Signal Processing and Machine Learning for Finance Report

Savraj Sian

CID: 01847921

April 8, 2024

Contents

1 Regression Methods

1.1 Processing Stock Price Data in Python

1.1.1 Log prices

The price data of the SPX index from the *priceData.csv* file was imported and the logarithm of the prices was taken, which compresses the data. Figure 1 shows the difference between the normal and log prices.

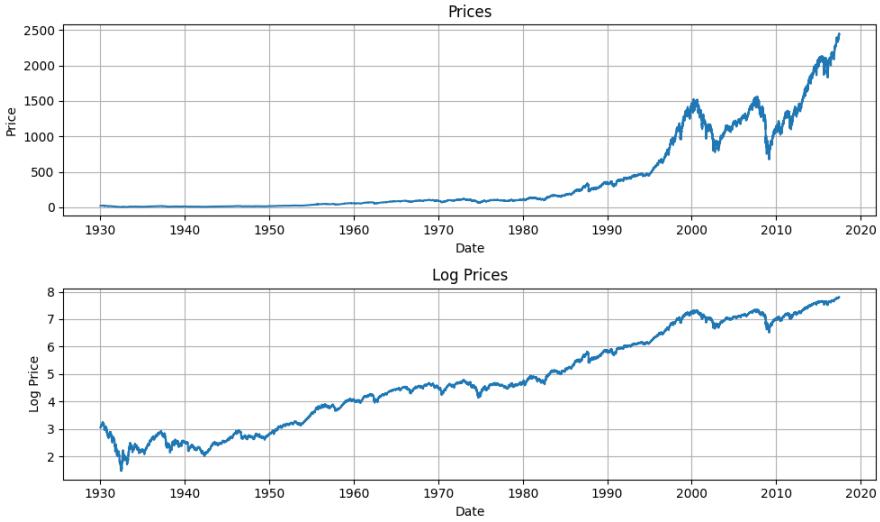

Figure 1: SPX prices

1.1.2 Sliding window and stationarity

The rolling mean and standard deviation for the normal and log prices using a sliding window length of 252 days, in 1 day increments. Figure 2 shows these values for the normal and log prices. The rolling mean allows us to clearly see that there is an upwards trend in the mean and we can therefore conclude that the price time series are non-stationary since there it does not have a constant mean.

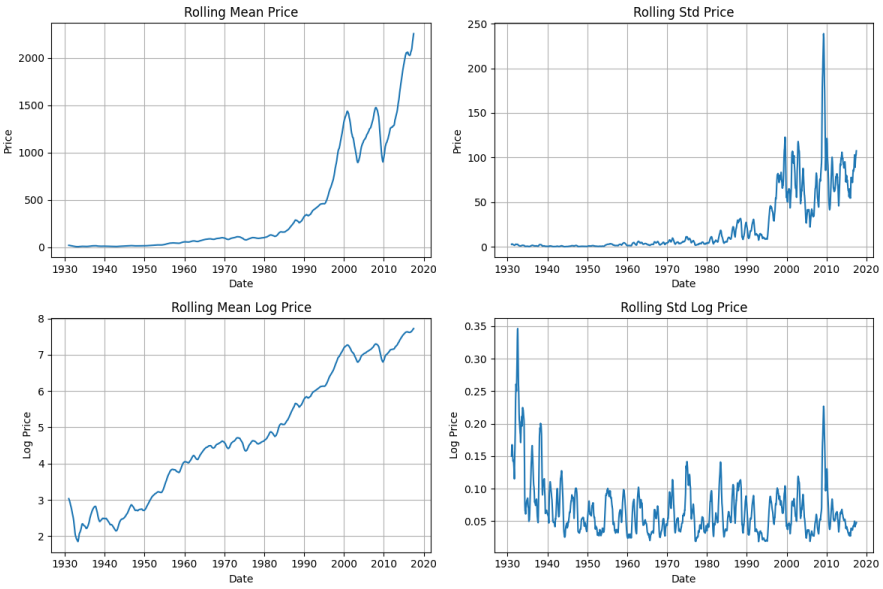

Figure 2: Rolling mean and standard deviation

1.1.3 Simple and log returns

The simple and log returns can be computed using the following formulas, where $p[t]$ is the price at time t, $R[t]$ is the simple return and $r[t]$ is the logarithmic return:

$$R[t] = \frac{p[t] - p[t-1]}{p[t-1]} = \frac{p[t]}{p[t-1]} - 1 \tag{1}$$

$$r[t] = \log\left[\frac{p[t]}{p[t-1]}\right] = \log[p[t] - \log[p[t-1]]] \tag{2}$$

Figure 3 contains the plots of the simple and log returns, which we can see oscillate around zero. The sliding means shown in Figure 4 shows more clearly the oscillatory behaviour. As evidenced by the mostly constant mean, the returns are a stationary time series.

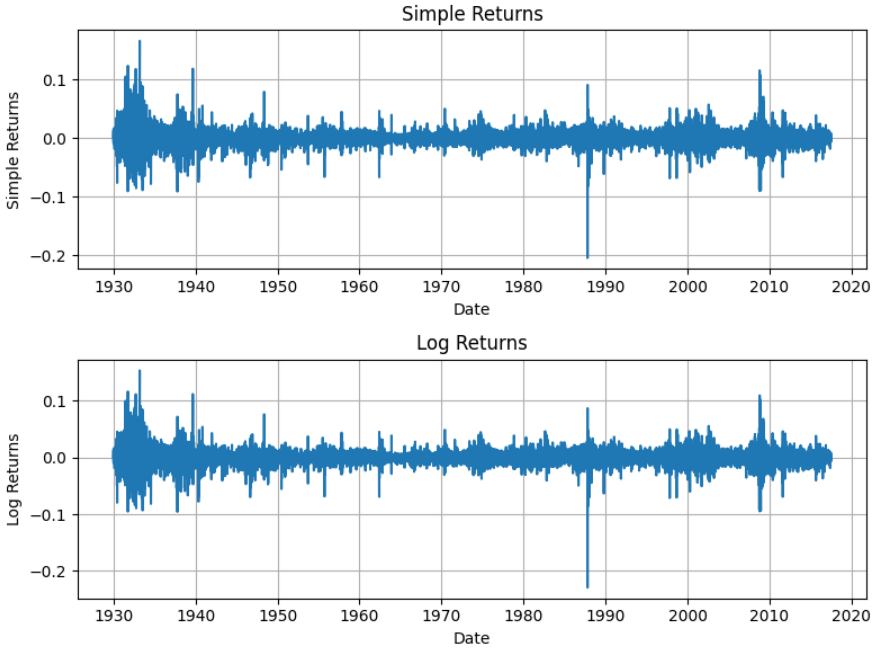

Figure 3: Simple and log returns

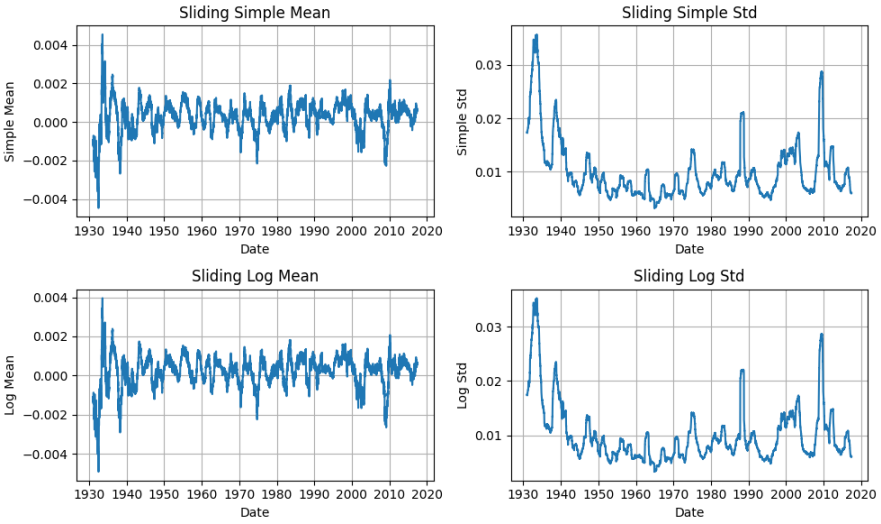

Figure 4: Rolling mean and standard deviation of simple and log returns

1.1.4 Advantages of using log returns

There are several reasons to use log returns over simple returns; over short period of time, prices are distributed log-normally, so taking the log returns are normally distributed. They also have time additivity meaning if an asset dips 1% today but gains 1% tomorrow, the value remains the same which is not the case with simple

returns. Taking the logarithm has mathematical advantages where it adds numerical stability and tractability for calculations like calculus, and it is a monotonic function that preserves relative ordering such that if $a < b$, then $log(a) < log(b)$. The Gaussian behaviour can be seen in Figure 5.

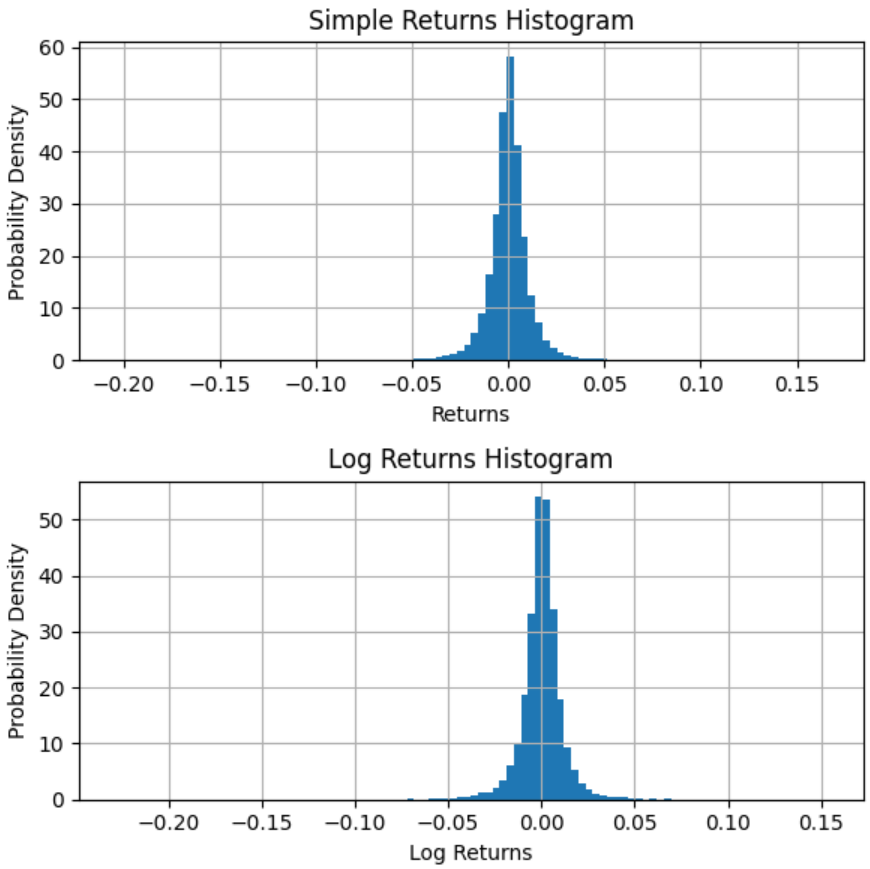

Figure 5: Rolling mean and standard deviation of simple and log returns

The Jacque-Bera test tests whether the sample data has the skewness and kurtosis matching a normal distribution. This test can therefore be used on the returns to quantify how Gaussian they are. Prices are only distributed log-normally over short periods of time, which explains the results of this test over the whole dataset; for simple returns the statistic was 257540 and the p-value was 0, and for the log returns the statistic was 309258 and the p-value was 0. The null hypothesis for this test is that the data is normally distributed, so a p-value of less than 0.05 can be considered a significant result and the null hypothesis can be rejected, i.e. the data is not normally distributed. However, when performing this test on smaller windows, the results are different. On 10 data points, the statistics were 0.72 and 0.70 and the p-values were and 0.70 and 0.71 for the simple and log returns respectively; this test shows a normal distibution. This can be seen in Figure 6, where the statistic value increases as the number of data points increases.

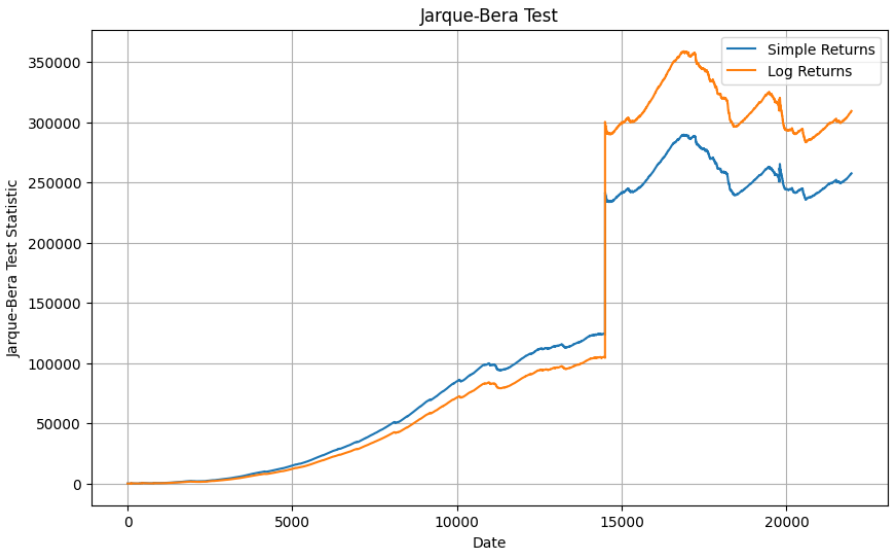

Figure 6: Jarque-Bera test for different number of data points

1.1.5 Stock price change

The example of purchasing a stock for £1, the next day its value going up to £2 and the following day it dropping back to £1 illustrates a key difference in simple and log returns. The stock's value has stayed the same, but according the simple returns which would be $[1, -0.5]$ do not reflect this whereas the log returns $[0.693, -0.693]$ sum to 0, showing there is no change. Therefore log returns are a better metric of showing price changes.

1.1.6 When to use simple returns

Assuming log-normality of prices over long timescales is unrealistic; log-normal distributions assume a positive skew but due to crashes, the majority of financial data is negatively skewed over long timescales. Also, log returns do not add linearly across assets so in the context of a portfolio, they are not suitable for assessing returns.

1.2 ARMA vs. ARIMA Models for Financial Applications

An autoregressive moving average process, ARMA(p,q), is a stochastic process consisting of an autoregressive (AR) component that regresses the variable $x[t]$ on its own lagged values $x[t-1], ..., x[t-p]$ and this is used to explain the momentum and mean reversion effects seen in financial markets. The moving average (MA) part models the error term as a linear combination of error terms at various times in the past, $y[t-1], ..., y[t-q]$ and aims to capture 'shocks' in the market caused by things like wars or bad earnings news.

The formula for an ARMA model is:

$$x[t] = \sum_{i=1}^{p} a_i x[t-i] + \sum_{i=1}^{q} b_i \eta[t-i] + \eta[t] \tag{3}$$

1.2.1 S&P 500 and ARMA vs ARIMA

ARMA models assume stationary data, so to see if the provided S&P 500 data has this quality, the same process as section 1.1 will be followed. Figure 7 shows the log close prices for the index and Figure 8 shows the rolling mean and standard deviation. The latter shows the increasing mean and a rising and falling standard deviation, so the prices are non-stationary. Due to this, ARMA models would be unsuitable given that they require stationarity so an ARIMA model would be preferred due to it performing some differencing on the time series to remove sources of non-stationarity.

Figure 7: S&P 500 log close prices

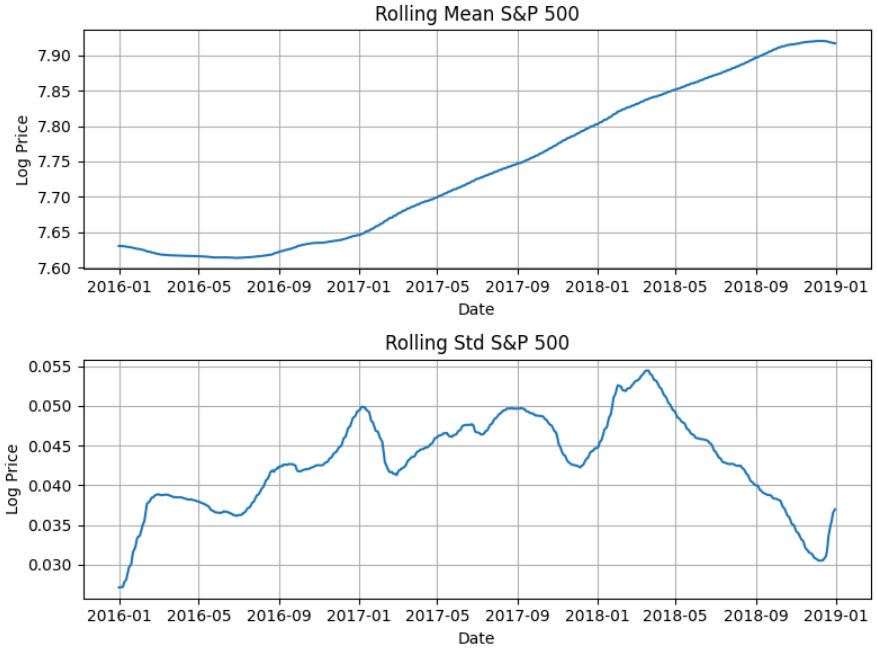

Figure 8: S&P 500 rolling mean and standard deviation

1.2.2 Fitting an ARMA(1,0) model

An ARMA(1,0) model was fitted to the S&P 500 log close price data and the prediction is plotted alongside the actual prices in Figure 9, where the discrepancies are easier to see in the lower zoomed in plot.

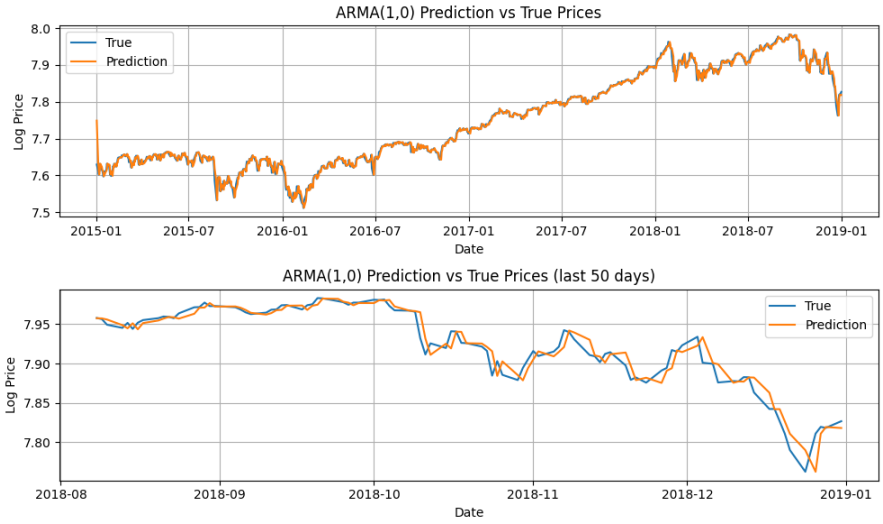

Figure 9: ARMA(1,0) prediction vs actual prices

The prediction follows the true prices closely, although there are small errors. Figure 10 shows a plot of the residuals and their histogram. The residuals fluctuate around zero and the histogram shows this since it is zero centered. It resembles a normal distribution, though it is slightly skewed; if the residuals are normally distributed, it suggests that the model has successfully captured the underlying process, leaving only the random noise which should follow a normal distribution. These findings can be useful in practice, as it shows that a simple model, despite the fact that the stationary requirement has not been met in this case, is capable of providing shorter term forecasts. But the application of the ARMA(1,0) model has its limitations, as due to the stationarity assumption, there is no guarantee that it would continue to work well over a longer time frame.

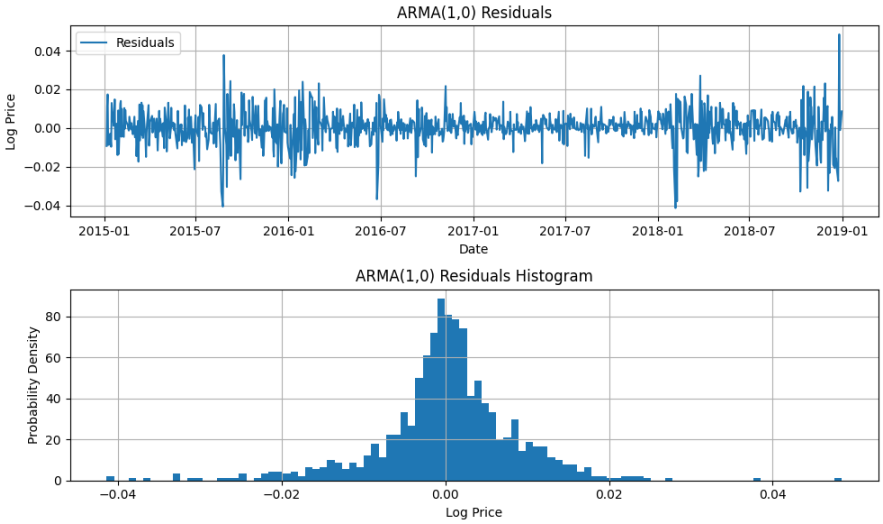

Figure 10: ARMA(1,0) residuals

The model parameters are as follows: $const = 7.748867, ar.L1 = 0.997354, sigma2 = 0.000074$. The ar.L1 term (the coefficient for the first lag in the AR part of the model) is close to 1, which indicates a high correlation with the previous value. It also suggests the time-series is close to a random walk, which is part of the efficient market hypothesis. The σ^2 term is the variance of the error, and this can be seen in the residuals plot, where the amplitude of the oscillations is low. However, the fact that the model is predicting the values to be very similar to the previous term could be a contributing factor to the low variance. Overall the ARMA model captures the general trends, but not the exact values. The constant is 7.748867 and can be seen as the mean that the model fluctuates around. So the formula for this model is:

$$x[t] = 7.748867 + 0.997354x[t-1] + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, 0.000074) \tag{4}$$

1.2.3 Fitting an ARIMA(1,1,0) model

The same process as the ARMA model was followed to fit an ARIMA(1,1,0) model. This model has the same AR process as the ARMA(1,0) model, but a differencing operation of $x[t] - x[t-1]$ is done beforehand. We can see that this model also predicts the prices well, with slightly better accuracy in the less volatile prices movements; this is easiest to see in the 2018-08 to 2018-10 section of Figures 11 and 9.

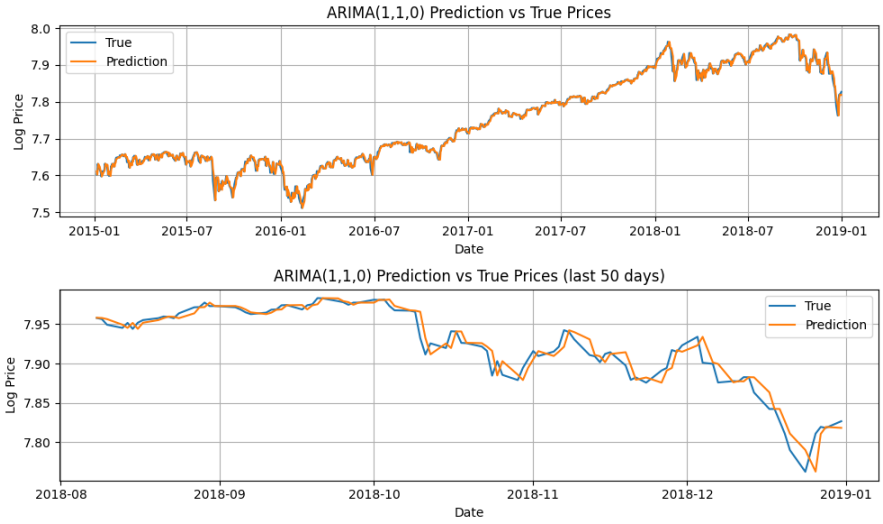

Figure 11: ARIMA(1,1,0) prediction vs actual prices

The residuals once again are centered around zero and the histogram resembles a normal distribution with a slight skew as seen in Figure 12. These properties of the prediction and residuals indicate that this model also adequately models the underlying process.

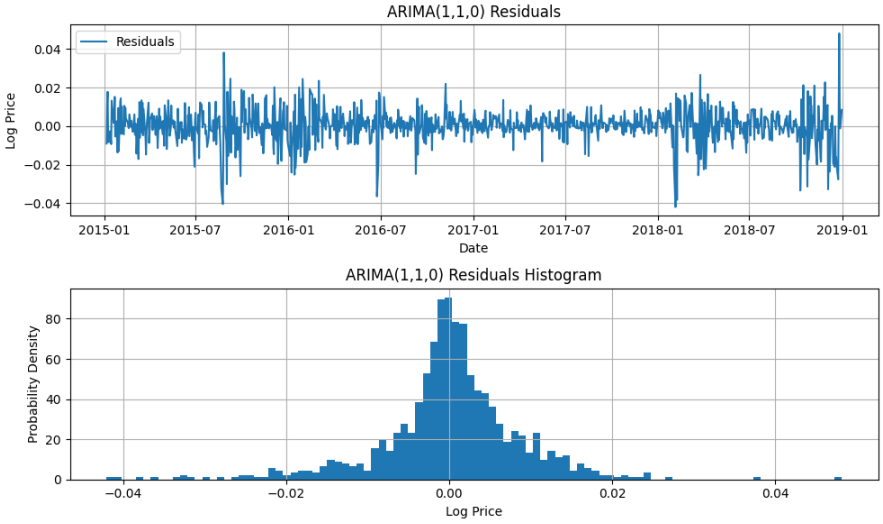

Figure 12: ARIMA(1,1,0) residuals

The parameters are $ar.L1 = -0.008170, sigma2 = 0.000074$. The small coefficient means the importance given to the previous parameter is low. The equation for this process is as follows, where $y[t] = x[t] - x[t-1]$:

$$y[t] = const - 0.008170y[t] + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, 0.000074) \tag{5}$$

ARIMA is the more physically meaningful analysis since it does not require stationarity (although the ARMA model performed just as well). However, the ARIMA model is using log returns in its prediction with the $x[t] - x[t-1]$ term (the prices are log prices), and from earlier in this report we have seen that the log returns are stationary, so the data going into the AR part is stationary.

1.2.4 Necessity of taking log prices for ARIMA

The same ARIMA analysis was done with regular prices, and the results are in Figures 13 and 14. As mentioned above, taking the log prices means the ARIMA model ends up with log returns, which due to the differencing are stationary (a necessity for the ARMA model). The magnitude of the errors is much larger due to the lack of the compressive log function and the histogram is less symmetrical and is not quite centered around zero. This is due to the differencing no longer guaranteeing a stationary input to the model. The mean error for the ARIMA model in question 1.2.3 was 1.977×10^{-4}, but the mean error for this non-log prices ARIMA model was 2.495; over 12000 times the error compared to when using log prices. It is worth noting that this mean error is the mean of the residuals and numerically shows that the mean of the histogram has shifted to the right when not using log prices. Also, by taking the logarithm of prices, you can linearize any exponential growth which better aligns with the linear nature of ARIMA, and this is again shown in the smaller errors when using log prices.

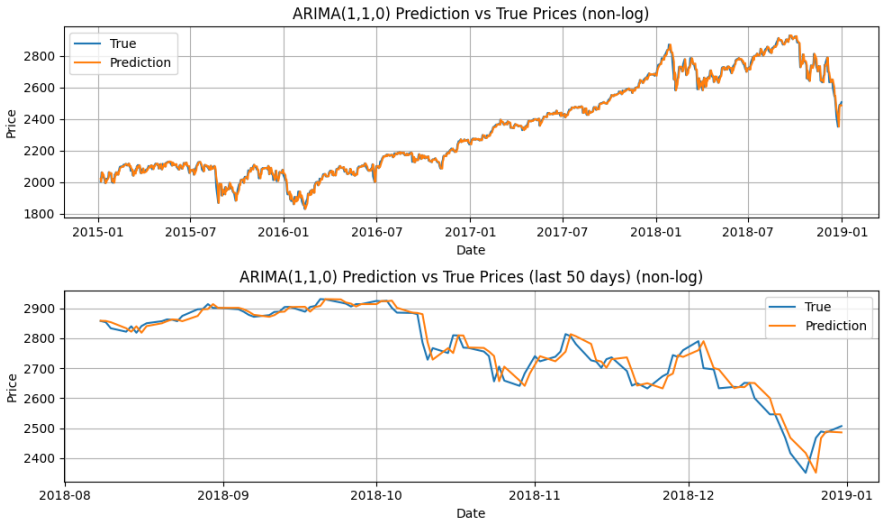

Figure 13: ARIMA(1,1,0) prediction vs actual non-log prices

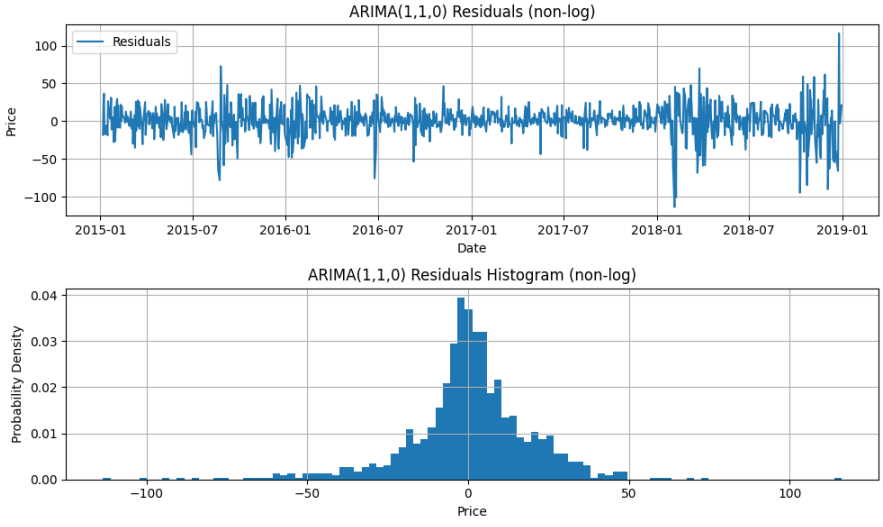

Figure 14: ARIMA(1,1,0) non-log residuals

1.3 Vector Autoregressive (VAR) Models

A multivariate extension of the AR processes, VAR(p) is given by:

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1\mathbf{y}_{t-1} + ... + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{e}_t \tag{6}$$

which can be written as:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a^1_{1,1} & a^1_{1,2} & \cdots & a^1_{1,k} \\ a^1_{2,1} & a^1_{2,2} & \cdots & a^1_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^1_{k,1} & a^1_{k,2} & \cdots & a^1_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a^p_{1,1} & a^p_{1,2} & \cdots & a^p_{1,k} \\ a^p_{2,1} & a^p_{2,2} & \cdots & a^p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^p_{k,1} & a^p_{k,2} & \cdots & a^p_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix} \tag{7}$$

1.3.1 Concise matrix form

The above two equation can be written in the concise matrix form of $\mathbf{Y} = \mathbf{BZ} + \mathbf{U}$, where:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_T \end{bmatrix} \tag{8}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{c} & \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_p \end{bmatrix} \tag{9}$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_T \end{bmatrix} \tag{10}$$

If we let $z_t = [1, y_{t-1}, \ldots, y_{t-p}]^T$, then:

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{11}$$

In the above equations $Y \in \mathbb{R}^{K \times T}, B \in \mathbb{R}^{K \times (KP+1)}, Z \in \mathbb{R}^{(KP+1) \times T}$ and $U \in \mathbb{R}^{K \times T}$.

1.3.2 Deriving optimal coefficients

To get the optimal set of coefficients, B_{opt}, you need to aim to minimise the error contained in U through least squares:

$$\begin{aligned} J = \min_B |U||_F^2 = \min_B ||Y - BZ||_F^2 = tr((Y - BZ)^T(Y - BZ)) & \tag{12} \\ = tr(Y^TY) - tr(Y^TBZ) - tr(Z^TB^TY) + tr(Z^TB^TBZ) & \tag{13} \\ = tr(Y^TY) - 2tr(Y^TBZ) + tr(BZZ^TB^T) & \tag{14} \end{aligned}$$

If we let $B = A$ and $ZZ^T = X$, we can use the matrix derivative identity $\frac{\partial}{\partial A}(tr(XAX^T)) = X(A + A^T)$ and also have the identity $\frac{\partial}{\partial X}(tr(AXB)) = A^TB^T$, allowing us to differentiate (and minimize) with respect to B:

$$\begin{aligned} \frac{J}{\partial B} = -2YZ^T + B(ZZ^T + ZZ^T) = -2YZ^T + 2BZZ^T = 0 & \tag{15} \\ BZZ^T = YZ^T & \tag{16} \\ B_{opt} = YZ^T(ZZ^T)^{-1} & \tag{17} \end{aligned}$$

1.3.3 Eigenvalues of a VAR(1) process

A VAR(1) is written as:

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

For the previous time instant it is:

$$\mathbf{y}_{t-1} = \mathbf{A}\mathbf{y}_{t-2} + \mathbf{e}_{t-1} \tag{19}$$

So you can write the following function for recursion:

$$\mathbf{y}_t = \mathbf{A}^t \mathbf{y}_0 + \sum_{i=0}^{t-1} \mathbf{A}^i \mathbf{e}_{t-i} \tag{20}$$

Due to the repeated multiplication of the matrix A, its eigenvalues need to have an absolute value less than 1. Performing eigendecomposition on A results in $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$, where $\mathbf{\Lambda}$ is the square eigenvalue matrix, where the eigenvalues, λ, are along the diagonal with all other values equal to 0. With the $\mathbf{A}^t$ operation now viewed as $(\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1})^t$, we can see that if $\lambda_i < |1|, \forall i$ then the multiplication will not diverge and will in fact tend to zero with sufficiently high t.

1.3.4 VAR(1) with selection of stocks

The stocks with tickers CAG, MAR, LIN, HCP and MAT were selected from S&P 500 data from 2015 to 2019. The plot of their prices and detrended prices is shown in Figures 15 and 16 respectively. The detrended prices were calculated using an MA(66) model. The window size of 66 corresponds to one quarter.

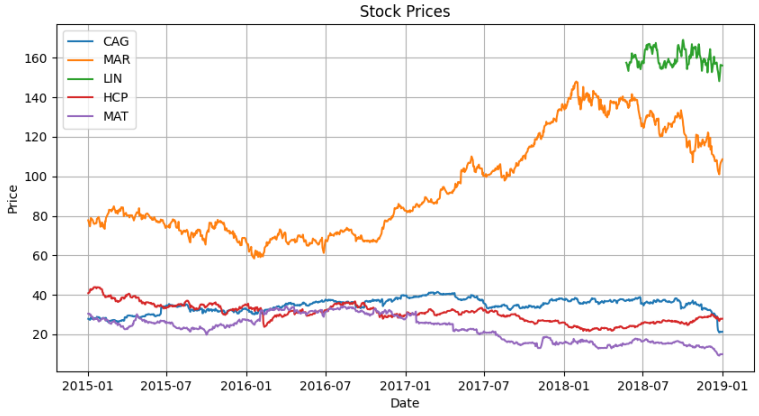

Figure 15: Stock prices

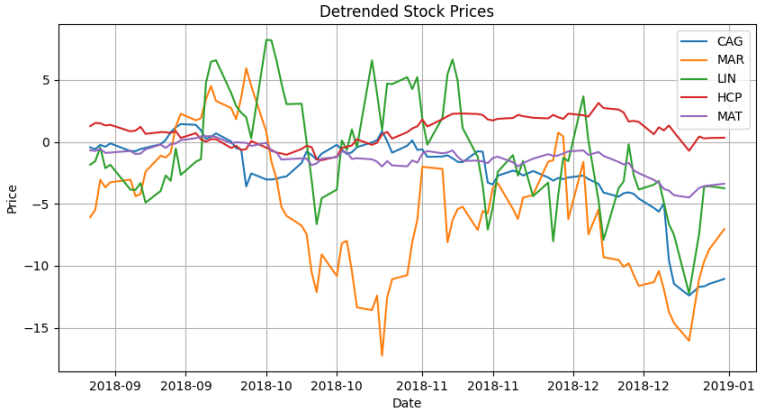

Figure 16: Detrended stock prices

Fitting a VAR(1) model to these stocks revealed that the absolute values of the eigenvalues of matrix A were 0.726, 0.726, 1.006, 0.861 and 0.911. Given that one of these values is above one, this may result in an unstable model so it is unsuitable for use with these five stocks. I also plotted the correlations of the detrended stock prices and residuals from the VAR(1) model and the results are in Figure 17. This shows that there is a strong positive correlation between CAG, MAT, and MAR for the prices and a moderately positive correlation between MAT and MAR in residuals. This makes sense given that MAT and MAR are in the consumer discretionary sector and CAG is in the consumer staples sector. The model parameters are displayed in Table 1, and these parameters reveal another relationship between stocks; LIN plays a large part in predicting the price of MAT. Therefore constructing a portfolio with these five stocks would not be advisable given the dependencies between them.

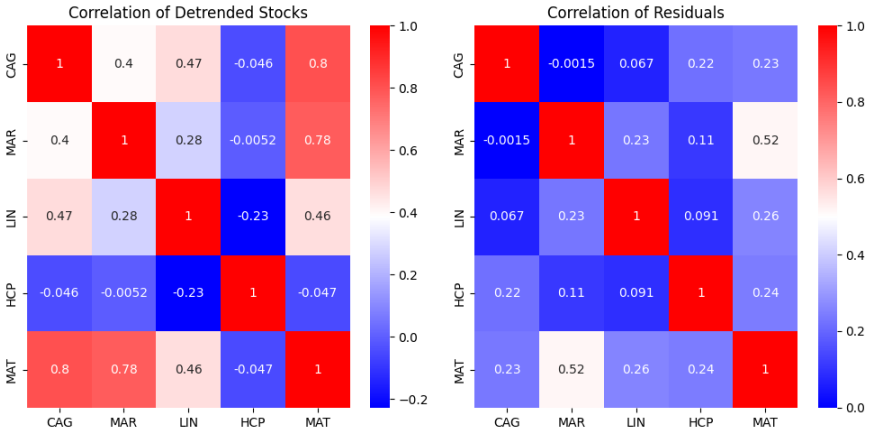

Figure 17: Correlation of prices and residuals

	CAG	MAR	LIN	HCP	MAT
const	0.143943	-0.215421	1.093664	0.034222	-0.005395
L1.CAG	0.872786	0.113179	-0.281265	0.011912	0.058776
L1.MAR	-0.063745	0.895820	-0.184820	-0.005004	0.022917
L1.LIN	0.000134	-0.111678	0.704023	0.004982	-0.025557
L1.HCP	-0.084776	-0.083831	-0.401417	0.931708	-0.046406
L1.MAT	0.643072	0.094931	2.033036	-0.012884	0.802974

Table 1: VAR(1) model parameters

1.3.5 VAR(1) with a sector-based portfolio

The stocks are now grouped by their sector, with the VAR(1) model applied to each group. The mean detrended price per sector is shown in Figure 18. The maximum absolute eigenvalues for matrix A for each sector is shown in Table 2, where we can see that all values are less than 1, except for the Financials sector. Therefore, the VAR(1) model is feasible for all sectors apart from this one. Since there are many sectors and a large number of stocks per sector, it is inefficient to plot correlations and study the model parameters to find dependencies. However, we know that stocks in the same sector will be correlated since they will be affected by similar factors like supply chain issues and sector news. For example, in the Industrials sector, the correlation between the detrended prices of PH and ROK is 0.922. As a result, building a portfolio consisting of stocks in the same sector is not a good idea. However, building a portfolio consisting of stocks that are representative of their sector is a much better idea, as there are likely to be fewer dependencies. Going back to correlations, the detrended price correlation between AME and GE in Industrials is just 0.001. So, constructing a portfolio consisting of all sectors, with minimal correlations of stocks within each sector and also minimal correlation across sectors would be the best approach. Of course, one should consider the outlook and predicted performance of each stock in making this decision too.

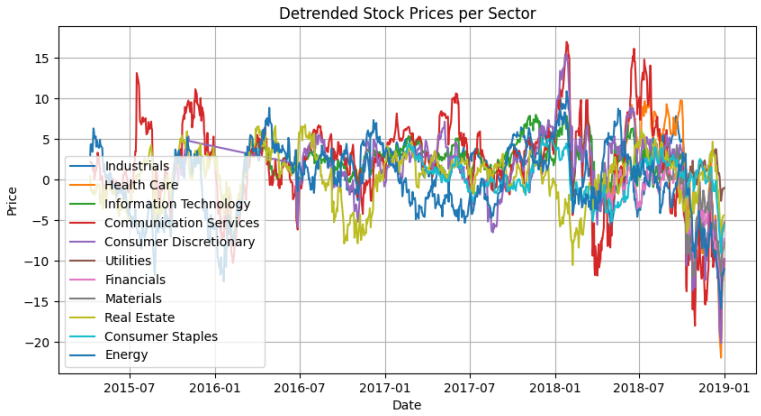

Figure 18: Mean detrended stock price per sector

Sector	Maximum absolute eigenvalue
Industrials	0.992
Health Care	0.994
Information Technology	0.993
Communication Services	0.982
Consumer Discretionary	0.991
Utilities	0.986
Financials	1.004
Materials	0.992
Real Estate	0.983
Consumer Staples	0.992
Energy	0.986

Table 2: VAR(1) eigenvalues per sector

2 Bond Pricing

2.1 Examples of Bond Pricing

2.1.1 Percentage return per annum

An investor receives \$1100 in 1 year from an investment of \$1000 now. The formula for calculating the price from a compounding investment is:

$$R = W\Big(1 + \frac{r}{m}\Big)^{mt} \tag{21}$$

where R is the final amount, W is the initial investment, r is the interest rate, m is the number of compounding periods for a specified time period and t is the number of time periods elapsed. Given $R = 1100$, $W = 1000$ and $t = 1$, we can rearrange the above formula to find r_m (which can be interpreted as the return per annum here):

$$r_m = m\left(\Big(\frac{R}{C}\Big)^{\frac{1}{m}} - 1\right) = m\left(\Big(\frac{1100}{1000}\Big)^{\frac{1}{m}} - 1\right) \tag{22}$$

For annual compounding, $m = 1$ so $r_1 = 0.1 = 10\%$. For semiannual compounding, $m = 2$ and $r_2 = 0.0976 = 9.76\%$. In monthly compounding, $m = 12$ so $r_{12} = 0.0957 = 9.57\%$. When using continuous compounding, m tends to infinity and the term $\Big(1 + \frac{r}{m}\Big)^{mt}$ becomes e^{rt}. Therefore, the price formula becomes $R = We^{rt}$ and r for continuous compounding can be found with:

$$r_{cont} = ln\Big(\frac{R}{C}\Big) = ln\Big(\frac{1100}{1000}\Big) = 0.0953 = 9.53\% \tag{23}$$

2.1.2 Continuous equivalent

To find the rate of interest with continuous compounding equivalent to 15% per annum with monthly compounding, we first compute the effective interest rate for the monthly compounding:

$$r_{eff} = \Big(1 + \frac{r}{m}\Big) - 1 = \Big(1 + \frac{0.15}{12}\Big) - 1 = 0.16075 \tag{24}$$

Since this is equivalent to the continuous effective interest rate, we equate the two:

$$e^r - 1 = 0.16075 \tag{25}$$

$$r = ln(1 + 0.16075) = 0.1491 = 14.91\% \tag{26}$$

2.1.3 Quarterly compounding

A deposit account pays 12% per annum with continuous compounding, but interest is actually paid quarterly. To find the interest paid each quarter on a $10,000 deposit, first follow a similar process to the last question. The effective interest rate for continuous compounding is:

$$r_{eff} = e^r - 1 = e^{0.12} - 1 = 0.12750 \tag{27}$$

The quarterly effective interest rate is:

$$(1 + r_{eff})^{\frac{1}{4}} - 1 = (1 + 0.12750)^{\frac{1}{4}} - 1 = 0.03046 \tag{28}$$

The interest paid each quarter is calculated by doing *amount in account* $\times 0.03046$ (the interest paid in previous quarters contributes to the later interest payments).

Month	3	6	9	12
Interest paid($)	304.55	313.83	323.39	333.23

Table 3: Interest Paid Each Quarter

2.2 Forward Rates

Suppose that the one–year interest rate, r_1 is 5%, and the two–year interest rate, r_2 is 7%. If you invest $100 for one year, your investment grows to 100 × 1.05 = $105; if you invest for two years, it grows to 100 × 1.072 = $114.49. The extra return that you earn for that second year is 1.072/1.05-1 = 0.090, or 9.0%.

2.2.1 Happy to earn the extra 9%?

The extra 9% is the forward rate implied by the market for investing a certain amount for 1 year, starting in 1 year from now. This transformation of the spot curve follows expectation dynamics. So investing for two years will return the same amount as investing for 1 year at 5% and then with the forward rate for 1 year using the amount returned from the 5% investment. Therefore, whether you choose to invest for 2 years depends on how you expect the spot curve to change in the next year. However, the flexibility of having access to your funds may be a key reason for someone to choose the one year option.

2.2.2 Investment strategies

The forward rate is derived such that you cannot use arbitrage; $(1 + r_2)^2 = (1 + r_1)(1 + f)$. Therefore, the choice of the one vs two year strategy if looking from a purely rates perspective, depends on if you think the 1 year spot rate will increase or decrease compared to the market. If you think it will increase beyond the current implied forward rate, it makes sense to choose the 1 year strategy and reinvest after that, otherwise you should take advantage of the higher current forward rate. If you take the 5% one year option, then this allows you to reevaluate your options where you might find a better alternative for you at that time.

2.2.3 Advantages and disadvantages of the forward rate

A longer investment is more risky since there is more time for unforeseen events to impact it, however this means they tend to have a higher return than shorter term ones (the forward rate previously seen is higher than the individual spot rates). So, to compensate for the increased risk, you get a higher return with the forward rate than using a two 1-year strategy. But this higher rate is fixed, so depending on what happens in the future, this could end up being an advantage or disadvantage as explained in the previous question.

2.2.4 From a 1-year to 2-year investment

The following formula describes the forward rate derivation:

$$(1 + r_j)^j = (1 + r_i)^i (1 + f_{i,j})^{j-i} \tag{29}$$

In this situation $j = 2$ and $i = 1$, so it simplifies to $(1 + r_2)^2 = (1 + r_1)(1 + f)$. To go from a 1-year to a 2-year investment, using $r_1 = 5\%$ and $r_2 = 7\%$, you need a forward rate of 9%. This calculation depends on the current spot rates.

2.3 Duration of a Coupon Bearing Bond

2.3.1 Macaulay Duration

The following table is for a 1% 7-year bond with annual payments.

Year	1	2	3	4	5	6	7	Total
Payment	\$10	\$10	\$10	\$10	\$10	\$10	\$1010	\$1070.00
PV(C_t) at 5%	\$9.52	\$9.07	\$8.64	\$8.23	\$7.84	\$7.46	\$717.79	PV = \$768.55
Fraction of PV: [PV(C_t)/PV]	0.0124	0.0118	0.0112	0.0107	0.0102	0.0097	0.9340	1
Year × Fraction of PV: [$t \times$ PV(C_t)/PV]	0.0124	0.0236	0.0337	0.0428	0.0510	0.0583	6.5377	6.76

Table 4: Duration for a 1% 7-year bond

2.3.2 Modified duration

Modified duration is calculated as $D_m = \frac{\text{Macaulay duration}}{1+\text{yield}}$ and is defined as the negative of the derivative of the price of a bond with respect to the yield to maturity. A bond's yield is the interest rate at which the present value (PV) of the payments is equal to the current price, and this is given as 5% in the table. Therefore, the modified duration is $D_m = \frac{6.76}{1+0.05} = 6.438\%$. The difference in calculation compared to the Macaulay duration is that the yield is taken into account with the division, so it gives a percentage change in bond price for a 1% change in yield.

2.3.3 Why (modified) duration is useful

Duration and modified duration are useful measures to protect a portfolio against unexpected interest rate changes as they tell you how you can expect the price to change in this scenario. Macaulay duration is a weighted average of times and its unit is years, whereas modified duration gives a percentage change in price for a 1% change in yield. A larger duration means the bond's price will be affected more by interest rates, so you can use these to see how badly your bonds will be affected; the modified duration could be more useful here since it gives you a relative measure. You can use the Macaulay duration of bonds to immunize a portfolio, where you try to form a portfolio with the same price and duration (and sometimes convexity too) as an obligation so that you are protected against interest rate changes. In the context of a pension plan, you are obligated to pay out a certain amount to customers.

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

The CAPM model is constructed as such:

$$E[R_i] = R_f + \beta_i(E[R_m] - R_f) \tag{30}$$

where $E[R_i]$ is the expected return of asset i, R_f is the risk-free rate and β_i is the sensitivity of the expected excess asset returns to the expected market returns, denoted $E[R_m]$. The sensitivity is defined as:

$$\beta_i = \frac{Cov(R_i, R_m)}{Var(R_m)} \tag{31}$$

2.4.1 Market returns

The market returns were estimated by taking the average of all stock returns on each day. Some stocks have missing entries and these are removed from the calculation. Figure 19 shows the returns for all companies over time, and the estimated market returns by taking an equally weighted average. The overall market return average was 4.72×10^{-5} per day, with a standard deviation of 0.006650.

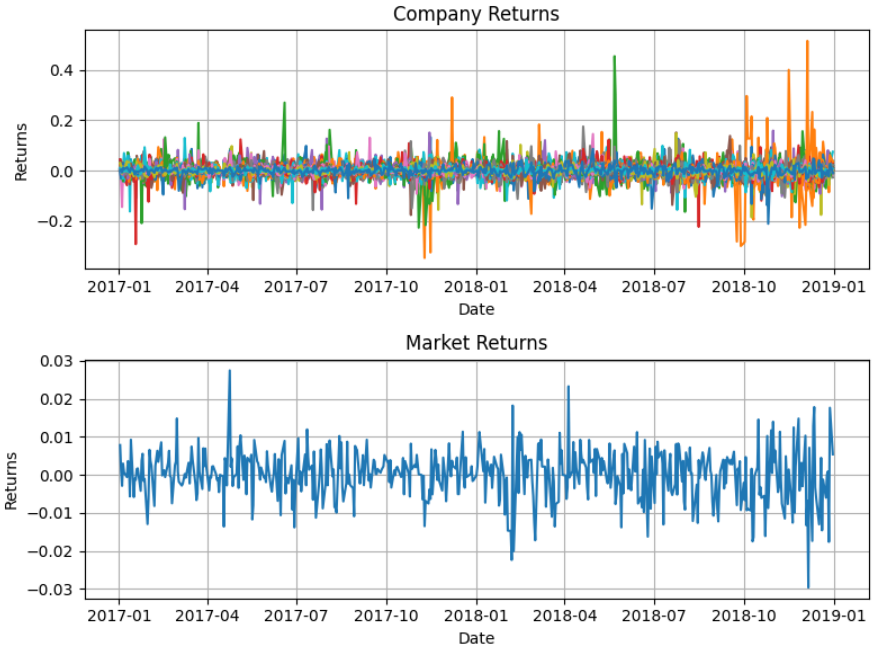

Figure 19: Market returns

2.4.2 Rolling window β

A rolling window size of 22 days (1 month) was used for this, and the formula given in equation 31 was used to calculate this value for each stock. Figure 20 shows the value of the beta over time for each stock, and a histogram of the standard deviation of all stock's beta over time. The β value of an asset is a measure it's volatility compared to the market as a whole. So a β of less than 1 shows the asset is less volatile than the market and would not add a lot of risk to a portfolio, although the returns are likely to be lower. The opposite goes for a value of greater than 1. A β of 1 indicates the asset is strongly correlated with the market so adding it to a portfolio would not add risk, but would not increase the chances of excess returns either. The market has a β of 1 since it is the 'benchmark' for the calculation. Most stocks have their β standard deviation less than 1, however there are stocks with a more volatile β, so most stocks keep a relatively stable value but some change a lot.

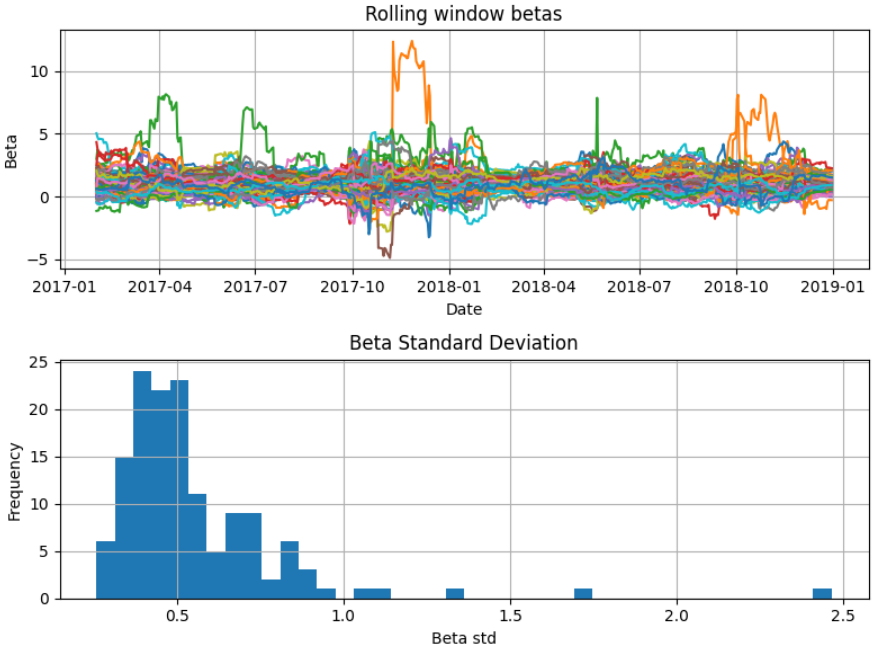

Figure 20: Betas and standard deviation of betas

2.4.3 Cap-weighted market return

The cap-weighted market return takes into account the market capitalization of each company when calculating the market return. It is defined like this:

$$R_m = \sum_i \frac{mcap_i \times ret_i}{\sum_i mcap_i} \tag{32}$$

where $mcap_i$ is the market cap of asset i, and the weighting coefficient is $\sum_i mcap_i$. Stocks with a lower market cap tend to be more volatile and therefore more risky, with greater potential for higher returns while stocks with a large market cap will fluctuate less, but yield smaller returns. The weighting coefficient means this is taken into account when calculating the market return, as shown in Figure 21 where we can see a slight reduction in volatility and magnitude. The overall weighted market return average was 1.88×10^{-4} with a standard deviation of 0.006599.

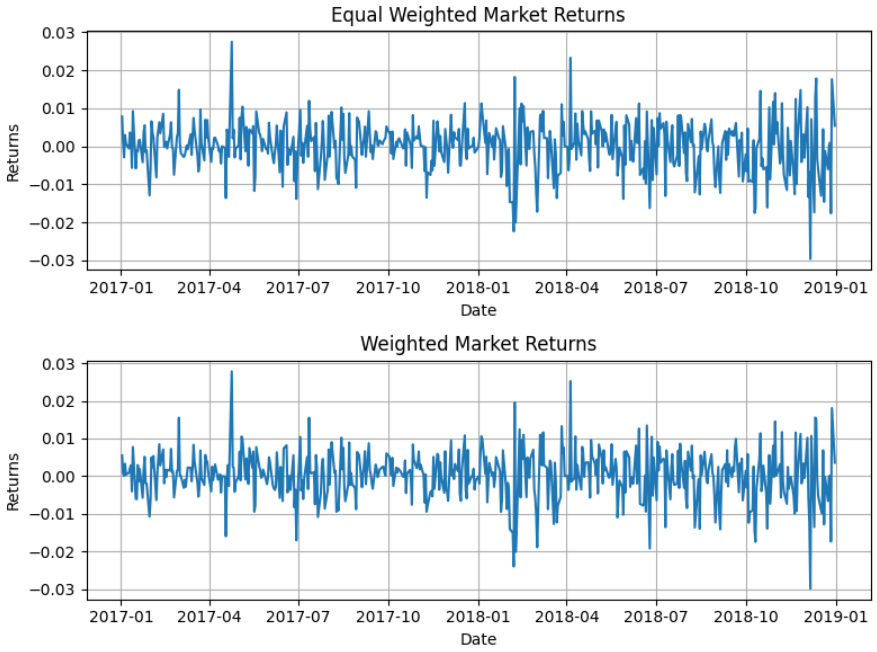

Figure 21: Weighted and equal-weighted market returns

2.4.4 Weighted rolling window β

The cap-weighted market return was used to recalculate β and the results are shown in Figure 22. The average β value fell to 0.961, compared to an average value of 1 for the equal weighted β in question 2. The standard deviation plot also shows a reduction in the volatility of the betas.

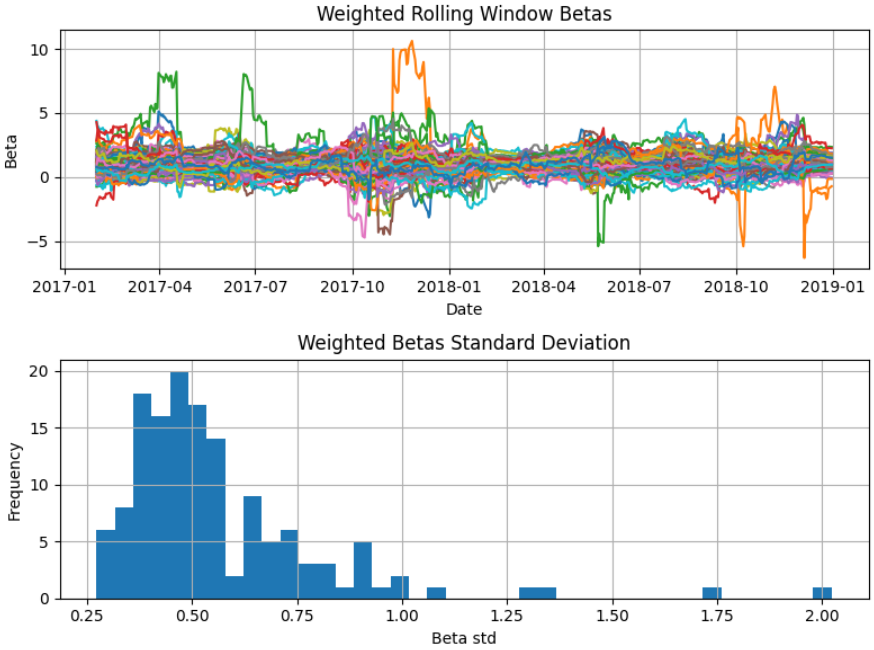

Figure 22: Cap-weighted rolling betas and their standard deviation

2.4.5 APT

Arbitrage pricing theory (APT) says that an asset's returns follow a factor structure. Here, we assume this hold for a two-factor model:

$$r_i = a + \beta_{mi} R_m + \beta_{si} R_s + \epsilon_i \tag{33}$$

where r_i is the return for asset i, ϵ_i is the residual, a is a constant, and β_{mi} and β_{si} are sensitivities to the factors R_m and R_s respectively.

2.4.5.a

To estimate a, R_m and R_s, a least squares regression can be used by going through each day individually. You can create the formula $Y = X\theta$, with the variables defined as follows:

$$Y = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}, \qquad X = \begin{bmatrix} \beta_m & \beta_s \end{bmatrix}, \qquad \theta = \begin{bmatrix} a \\ R_m \\ R_s \end{bmatrix} \tag{34}$$

where the above is written for 1 day. β_m was calculated in question 4 and β_s is the exposure to size and was calculated as the logarithm of the company's market cap. The solution to the regression is found by minimising $||X\theta - Y||_2^2$, and the result of this is $\hat{\theta} = (X^T X)^{-1} X^T Y$.

2.4.5.b

The magnitude and density distribution of the 3 parameters is shown in Figure 23. We can see that a has the largest magnitude and also the largest variance, followed by R_m and then R_s. So the return due to the market return is larger than the return due to the size factor. While the variance of R_s is the lowest, its magnitude is the lowest too, so relative to its size it has a sizeable variance.

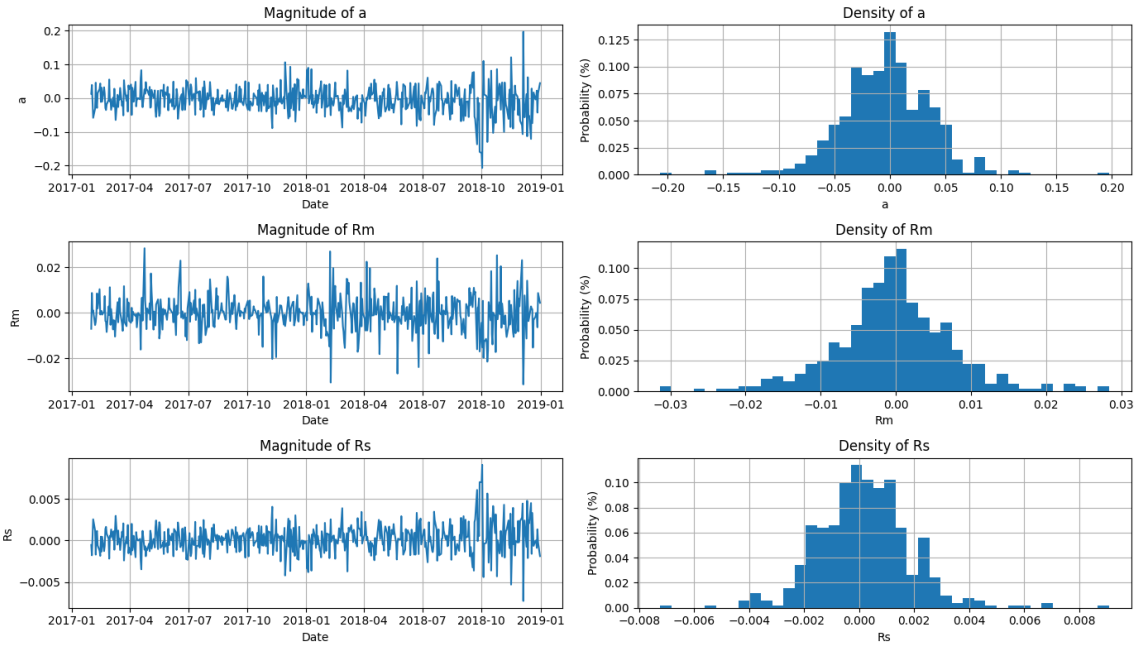

Figure 23: Magnitude and variance of a, R_m and R_s

2.4.5.c

The correlation between ϵ_i, the specific return, and r_i, the actual company return, was computed over time. Figure 24 shows a histogram of the correlations and we can see that the correlations are mostly around 0.8, indicating a strong positive correlation between the two. This means that there are not enough factors in the model to explain the returns; ideally the residuals (ϵ_i) are uncorrelated and random, meaning that the model has captured the systemic factors.

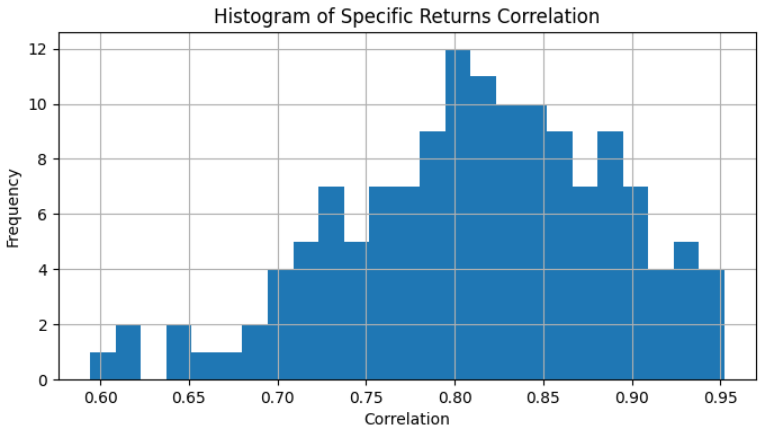

Figure 24: Specific and actual return correlation histogram

2.4.5.d

The matrix R was constructed by horizontally concatenating R_m and R_s such that they each form a column. The covariance of this matrix was found using a rolling window of 22 days, resulting in a 2x2 matrix for each day. Magnitude and stability information is contained in Figure 25. The magnitude of each covariance matrix is very small, but also fluctuates a lot. This means that there is little correlation between the two return vectors, which is ideal since uncorrelated factors are desirable. The condition number tends to be relatively high too, with the eigenvalues of order 10^{-5}. This indicates that the matrices are not very stable, but it is still possible to invert them all.

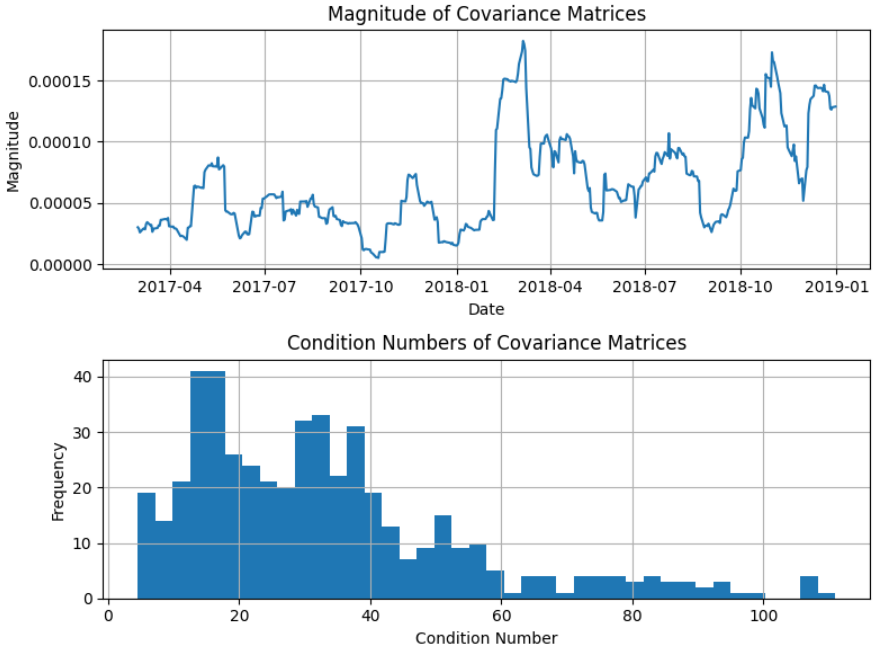

Figure 25: Magnitude and condition numbers of the covariance matrices

2.4.5.e

The matrix E was formed by taking the ϵ values from question 2.4.5.a. In the question PDF, it says that this should be 500x157, since there are 500 days remaining after the rolling window and 157 totals in the .csv file, but I removed stocks with missing (NaN) values at the beginning of this question, so I am left with 141 stocks. To perform principle component analysis (PCA) on E, first I found the covariance matrix, and then got the eigenvalues and eigenvectors of that. Figure 26 shows a scree plot and the variance that can be explained by differing numbers of components. Only 7.37% can be explained by the first component, and 12.39% by the first two. This again shows why the two factor model was insufficient. In fact, to explain 95% of the variance, 94 components are needed. The scree plot shows that after around the first 20 components, each added component starts to contribute less and this is reflected in the cumulative graph where we see a decrease in gradient here.

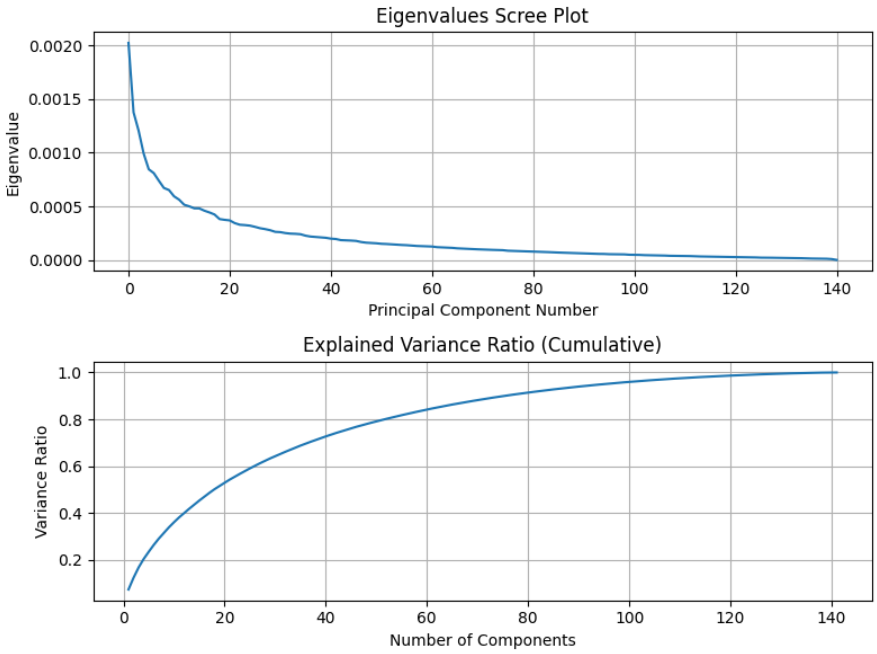

Figure 26: Scree plot and explained variance

3 Portfolio Optimization

3.1 Adaptive Minimum-Variance Portfolio Optimization

3.1.1 Minimum-variance weights and variance

The weights of a portfolio describe the proportion of capital allocated to each asset. The expected return of a portfolio is $\bar{\mu} = \mathbb{E}[\bar{r}[t]] = \mathbf{w^T}\boldsymbol{\mu}$, where w is the weights and $\bar{r}$ is the return of the portfolio at time t, calculated from the expected return of each asset. The variance of a portfolio is $\bar{\sigma}^2 = \mathbb{E}[(\bar{r}[t] - \bar{\mu})^2] = \mathbf{w^T C w}$, where $\mathbf{C}$ is the covariance matrix for the assets.

The optimal weights can be derived by solving the following problem (assuming shorting is allowed):

$$\min_w \quad J(\mathbf{w}, \mathbf{C}) = \frac{1}{2}\mathbf{w^T C w} \tag{35}$$

$$\text{subject to} \quad \mathbf{w^T 1} = 1 \tag{36}$$

This can be written as the following Lagrangian:

$$\min_{\mathbf{w},\lambda} \quad J(\mathbf{w}, \lambda, \mathbf{C}) = \frac{1}{2}\mathbf{w^T C w} - \lambda(\mathbf{w^T 1} - 1) \tag{37}$$

Taking the derivative with respect to $\mathbf{w}$ and λ, and setting to 0:

$$\frac{\partial J}{\partial \mathbf{w}} = \mathbf{C w} - \lambda \mathbf{1} = 0 \tag{38}$$

$$\mathbf{w} = \lambda \mathbf{C}^{-1}\mathbf{1} \tag{39}$$

$$\frac{\partial J}{\partial \lambda} = \mathbf{w}^T\mathbf{1} - 1 = 0 \tag{40}$$

$$\mathbf{w}^T\mathbf{1} = 1 \tag{41}$$

Substituting Equation 39 into 41 to find an expression for λ (the transpose of a covariance matrix is equal to the matrix itself):

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

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

$$\lambda = \frac{1}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}} \tag{44}$$

Therefore the optimal weights are:

$$\mathbf{w}^* = \frac{1}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}\mathbf{C}^{-1}\mathbf{1} \tag{45}$$

The theoretical variance of the returns if applying these weights is:

$$\sigma^2 = \mathbf{w^T C w} = \mathbf{1}^T\mathbf{C}^{-1}\frac{1}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}\mathbf{C}\frac{1}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}\mathbf{C}^{-1}\mathbf{1} \tag{46}$$

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

$$= \frac{1}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}\frac{1}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1} \tag{48}$$

$$= \frac{1}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}} \tag{49}$$

3.1.2 Equal vs minimum-variance portfolio

Data for 10 stocks was split 50/50 into train and test data, where the train data was 2017 and the train data was 2018. An equally weighted portfolio was created alongside a minimum variance portfolio, where the weights were determined on the train data using the result derived in the previous question. Figure 27 shows the returns and cumulative returns for the two portfolios on the train and test data. The returns for the minimum variance portfolio are slightly less volatile on the train and test data, as expected. On the train data, the cumulative returns are consistently superior over time for the minimum variance portfolio. However, on the test data the minimum variance portfolio performs worse for most of the year, with larger dips, although at the end the cumulative returns become slightly higher than the equal-weighted portfolio. Since the weights were calculated for a different time period (where the overall trend appeared to be upwards), the same weights when applied to the test period (where the trend appeared to be downward) no longer provided a consistently better performance than the equal-weighted portfolio.

The theoretical optimal variance from the minimum variance portfolio in the train data was 2.862×10^{-5}, and naturally this was achieved. However, in the test data, the minimum variance portfolio had a variance of 8.192×10^{-5}. Table 5 contains more detailed information; the mean and cumulative returns, variance figures and Sharpe ratios confirm my earlier statements (the Sharpe ratios were calculated without a risk-free rate, since I am just comparing them).

Figure 27: Equal and minimum variance portfolio returns and cumulative returns

Table 5: Portfolio Performance Analysis

Portfolio	Mean Return	Cumulative Return	Variance	Sharpe Ratio
Train Analysis				
Equal Weighted	0.000860	0.223484	0.000037	0.140374
Minimum Variance	0.000958	0.249062	0.000029	0.179072
Test Analysis				
Equal Weighted	-0.000473	-0.123503	0.000079	-0.053126
Minimum Variance	-0.000451	-0.117582	0.000082	-0.049774

3.1.3 Adaptive minimum-variance portfolio

An adaptive minimum-variance portfolio was created by using empirical estimates of the mean vector and covariance matrix with rolling windows. Rolling window sizes of 22, 66, 132 and 252 days (corresponding to 1, 3, 6 and 12 months) were used to assess the effect changing the size has. Figure 28 contains the cumulative returns for these window sizes. The smaller the rolling window size, the better the cumulative return performance over time with the 1 month window having the best return. It is worth pointing out that in reality, the frequent

position size changes caused by changing weights would incur transaction costs and there are also additional costs associated with shorting such as stock borrowing fees, so the actual returns would be less.

The recursive weight updates allows the model to react to shorter term trends and means that where the equally weighted and non-adaptive minimum-variance portfolio generated negative cumulative returns, the 1 and 3 month window portfolios generated positive returns. The estimate of the covariance matrix uses more recent, relevant data so it can reflect shorter-term market dynamics and adjust to changes in asset relationships quicker. However, there should be a balance between wanting to uphold longer term market trends and act on the shorter term ones. Also, this approach does not necessarily improve the model's ability to predict future returns as this depends on how well past trends and asset covariance relate to future ones.

The exact performance figures are in Table 6, where we can see the mean returns, cumulative returns and Sharpe ratios are better for all time windows than the equal-weighted and non-weighted minimum-variance portfolios. However, the variances, particularly for the shorter windows, are higher due to the windowing and this can be seen in the more severe spikes in the shorter window cumulative return graphs. This could be due to factors previously mentioned and there also being less data points used in the calculations, meaning the data is noisier.

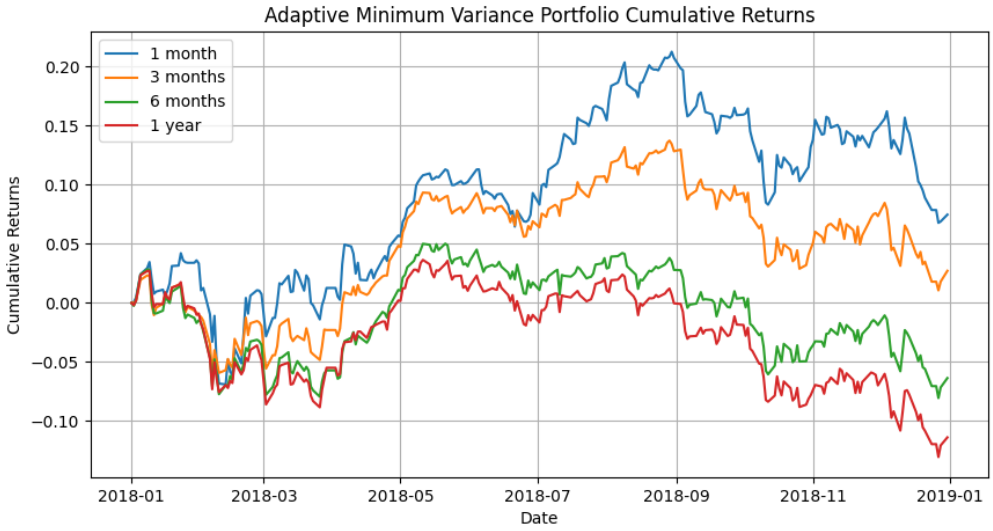

Figure 28: Cumulative returns for different window sizes

Table 6: Adaptive Portfolio Analysis for Different Window Sizes

Window Size	**Mean Return**	**Cumulative Return**	**Variance**	**Sharpe Ratio**
1 month	0.000286	0.074774	0.000108	0.027537
3 months	0.000104	0.02719	0.000074	0.012113
6 months	-0.000244	-0.063607	0.00007	-0.029152
12 months	-0.000436	-0.113804	0.000066	-0.053534

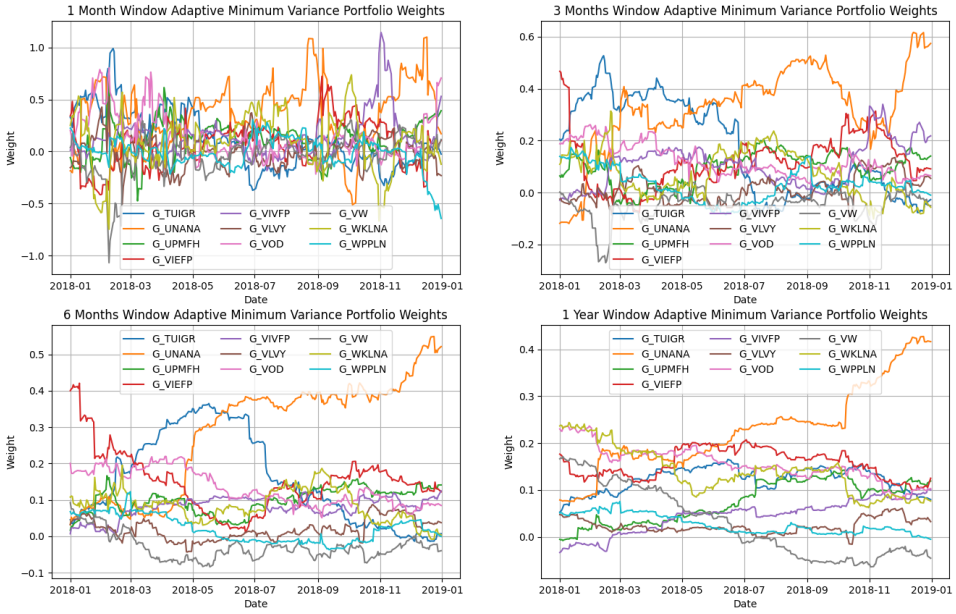

Figure 29: Weights per stock for different window sizes

4 Robust Statistics and Non Linear Methods

4.1 Data Import and Exploratory Data Analysis

4.1.1 Descriptive statistics

Key descriptive statistics were generated for three stocks, AAPL, IBM and JPM, and the DJI index. The following statistics were calculated for each:

- Mean: The average value, indicating the central tendency of the data.
- Median: Also indicates central tendency, but is less affected by outliers.
- Maximum: The highest value
- Minimum: The lowest value in the dataset.
- 25th percentile: A measure of dispersion - the first quartile, below which 25% of the data lies.
- 75 percentile: A measure of dispersion - the third quartile, below which 75% of the data lies.
- Interquartile Range (IQR): The range between the first and third quartiles, measuring the spread of the middle 50% of the data.
- Standard Deviation: A measure of data dispersion around the mean.
- Skewness: Indicates the asymmetry of the data distribution.
- Kurtosis: Measures the how heavy or light tailed the distribution is.

The statistics are shown in Table 7.

Table 7: Descriptive Statistics for Selected Stocks

Statistic	Open	High	Low	Close	Adj Close	Volume
			AAPL			
Mean	187.69	189.56	185.82	187.71	186.17	32704750.20
Median	186.29	187.40	184.94	186.12	184.35	29184000.00
Max	230.78	233.47	229.78	232.07	230.28	96246700.00
Min	143.98	145.72	142.00	142.19	141.58	12513900.00
25%	171.32	173.04	169.72	171.00	170.21	22887350.00
75%	207.32	209.38	205.78	207.76	205.90	39199050.00
IQR	36.00	36.34	36.06	36.76	35.69	16311700.00
Stddev	22.15	22.28	22.01	22.16	21.90	14179721.59
Skewness	0.26	0.30	0.22	0.26	0.29	1.74
Kurtosis	-0.91	-0.92	-0.92	-0.93	-0.93	4.35
			IBM			
Mean	138.45	139.49	137.33	138.36	134.90	5198937.45
Median	142.81	143.99	142.06	142.71	138.57	4237900.00
Max	160.06	162.00	159.64	160.91	153.67	22063700.00
Min	108.00	111.00	105.94	107.57	106.33	1963200.00
25%	131.27	132.56	129.22	130.88	127.75	3436050.00
75%	146.65	147.28	145.57	146.39	141.85	5389000.00
IQR	15.38	14.72	16.34	15.50	14.10	1952950.00
Stddev	12.11	11.91	12.20	12.03	10.67	3328955.53
Skewness	-0.68	-0.62	-0.71	-0.68	-0.81	3.19
Kurtosis	-0.59	-0.62	-0.56	-0.58	-0.42	11.80
			JPM			
Mean	108.71	109.65	107.68	108.61	107.26	14700689.24
Median	109.18	110.53	107.79	109.02	107.22	13633000.00
Max	119.13	119.24	118.08	118.63	116.86	41313900.00
Min	92.69	94.22	91.11	92.14	91.40	6488400.00
25%	104.55	105.42	103.70	104.51	104.14	10812900.00
75%	113.37	114.26	112.54	113.34	111.36	17046500.00
IQR	8.81	8.85	8.85	8.83	7.22	6233600.00
Stddev	5.36	5.20	5.43	5.30	4.83	5349770.56
Skewness	-0.42	-0.38	-0.38	-0.37	-0.34	1.69
Kurtosis	-0.32	-0.54	-0.27	-0.40	-0.11	4.43
			DJI			
Mean	25001.26	25142.04	24846.00	24999.15	24999.15	332889442.23
Median	25025.58	25124.10	24883.04	25044.29	25044.29	313790000.00
Max	26833.47	26951.81	26789.08	26828.39	26828.39	900510000.00
Min	21857.73	22339.87	21712.53	21792.20	21792.20	155940000.00
25%	24485.61	24614.27	24275.59	24433.09	24433.09	270100000.00
75%	25595.04	25692.09	25480.01	25591.25	25591.25	379030000.00
IQR	1109.43	1077.82	1204.42	1158.16	1158.16	108930000.00
Stddev	858.83	815.20	903.30	859.13	859.13	94078038.14
Skewness	-0.37	-0.24	-0.46	-0.38	-0.38	1.74
Kurtosis	0.49	0.12	0.56	0.40	0.40	5.86

4.1.2 Densities of adjusted close and 1-day returns

Histograms and the PDF of the adjusted close price and 1-day returns were plotted for AAPL, IBM, JPM and DJI. The PDF was approximated using Seaborn's non-parametric kernel density estimation (KDE). The bandwidth (window size) was adjusted to strike a balance with smoothness. The results are shown in Figure 30. AAPL is the most volatile, with the largest variation in close price and returns. It also has three clear peaks in it's price PDF. IBM is the next most volatile, followed by DJI and JPM. Given DJI is an index, a lower volatility is expected. The return plots resemble a more typical bell-curve. This could be due to the fact returns are stationary, but prices are not so their distribution can change over time. It is often assumed returns are mostly normally distributed due to the efficient market hypothesis, although in practice this is not the case, as seen by the skew and tails in some of the return PDFs and histograms.

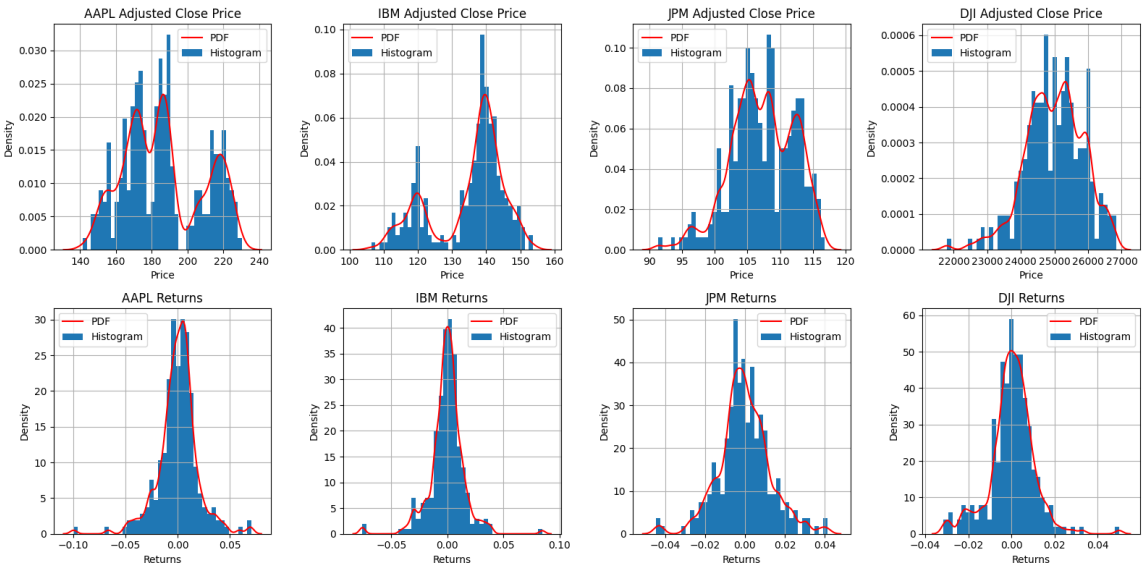

Figure 30: Histograms and PDFs for the stocks and index

4.1.3 Rolling mean vs rolling median

The rolling mean and rolling median were plotted for the adjusted close price using a 5-day window. For the rolling mean, mean ± 1.5 x standard deviation was also plotted, and for the rolling median, median ±1.5 x median absolute deviation (MAD) was plotted. The plots are shown in Figures 31 and 32. The adjusted close prices outside of the cyan shaded regions can be considered as outliers to typical price movements. The 'window' of inliers is tighter for MAD compared to the rolling mean standard deviation method, since the mean is less robust to outliers compared to the median. These plots highlight the difference in robustness between the two metrics, which can be seen clearly in DJI plots just before January 2019. This is also reflected in Table 8.

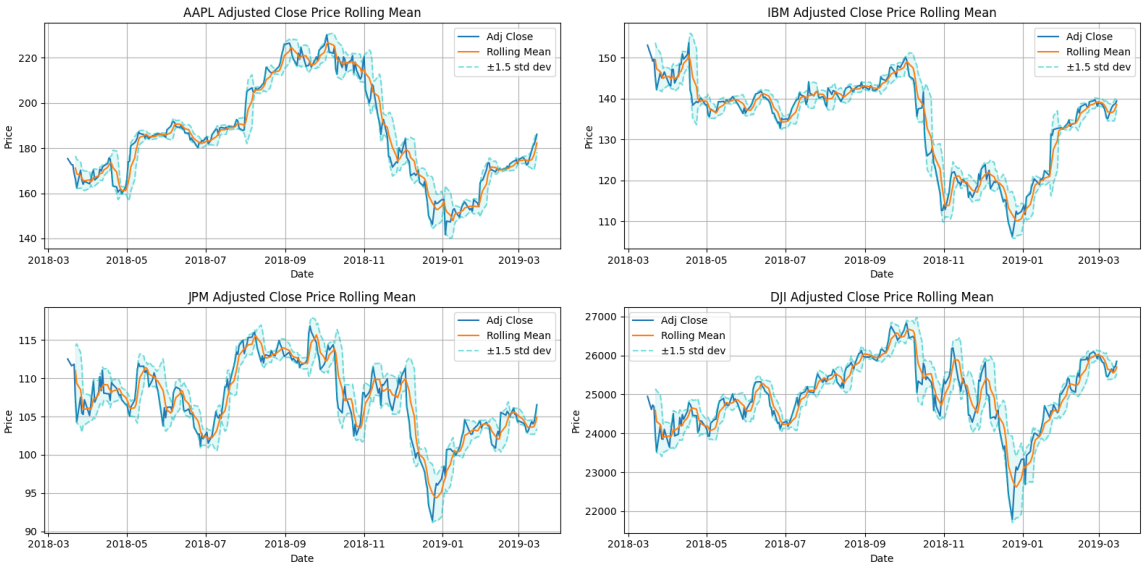

Figure 31: Rolling mean and outlier detection

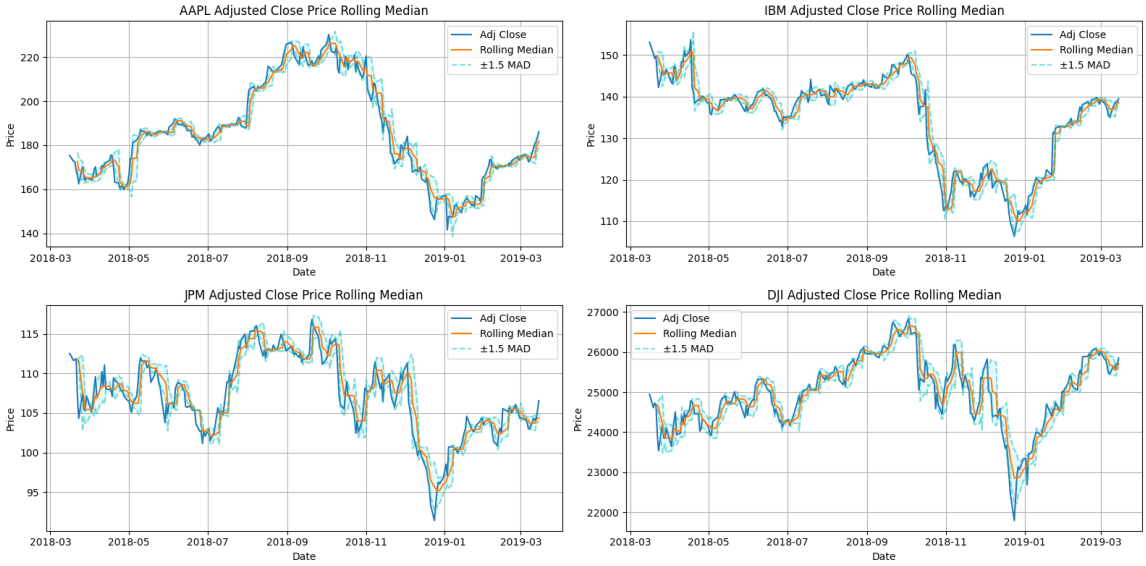

Figure 32: Rolling median and outlier detection

Stock	Outliers (Mean Method)	Outliers (Median Method)
AAPL	30	103
IBM	31	94
JPM	33	105
DJI	30	97

Table 8: Outliers identified using the mean and median methods

4.1.4 Introducing outlier points

Four artificial outliers were added at the dates 2018-05-14, 2018-09-14, 2018-12-14 and 2019-01-14. The same analysis as the previous question was done and is shown in Figures 33 and 34, and the number of outliers is in Table 9. The superior robustness of the rolling median is again highlighted as we do not see the cyan window spike for the median, but it does with the mean. The outlier count increased by 1 for the mean and decreased by 6 for the median.

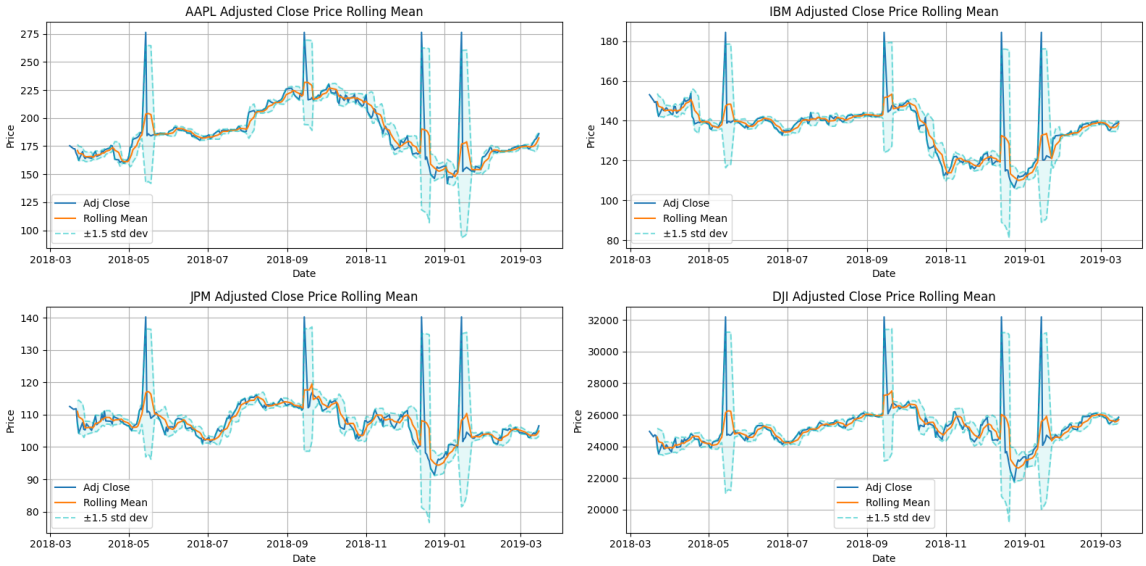

Figure 33: Rolling mean with added outliers

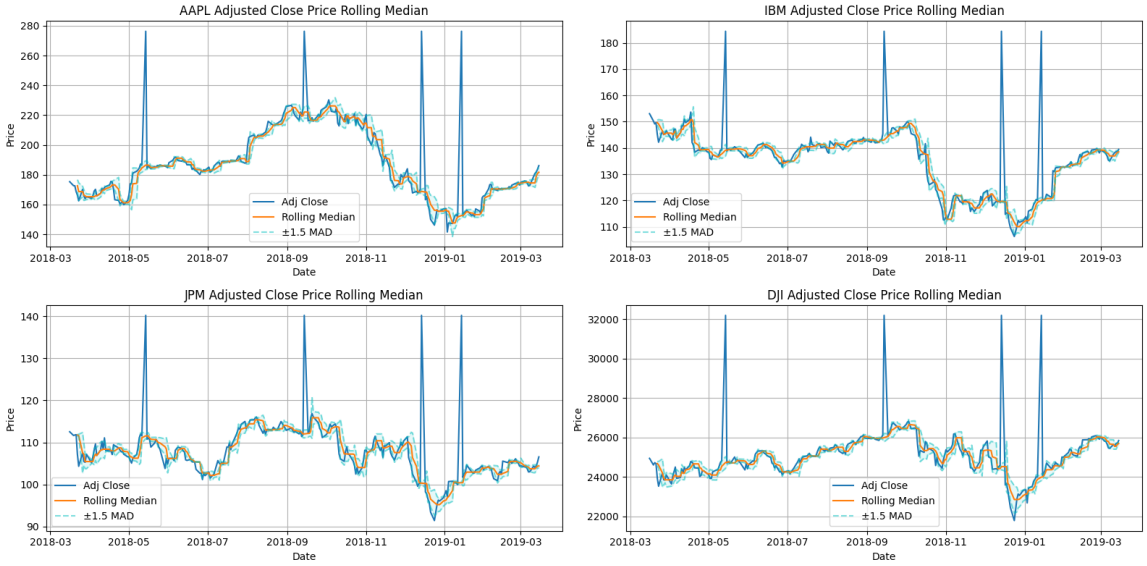

Figure 34: Rolling median with added outliers

Stock	Outliers (Mean Method)	Outliers (Median Method)
AAPL	32	102
IBM	31	93
JPM	33	101
DJI	29	96

Table 9: Outliers identified using the mean and median methods with added outliers

4.1.5 Box plots

Figure 35 shows box plots for the adjusted close price for AAPL, IBM, JPM and DJI. The orange line shows the median, the green dotted line shows the mean, the left of the black box is the 25th percentile and the right is the 75th percentile, meaning the box represents the middle 50% of the data with the width being the IQR. The blue lines extend out to the whiskers which are ±1.5 x IQR, and the circle points beyond these lines are outliers from this range. The 'notches' represent a 95% confidence interval for the median. When comparing medians, if the notches no not overlap then you can conclude that they differ with 95% confidence, and this is the case here. All of the plots display a skew, with IBM having the largest where the median is clearly towards the upper quartile (meaning a negative skew), and this is consistent with the values in Table 7. JPM and DJI's mean and median are nearly identical, indicating a the data may be somewhat symmetrically distributed but these values clearly differ for AAPL and IBM. This aligns with the plots in Figure 30. AAPL has the largest relative IQR, meaning its middle 50% is more spread out and this is a sign of greater volatility. It also has no outliers, meaning that whilst it showed the most volatility, the price movements may have been slightly less erratic than the others. However, this could also be possibly due to the larger IQR granting a higher threshold for outlier detection.

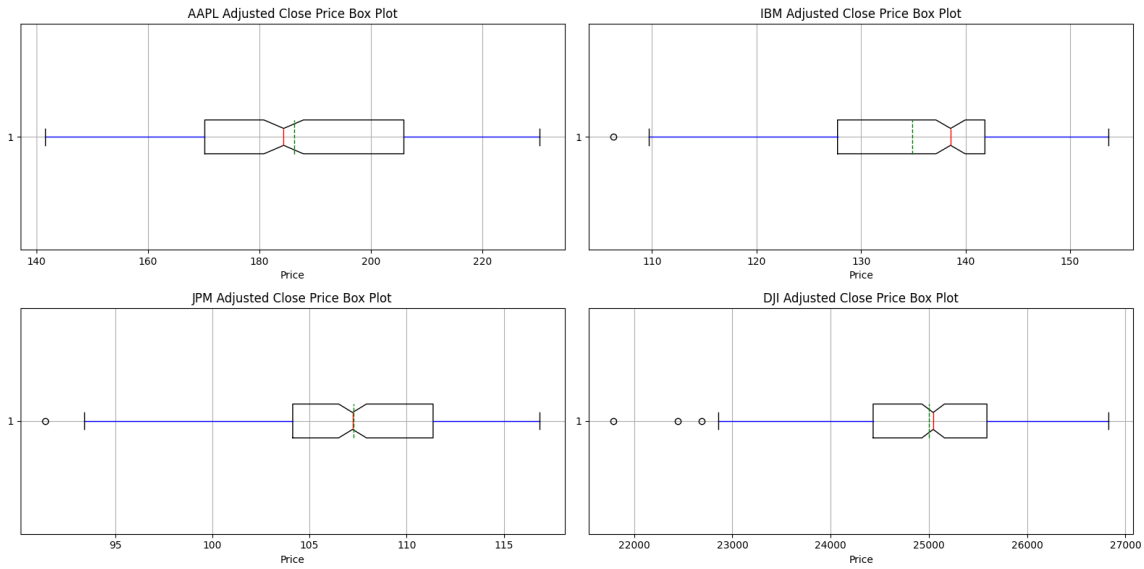

Figure 35: Box plots for the adjusted close prices

4.2 Robust Estimators

4.2.1 Implementing median, IQR and MAD

I implemented custom versions for calculating the median, IQR and MAD. Below is the code for the median, where the distinction is made between needing to just select the middle value for an odd length array and take the average of the middle two values for an even length array:

```
def calc_median(x):
    sorted = x.sort_values()
    n = len(sorted)
    if n % 2 == 0:
        median = (sorted.iloc[n//2] + sorted.iloc[n//2 - 1]) / 2
    else:
        median = sorted.iloc[n//2]
return median
```

The code for calculating the IQR is much longer so only the part for odd length arrays is presented below. I included an option for using the exclusive or inclusive method to calculate the IQR, where the exclusive version does not use the middle value but the inclusive does. Scipy's IQR function when using midpoint interpolation seems to use the inclusive method with odd length arrays. My IQR version for even length arrays includes an inclusive version for completeness, but this is usually not used in practice.

```
if exclusive:
    lowerhalf = sorted[:n//2]
    upperhalf = sorted[n//2+1:]
    q1 = lowerhalf[n//4]
    q3 = upperhalf[n//4]
    IQR = q3 - q1
else:
    median = sorted[n//2]
    lowerhalf = np.append(sorted[:n//2], median)
    upperhalf = np.insert(sorted[n//2+1:], 0, median)
    q1 = calc_median(pd.Series(lowerhalf))
    q3 = calc_median(pd.Series(upperhalf))
    IQR = q3 - q1
```

Calculating the MAD is simpler and is shown below. It finds the median of the absolute deviation from the median, and uses my custom function to calculate the median.

```
def calc_MAD(x):
```

```
median = calc_median(x)
MAD = calc_median((x - median).abs())
```

Table 10 contains a comparison between my estimators and Panda's median implementation, and Scipy's IQR and MAD implementations. All of my versions yield the same results when using inclusive IQR since this seems to be the default for Scipy as mentioned earlier.

Stock	Median	My Median	IQR	My IQR (exclusive)	My IQR (inclusive)	MAD	My MAD
AAPL	184.35	184.35	35.69	35.72	35.69	15.48	15.48
IBM	138.57	138.57	14.10	14.44	14.10	4.49	4.49
JPM	107.22	107.22	7.22	7.28	7.22	3.45	3.45
DJI	25044.29	25044.29	1158.16	1175.48	1158.16	590.72	590.72

Table 10: Comparison of estimators

4.2.2 Computational efficiency

For the median, the input has to be sorted first which is $O(nlog(n))$, where n is the length of the input array, since Pandas uses 'mergesort'. Calculating the median itself is just some lookups and arithmetic, which can be assumed to be $O(1)$, therefore the overall complexity for this function is $O(nlog(n))$.

The IQR also needs to sort the input first which is $O(nlog(n))$. Following this there are two array splits for the upper and lower halves. The median function is called on these two arrays and each of these is $O(nlog(n))$. Calculating the final value is a subtraction which is $O(1)$, so the final complexity for this function is $O(nlog(n))$.

The MAD does not need to sort values, but it does involve calling the median function twice, and some arithmetic and comparisons. The median function calls here dominate the complexity, so the complexity of the MAD function is also $O(nlog(n))$.

4.2.3 Breakdown points

The breakdown point of an estimator is the maximum fraction of outliers that an estimator can tolerate and it is defined between 0 and 0.5, since the proportion of outliers cannot exceed 0.5 by definition. Since the median is the middle of a sorted array, it can have $\frac{\frac{n-1}{2}}{n}$ values that will not affect the median. This can be rewritten as $\frac{1}{2} - \frac{1}{2n}$, and as n goes to infinity, this becomes $\frac{1}{2}$. Therefore, the breakdown point of the median is 0.5. The IQR effectively involves the median but on arrays half the length of the input, so its breakdown point is 0.25. Calculating the MAD requires a calculation of the median, and then the median on the absolute deviations. There is nothing here to worsen the breakdown point compared to the median, therefore the breakdown point of the MAD is 0.5.

4.3 Robust and OLS regression

4.3.1 OLS regression

Each stock's one-day return was regressed again DJI's returns using OLS. The regression takes the form $Y = X\theta + \epsilon$, where Y is stock returns vector, X contains a column of ones and another column of DJI's returns, and ϵ is the vector of errors. The solution takes the form $\theta = (X^T X)^{-1} X^T Y$. I did this manually in question 2.4.5, so this time I used Sklearn's LinearRegression. The results are shown in Figures 36 and 37. The R^2 value for JPM is the highest, but none of the predicted returns are particularly accurate; this could be due to the model not being complex enough to capture the underlying dynamics, or due to the outliers that we saw earlier in Section 4.

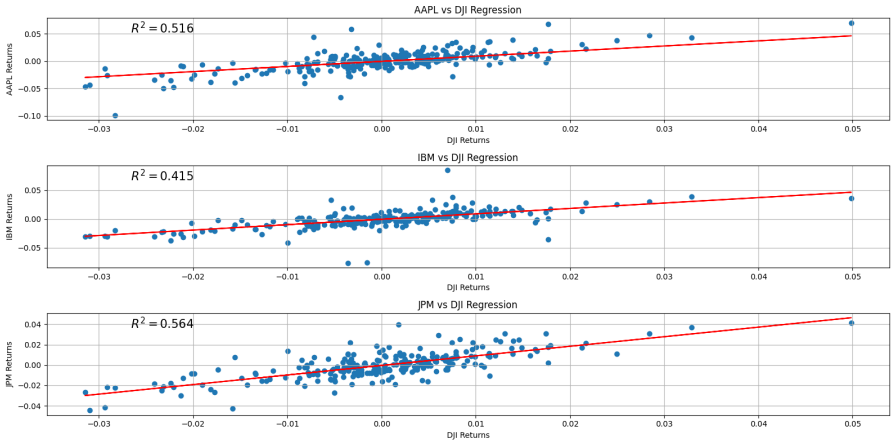

Figure 36: Stocks vs DJI OLS regression

Figure 37: Predicted stock returns from OLS regression

4.3.2 Huber Regression

Using Huber regression mitigates the effect of outliers on the regression. There is an error threshold, σ, which determines whether to use the usual L2 loss, or whether to use an L1 loss. If $Y - X\theta > \sigma$, you use the L2 loss, else you use the L1 loss. This lessens the impact of large outliers since the large errors caused by these are now put into a linear rather than exponential function. You are able to fine-tune the threshold for which a point is considered an outlier with σ. The results are shown in Figures 38 and 39, where we see little has changed with the R^2 values, or the returns prediction. Therefore, the cause of the poor prediction is likely due to the lack of complexity in the model, since this model suppresses the effect of outliers. I tried changing σ, but there was a very small change in the results so I left it as the default value. The final parameter values are shown in Table 11, where we again see a small change.

Figure 38: Predicted stock returns from Huber regression

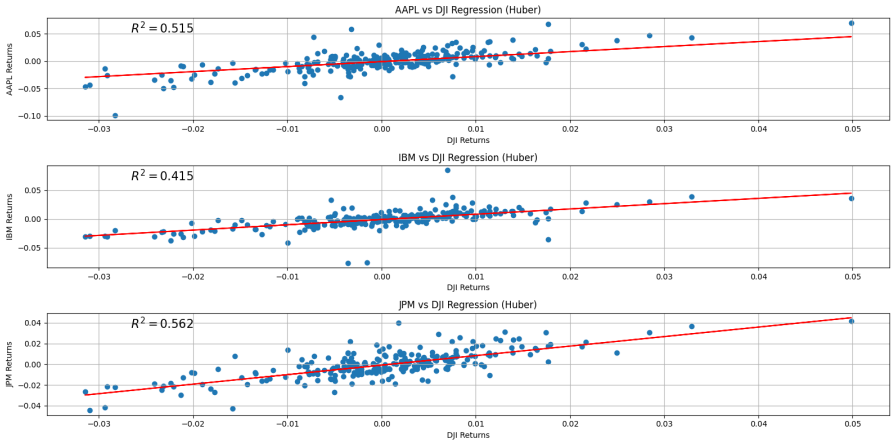

Figure 39: Stocks vs DJI Huber regression

Stock	OLS		Huber	
	Beta	Alpha	Beta	Alpha
AAPL	1.326490	0.000104	1.280361	0.000066
IBM	0.961954	-0.000624	0.954721	-0.000471
JPM	0.938374	-0.000420	0.936419	-0.000554

Table 11: Comparison of OLS and Huber parameters

4.3.3 Effect of outliers

The effect of adding outliers to OLS and Huber regressions was tested by artificially two vertical outliers and two leverage outliers separately, and performing the regressions on these two cases. The vertical outliers were created through adding points on the stock returns, and the leverage outliers were created by adding points on the index returns. The results are shown in Figures 40 and 41. The OLS regression was less affected by vertical outliers than leverage outliers, which had a drastic effect. The leverage outliers had significantly less of an effect on the Huber regression and the vertical outliers had a small effect. The larger effect from leverage outliers can be explained by the fact that vertical outliers lie in the same x-axis space, but leverage outliers live in a different x and y-axis space. This is exacerbated by the exponential nature of the L2 loss used in OLS, but this is mitigated in the Huber loss. Table 12 shows the parameters for the regressions with the two types of outliers. Compared to the original parameters in Table 11, the Huber parameters in general have changed less.

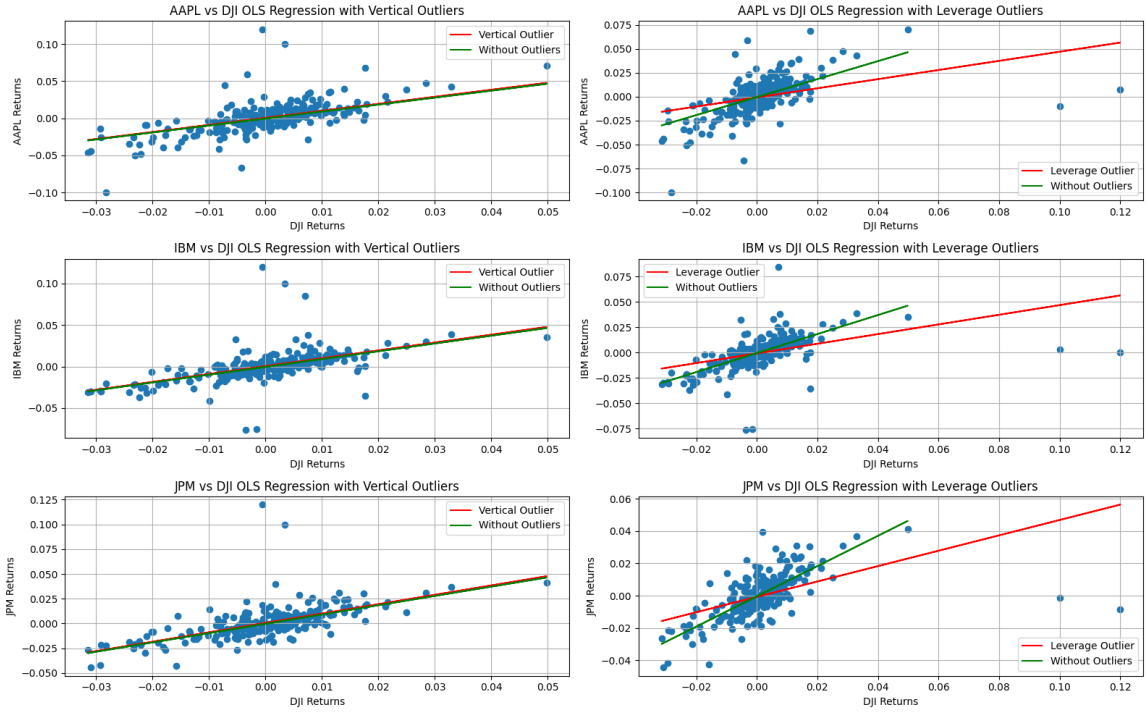

Figure 40: Stocks vs DJI OLS regression with outliers

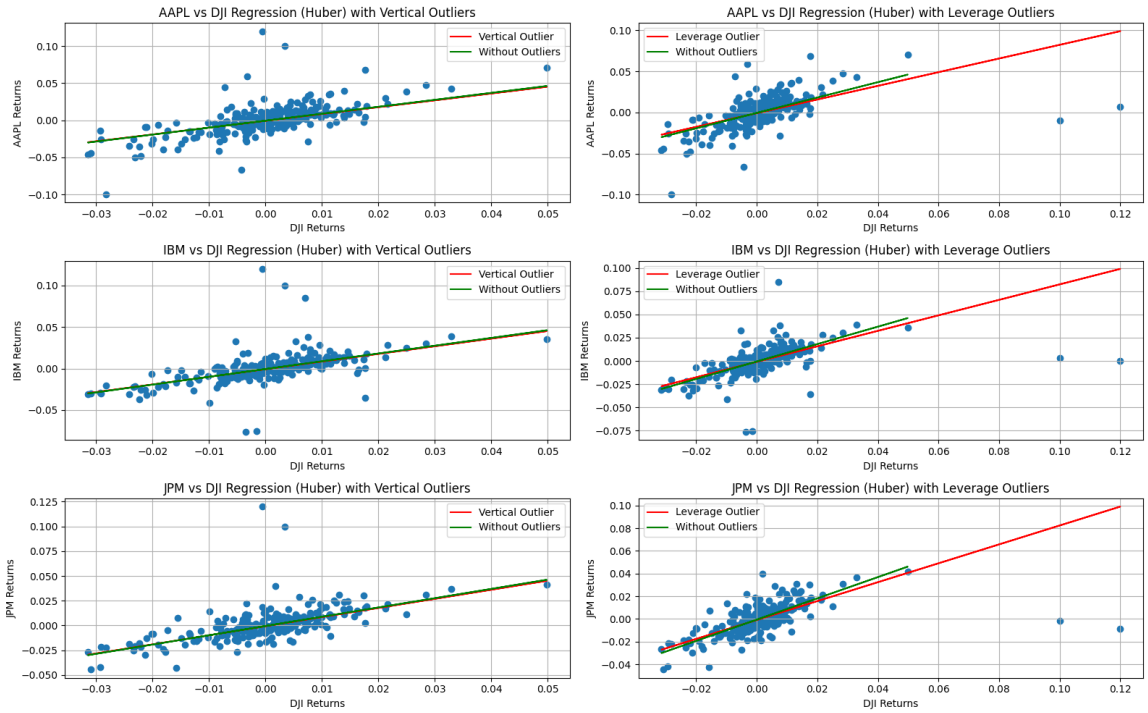

Figure 41: Stocks vs DJI Huber regression with outliers

Stock	OLS				Huber			
	Vertical Outliers		Leverage Outliers		Vertical Outliers		Leverage Outliers	
	Beta	Alpha	Beta	Alpha	Beta	Alpha	Beta	Alpha
AAPL	1.336362	0.000993	0.701762	-0.000382	1.275623	-0.000085	1.139413	-0.000223
IBM	0.970082	0.000241	0.519353	-0.000988	0.972691	-0.000574	0.887819	-0.000701
JPM	0.946763	0.000499	0.476620	-0.000743	0.918557	-0.000733	0.832299	-0.000879

Table 12: Comparison of regression parameters with outliers

4.4 Robust Trading Strategies

4.4.1 Moving average

The moving average (MA) crossover trading strategy dictates that when the 20-day MA > 50-day MA, you long (buy) the asset, and when the 20-day MA < 50-day MA, you short (sell) the asset. The 20 and 50 day MAs were plotted for the adjusted close prices of AAPL, IBM, JPM and DJI. The same was also done for corrupted close prices, with random outliers added. The results are in Figure 42 with the buy and sell sections highlighted in green and red respectively, and the crossover points annotated with green and red arrows indicating to buy and sell. The presence of outliers has affected the strategy, which is clear to see when looking at the number of arrows present in the outlier plots compared to the non-corrupted returns.

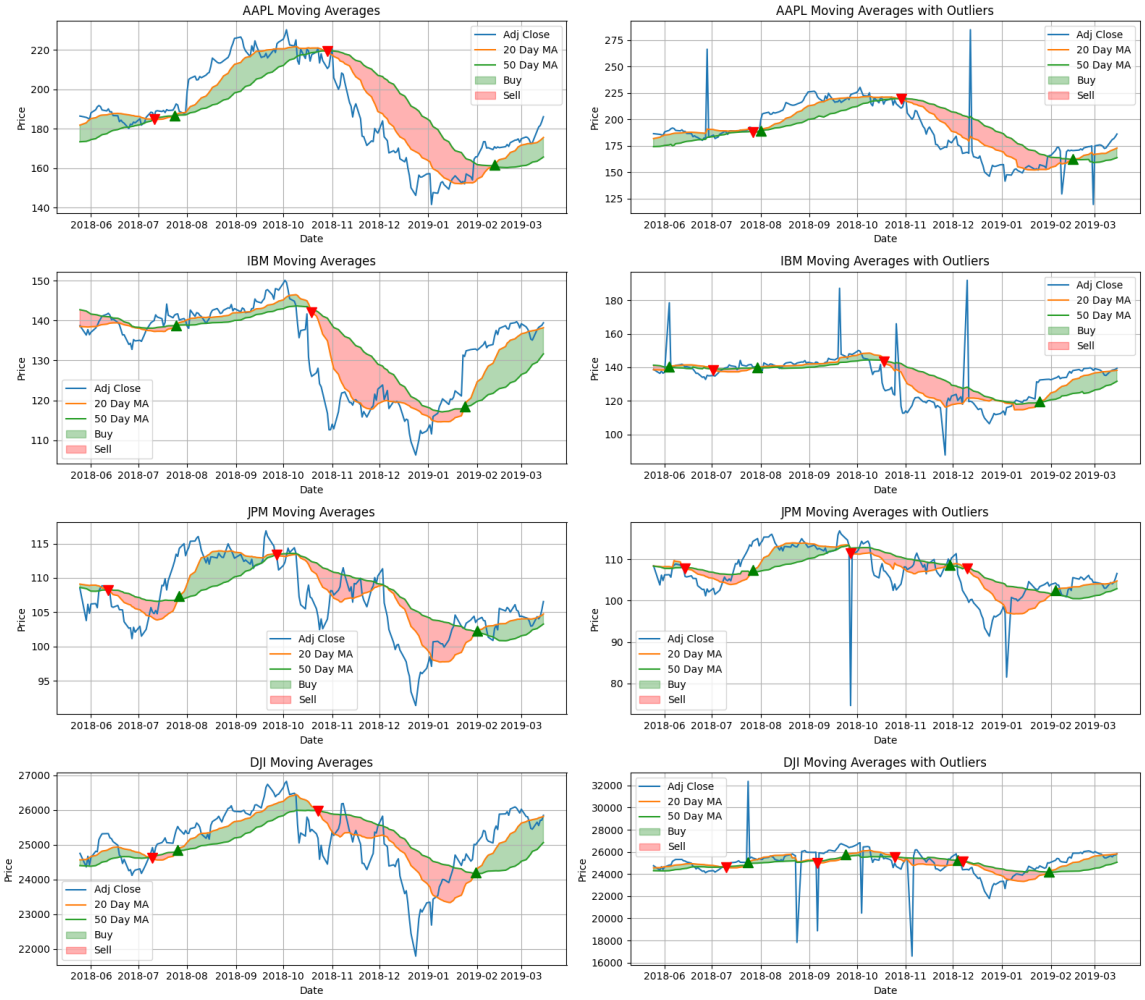

Figure 42: 20 and 50 day MAs with and without outliers

4.4.2 Moving median

The same was done using a 20 and 50-day moving median (MM), and the results of this are shown in Figure 43. As seen previously in section 4, the median is more robust than the mean; this time only IBM was badly

affected by the outliers since it is the only asset to have additional buy and sell indicators, whereas with the MA strategy, IBM, JPM and DJI all had additional indicators. Table 13 shows the difference in proportion of time spent in the buy position between the normal close and corrupted close prices. The difference in sell time is just the negative of the difference in buy time, so is excluded for simplicity. This table quantifies that the MM was less affected by the outliers and is more robust since it had smaller changes, in fact, the MM strategy for JPM was completely unaffected by outliers.

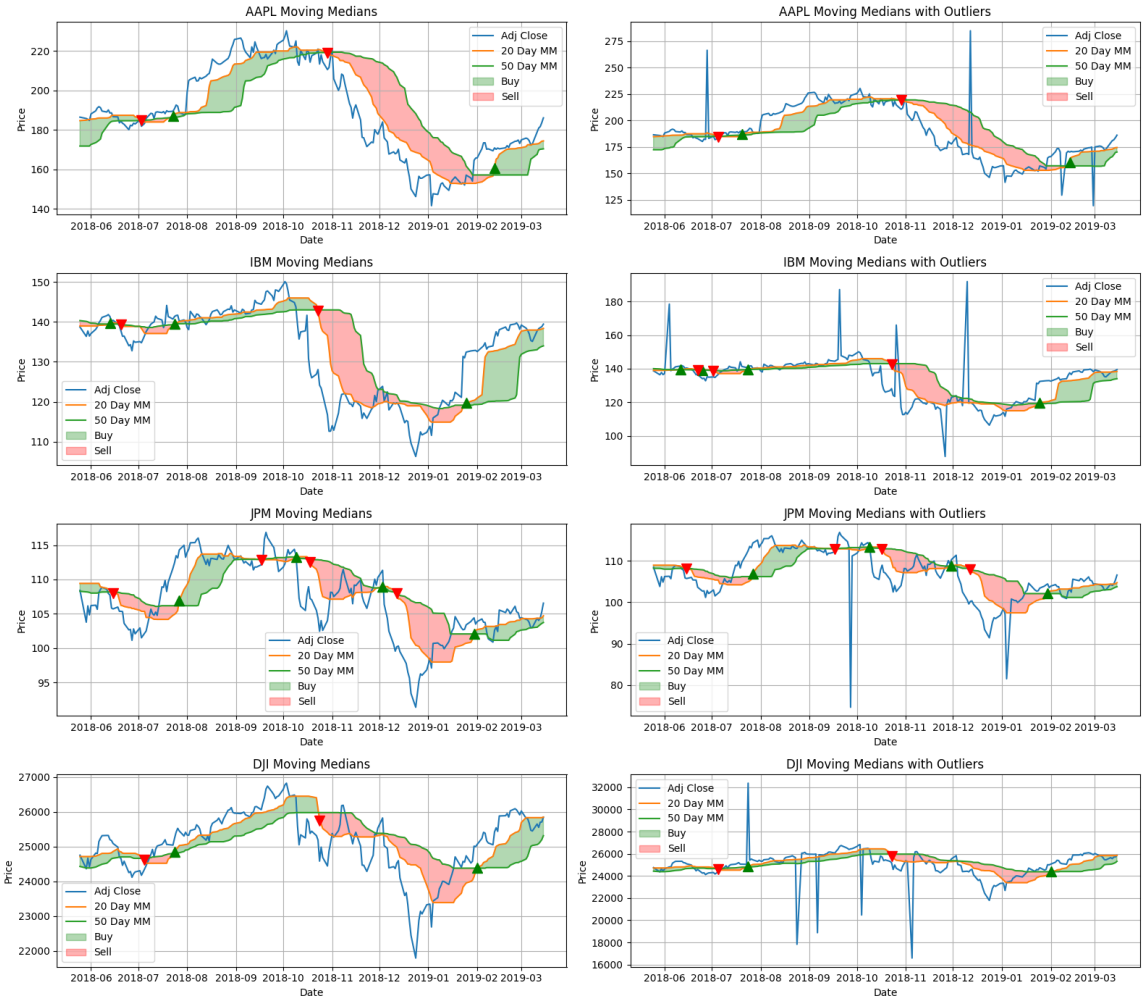

Figure 43: 20 and 50 day MMs with and without outliers

Stock	Moving Mean	Moving Median
AAPL	-0.014851	-0.004950
IBM	-0.074257	-0.044554
JPM	-0.034653	0.000000
DJI	0.029703	0.004950

Table 13: Difference in time spent in buy between corrupted and uncorrupted close prices

5 Graphs in Finance

5.1 Choosing Stocks

This section involves apply graphs to modelling and visualizing the relationships between stocks with the S&P 500. I chose to analyse stocks within the 'Information Technology' GICS sector, more specifically stocks that are semiconductor related. Due to the relational nature of graphs, I thought it would be interesting to also see the relationships between the 'Semiconductors' and 'Semiconductor Equipment' GICS sub-industry. I noticed that most of these stocks now were headquartered in California, so I chose all of the California-based stocks

to eliminate one variable from the relationships between the 'Semiconductors' and 'Semiconductor Equipment' sub-industries. However, I still wanted to look at geographical relationships, and I saw that there were two 'Semiconductors' stocks in Massachusetts, so I added these. I added two more 'Semiconductors' stocks that were based elsewhere (one in Idaho and the other in Texas) for further geographical analysis. However, I now had more than 10 stocks, so I used the Internet Archive's Wayback Machine to access a site giving the weights of each S&P 500 component in 2017. By webscraping this webpage, I could effectively sort the stocks by market cap, and I excluded the lowest weighted stocks to arrive at the final list of 10 stocks. The reason for excluding these are the higher-weighted stocks are likely to have a greater influence in market trends and be more representative of the sector as a whole. The table below shows the information for my selected stocks:

Security	Symbol	GICS Sector	GICS Sub Industry	Headquarters Location	S&P Weight
Analog Devices, Inc.	ADI	Information Technology	Semiconductors	Norwood, Massachusetts	0.137413
Applied Materials Inc.	AMAT	Information Technology	Semiconductor Equipment	Santa Clara, California	0.213898
Broadcom	AVGO	Information Technology	Semiconductors	San Jose, California	0.449496
Intel Corp.	INTC	Information Technology	Semiconductors	Santa Clara, California	0.764670
Lam Research	LRCX	Information Technology	Semiconductor Equipment	Fremont, California	0.110382
Micron Technology	MU	Information Technology	Semiconductors	Boise, Idaho	0.149126
Nvidia Corporation	NVDA	Information Technology	Semiconductors	Santa Clara, California	0.413936
QUALCOMM Inc.	QCOM	Information Technology	Semiconductors	San Diego, California	0.392207
Skyworks Solutions	SWKS	Information Technology	Semiconductors	Woburn, Massachusetts	0.085308
Texas Instruments	TXN	Information Technology	Semiconductors	Dallas, Texas	0.369813

Table 14: Selected stocks and their information

5.2 Constructing the Network

To construct the graph network, a correlation matrix is created and is shown in Figure 44. The highest correlation is between AMAT and LRCX, with a value of 0.824, and the lowest is 0.361, between QCOM and NVDA. The correlations generally fall between 0.4 and 0.6. The mean correlation of each stock is displayed on the right, using the same colour scale as the correlation matrix.

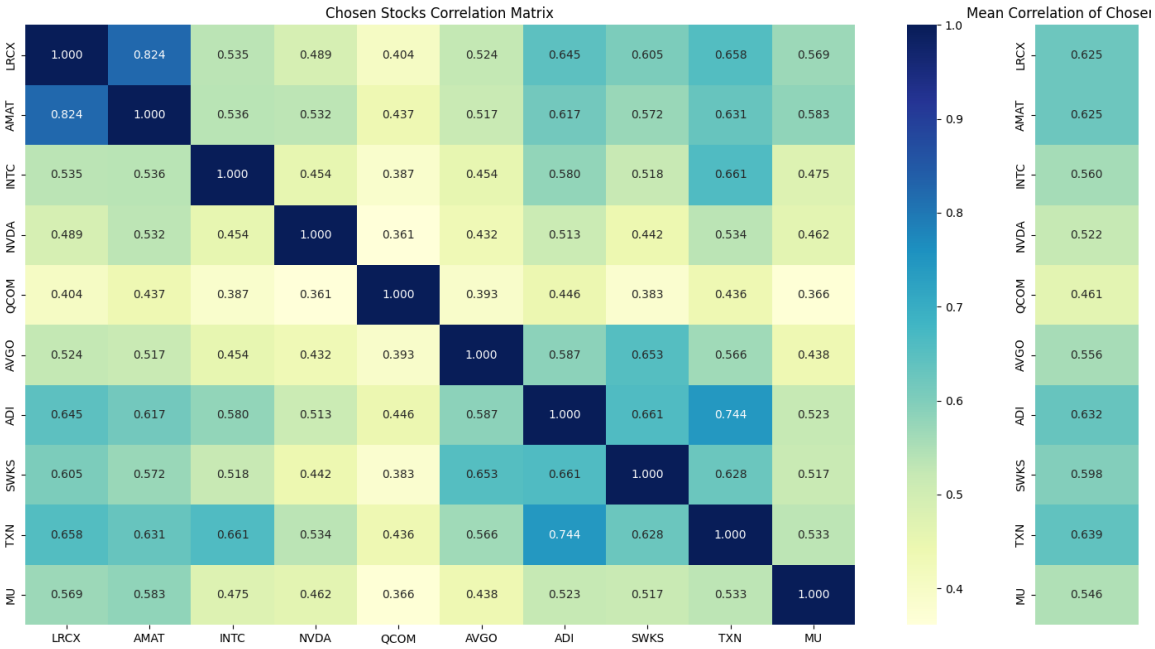

Figure 44: Correlation matrix for stocks and mean correlations

The graph based on this matrix is shown in Figure 45. The size of the nodes is based on the stock's mean correlation, indicating it's 'importance' relative to the others. The width of the links is related to the magnitude of the correlation between stocks, and is recalculated relatively within each subplot to help show the difference between each link. The size and widths were normalised to be between a predefined range to make the differences clearer. The layout of the graph is done according to the 'spring layout' option in `networkx`. This positions nodes using a force directed algorithm where there is an attractive force is between connected nodes that is proportional to the weight of the edge, and a repulsive force between all nodes. This results in edges with larger weights being shorter. I considered using the 'Kamada Kawai layout', which places a 'spring' between all nodes

with the spring being the same as the graph distance between nodes, resulting in edges with larger weights being longer. For this context, I decided having more strongly connected nodes closer would aid interpretability. It is worth pointing out that the 'spring layout' is not deterministic, so the layout changes upon re-runs of the code cell.

In the generation of the graph, the correlation matrix plays a similar role to an adjacency matrix, where the magnitude of the element (i, j) indicates the strength of the link between asset i and asset j. The strengths of the links is what is used as the weights of the edges as discussed in the previous paragraph.

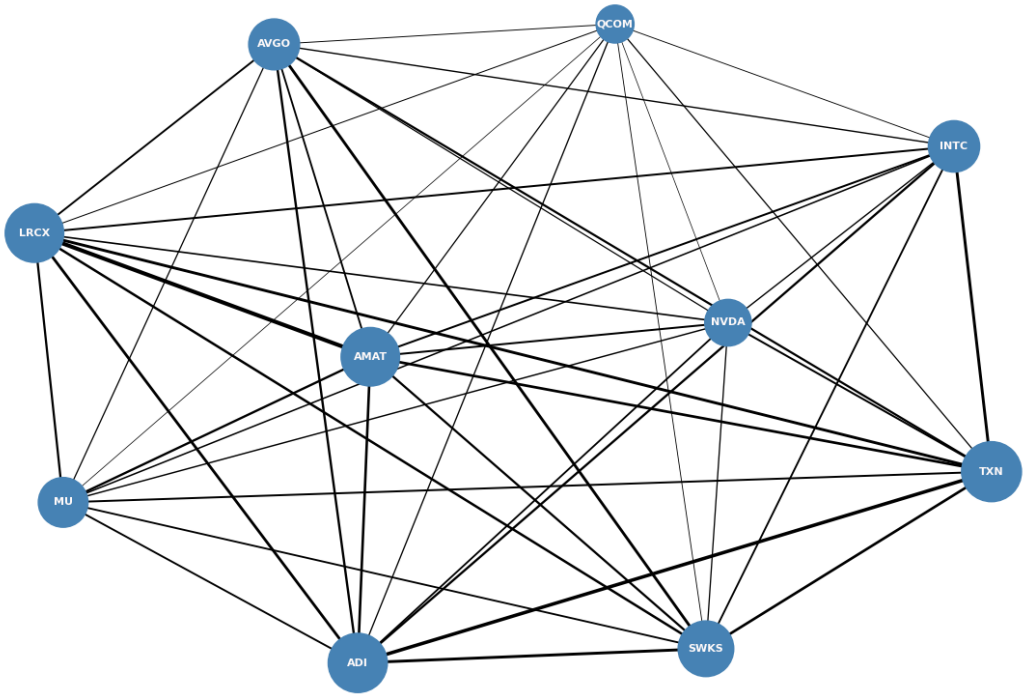

Figure 45: Correlation matrix for stocks and mean correlations

5.3 Network Analysis

The strongest correlation is between AMAT and LRCX, who are both semiconductor equipment companies. This high positive correlation makes sense given that they are serving the same market and will be similarly affected by industry headwinds/tailwinds. When looking at the relationships between the semiconductor and semiconductor equipment companies, AMAT and TXN have a particularly strong link. Upon further investigation, I found that TXN uses AMAT as a supplier so this explains this higher correlation. When looking at the geographical aspect, there does not seem to be any clear pattern dictating that the semiconductor companies within California act in a similar manner due to their location; their movements seem to be more related to factors like their targetted semiconductor market; for example NVDA and QCOM target different customers with different products and have the weakest correlation overall, despite both being in California. The companies headquartered outside of California do not appear to display any geographical pattern either. The companies with a higher market cap would probably display a stronger correlation with the overall S&P 500, but when looking between the companies that make up the index, the ones with the higher mean correlations (larger nodes on the graphs) do not have higher market caps.

Figure 46 shows graphs for different cutoff thresholds of minimum correlation needed for the link to be displayed (nodes with no links remaining are removed). This allows us to focus on the companies that play a bigger role and more closely look at the number and type of connections as the threshold rises. All correlations are above 0.3, so the top left graph contains all links and nodes like in Figure 45. QCOM is the smallest node and has the weakest connections (seen as the threshold rises to 0.4), and is the first node to disappear as the threshold increases to 0.5. This is it because it generally displays much weaker correlations with the rest of the chosen stocks, likely due to its different target customers, its products and its system-on-chip products are always in demand. TXN and ADI are the largest nodes since they have the highest mean correlations and are prominently placed within the plots, however, AMAT and LRCX are the final remaining pair due to them having the highest

correlation as mentioned earlier.

The correlation would remain the same regardless of data ordering as it measures the strength and direction of a linear relationship between two variables based on their relative positions, not their sequence in the dataset, so re-ordering the time series or indices would not change the outcome. Although, the pairing of data points between each stock needs to be kept for this to hold, i.e. all stocks would need to be reordered identically. However, due to the non-deterministic nature of 'spring-layout' as mentioned earlier, the displaying of the graph node positions would change. An example of this is that top left graph of Figure 46, and Figure 45 show the same data, but with the nodes in different positions. But the weights of the connections (correlations between pairs of stocks) and node sizes (mean correlations) are deterministic and would remain the same upon any re-ordering, as long as the re-ordering of the close prices (and therefore returns) is done such that all of the stock prices on a given day are consistently aligned.

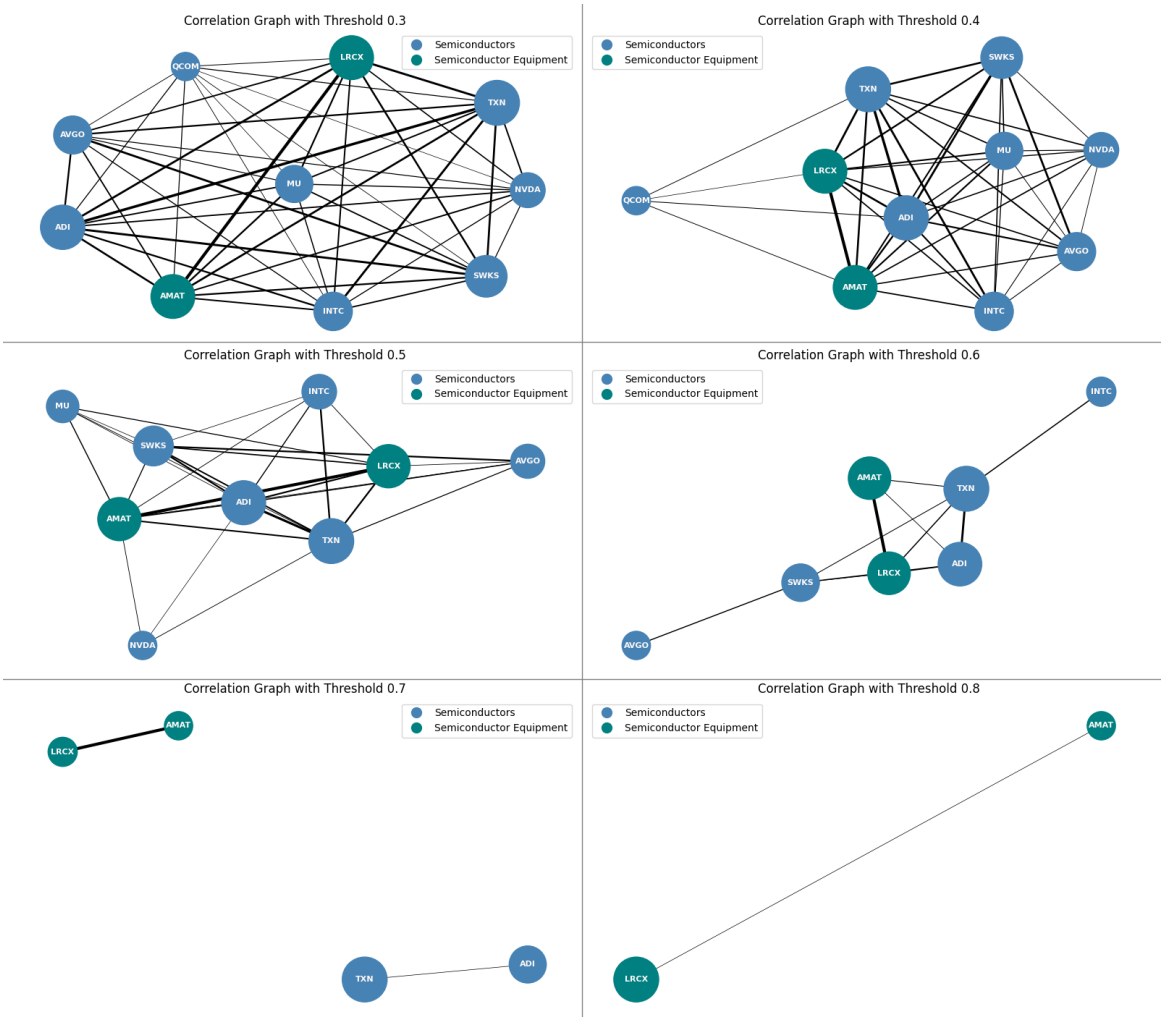

Figure 46: Correlation matrix for stocks and mean correlations

5.4 Using a different distance metric

I initially decided to try and use mutual information (MI) as a distance metric, since it would in theory take into account nonlinear relationships that the correlation cannot pick up. Since it quantifies the amount of information shared between the log returns of two stocks, it seemed to be a suitable metric to use. I hoped it would reveal some new links in the graph that were not present when using correlation as the distance metric, however, when looking the the graphs for different MI threshold, I saw that they were the same as for correlation, with the same relative strengths of links. I have kept the code and graphs for reference in my .ipynb file but exclude them here since I am not analysing them.

Since I wanted to create some graphs that showed some new information, I switched to using a periodogram based measure to analyse frequency components of the time series. This allows you to see any cyclic properties of stocks, which may be present due to factors like sector/economic cycles or trends, supply chain dynamics, breakthroughs, product release timelines, and earnings releases. There are two dissimilarities for periodograms

in the in the 'tdist' package for R and I chose to use the integrated periodogram based dissimilarity rather than the periodogram based similarity. There were a few reasons for this; the periodogram is an inconsistent estimator (although asymptotically unbiased) of the spectral density, but the integrated periodogram is a consistent estimator of the spectral distribution. It also gives more holistic view of the spectral properties across the frequency spectrum as this measure looks more at similarities or differences in the overall shape of the spectral distribution. I recreated the 'IntPerDistance' function from 'tdist' in Python. This calculates the integrated periodogram distance between two time series, using Scipy's periodogram function to get the spectra.

To construct the graph, a matrix was made in a similar manner to the correlation matrix, and is displayed in Figure 47. The three stocks with the highest mean distances, INTC, TXN and MU, are not at the bottom of the mean correlations shown in 44, proving that this method is in fact looking at different properties.

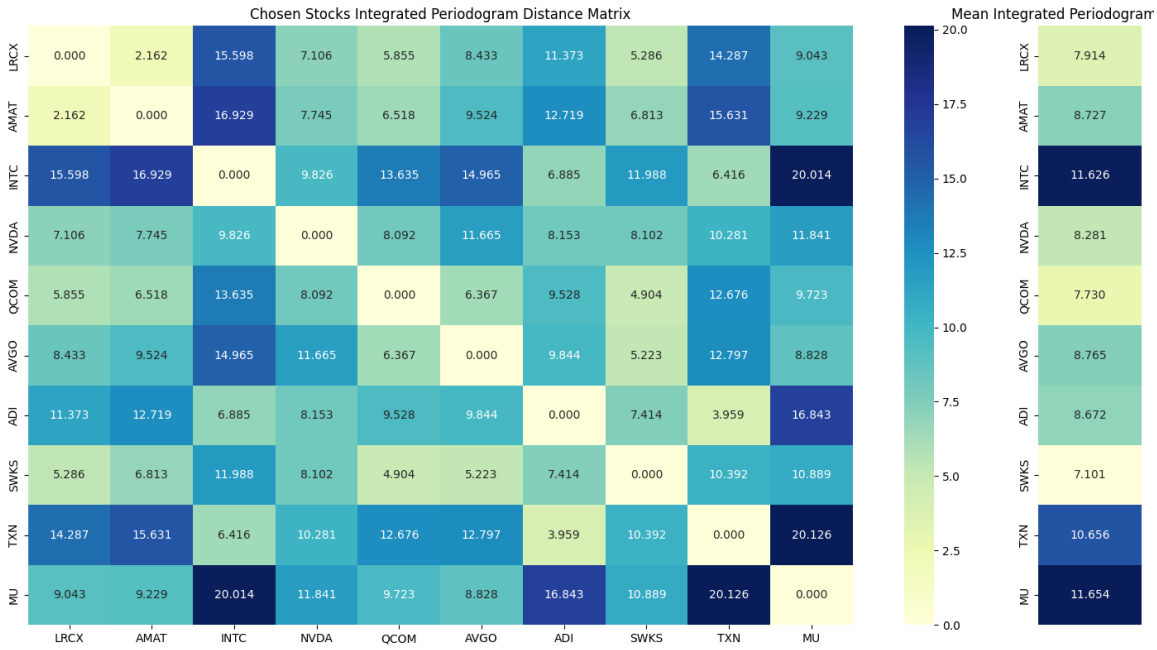

Figure 47: Matrix for stocks

There are two main ways to create a graph out of this matrix; since this is a dissimilarity measure, you can either look directly at this and focus on how different the stocks are, as shown in Figure 48, or how similar they are, as shown in Figure 49. The matrix was once again used in a similar manner to an adjacency matrix. The dissimilarity graphs retain nodes as the distance threshold increases, so you are left with the most dissimilar stocks at the highest threshold (bottom right graph), and the node size for a stock is larger for a larger mean distance. The similarity graphs retain nodes as the distance threshold decreases so that you have the most similar stocks at the end (bottom right graph), and the node size is larger for a smaller mean distance. The link widths are always proportional to the size of the distance between two stocks. It is easier to compare the similarity graph to the correlation graph, where we see the most similar stocks are the same, but the graphs are less interconnected as the threshold decreases in the similarity graph, for example, for a threshold < 6, ADI and TXN are separated. The similarity graphs seem to indicate that the stocks that have similar cyclic/periodic properties are slightly more clustered. As with the correlation graph, the two most similar stocks are AMAT and LRCX, and ADI and TXN.

We can also look at the most dissimilar stocks in Figure 48, where we see that TXN and MU are the most dissimilar. Given their completely different product lineup of chips and memory respectively, this is not surprising. This was not the case in the correlation matrix, which had the weakest link as QCOM and NVDA. That particular pair had a dissimilarity score of 8.092, just under the overall mean score of 9.113, once again proving that the integrated periodogram measure is looking at different factors.

The topology of the graph is dictated by the nature of the data, as if a stock's returns contains strong periodic components or dominant frequencies, these will become steep increases in the integrated periodogram, which is taken into account in the score and therefore also the strength of the links and size of the nodes. Reordering the vertices would not have much of an effect other than perhaps hindering readability. Data order is important for the periodogram, as it is estimating the spectral density, which is inherently tied to the ordering of the time series. Reordering the data disrupts any cyclic patterns or trends, and the periodogram is calculated using the

Fourier transform which assumes periodicity (and continuity) in its input according to the time order of the data. Since the integrated periodogram effectively cumulatively sums the spectral density across frequencies (this is how it is implemented in code), any change in the periodogram directly impacts the integrated periodogram.

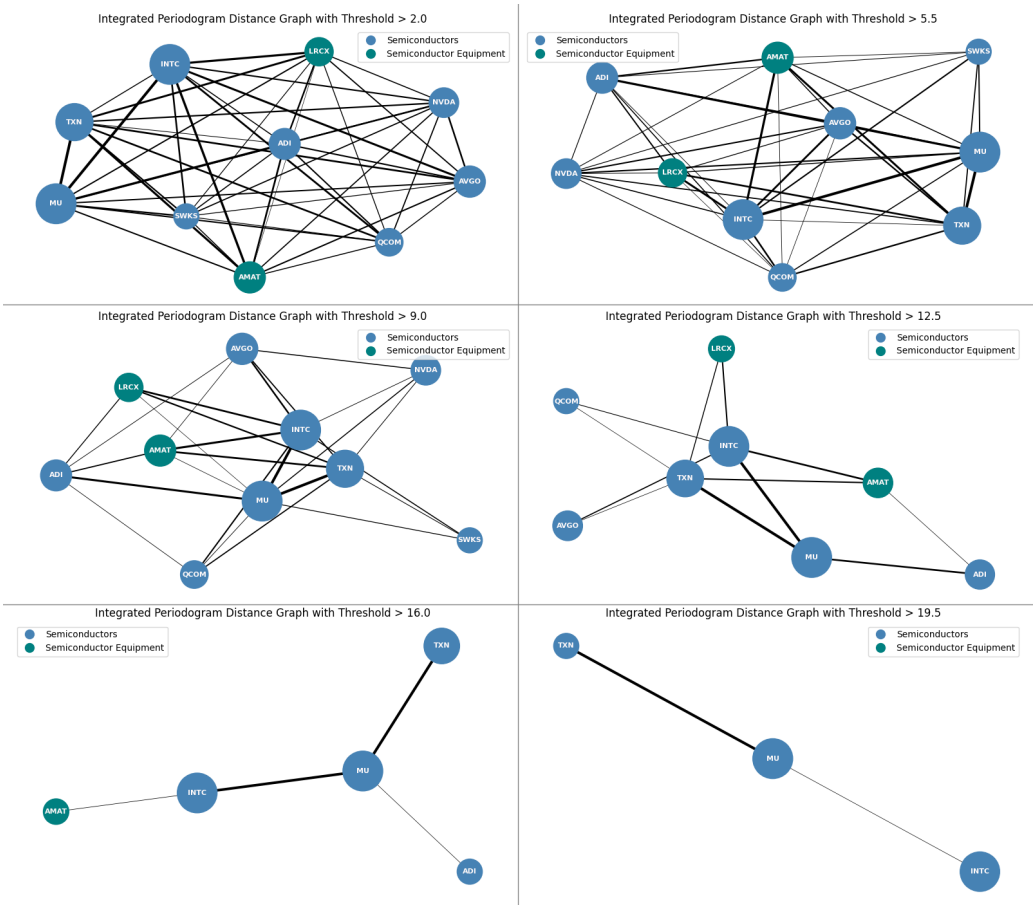

Figure 48: Dissimilarity graphs

5.5 Raw Prices Instead of Log Returns

If you were to do the same analysis as I have in Section 5 with raw prices, several issues would arise. The correlation matrix would weight early prices more heavily for the following reason; if you rewrite the prices, $p_0, p_1, \ldots, p_{T-1}, p_T$, using the returns, $r_t = p_t - p_{t-1}$, then you get $p_0, p_0 + r_1, p_0 + r_1 + r_2, \ldots, p_0 + r_1 + \ldots + r_T$. Here, r_1 is included in all of the prices apart from the first, but r_T only contributes to the last. Therefore, early changes in the prices would hold more weight than later changes in the correlation calculation. However, when using returns, each has equal importance in the calculation. So, the correlations found would give an distorted view of the stocks' relationships over time. Another potential issue comes from the fact that returns are considered to be stationary but prices are non-stationary. The non-stationary nature could lead to spurious correlations as this data contains trends so two stocks could be found to have a much higher correlation than they actually do, due to the fact that they happen to be both trending upwards. Using returns somewhat mitigates this as it effectively applies a form of detrending since you are looking at percentage change between adjacent time steps, although the returns can still exhibit trends. This is illustrated in Figure 50 which contains the decomposition of LRCX into a trend component, seasonal component and residuals, using prices and log returns. The trend and seasonal components are far less strong for the returns, and the residuals appear to be more similar to noise.

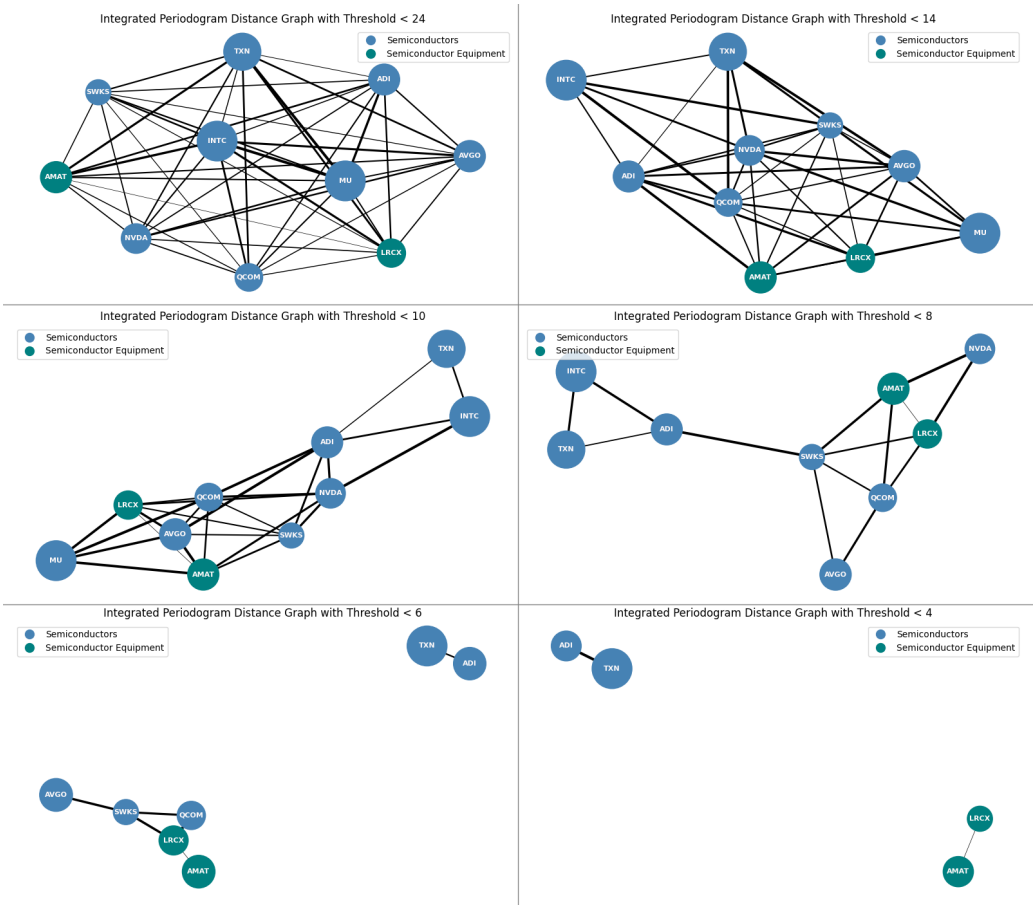

Figure 49: Similarity graphs

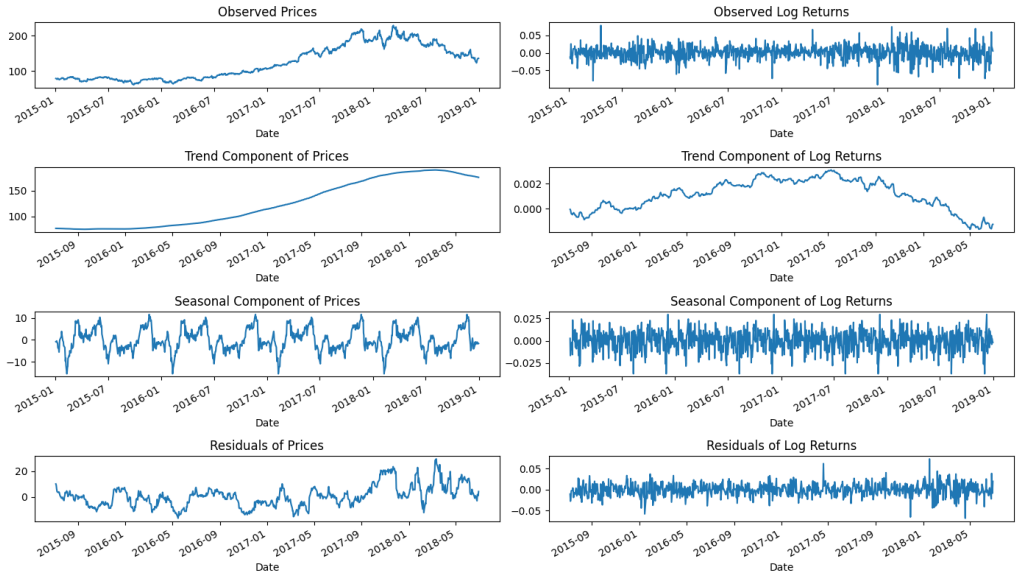

Figure 50: Decomposition of LRCX

The non-stationary of the prices is a much bigger issue in the integrated periodogram analysis; raw prices are non-stationary, but log returns can be considered stationary. Stationarity is essential for this since it uses the Fourier transform which assumes this property since it aims to decompose the signal into its constituent frequencies, assuming that these frequency components remain constant over time. Since non-stationarity violates the Fourier transform's underlying assumptions, this can lead to misinterpretations of the periodogram. Also, the price data has a non-zero mean as it typically shows some sort of trend. The result of this would be a big peak at a frequency of 0 in the periodogram as the Fourier transform of a non-zero mean signal will have

a zero-frequency (DC) component. The integrated periodogram would show a sharp increase and then remain largely constant over time due to the dominance of the zero-frequency component. This would in turn affect the weights assigned to each link in the graphs as they would now be largely related to the magnitude of the mean/trend of the prices due to the way the distances are calculated.

If a time series shows cyclic patterns that are detectable in its spectral density, then re-ordering the series will destroy these patterns meaning the periodogram will not accurately reflect the original series' cyclicality. Therefore, the distances calculated between the integrated periodograms would represent something completely different, making the weights and node sizes misleading and wrong.