Signal Processing and Machine Learning for Finance 2021-2022

Final Report

Matthew Scott
01510365

Electrical and Electronic Engineering
Imperial College London
11/04/2022

1 Regression Methods

Python has a diverse and well-constructed toolbox for signal processing, making it an excellent choice of programming language. Regression is a technique that is applied for 'fitting' data to a function to find the input and output relationship of a dataset. The use of the SPX index is discussed below:

1.1 Processing stock price data in Python

1.1.1

The price of the SPX index fluctuates over time. Its development from 1930 to 2009 can be seen in Figure 1

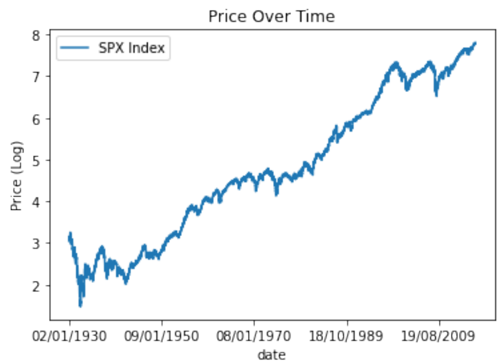

Figure 1: SPX index prices over time

1.1.2

To assess the stationarity of such data, first- and second-order statistics in the form of *mean* and *variance* are often used. Whilst any (logical) time-window can be used, the results of applying a 252-day rolling window to both the raw and *log* prices of the time series are shown in Figures 2 and 3 to find the rolling mean and rolling variance respectively.

Noticeably, Figure 3 illustrates a rapid fluctuation in variance with a clear trend in mean in Figure 2 that shows an overall increase. For this reason, both time series are clearly highly time-dependent and are hence **not** stationary processes.

1.1.3

Further investigating the first- and second-order statistics of the SPX returns (using the same 252 day window) yields Figures 4 and 5 which show the rolling means and variances for simple and log returns respectively.

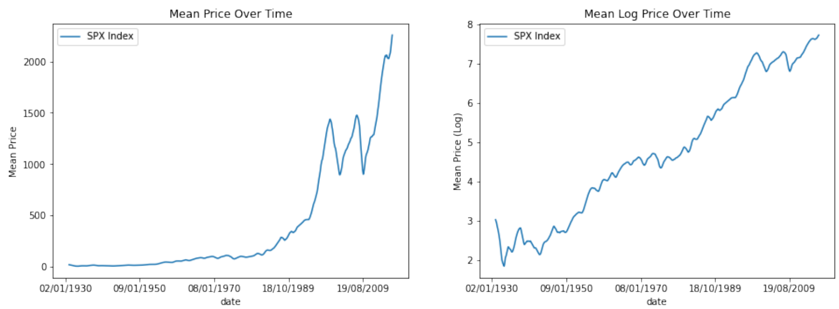

Figure 2: Mean SPX prices over time

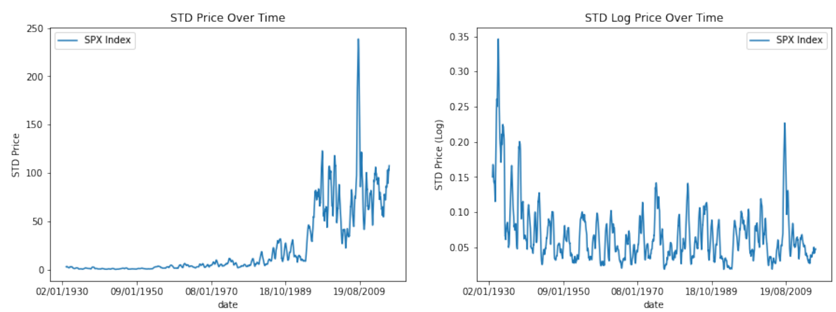

Figure 3: Standard deviation of SPX prices over time

In both cases, it can be seen that there are no visible trends or seasonal components in the mean. For this reason, the returns for both time series **are** stationary.

1.1.4

The choice between log and simple returns varies based on their progressive properties. Simple returns use multiplicative (geometric) progression whereas log returns use additive (arithmetic) progression. This means that the returns between any two time points, $r(t_x, t_y)$, are time consistent, as shown in Equation 1

$$r(t, t-2) = r(t, t-1) + r(t-1, t-2) \tag{1}$$

Furthermore, financial returns are assumed to be distributed *log-normally*, meaning that the logarithm of the returns is distributed normally. For this reason, the sum of log-returns also yields a normal distribution, which is as to be expected.

Using the Jarque-Bera test allows for a test of the null hypothesis that a dataset is normally distributed (Gaussian). Applying this test to the SPX time

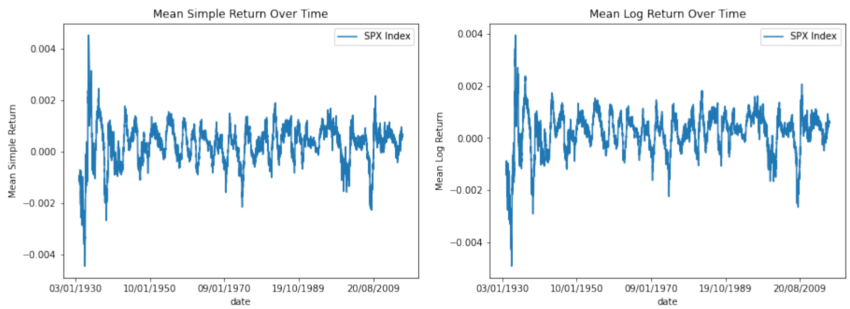

Figure 4: Mean SPX return over time

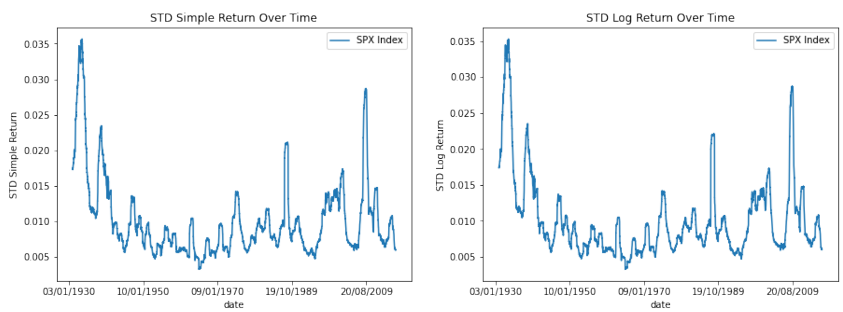

Figure 5: Standard deviation of SPX return over time

series yields a score of 0, meaning that the null hypothesis cannot be rejected at the 5% significance level. This means that the data *is* Gaussian, which supports the Central Limit Theorem (CLT), as such a large dataset will ultimately tend to a normal distribution.

1.1.5

Using the example of a £1 stock, the simple and log returns can be mathematically seen in Table 6

Price	£1	£2	£1
Simple Return	N/A	+100%	-50%
Log Return	N/A	+69.3%	-69.3%

Figure 6: Simple and log returns for a simple stock

The main concept to yield from log returns is that, when a stock returns to a previous price, the total percentage change is 0%. This is very convenient for time series analysis.

1.1.6

Log returns are unsuitable when considering a weighted portfolio. This is because they are not a linear function of the portfolio weights, so the total return from the portfolio cannot simply be calculated with $\mathbf{w^T r}$.

1.2 ARMA vs ARIMA Models for Financial Applications

1.2.1

The closing price of the S&P 500 gives an opportunity to use regression to attempt to model the fluctuations seen in Figure 7. The second order statistics can subsequently be seen in Figure 8, which shows a clear trend in the rolling mean. For this reason, the data is non-stationary, so requires *differencing* in order to achieve stationarity. An **ARIMA** model is hence best suited to this problem.

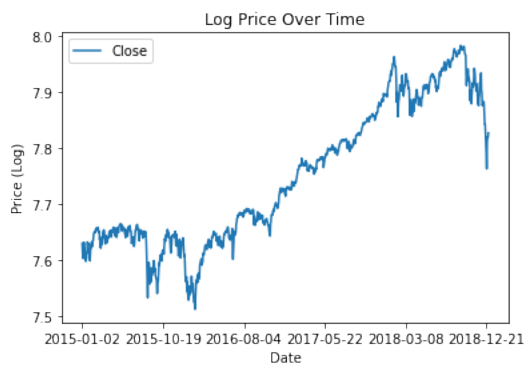

Figure 7: Log prices of S&P 500

1.2.2

Applying an ARMA(1,0) to the dataset yields the prediction in Figure 9.

It is clear that the ARMA(1,0) (or simply AR(1)) model is effective in fitting the dataset, showing very little error in capturing the overall trend and noise of the time series.

This analysis can be potentially useful, as the AR model can also be used for forecasting. This means that predictions can be made about the future of the stock market with the caveat of preventing long-term forecasting. The AR model is unable to capture random future events (such as a crash), as it must rely on past data alone. For this reason, the model cannot be used to forecast far into the future, but can be used for short predictions.

This model takes the form:

$$x[t] = -0.00816978 * x[t-1] + \epsilon_t \tag{2}$$

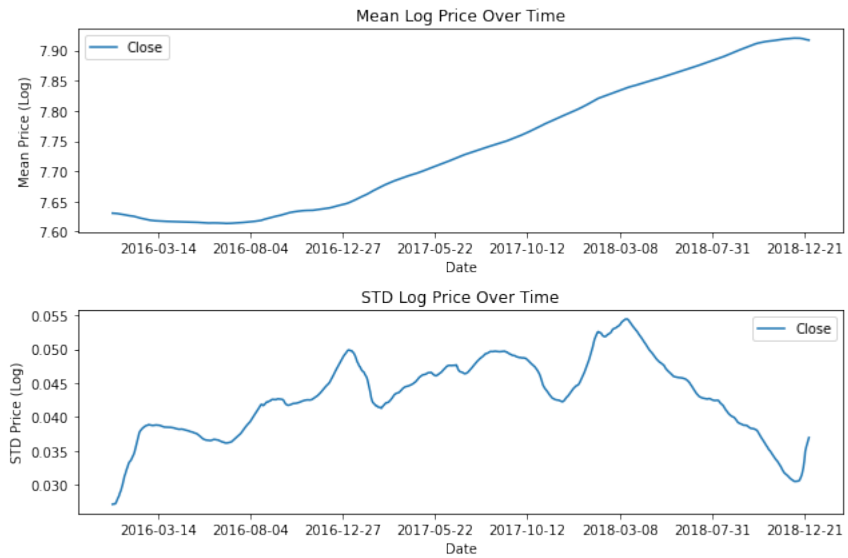

Figure 8: Mean and standard deviation of S&P 500 prices

yielding a MAE of 0.598%

1.2.3

By instead introducing a non-seasonal difference of 1 to the ARMA(1,0) model, an ARIMA(1,1,0) model is formed. Applying this regressor yields the prediction in Figure 10. This yields near identical results to the ARMA(1,0) analysis, showing that the effect of differencing was not necessary.

Ultimately, the ARIMA(1,1,0) analysis is more meaningful, as it better reflects the steps that need to be taken to produce a reliable model. Financial data is dependent on inflation over time, so will likely show trend and seasonal components. For this reason, differencing should generally be applied to make a series stationary.

This model also takes the form:

$$x[t] = -0.00816978 * x[t-1] + \epsilon_t \tag{3}$$

however yields an increased MAE of 1.343%

1.2.4

The unstable variance of the series means that a log transform must be applied to the series data. This is because, by applying the transformation, the error term in price from each day $x[t]-x[t-1]$ that the auto regressor models becomes:

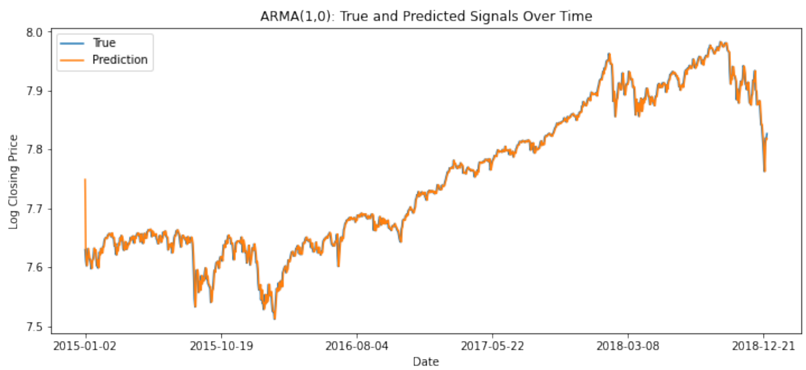

Figure 9: True and predicted S&P signals using ARMA(1,0) model

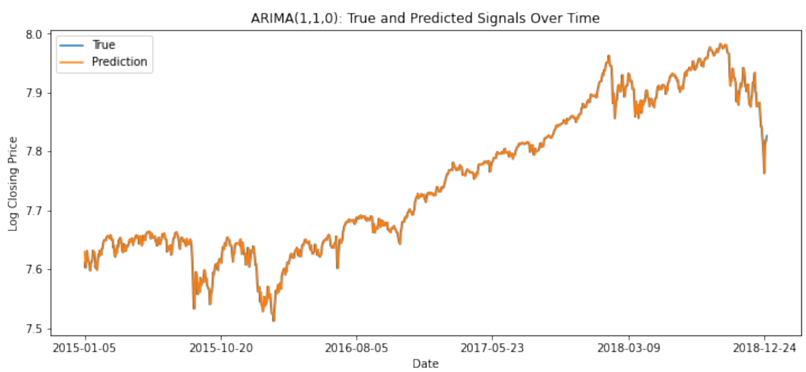

Figure 10: True and predicted S&P signals using ARMA(1,1,0) model

$$\epsilon_t = log(x[t]) - log(x[t-1]) = log(\frac{x[t]}{x[t-1]}) \tag{4}$$

which reduces the variance in the autoregressive model error.

1.3 Vector Autoregressive (VAR) Models

VAR models are a multivariate extension of the AR processes, given by:

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1\mathbf{y}_{t-1} + \mathbf{A}_2\mathbf{y}_{t-2} + ... + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{e}_t \tag{5}$$

1.3.1

Defining a matrix $\mathbf{B} = [\mathbf{c} \quad \mathbf{A}_1 \quad \mathbf{A}_2 ... \mathbf{A}_p\ ]$ allows for factoring the equation as:

$$\mathbf{y}_t = \mathbf{B} \cdot [1 \quad \mathbf{y}_{t-1} \quad \mathbf{y}_{t-2} ... \mathbf{y}_{t-p}]^T + \mathbf{e}_t \tag{6}$$

which allows for each $\mathbf{y}_t$ to be packed into a matrix:

$$\mathbf{Y} = [\mathbf{y}_p \quad \mathbf{y}_{p+1} \quad \mathbf{y}_{p+2} ... \mathbf{y}_{p+T}] \tag{7}$$

and each factored column vector:

$$\mathbf{Z} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{y}_{p-1} & \mathbf{y}_{p-2} & \cdots & \mathbf{y}_{p+T-1} \\ \mathbf{y}_{p-2} & \mathbf{y}_{p-3} & \cdots & \mathbf{y}_{p+T-2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_T \end{bmatrix} \tag{8}$$

leaving the vector of error vectors:

$$\mathbf{U} = [\mathbf{e}_p \quad \mathbf{e}_{p+1} \quad \mathbf{e}_{p+2} ... \mathbf{e}_{p+T}] \tag{9}$$

to yield the final equation:

$$\mathbf{Y} = \mathbf{B}\mathbf{Z} + \mathbf{U} \tag{10}$$

1.3.2

The optimality conditions on $\mathbf{B}$ can be derived by minimising the cost function of the error vector, taking this function as $\mathbf{U}^{\mathbf{T}}\mathbf{U}$, such that:

$$\mathbf{B}^* = \min_{\mathbf{B}} (\mathbf{Y} - \mathbf{B}\mathbf{Z})^T (\mathbf{Y} - \mathbf{B}\mathbf{Z}) \tag{11}$$

which poses a standard optimisation problem that can be solved through matrix calculus, where:

$$\mathbf{B}^* = \frac{\partial}{\partial \mathbf{B}} (\mathbf{Y} - \mathbf{B}\mathbf{Z})^T (\mathbf{Y} - \mathbf{B}\mathbf{Z}) \tag{12}$$

which can be expanded through the symmetry of the matrices as:

$$\mathbf{B}^* = \frac{\partial}{\partial \mathbf{B}} \mathbf{Y}^{\mathbf{T}}\mathbf{Y} - 2\mathbf{Y}^{\mathbf{T}}\mathbf{B}\mathbf{Z} + (\mathbf{B}\mathbf{Z})^{\mathbf{T}}(\mathbf{B}\mathbf{Z}) \tag{13}$$

to yield:

$$-2\mathbf{Y}^{\mathbf{T}}\mathbf{Z} + 2(\mathbf{B}\mathbf{Z})\mathbf{Z}^{\mathbf{T}} = 0 \tag{14}$$

where, after rearranging:

$$\mathbf{B}^* = \mathbf{Y}^{\mathbf{T}}\mathbf{Z}(\mathbf{Z}\mathbf{Z}^{\mathbf{T}})^{-1} \tag{15}$$

1.3.3

Defining a VAR(1) process as:

$$\mathbf{y}_t = \mathbf{A}\mathbf{y}_{t-1} + \mathbf{e}_t \tag{16}$$

it can be continued as:

$$\mathbf{y}_{t-1} = \mathbf{A}\mathbf{y}_{t-2} + \mathbf{e}_{t-1} \tag{17}$$

and hence defines a recursive relationship with:

$$\mathbf{y}_t = \mathbf{A}(\mathbf{A}\mathbf{y}_{t-2} + \mathbf{e}_{t-1}) + \mathbf{e}_t \tag{18}$$

which can be reduced to:

$$\mathbf{y}_t = \mathbf{A}^n\mathbf{y}_{t-n} + \sum_{i=0}^{n-1}\mathbf{A}^i\mathbf{e}_{t-i} \tag{19}$$

where, clearly, the equation will tend to ∞ if $\mathbf{A}^n > 1$. Performing SVD on $\mathbf{A}^n$ yields a product of the form $\mathbf{U\Lambda^nV^T}$, with:

$$\mathbf{\Lambda}^n = \begin{bmatrix} \lambda_1^n & 0 & \cdots & 0 \\ 0 & \lambda_2^n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p^n \end{bmatrix} \tag{20}$$

demonstrating that all eigenvalues must have an absolute value less than 1 to avoid divergence.

1.3.4

This VAR model can be applied to a set of stocks in a portfolio to identify if it is stable.

The stocks in Figure 11 are detrended and subsequently fit to a VAR(1) model. This yields a matrix $\mathbf{A}$ with eigenvalues:

$$\begin{bmatrix} 0.72609393 & 0.72609393 & 1.00635964 & 0.86051894 & 0.91144512 \end{bmatrix} \tag{21}$$

showing that this is a poor choice of portfolio, as the largest eigenvalue has an absolute value greater than 1, leading to instability.

1.3.5

This approach can also be applied by investigating a portfolio consisting of stocks in the same sector. By performing the same analysis as in Section 1.3.4, the following table is generated, listing the largest and smallest eigenvalues for the VAR matrix:

From this table, it is clear to see that the idea to create a portfolio by sector is a sound idea, apart from in the financial sector as this would lead to instability.

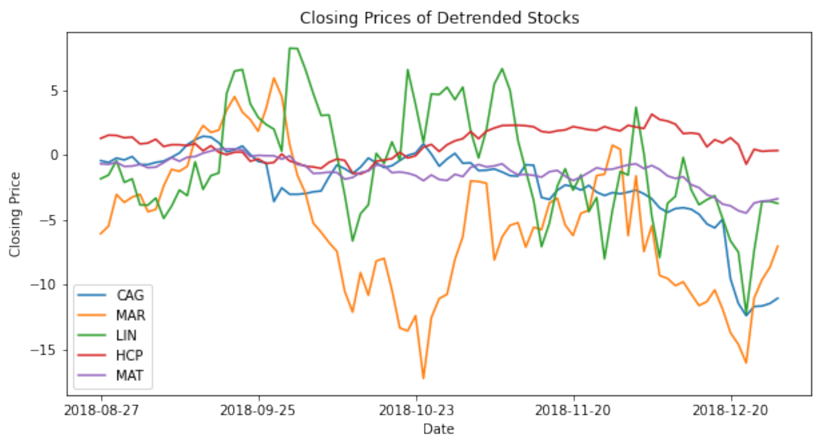

Figure 11: Closing prices of detrended stocks

Sector	Min	Max
Industrials	0.371246	0.991721
Health Care	0.383165	0.994153
Information Technology	0.374081	0.992738
Communication Services	0.752488	0.982263
Consumer Discretionary	0.447563	0.99065
Utilities	0.399328	0.985648
Financials	0.453489	1.00434
Materials	0.284215	0.991744
Real Estate	0.763563	0.982785
Consumer Staples	0.546458	0.991508
Energy	0.825707	0.985577

Figure 12: Min and max eigenvalues for portfolios of each sector

2 Bond Pricing

2.1 Examples of bond pricing

2.1.1

The returns, R on a principal investment, P are equal to:

$$R = P\left(1 + \frac{r}{n}\right)^n \tag{22}$$

The compounded return, r from an investment is hence expressed as:

$$r_n = n\left[\left(\frac{R}{P}\right)^{\frac{1}{n}} - 1\right] \tag{23}$$

which, when $P = 1000$ and $R = 1100$ marks the investment and return, yields:

(a) 10%, with annual compounding

(b) 9.76%, with semi-annual compounding

(c) 9.57%, with monthly compounding

With continuous compounding, however, the limit as $n \to \infty$ must be considered:

$$\lim_{n\to\infty}\left(1+\frac{r}{n}\right)^n = e^r \tag{24}$$

hence $1.1 = e^r$ and

(d) $ln(1.1) = 9.53\%$, with continuous compounding

2.1.2

15% interest with monthly compounding gives:

$$ROI = \left(1+\frac{0.15}{12}\right)^1 2 = 1.1607... \tag{25}$$

where, to achieve an equal ROI with continuous compounding:

$$ROI = e^r \implies r = 14.91\% \tag{26}$$

2.1.3

As before:

$$e^{0.12} = ROI = \left(1+\frac{r}{4}\right)^4 \implies r = 12.18\% \tag{27}$$

hence, the interest *accumulated* at each quarter, Q_n is:

$$Q_n = 10000 * \left(1+\frac{0.1218}{4}\right)^n - 10000 \tag{28}$$

giving payouts:

(Q1) 304.55

(Q2) 312.82

(Q3) 323.37

(Q4) 333.23

2.2 Forward Rates

(a) The 'additional' 9% is a reflection of the spot rate forecast 1 year in the future. Assuming that this forecast holds (such that there are no unforeseen developments in the market), there is no difference between a 2-year investment and two 1-year investments. Whereas the former offers security, the latter gives more flexibility to the investor, so both options are appealing depending on the financial situation.

(b) Assuming a no-arbitrage situation, all investment strategies should be equal to prevent "free money." These strategies are hence all equal mathematically, but differ based on the speed of return, where lower percentage returns offer faster rewards at the cost of security.

(c) Spot rate forecasts are always calculated based on current interest rates, and are therefore affected by interest rate inflation, so a 9% forward rate is disadvantageous in this regard, as it offers no robustness against this.

(d) No extra money has to be put in to change investment schemes

2.3 Duration of a coupon-bearing bond

Since each term in the summation is a (year * fraction of PV), the final row of the table yields all the information needed. Hence, the duration is:

$$0.0124 + 0.0236 + ... + 6.5377 = 6.76 years \tag{29}$$

The modified duration divides the existing duration by (1 + YTM):

$$D_M = \frac{6.7595}{1.05} = 6.44 years \tag{30}$$

Broadly, the Macaulay duration measures the weighted average time an investor must hold a bond until the present value of the bond's cash flows is equal to the amount paid for the bond, whereas the modified duration measures how much a change in the interest rates impacts the price of a bond.

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

2.4.1

The average market returns can be found by averaging the company returns at each time step. This produces Figure 13

2.4.2

The rolling β of a stock, i wrt. the market, m, is defined as:

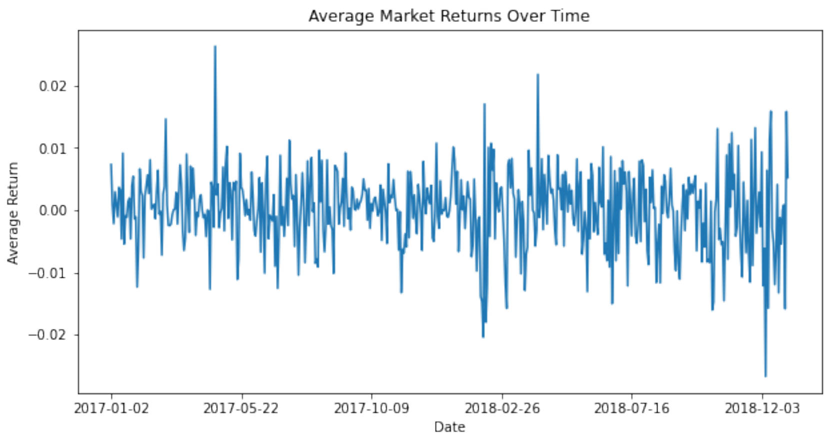

Figure 13: Estimated market returns per day

$$\beta_i = \frac{cov(R_i, R_m)}{var(R_m)} \tag{31}$$

Thus, by calculating the variance of the market during this window, as well as its covariance with each stock, the resulting plot in Figure 14 is generated. Visually, most stocks oscillate around the line $\beta = 1$, which demonstrates perfect correlation with the market. The large number of spikes in the plot imply that these stocks tend to be more volatile than the market. This is supported by the numerical data in Figure 15

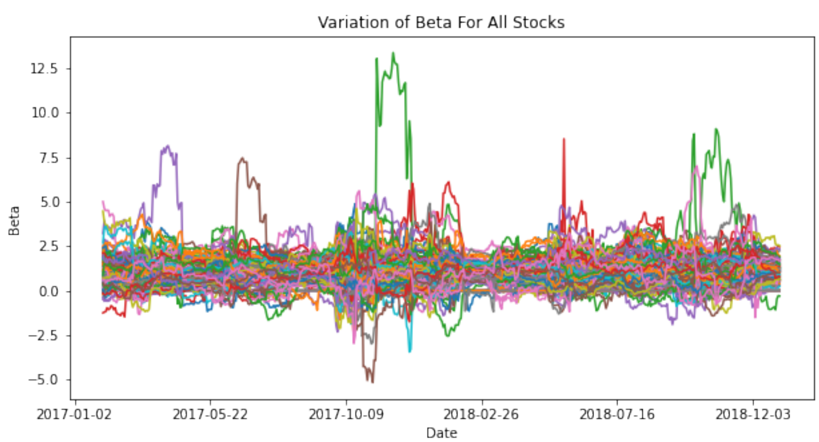

Figure 14: Rolling β for all stocks

Statistic	Value
Mean	1.00
STD	0.7743

Figure 15: Statistics of stock β data

2.4.3

The cap-weighted market return can be calculated such that, for each day:

$$R_{m,t} = \sum_i \frac{mcap_i \times ret_i}{\sum_i mcap_i} \tag{32}$$

Which, when applied to the dataset yields the plot in Figure 16. Notably, the weighting coefficient $\sum_i mcap_i$ creates what is termed the "market portfolio," where the return of each stock is weighted by the proportion of its market cap. This leads into the notion of the "one-fund theorem," where, assuming an arbitrage-free market with optimal trading, the optimal portfolio **is** the market portfolio.

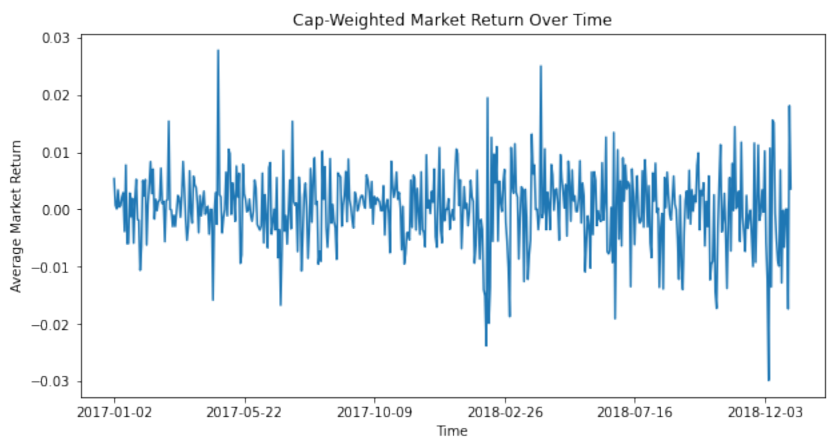

Figure 16: Cap-weighted market return

2.4.4

This cap weighted market return can be used to generate a cap weighted rolling β_m. The results of using this metric create the series in Figure 17, with statistics shown in Figure 18

Introducing cap-weighted returns generates the efficient (market) portfolio that sports a smaller mean β_m than with an equally weighted market portfolio. This means that stocks are generally less volatile than the overall market. Fur-

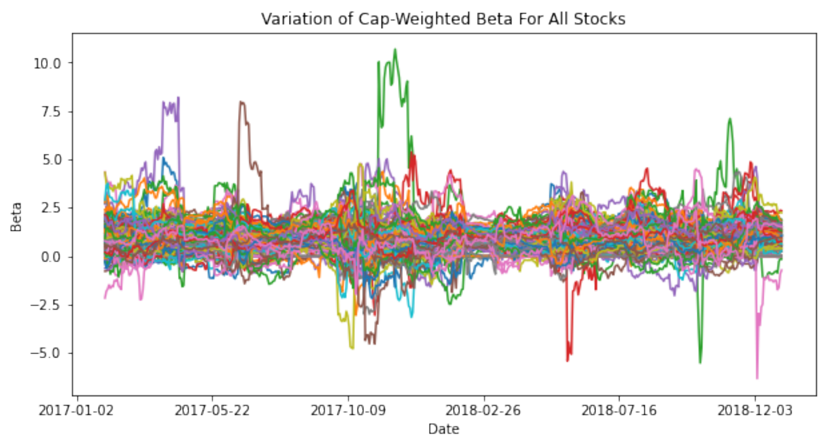

Figure 17: Cap weighted rolling β

Statistic	Value
Mean	0.907
STD	0.7162

Figure 18: Statistics of stock β_m data

thermore, there is less variation in the volatility of stocks, as the variance of β_m is again smaller than in an equally weighted portfolio.

2.4.5

(a) Arbitrage Pricing Theory (APT) is a pricing model that aims to represent the return of a risky asset as a function of "macroeconomic variables that capture systematic risk." For this implementation, for each day, the return of a stock i on day t, $r_{i,t}$ is:

$$r_{i,t} = \begin{bmatrix} 1 & b_{m_{i,t}} & b_{s_{i,t}} \end{bmatrix} \cdot \begin{bmatrix} a_t \\ R_{m_t} \\ R_{s_t} \end{bmatrix} + \epsilon_{i,t} \tag{33}$$

where each stock can be stacked into vector form to yield:

$$\begin{bmatrix} r_{1,t} \\ r_{2,t} \\ \vdots \\ r_{n,t} \end{bmatrix} = \begin{bmatrix} 1 & b_{m_{1,t}} & b_{s_{1,t}} \\ 1 & b_{m_{2,t}} & b_{s_{2,t}} \\ \vdots & \vdots & \vdots \\ 1 & b_{m_{n,t}} & b_{s_{n,t}} \end{bmatrix} \cdot \begin{bmatrix} a_t \\ R_{m_t} \\ R_{s_t} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \\ \vdots \\ \epsilon_{n_t} \end{bmatrix} \tag{34}$$

producing a matrix equation in the form:

$$\mathbf{r} = \mathbf{A}\mathbf{x} + \mathbf{e} \tag{35}$$

which is solved through the optimisation problem:

$$\mathbf{x}^* = \min_{\mathbf{x}} \ (\mathbf{r} - \mathbf{A}\mathbf{x})^{\mathbf{T}}(\mathbf{r} - \mathbf{A}\mathbf{x}) \tag{36}$$

to yield:

$$\mathbf{x}^* = (\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{r} \tag{37}$$

which can be plotted over time to yield the result in Figure 19.

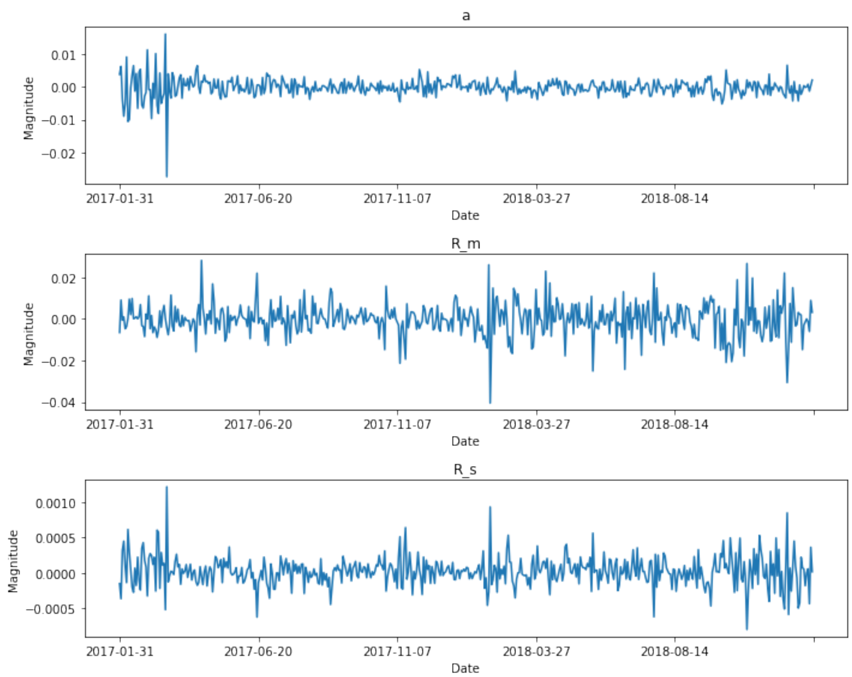

Figure 19: Magnitude of APT estimators

(b) The distributions of a, R_m and R_s are shown in Figure 20.

The distributions of all parameters are strongly 0 centred with low variance and magnitude, meaning that they contribute little to the overall estimator. This suggests that the residual, ϵ_i contributes the most to the overall estimator and has a high correlation with the return.

(c) By taking the correlation through time of the residual and true return, the "specific return" can be identified with probability distribution shown

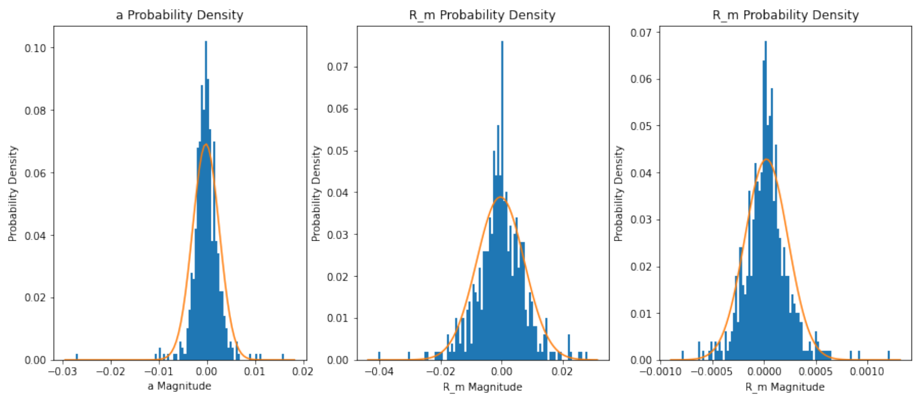

Figure 20: Probability distribution of APT estimators

in Figure 21. This shows, as expected from Section 2.4.5c, a very high correlation centred at +0.8 with artificially high variance from the outlying stocks with missing data.

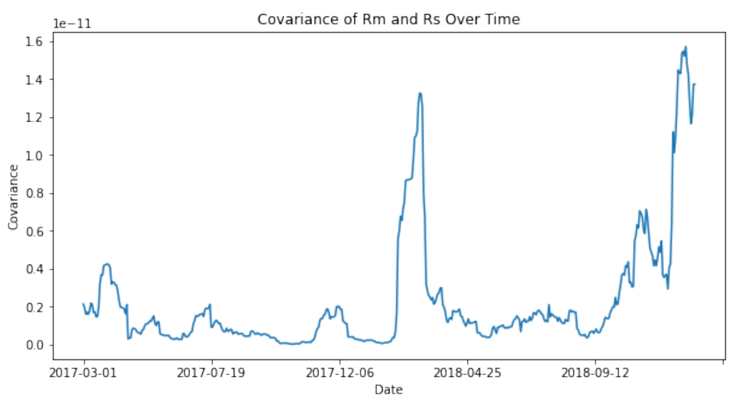

Figure 21: Covariances of APT paramters over time

This allows the conclusion to be made that the 2-factor model is **unsuitable** in modelling the returns of the stock.

(d) The rolling covariance between the estimators R_m and R_s can also be investigated as a time series and is shown in Figure 21. Whilst peaks can be seen, the overall magnitude of the covariance is of the order 10^{-11}, meaning that these parameters are completely uncorrelated.

(e) As established above, the majority of the 'information' regarding the APT estimator is held within the residual. For this reason, investigating the covariances of the different residuals is a logical step, however this yields

19-dimensional data which suggests that dimensionality reduction should be applied in the form of PCA.

PCA finds a set of orthogonal vectors that maximise the projected variances of the dataset, with the first principal component, PC1, representing the largest variance and decreasing with each subsequent component. Figure 22 illustrates the variance held within the first 30 principal components.

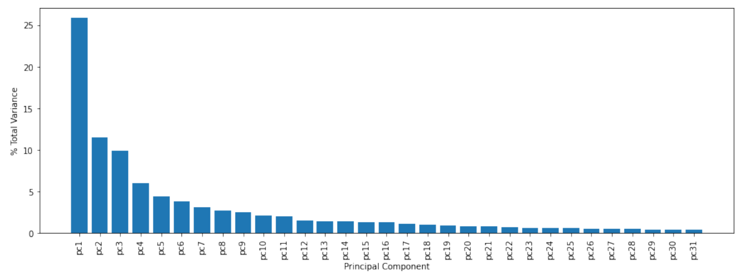

Figure 22: Percentage of variance held in each principal component

From Figure 22, it is clear that approximately 25% of the total variance is held within the first principal component and that the distributed variance of the dataset can be preserved by using only 3 dimensions, where the cumulative variance reaches 47.3%. This shows that the residuals are highly correlated in 1 direction, meaning that they serve as a good estimator for other trends in the stock market.

3 Portfolio Optimization

3.1 Adaptive minimum-variance portfolio optimization

3.1.1

To find the optimal weights of the minimum variance portfolio we solve the Lagrange optimisation:

$$\min_{\mathbf{w},\lambda} \; J'(\mathbf{w},\lambda,\mathbf{C}) = \frac{1}{2}\mathbf{w}^T\mathbf{C}\mathbf{w} - \lambda(\mathbf{w}^T\mathbf{1} - 1) \tag{38}$$

by differentiating the function with respect to $\mathbf{w}$:

$$\frac{\partial}{\partial \mathbf{w}}(\frac{1}{2}\mathbf{w}^T\mathbf{w} - \lambda(\mathbf{w}^T\mathbf{1} - 1)) = 0 \tag{39}$$

then equating to 0 to solve for both λ and $\mathbf{w}$

$$\mathbf{C}\mathbf{w} - \lambda\mathbf{1} = 0 \tag{40}$$

$$\mathbf{w} = \mathbf{C}^{-1}\lambda\mathbf{1} \tag{41}$$

$$\mathbf{w}^T\mathbf{1} = \mathbf{1}\lambda\mathbf{C}^{-1}\mathbf{1} \tag{42}$$

$$\lambda = \frac{\mathbf{w}^T\mathbf{1}}{\mathbf{1}\mathbf{C}^{-1}\mathbf{1}} \tag{43}$$

where, using the weight constraint:

$$\mathbf{w}^T\mathbf{1} = 1 \tag{44}$$

λ can be evaluated as:

$$\lambda = \frac{1}{\mathbf{1}\mathbf{C}^{-1}\mathbf{1}} \tag{45}$$

and by substituting into equation 40:

$$\mathbf{C}\mathbf{w} = \frac{\mathbf{1}}{\mathbf{1}\mathbf{C}^{-1}\mathbf{1}} \tag{46}$$

yielding:

$$\mathbf{w}^* = \frac{\mathbf{C}^{-1}}{\mathbf{1}\mathbf{C}^{-1}\mathbf{1}} \tag{47}$$

To derive the theoretical variance of returns for $\mathbf{w}^*$, it is substituted into Equation 48

$$\sigma^{2^*} = \mathbf{w}^{*^T}\mathbf{C}\mathbf{w}^* \tag{48}$$

to get, after simplification:

$$\sigma^{2^*} = \frac{\mathbf{1}\mathbf{C}^{-1}\mathbf{C}\mathbf{C}^{-1}\mathbf{1}}{(\mathbf{1}\mathbf{C}^{-1}\mathbf{1})^2} \tag{49}$$

$$= \frac{1}{\mathbf{1}\mathbf{C}^{-1}\mathbf{1}} \tag{50}$$

3.1.2

This theoretical analysis can be applied to a stock market to generate two portfolios - one comprising equally weighted stocks and one applying the weights found from the minimum variance optimiser. The results of such weightings are shown in Figure 23, where a model is generated using a test and training split of 50%.

The performance can be assessed using two metrics; the cumulative return shows the final return after the algorithm has been applied over the entire time series and the variance illustrates the volatility of such returns. For the minimum variance portfolio, the observed variance matches the theoretical variance and is lower than the variance of the even weighted portfolio.

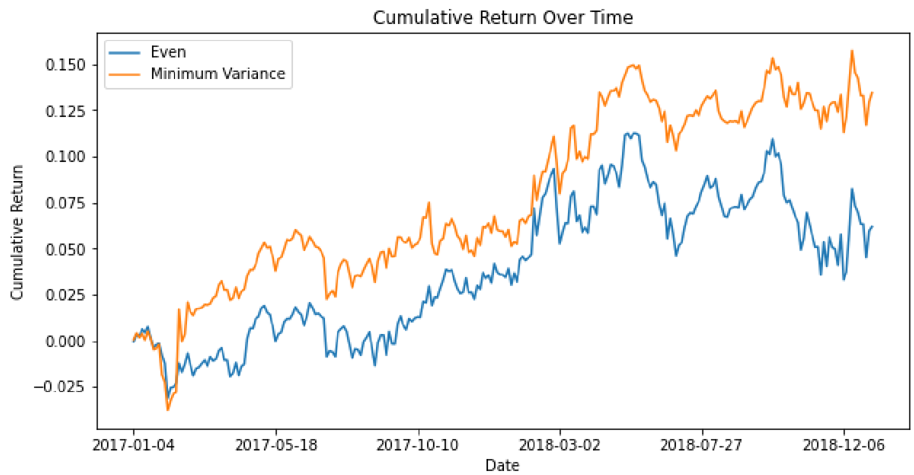

Figure 23: Cumulative returns of different weighted portfolios

This culminates in a 0.03% higher mean return and 5.4% higher cumulative return that the even weighted case, showing that the choice of a minimum variance portfolio is a clear improvement.

Weighting	Mean Return	Theoretical Variance	Observed Variance
Even	0.000237	5.97484e-05	6e-5
Min Variance	0.000516	5.655736e-05	5.7e-5

Figure 24: Statistics of differently weighted portfolios

3.1.3

For further increased robustness against seasonal data, a recursive minimum variance model can be implemented that generates a weighting for each time step based on a rolling window of previous data.

A covariance tensor is formed, that stores the covariance of all stocks for each time step in the window for a certain day. These covariance matrices are then evenly aggregated to produce a mean covariance matrix for determining the ideal weights.

The effects of applying this algorithm yield the results seen in Figure 25, where the cumulative returns for the recursive model are 30% higher than the even weighted case and 15% higher than the minimum variance portfolio. The observed variance is also significantly lower than the other cases, with a value 67% smaller than the minimum variance portfolio.

This can be attributed to the algorithm applied: instead of using a single set of weights across all time steps, an optimal set of weights is recursively

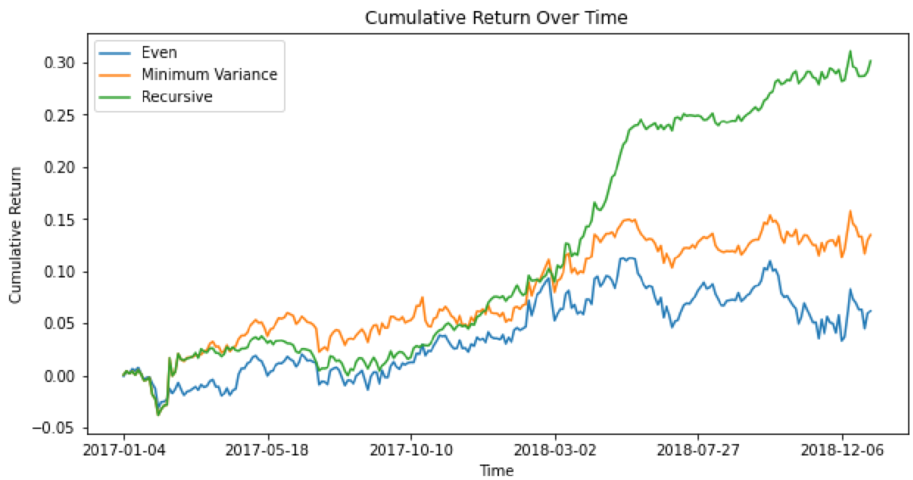

Figure 25: Cumulative returns of different weighted portfolios

Weighting	Mean Return	Theoretical Variance	Observed Variance
Even	0.000237	5.97484e-05	6e-5
Min Variance	0.000516	5.655736e-05	5.7e-5
Recursive	0.000579	N/A	1.9e-5

Figure 26: Statistics of differently weighted portfolios

generated for each time step. This ensures that the dataset isn't overfitted to and can respond well to anomalies.

A possible change to the recursive model is through the means of aggregating the matrices in the covariance tensor. Instead of offering an even split, a weighting could be calculated to maximise returns at any time step. This will help to robustify the model against anomalies/outliers, as covariance matrices that stem from an anomaly will no longer corrupt the aggregated matrix.

4 Robust Statistics and Non Linear Methods

4.1 Data Import and Exploratory Data Analysis

4.1.1

Three stocks, Apple, IBM and JP Morgan, are assessed in terms of their statistical parameters: mean, median and standard deviation. Their respective data is illustrated in Figures 27, 28 and 29

Measure	Open	High	Low	Close	Adj. Close	Volume	Return
Mean	187.69	189.56	185.82	187.71	186.17	3.27e+07	4.26e-4
Median	186.29	187.40	184.94	186.12	184.35	2.92e+07	1.61e-3
STD	22.15	22.28	22.01	22.16	21.90	1.42e+07	1.93e-2

Figure 27: Apple Stock statistics

Measure	Open	High	Low	Close	Adj. Close	Volume	Return
Mean	138.45	139.49	137.33	138.36	134.90	5.20e+06	-2.52e-4
Median	142.81	143.99	142.06	142.71	138.57	4.24e+06	4.09e-4
STD	12.11	11.91	12.20	12.03	10.67	3.33e+06	1.56e-2

Figure 28: IBM Stock statistics

4.1.2

Generating the probability density function (PDF) for each stock yields the set of graphs seen in Figure 30. The main difference between the PDF of adjusted close and returns is the distribution: Section 1 has illustrated that returns are Gaussian, however prices are not.

Measure	Open	High	Low	Close	Adj. Close	Volume	Return
Mean	108.70	109.65	107.68	108.60	107.26	1.47e+07	-1.3e-4
Median	109.18	110.53	107.79	109.02	107.22	1.36+07	-6e-4
STD	5.36	5.20	5.43	5.30	4.83	5.35e+ 06	1.3e-2

Figure 29: JP Morgan Stock statistics

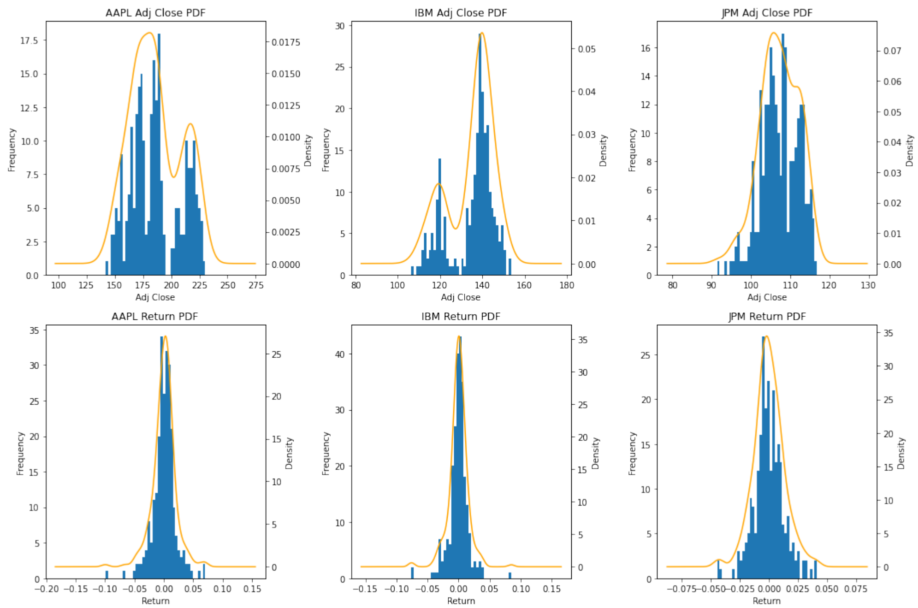

Figure 30: PDF of Adj. Close and Returns of 3 stocks

For this reason, the Gaussian kernel is unable to fit accurately to the price data and creates and arbitrary distribution based on the peaks.

4.1.3

Anomalous data can be detected through various metrics. One approach is taking the rolling mean (a 5-day window, in this case) of a sample and plotting 1.5 standard deviations (SDs) above and below it. Any data that lies outside of this confidence integral is regarded as an anomaly. The anomaly analysis for all stocks can be seen in Figure 31

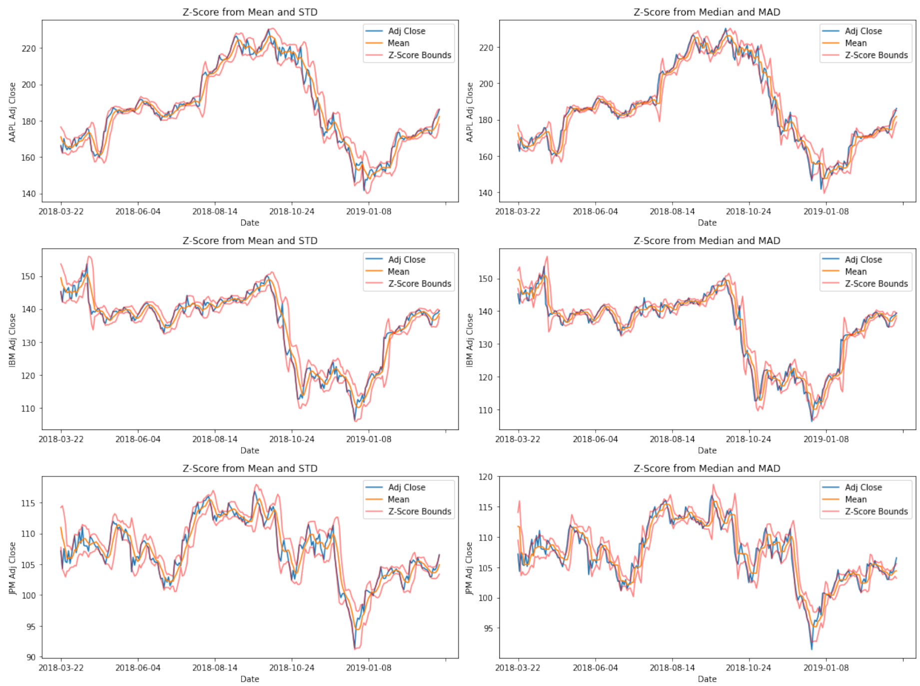

Figure 31: Z-Score confidence regions for AAPL, IBM and JPM

Alternatively, the mean absolute deviation (MAD) can also be applied as an anomaly detecting metric by taking 1.5 mean absolute deviations above and below the rolling median. This metric is also shown in Figure 31.

The main difference between the two metrics is the sharpness of the peaks. The SD metric has rounded peaks, whereas the MAD has sharp peaks. This suggests that the the MAD is less likely to misclassify an anomaly, as a more precise area is covered.

4.1.4

Outlier points are artificially introduced to the dataset at 4 evenly spaced points at values equal to 1.2× the maximum value of the series. Rerunning the anomaly detection analysis on the altered dataset gives Figure 32

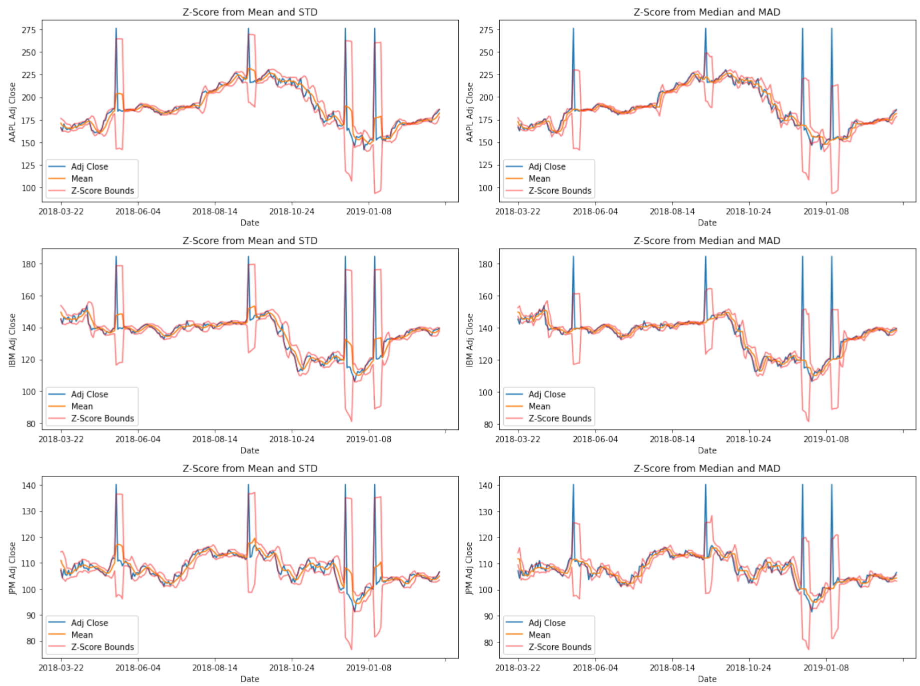

Figure 32: Z-Score confidence regions for AAPL, IBM and JPM with outliers introduced

Introducing outliers heavily impacts the Z-Score region around these points, as the magnitude of the region rapidly changes, often classifying previously acceptable points as outliers. The MAD estimator is more robust against artificial spikes, however, as the magnitude of the confidence region changes less than the SD estimator.

4.1.5

Box plots serve as an effective way of visualising the mean, IQR and range of a dataset, with the central green line in Figure 33 representing the mean, the edges of the 'box' representing the lower and upper quartiles and the tail representing the minimum and maximum values.

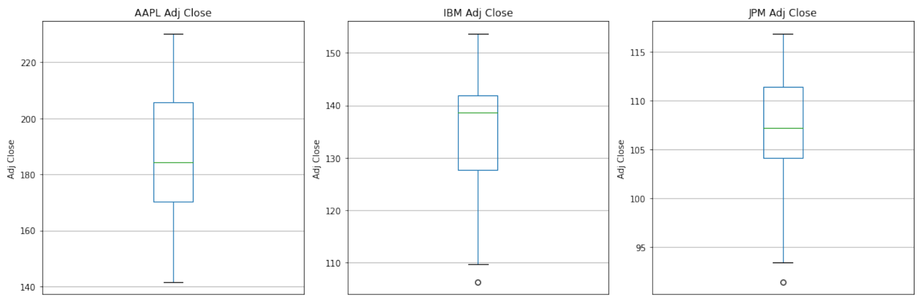

Figure 33: Adj. close box plots for AAPL, IBM and JPM

The `matplotlib` implementation defines outliers as any points lying outside 1.5× the IQR and are hence represented by 'bubbles' outside of the tail.

4.2 Robust Estimators

4.2.1

Below shows an implementation of 3 robust estimators, the median, IQR and MAD, without using in-built Python libraries.

```
def get_median(series):
    series = series.sort_values(ignore_index=True)

    mid_index = (len(series) + 1) / 2 - 1
    up_index = math.ceil(mid_index)
    dn_index = math.floor(mid_index)

    return (series[up_index] + series[dn_index]) / 2}

def get_iqr(series):
    series = series.sort_values(ignore_index=True)
    q1_index = (len(series) + 1) * 1 / 4 - 1
    q1_up_index = math.ceil(q1_index)
    q1_dn_index = math.floor(q1_index)

    q2_index = (len(series) + 1) * 3 / 4 - 1
    q2_up_index = math.ceil(q2_index)
    q2_dn_index = math.floor(q2_index)

    q1 = (series[q1_up_index] + series[q1_dn_index]) / 2
    q2 = (series[q2_up_index] + series[q2_dn_index]) / 2

    return q2 - q1
```

```python
def get_mad(series):
    return get_median(abs(series - get_median(series)))
```

4.2.2

To assess the efficiency of the different implementations, they are compared against the speed of the pure library implementations by averaging the time to execute the function over 10,000 cycles. The results are shown in Figure 34

	Library	Self
Median	2.65e-4	1.65e-4
IQR	1.67e-4	1.74e-4
MAD	1.63e-4	5.86e-4

Figure 34: Average time to execute function over 10,000 cycles

Whilst the self implementation produces an algorithm faster than the library for the median, the other metrics are substantially slower.

4.2.3

An estimator's breakdown point is the proportion of outliers that can constitute a dataset before the performance degrades. For these experiments, the original dataset is iteratively corrupted with each iteration setting the subsequent value to 1.2× the max value. When the estimator produces a value that is 30% worse than the true estimator, the breakdown point is reached.

	Library	Self
Median	0.498008	0.498008
IQR	0.179283	0.179283
MAD	0.135458	0.135458

Figure 35: Breakdown points of library vs self implementation of estimators

The median has the best tolerance to outliers, as it has a theoretical breakdown point of 0.5. This is well reflected in Table 35. The IQR and MAD are dependent on the scale of the data, so are more susceptible to arbitrarily high datapoints corrupting the dataset. This is reflected by the comparably lower breakdown points of these estimators.

4.3 Robust and OLS regression

4.3.1

As discussed in Section 1, regression serves to map a set of input values to a set of output values using various different techniques. Ordinary Least Squares (OLS) minimises the sum of squared vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation.

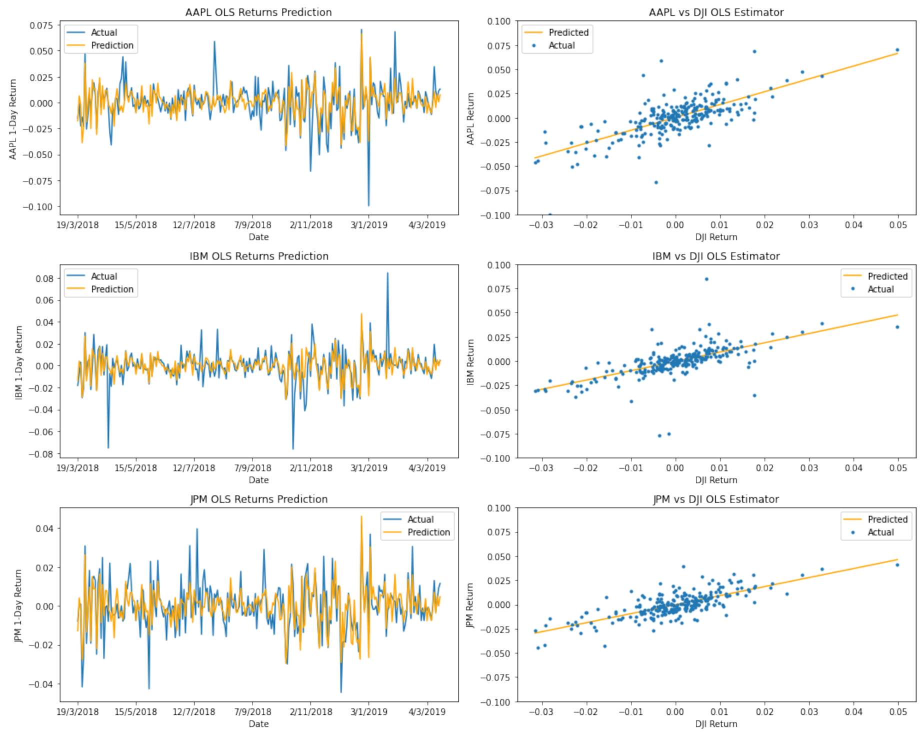

Figure 36: Regression of stock 1-day returns against DJI 1-day returns using OLS

By regressing each stock against the DJI, a regressor model can be formed. The time-series prediction and line of best fit for each model can be seen in Figure 36.

4.3.2

Alternatively, the "Huber Loss" is a loss function used in robust regression that is less sensitive to outliers in data than the squared error loss.

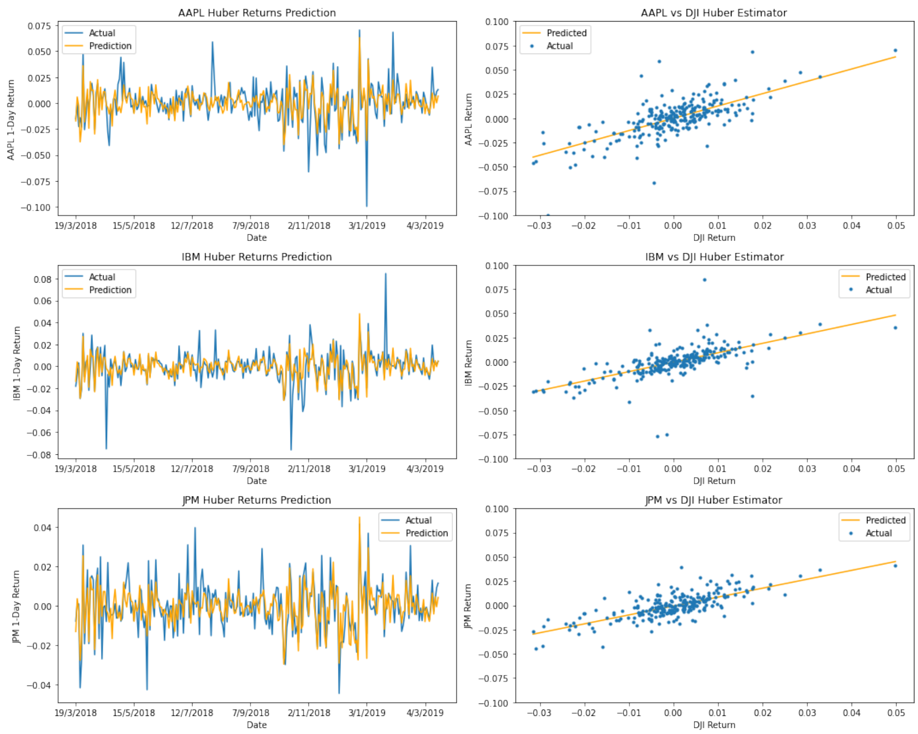

Figure 37: Regression of stock 1-day returns against DJI 1-day returns using Huber Regression

The results of each regressor are shown in Figure 37.

4.3.3

The robustness of the Huber estimator is investigated against the OLS regressor by artificially introducing a percentage of outliers into the dataset, thereby corrupting it. Figure 38 illustrates the differences between the performances of the two regressors.

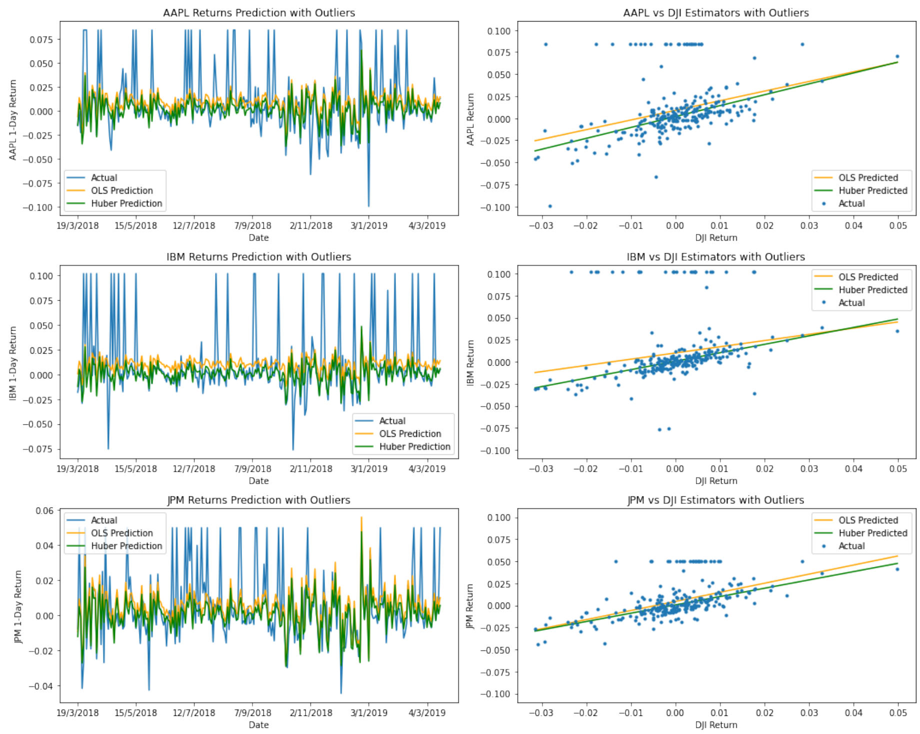

Figure 38: OLS and Huber regressors with artificial outliers

Clearly, the OLS regressor has a tendency to increase the gradient of the line of best fit in order to fit the proportion of outliers set at 1.2× the maximum value in the series. This is in contrast to the Huber regressor which develops an immunity and sees comparably greater gradient change.

This is reflected in the predicted time-series, where the OLS regressor becomes very sensitive to noise and constantly overshoots the Huber regressor. It can hence be concluded that the Huber regressor is robust to outliers and is a superior choice of regression method.

4.4 Robust Trading Strategies

4.4.1

A simple technique for trading based on time-series data is to inspect the 20 and 50 day moving averages (MA). If the 20-day MA is larger than the 50-day MA, then stock should be *bought*, otherwise it should be sold. The results of applying this algorithm to clean and artificially corrupted data are seen in Figure 39.

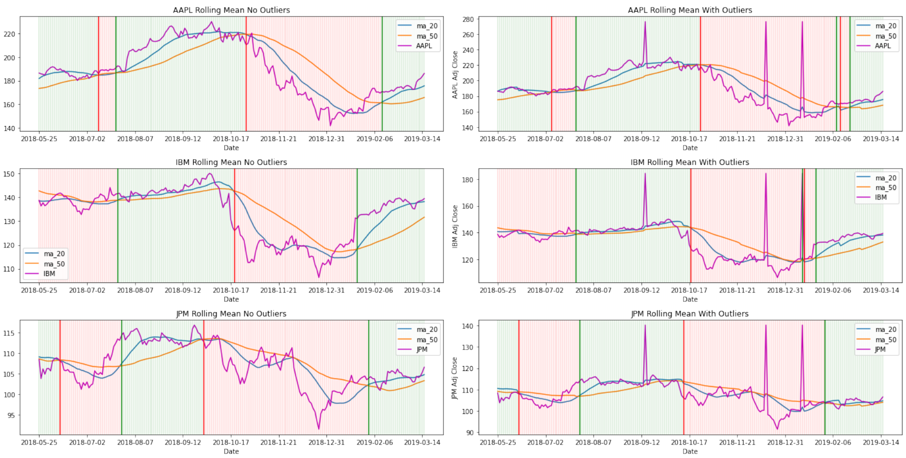

Figure 39: Adj. close rolling mean with buy/sell regions highlighted with (right) and without (left) outliers

Implementing this method shows a clear sensitivity to outliers, where any artefacts result in idiosyncratic buying or selling of a stock, even if this is an illogical decision based on the clean data.

4.4.2

An alternate approach is to use the same technique but with a rolling *median* instead of a rolling mean. This yields the plot in Figure 40

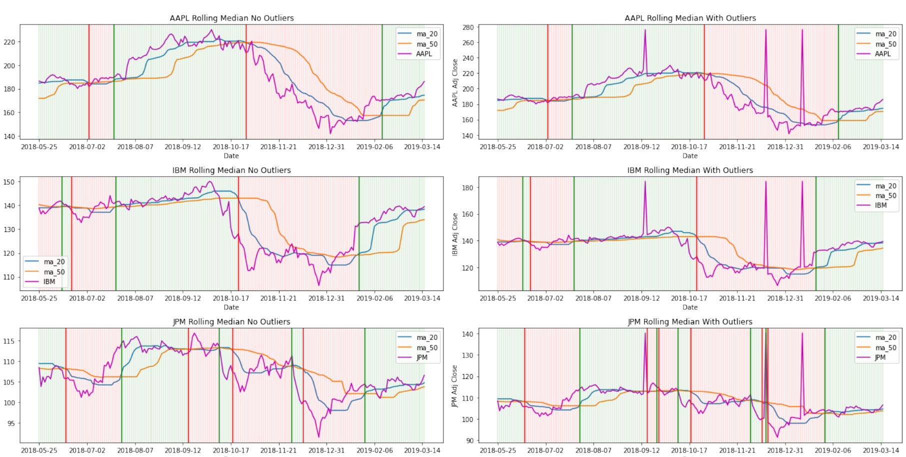

Figure 40: Adj. close rolling median with buy/sell regions highlighted with (right) and without (left) outliers

Using this method means that the stock becomes more robust to outliers, as there are comparably fewer instances where the recommended trade action for noisy data is different to the clean data, when compared to Figure 39.

5 Graphs in Finance

5.1

5.1.1

This set of experiments aims to investigate the topological relationships between the 10 *longest running financial stocks* in the S&P 500. For this reason, the stocks under investigation are displayed in Figure 41.

Symbol	GICS Sub Industry	Headquarters Location	Founded
AXP	Consumer Finance	New York, New York	1850
MTB	Regional Banks	Buffalo, New York	1856
FITB	Regional Banks	Cincinnati, Ohio	1858
HBAN	Regional Banks	Columbus, Ohio	1866
MET	Life & Health Insurance	New York, New York	1868
NTRS	Asset Management	Chicago, Illinois	1889
AMP	Asset Management	Minneapolis, Minnesota	1894
MCO	Financial Exchanges & Data	New York, New York	1909
AIG	Casualty Insurance	New York, New York	1919
ALL	Casualty Insurance	Northfield Township, Illinois	1931

Figure 41: Breakdown points of library vs self implementation of estimators

5.1.2

It is possible to construct a graph between the stocks based on a threshold of any difference metric. The first metric to investigate is the covariance between any two stocks, which can be analysed by forming a covariance matrix.

If the covariance between any two stocks is larger than 0.5 (meaning that they are strongly correlated), a link is formed between the stock nodes. This creates the graph seen in Figure 42

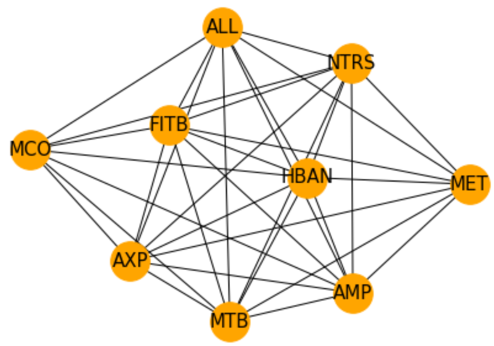

Figure 42: Graph connecting stocks with covariance > 0.5

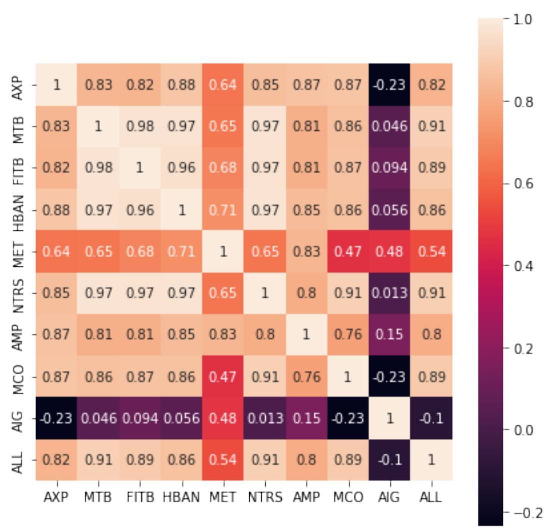

Figure 43: Covariance matrix for log data

Clearly, the role of the covariance matrix is to illustrate the similarities between the different time series. Due to all of the stocks being from the same sector and having been established for similar amounts of time, it is likely that the companies will be broadly similar in performance. This is reflected in Figure 43.

5.1.3

The topology of the graph naturally reflects the nature of the data, as fully uncorrelated data would result in a disconnected graph. Reordering the graph vertices, whilst changing the appearance of the graph would not change the topology. This is because the links between nodes are independent of the arrangement in 2-D space.

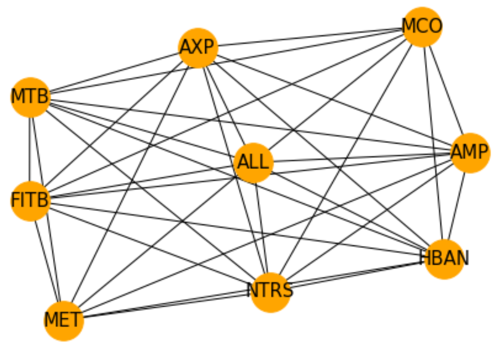

Figure 44: Graph connecting stocks with covariance > 0.5 with re-ordering of time-series data

As shown in Figure 44, reordering the time-series has no affect on the graph. Simply, this is because the covariance between two series is calculated as:

$$\frac{1}{N}\sum_{x,y}(x-\mu_x)(y-\mu_y) \tag{51}$$

meaning that, irrespective of order, the entire series will be iterated over with no change in mean. The covariance is mathematically independent of ordering.

5.1.4

Alternative distance metrics may be used to assess the similarity of a dataset. This section investigates whether two stocks that are similar in the time domain will retain similarity in the frequency domain. For this reason, the supplementary `fourier_distance(s1,s2,bins)` function is defined to calculate the Fourier distance between two series.

```
def fourier_distance(s1, s2, bins=None):

    if bins is None:
        bins = math.floor(len(s1) / 2) + 1

    if len(s1) != len(s2):
        raise Exception("Series must have equal length")

    fft1 = fft(s1)
    fft2 = fft(s2)
    series_sqr_diff = abs(fft1[1:bins] - fft2[1:bins]) ** 2
    return math.sqrt(series_sqr_diff.sum())
```

The Fourier distances are normalised by the largest value in the dataframe after the distances between all stocks is calculated. This yields the distance matrix in Figure 46, which is converted into the graph in Figure 45.

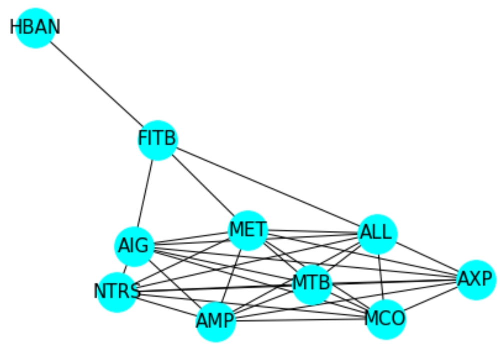

Figure 45: Graph connecting stocks with normalised Fourier distance < 0.5

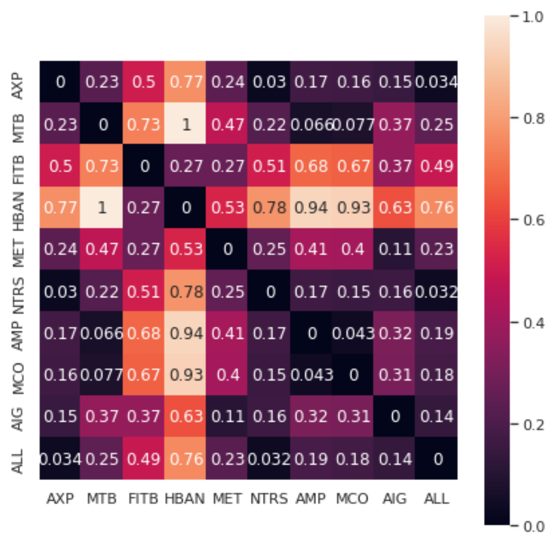

Figure 46: Fourier distance matrix for log data

Visually, this graph is very similar to Figure 44. This is expected, because the FFT will peak at points where the amplitude of a frequency component is high. Therefore, the FFT can be interpreted to show the seasonal (periodic) components of each signal at the different bins.

Whereas the correlation in Figure 44 can be interpreted as the correlation between signals in time, the analysis in Figure 45 yields the correlation between seasonal components in the frequency domain. If signals are well correlated in the time domain, they likely share seasonal components and will hence have a similar distribution in the frequency domain.

A caveat to this metric stems from the nature of the fast Fourier transform (FFT). Since the FFT of a series is ordering dependent, the Fourier distance will also be ordering dependent, so the time series must be preserved.

5.1.5

Finally, the differences in using raw (non-log) and log prices can be investigated by repeating the experiments from Sections 5.1.3 and 5.1.4 for a non-log dataset.

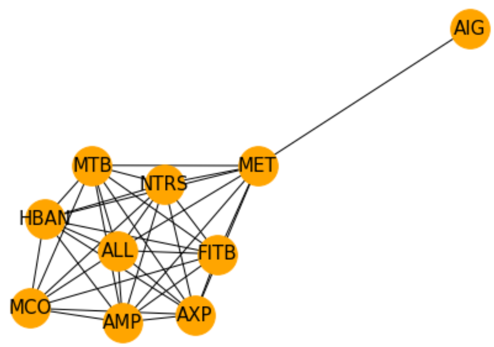

Figure 47: Graph connecting stocks with covariance > 0.5 with raw (non-log) data

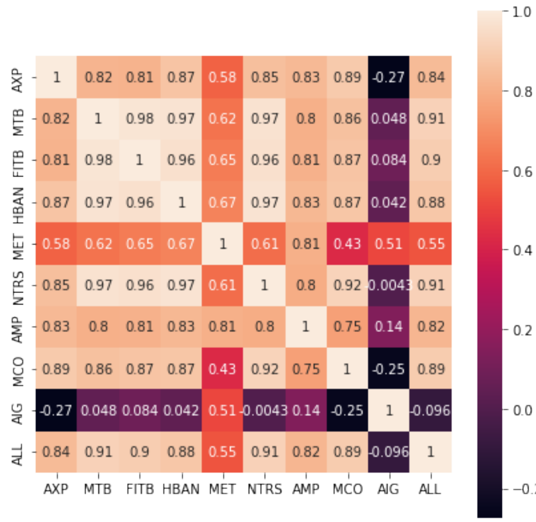

Figure 48: Covariance matrix for raw data

Serving as a non-linear transform, taking the log of a dataset means that the correlation will also change, since the spatial representation of the data also changes. Raw prices, unlike log prices, are non-stationary, so taking covariances will also fit any general trends and seasonal components (as will the Fourier distance), meaning that the connections between stocks will likely be strengthened (and easier to form for the same threshold).

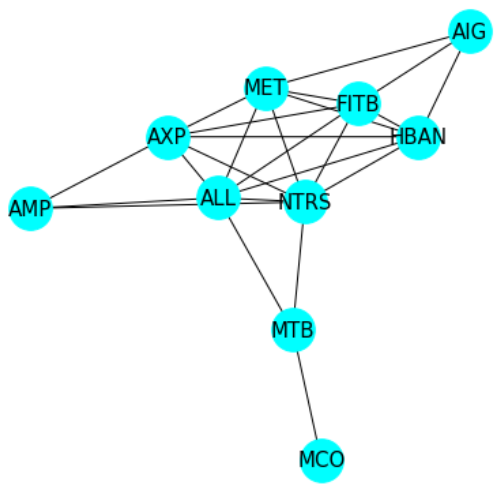

Figure 49: Graph connecting stocks with Fourier distance < 0.5 with raw (non-log) data

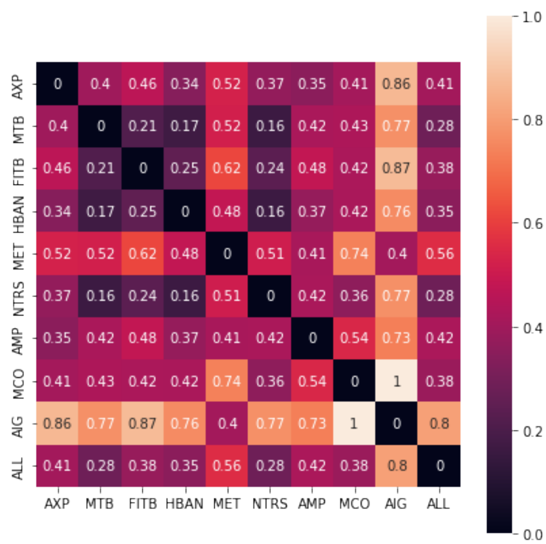

Figure 50: Fourier distance matrix for raw data