the nature of the data, as the correlations between stocks are determined by the underlying factors that drive their prices and therefore the log-returns. For example, stocks in the same industry or sector are likely to be more highly correlated than stocks in different industries or sectors. Similarly, macroeconomic factors such as interest rates, inflation, and economic growth can impact the correlations between stocks.

The re-ordering of graph vertices will only affect the visualization of the graph and it would not affect the underlying correlations between stocks.

Whether the re-ordering of the time series will affect the results depends on the type of re-ordering. If the time-series is reversed, i.e. the last value becomes the first, 2nd last value becomes the 2nd, etc. then this will not affect the results. This is because the correlation values will remain the same. However, if the re-ordering is random between the time-series, then this would affect the results as time-series datasets are not independent and identically distributed.

5.4

2 other distance metrics were analysed: euclidean and Dynamic Time Warping (DTW) and are illustrated in Figures 62 and 64 respectively.

The euclidean distance metric measures the straight-line distance between two points in a multi-dimensional space. Thus, a smaller value represents a higher degree of similarity.

DTW compares two time series by warping one of the series in the time domain to match the other series, while minimizing the difference between the two series. It is therefore non-linear and can be interpreted as a distance metric. In the context of stocks, DTW is particularly useful when comparing time series data that may be subject to different rates of change over time. For example, if two stocks have similar log-return movements over a given time period, but one stock experiences more volatility than the other, DTW can be used to align the two time series and compare them in a meaningful way.

The DTW heatmap shows that BHF has a roughly 3x distance value to the other stocks than any other. This is reflected by the high BHF edges weightings (thicker lines) in the DTW graph to the other nodes. In addition, AMD exhibits a large relative distance for all other stocks, being 12 or 13 in value compared to 9 for other stocks. These patterns were not seen in the correlation similarity heatmap, but are seen in the euclidean. This is likely due to DTW capturing non-linear relationships in the data compared to correlation, which is linear.

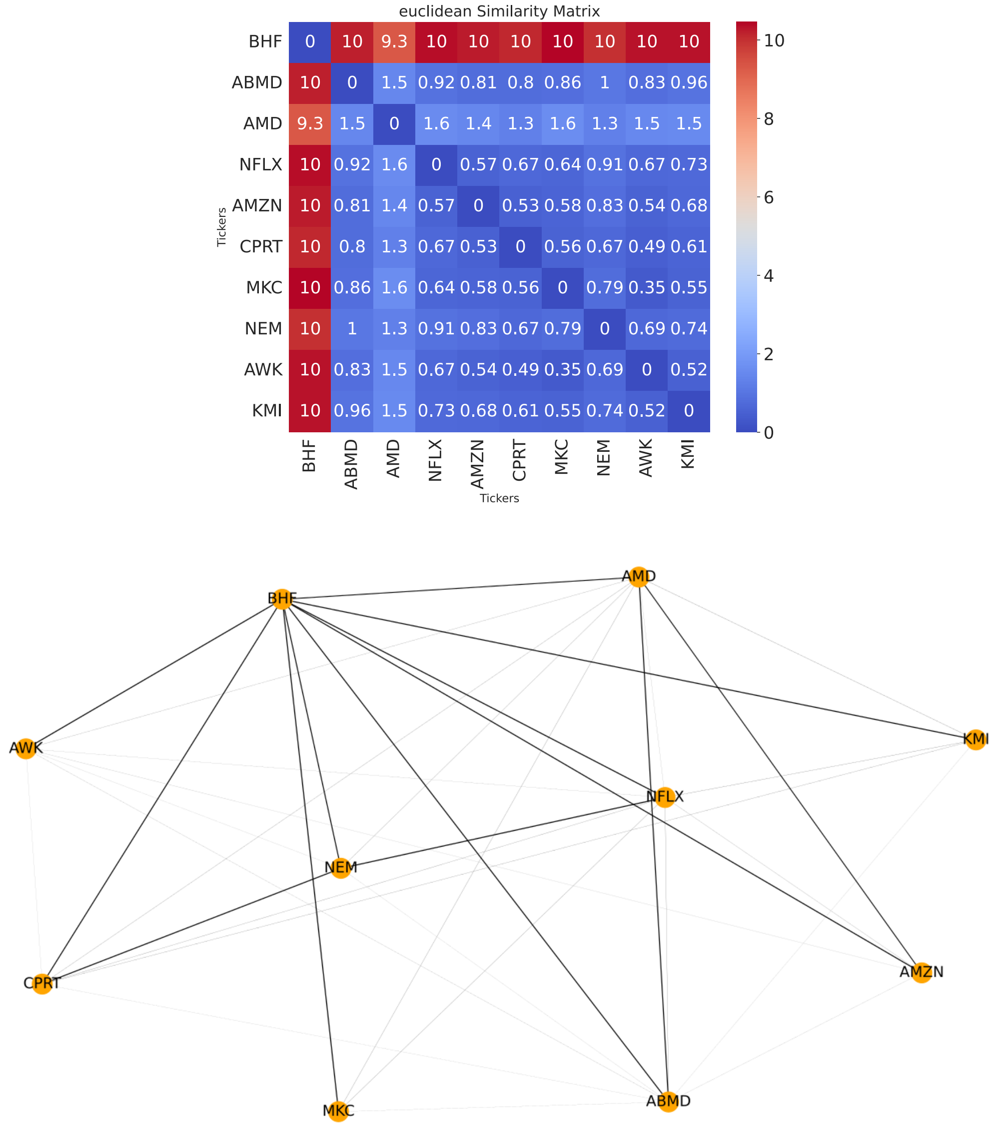


Figure 62: Chosen Stocks Log Returns Euclidean Heatmap and Graph

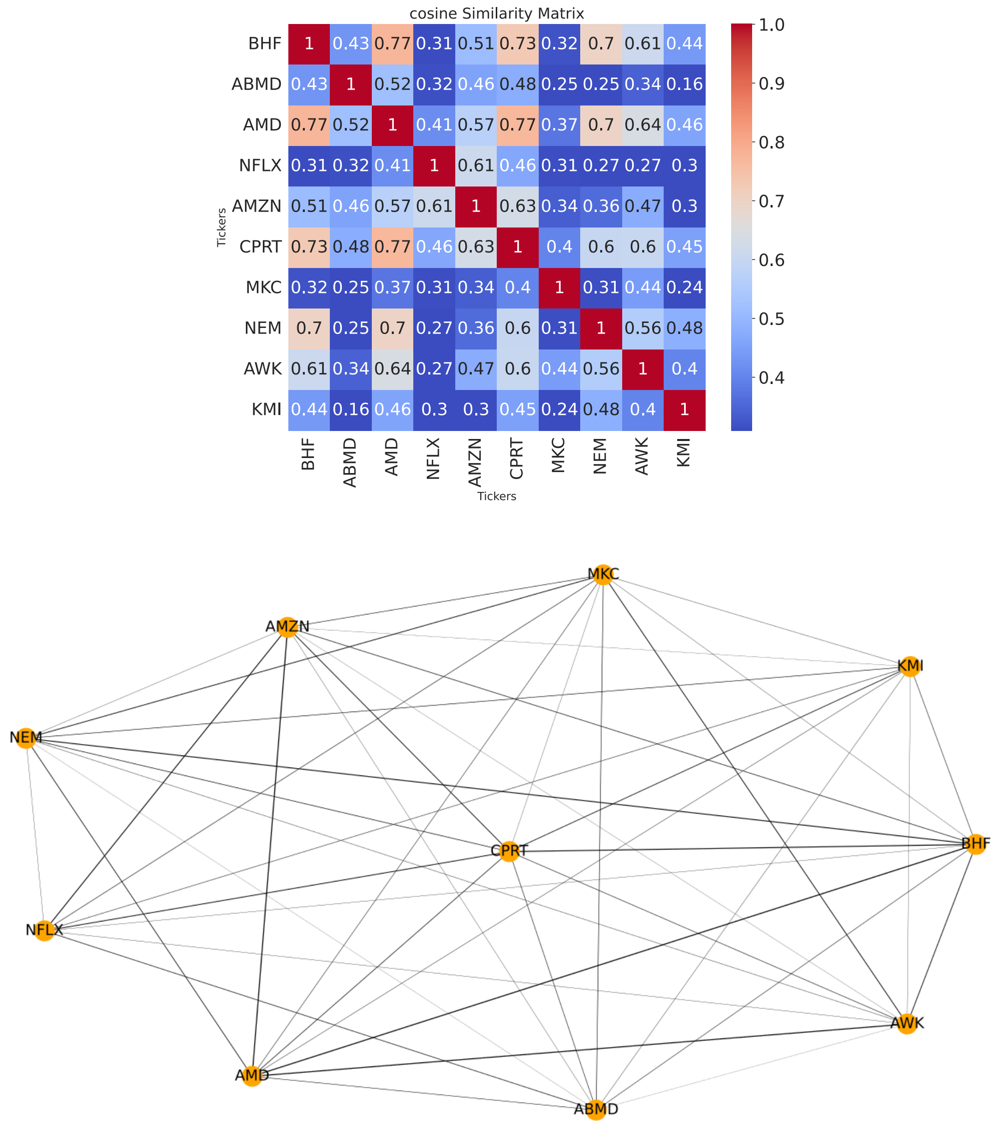


Figure 63: Chosen Stocks Log Returns Cosine Heatmap and Graph

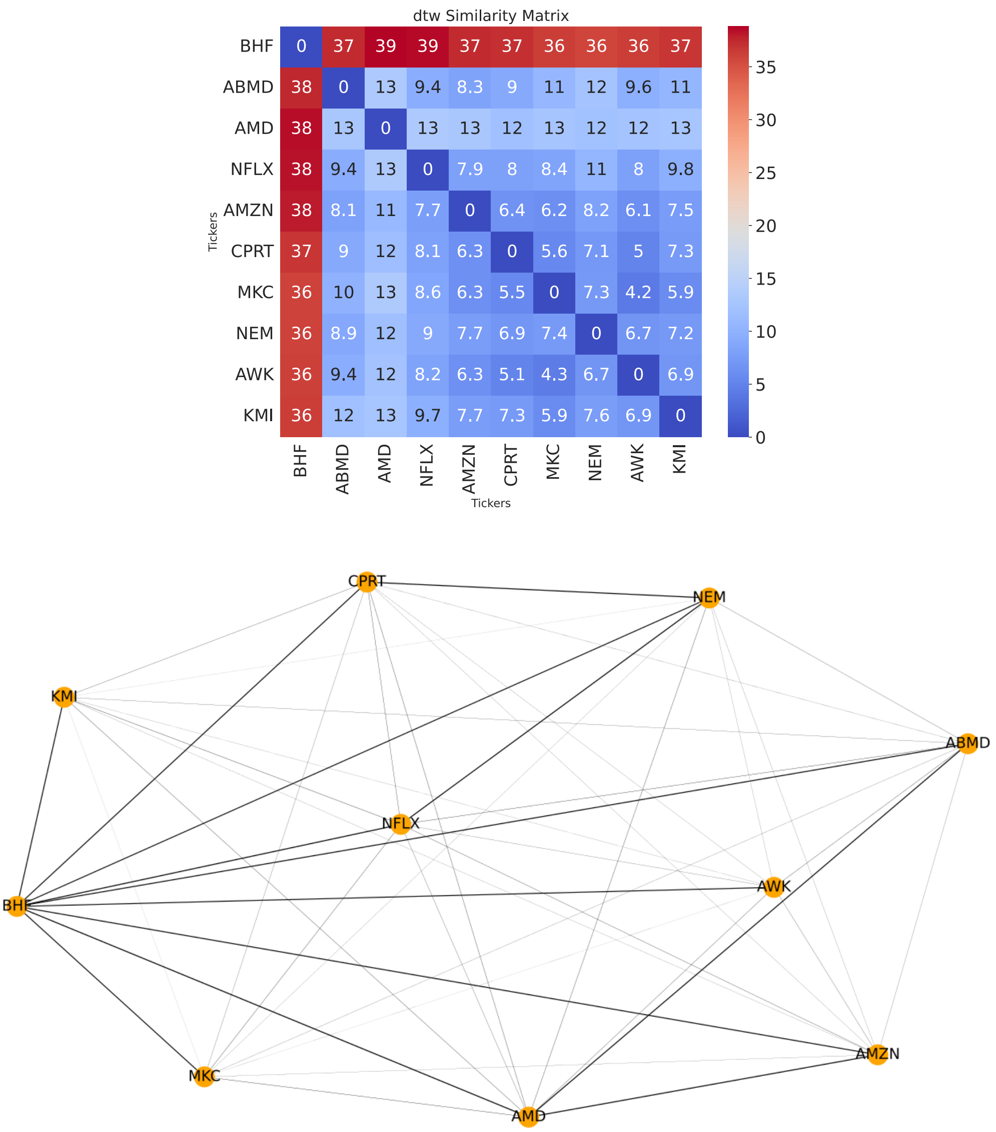


Figure 64: Chosen Stocks Log Returns DTW Heatmap and Graph

5.5

Firstly, assume the same criterion is used for choosing the 10 assets as in Section 5.1.

Cosine and correlation similarity may not be as appropriate for comparing raw price data, as the magnitudes of the data points can be much larger and more variable. In this case, other distance metrics such as Euclidean distance or DTW distance will be more appropriate. This is especially the case for S&P 500 stock data between 2015 and 2019, where there are general trends and seasonality in the data. This is shown by Figure 65 where many of the values are highly correlated between 0.7 and 1.0, while the KMI stock shows a large negative correlation between -0.8 and -1.

The correlation topology of the graph will still no longer be dictated by the nature of the graph. This is because almost all lines will show large values as a result of the positive trend of the stocks, masking the true nature of the data. This was not the case for log-returns as the differencing acted to remove the trend. However, the re-ordering of the graph vertices would not affect the results as it is permutation equivariant and so would only change the visualisation, similar to log-returns.

The affect of re-ordering the time series will depend on the type of re-ordering as discussed in Section 5.3: If the time-series is simply reversed, the results should not be affected with the distance metrics used. On the other hand, if the re-ordering is random, then this will affect the results as time-series is not independent.

Figure 66 shows the distance metric graphs chosen from Section 5.4 using raw prices. Similarly to when using log returns, the DTW and Euclidean graphs show similar properties. However, with raw prices, the AMZN stock has the highest edge weights to other stocks, compared to BHF for log returns. The cosine graph shows the same properties as the correlation graph.

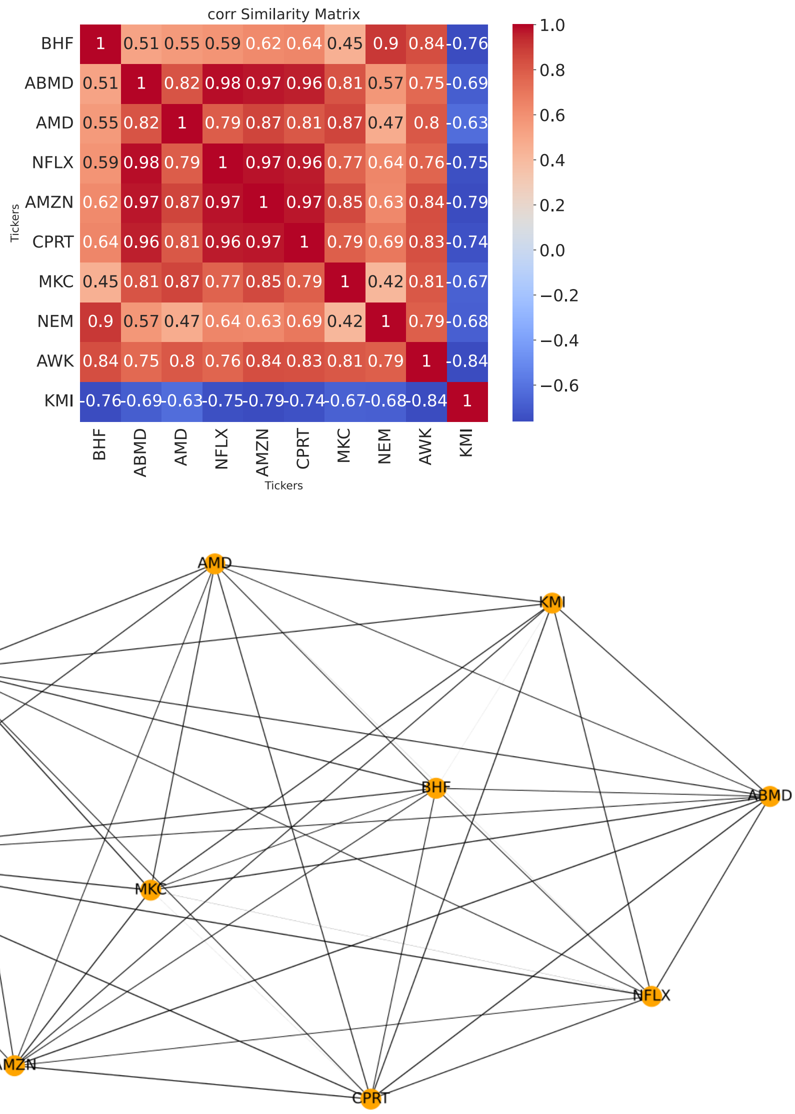


Figure 65: Chosen Stocks Raw Prices Correlation Heatmap and Graph

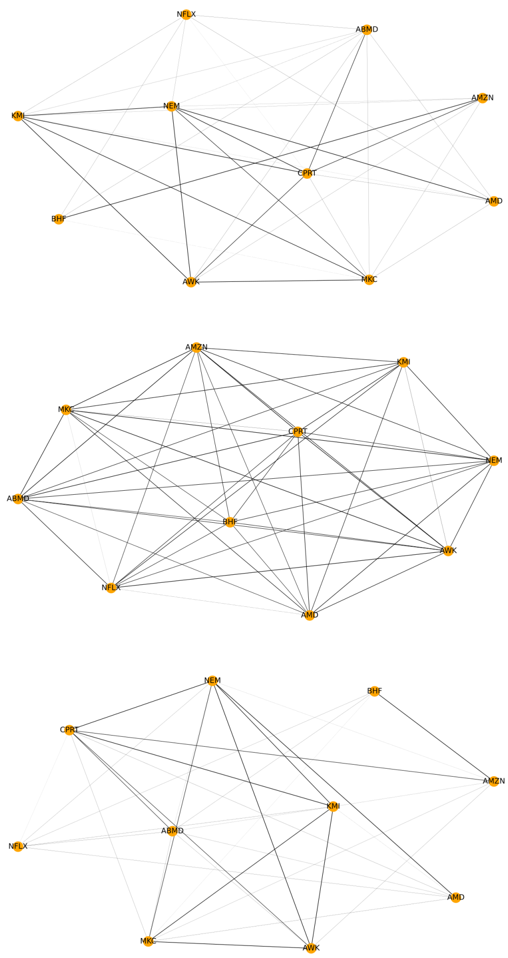


Figure 66: Chosen Stocks Raw Prices Distance Metrics Graphs. Top: Euclidean. Middle: Cosine. Bottom: DTW

A Appendix A: Robust Estimators Custom Functions

```
def custom_median(data):
    start_time = time.perf_counter()
    # sort 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.
    value = 0
    if len(data) % 2 == 0:
        value =  data[int(len(data) / 2)]
    else:
        median_low  = data[int(len(data) / 2) - 1]
        median_high = data[int(len(data) / 2)]
        value = (median_low + median_high) / 2

    end_time = time.perf_counter()
    execution_time = end_time - start_time
    print(f"Custom median execution time is: {execution_time}")
    return value

def custom_iqr(data):
    start_time = time.perf_counter()
    # sort 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.
    value = 0
    if len(data) % 2 == 0:
        quart1 = data[int(len(data) / 4)]
        quart3 = data[int(len(data) * 3 / 4)]
        value = 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)]

        value = (quart3_high + quart3_low) / 2 - (quart1_high + quart1_low) / 2

    end_time = time.perf_counter()
    execution_time = end_time - start_time
    print(f"Custom IQR execution time is: {execution_time}")
    return value
```

Figure 67: Robust Custom Estimators Code Part 1

```
def custom_mad(data):
    start_time = time.perf_counter()
    dev = abs(data - custom_median(data))

    value = custom_median(dev)

    end_time = time.perf_counter()
    execution_time = end_time - start_time
    print(f"Custom mad execution time is: {execution_time}")
    return value
```

Figure 68: Robust Custom Estimators Code Part 2