Imperial College London

Imperial College London

Department of Electrical and Electronic Engineering

ELEC97100: Signal Processing and Machine Learning for Finance

Author:

Raj Jain (**CID:** 01409529)

Examiner:

Professor Danilo Mandic

Date: 19 April, 2021

Contents

1 Regression Methods

1.1 Processing Stock Price Data in Python

1.1.1

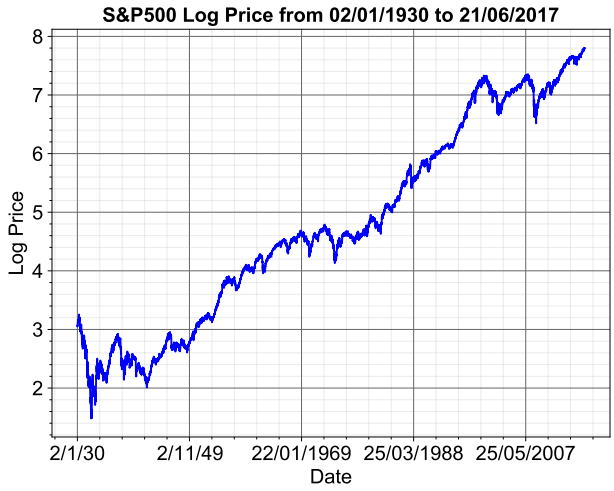
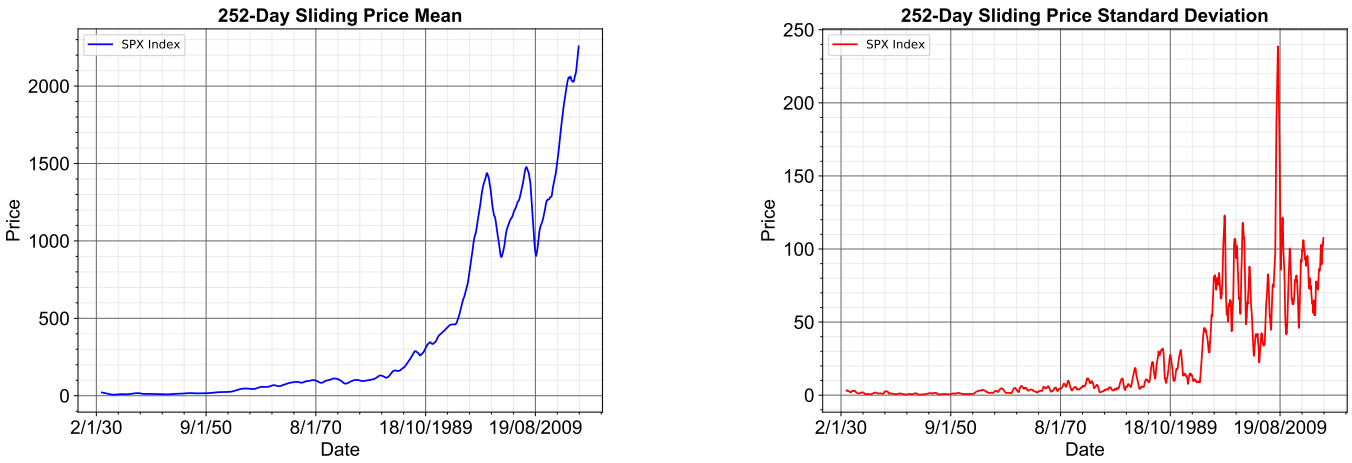


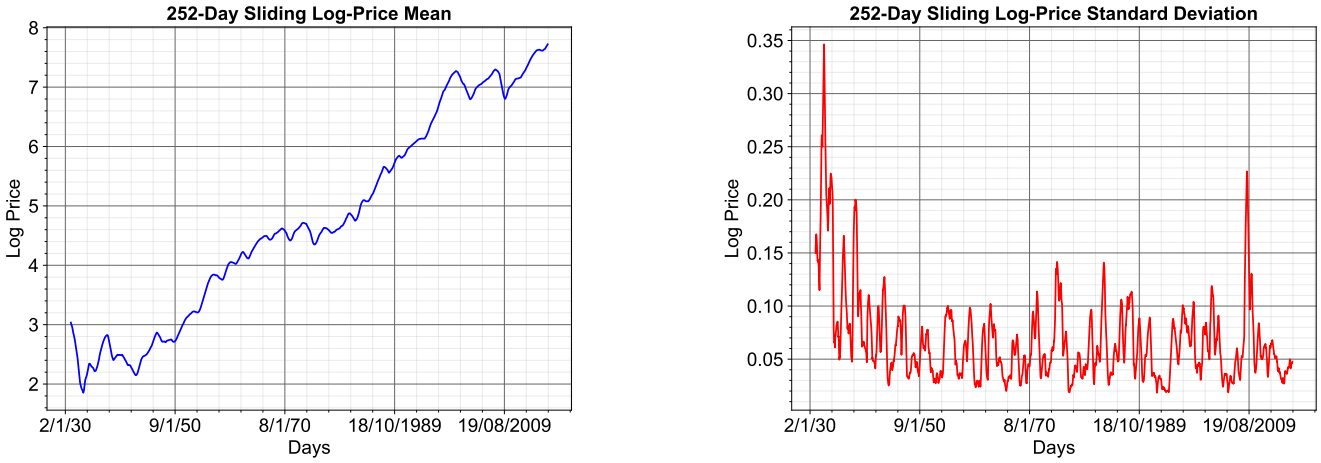
Figure 1: Plot of Natural Logarithm of Prices of S&P 500 from 02/01/1930 to 21/06/2017

1.1.2



(a) **(b)**

Figure 2: Plot of 252-Day Sliding Price Mean (a) and 252-Day Sliding Price Standard Deviation (b)



(a) **(b)**

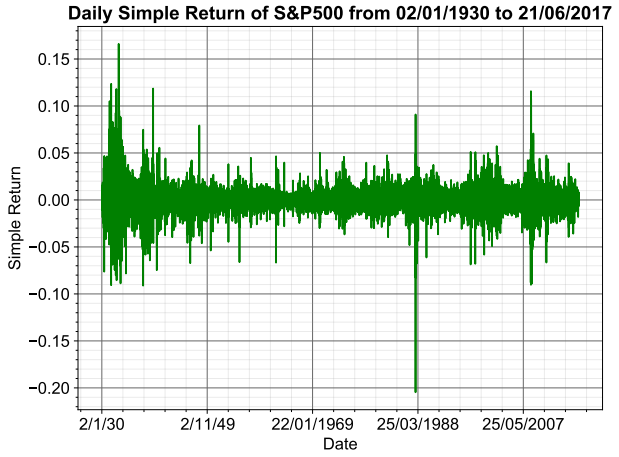
Figure 3: Plot of 252-Day Sliding Log-Price Mean (a) and 252-Day Sliding Log-Price Standard Deviation (b)

A time series X is said to be wide-sense stationary (WSS) if its mean and variance are time invariant. Although not explored here, the auto-correlation function of the X should also be a function of time difference, for a process to be wide-sense stationary. From Fig. 2a, the 252-day sliding mean of the S&P 500 price does not remain constant. Moreover, from Fig. 3a, the 252-day sliding mean of the natural logarithm of the S&P 500 price is not time invariant.

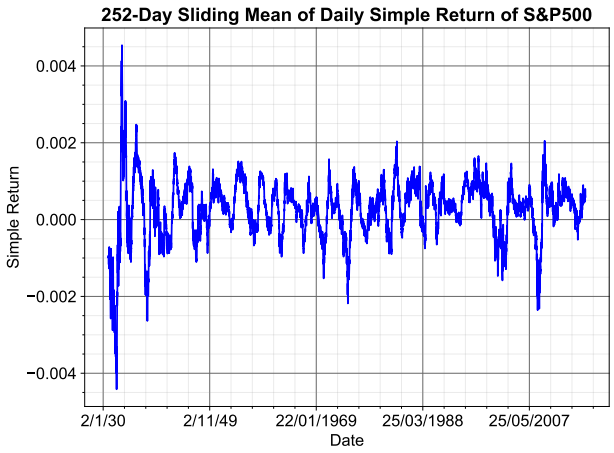
Fig. 2b shows the 252-day sliding standard deviation of the S&P 500 price. While the 252-day sliding standard deviation remains constant (between 0 and 10) from 02/01/1930 to 08/01/1970, there are significant changes within the price between 18/10/1989 and 19/08/2009. At the same time, there are significant changes within the 252-Day sliding standard deviation of the log-price of S&P 500, as shown in Fig. 3b. Therefore, the price time-series and log-price time series are not wide-sense stationary (WSS).

Finally, an interesting observation is that applying the 252-Day sliding window to the log-prices in Fig. 1 smoothes the log-price plot over time, as shown in Fig. 3.

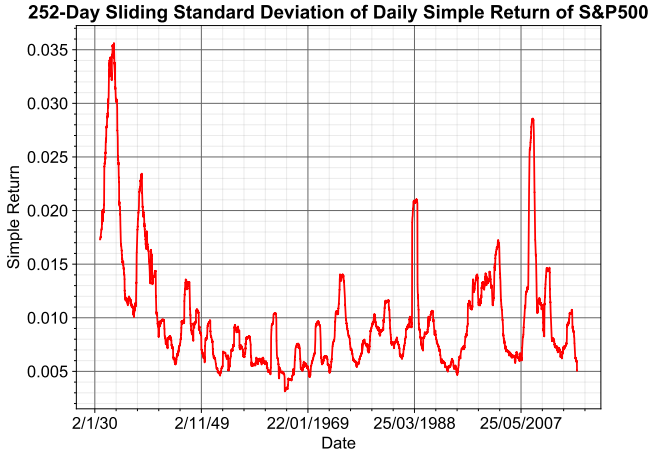
1.1.3



(a)



(b)



(c)

Figure 4: Plot of Daily Simple Return (a), 252-Day Sliding Mean (b) and 252-Day Sliding Standard Deviation (c) of Simple Returns

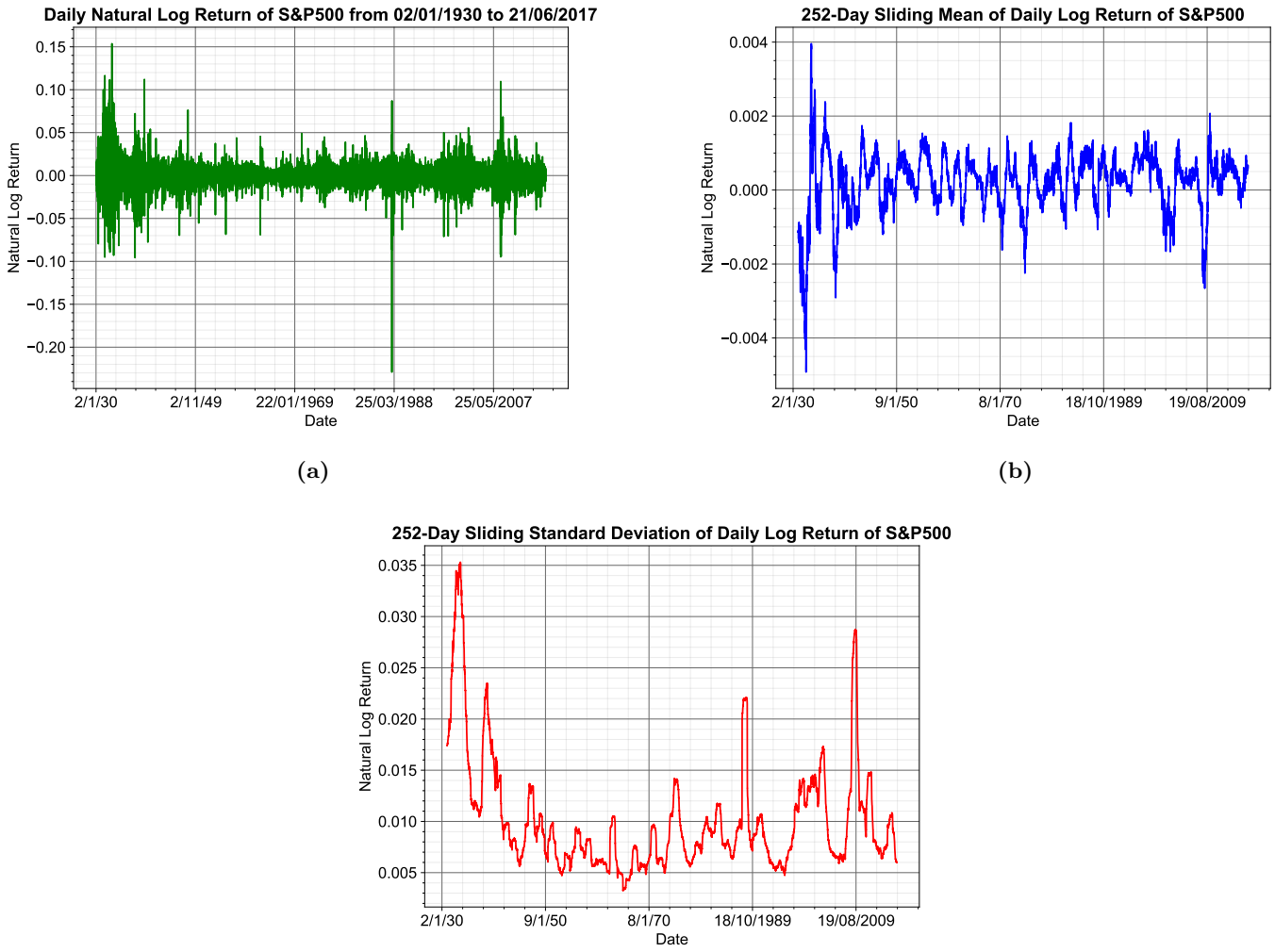


Figure 5: Plot of Daily Natural Log Return (a), 252-Day Sliding Mean (b) and 252-Day Sliding Standard Deviation (c) of Natural Log Returns

Simple return r_t is defined as the percentage change in the price from time $t-1$ to time t. It is given by the formula in Eq. (1).

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

In Eq. (1), p_t is the price of the asset at time t and p_{t-1} is the price of the asset at time $t-1$. On the other hand, logarithmic returns ρ_t at time t are defined in Eq. (2).

$$\rho_t = \ln\left(\frac{p_t}{p_{t-1}}\right) = \ln\left(R_t\right) = \ln\left(1 + r_t\right) \tag{2}$$

Fig. 4a shows the daily simple returns of the S&P 500 Index. A 252-day sliding window is used to compute the mean of daily simple returns and the resulting plot is shown in Fig. 4b. From Fig. 4b, the mean appears more time invariant, compared to the left plots in Fig. 2a and Fig. 3a. Additionally, the sliding mean of simple return plot in Fig. 4b appears to be more centered around 0. Therefore, the simple return time series is more stationary than the time series of the price. However, there are still variations in the 252-Day sliding standard deviation of the simple returns, as shown in Fig. 4c.

Similarly, a 252-day sliding window is used to compute the mean of daily log returns and the resulting plot is shown in Fig. 5b. From Fig. 5b, the sliding mean of log returns remains relatively constant compared to the sliding mean of log prices in Fig. 3a. Although there are variations in the sliding standard deviation of log returns shown in Fig. 5c, the log-returns time series is more stationary than the log-price time series.

An interesting observation from Fig. 4a is that the daily simple returns are very small, i.e. mostly lie between -0.1 and 0.1. Therefore, the daily log-returns will also very closely resemble the daily simple returns and this is observed in Fig. 5a. This is also mathematically shown in Eq. (3). Moreover, the 252-day sliding mean plots of daily simple return (Fig. 4b) and daily log return (Fig. 5b) are indistinguishable due to very little variations in the price index between t and $t-1$ (between two consecutive trading days).

$$\rho_t = \ln(1 + r_t) \approx r_t \quad for \quad r \ll 1 \tag{3}$$

1.1.4

The benefits of using natural log returns over simple returns for signal processing purposes is summarized below:

1. **Log-Normality of Prices:** Over short periods of time, the prices, p_t at any time t are considered to be distributed log-normally. This means that log-returns, i.e. $\rho_t \approx r_t$ in Eq. (3) have a normal distribution, as explained in Eq. (4).

$$\rho_t = \ln(1 + r_t) \approx r_t \quad for \quad r \ll 1 \implies 1 + r_t = e^{\ln\left(\frac{p_t}{p_{t-1}}\right)} \tag{4}$$

   This log-normality assumption of prices in short-term is crucial for using several Signal Processing techniques which assume normality and gaussianity.

2. **Time Additivity:** The compounding simple return over i periods is given by Eq. (5) where r_i is the simple return at time i.

$$\prod_i (1 + r_i) = (1 + r_1)(1 + r_2)\dots(1 + r_i) \tag{5}$$

   The assumption that r_i is log-normal implies that the compounding simple return in Eq. (5) is also log-normal. This is not desirable since the normality properties in signal processing cannot be exploited. Taking the natural logarithm of both sides in Eq. (5) leads to the result in Eq. (6).

$$\sum_i \ln(1 + r_i) = \ln(1 + r_1) + \ln(1 + r_2) + \dots + \ln(1 + r_n) \tag{6}$$

   $\ln(1+r_i)$ at time i is not only normally distributed but also the additive sum in Eq. (6) is normally distributed. This additivity and normal distribution nature also implies the symmetric nature of logarithmic returns. An example is discussed in Section 1.1.5.

3. **Mathematical Tractability and Numerical Stability:** Working with exponents and logarithms is easier for manipulation in Calculus. Moreover, addition of small numbers (Eq. (6)) is numerically stable but multiplying small numbers is not (Eq. (5)).

The "Jarque-Bera" (JB) test is a goodness-of-fit test of whether sample data have the skewness and kurtosis matching a normal distribution. The test statistic JB is given by Eq. (7).

$$JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right) \tag{7}$$

In Eq. (7), n is the number of observations, S and K are the skewness and kurtosis, respectively, of the data. In the JB test, a null hypothesis is assumed, i.e. that the sample data is normally distributed. The result of the JB test carried out in Python is a p-value. A p-value is the probability of obtaining the observed sample results when the null hypothesis is actually true.

When conducting the JB test on the Natural Logarithm of the S&P 500 prices, a p-value of 0 is obtained. Therefore, the probability of attaining sample results when the data is normally distributed is 0. Therefore, the

null hypothesis is rejected and the natural logarithm of the price data is not normally distributed. Finally, the JB Test Statistic Spread, which is the difference between JB simple returns and JB log-returns, is shown in Fig. 6.

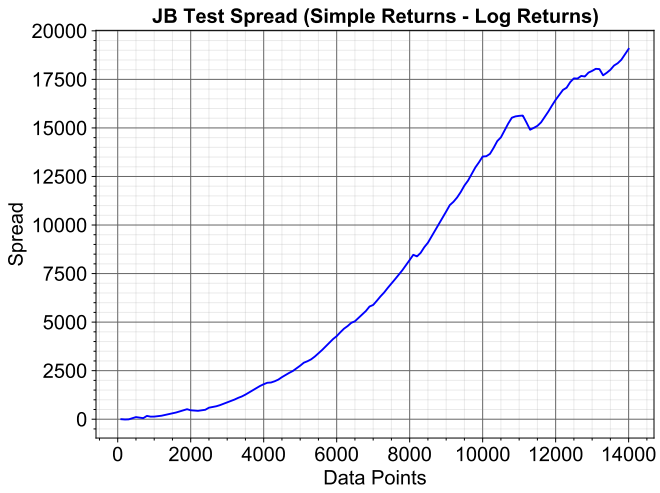


Figure 6: Jarque-Bera (JB) Test Spread, i.e. the difference between the JB Simple and JB Log-Returns

1.1.5

The simple returns for the stock price are given in Eq. (8) and Eq. (9).

$$S_1 = \frac{2-1}{1} = 1 \tag{8}$$

$$S_2 = \frac{-1}{2} = -0.5 \tag{9}$$

On the other hand, the natural logarithmic returns for the stock price are given in Eq. (10) and Eq. (11).

$$L_1 = \ln\left(\frac{2}{1}\right) = 0.693 \tag{10}$$

$$L_2 = \ln\left(\frac{1}{2}\right) = -0.693 \tag{11}$$

From Eq. (10) and Eq. (11), it can be concluded that logarithmic returns are symmetric unlike simple returns. This symmetric nature of log returns also implies that the stock price has not changed and this is further confirmed by the fact that the stock price returns to £1 after increasing to £2.

1.1.6

The logarithmic returns should not be used over simple returns in the following circumstances:

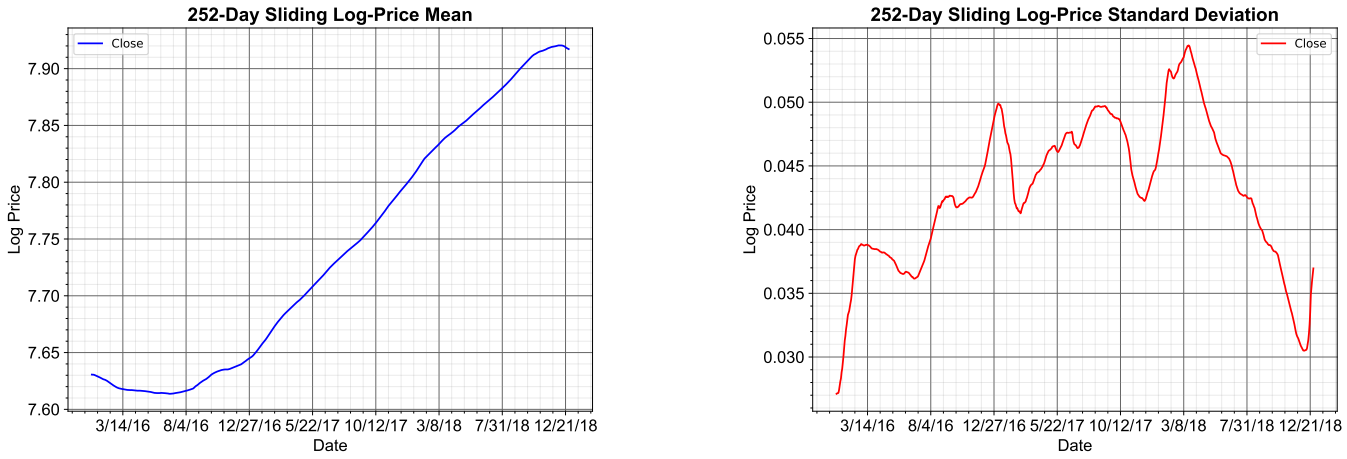
1. **Log-normality over long time-scales:** While it is assumed that prices vary log-normally over short periods, the same assumption does not hold for long periods. The prices are more skewed in the long run since the market is generally upward trending.

2. **Non-Linear in Portfolio:** Natural Logarithmic returns are not additive across portfolio whereas simple returns are linearly additive.

1.2 ARMA vs ARIMA Models for Financial Applications

1.2.1

Figure 7: Plot of Natural Logarithm of Closing Prices of S&P 500 from 02/01/2015 to 31/12/2018



(a) **(b)**

Figure 8: Plot of 252-Day Sliding Mean (a) and Standard Deviation (b) of Natural Logarithm of Closing Prices of S&P 500 from 02/01/2015 to 31/12/2018

An autoregressive process of order p, AR(p) is given by Section 1.2.1 where $x[t]$ is a random variable at time t.

$$x[t] = a_1 x[t-1] + \cdots a_p x[t-p] + \eta[t] = \sum_{i=1}^{p} a_i x[t-i] + \eta[t] \tag{12}$$

From , the future value at time t which is $x[t]$, depends on the weighted sum of past values and the noise, $\eta[t]$ at time t. Multiplying $x[t-k]$ on both sides of Section 1.2.1 yields the result shown in Eq. (13).

$$x[t-k]x[t] = a_1 x[t-k]x[t-1] + a_2 x[t-k]x[t-2] + \cdots + a_p x[t-k]x[t-p] + x[t-k]\eta[t] \tag{13}$$

It is important to note that $E(x[t-k]\eta[t])$ is equal to 0 for $k > 0$. Therefore, taking the expectation on both sides of Eq. (13), yields the result in Eq. (14). In Eq. (14), it is important to note that $r[-m] = r[m]$, i.e. the autocorrelation function is an even function.

$$r_{xx}[k] = a_1 r_{xx}[k-1] + a_2 r_{xx}[k-2] + \cdots + a_p r_{xx}[k-p] \tag{14}$$

For $k = 0$, the Eq. (14) is given in Eq. (15).

$$r_{xx}[0] = a_1 r_{xx}[1] + a_2 r_{xx}[2] + \cdots + a_p r_{xx}[p] + \sigma_\eta^2 \tag{15}$$

For $k = 1, 2, ...p$ and from the general $AR(p)$ autocorrelation function in Eq. (14), the following equations are obtained in Eq. (16).

$$\begin{aligned} r_{xx}[1] &= a_1 r_{xx}[0] + a_2 r_{xx}[1] + \cdots + a_p r_{xx}[p] + \sigma_\eta^2 \\ r_{xx}[2] &= a_1 r_{xx}[1] + a_2 r_{xx}[0] + \cdots + a_p r_{xx}[p-2] \\ &\vdots = \vdots \\ r_{xx}[p] &= a_1 r_{xx}[p-1] + a_2 r_{xx}[p-2] + \cdots + a_p r_{xx}[0] \end{aligned} \tag{16}$$

The set of equations in Eq. (16) are referred to as Yule-Walker or normal equations. The solution to Eq. (16) gives a set of autoregressive parameters, $a_1, ..., a_p$ or $\mathbf{a} = [a_1, \ldots, a_p]^T$. The equations in Eq. (16) can be compressed in a compact vector-matrix form as shown in Eq. (17).

$$\mathbf{r}_{xx} = \mathbf{R}_{xx}\mathbf{a} \quad \Rightarrow \quad \mathbf{a} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx} \tag{17}$$

In Eq. (17), the ACF matrix R_{xx} is a positive definite (Toeplitz) which guarantees matrix inversion.

An Autoregressive Moving-Average (ARMA) process, ARMA(p,q), is a stochastic process that is composed of an AR part which regresses the variable on its own lagged values and an MA part which models the error term as a linear combination of error terms occurring simultaneously at various times in the past. It is given by Eq. (18).

$$x[t] = \sum_{i=1}^{p} a_i x[t-i] + \sum_{i=1}^{q} b_i \eta[t-i] + \eta[t] \tag{18}$$

Multiplying both sides of Eq. (18) with $x[t-k]$ and taking the expectation, yields the following result shown in Eq. (19).

$$r_{xx}[k] = \sum_{i=1}^{p} a_i r_{xx}[k-i] + \sum_{i=0}^{q} b_i r_{x\eta}[t-i] \tag{19}$$

The usage of ARMA models is based on the assumption that the time series will be stationary. Wold's decomposition theorem states that any purely non-deterministic covariance-stationary process can be arbitrarily well approximated by an ARMA process. From Fig. 8a, the 252-Day sliding log-price mean is upward trending, while the 252-day sliding log-price standard deviation from Fig. 8b fluctuates. On a large scale, these aforementioned values do not fluctuate as much but they are still not constant. Therefore, the time series is not stationary and ARMA model is not appropriate for analysis.

For non-stationarity of data, a generalization of ARMA is used, i.e. Autoregressive Integrated Moving Average (ARIMA). An initial difference is applied to remove the elements of non-stationarity. ARIMA(p,d,q) is given by Eq. (20) where p and q are AR and MA lags, respectively.

$$y[t] = \sum_{i=1}^{p} a_i y[t-i] + \sum_{i=1}^{q} b_i \eta[t-i] + \eta[t] \tag{20}$$

In Eq. (20), $y[t]$ is the d-th difference of $x[t]$. For instance, if $d = 1$, then $y[t] = (x[t] - x[t-1])$.

1.2.2

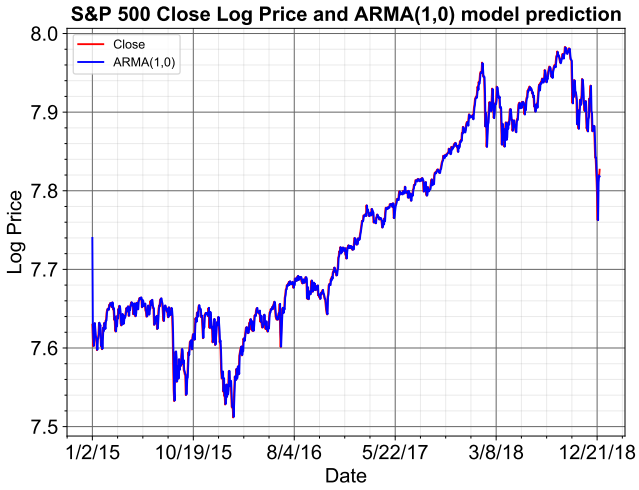


Figure 9: Plot of S&P 500 Close Log Price and ARMA(1,0) model prediction

The Natural Logarithm of S&P 500 close price and the ARMA(1,0) prediction model on the data is shown in Fig. 7. The ARMA(1,0) prediction model means that there is no moving average component and only an autoregressive component with order 1 is used. Therefore, the AR(1) process is given by Eq. (21).

$$x[t] = a_1 x[t-1] + \eta[t] \tag{21}$$

The model parameters, i.e. the values for a_1 and $\eta[t]$ in Eq. (21) is determined using model parameters function. Therefore, $\eta[t] = 7.74$ and $a_1 = 0.9974$. Using the AR(1) model to fit the time series means that only the previous value in the time series, at $t-1$, is used to predict the current value in the time series, at time t. Then, a constant is added to the prediction. From $a_1 = 0.9974$, it can be seen that a very significant weight is attached to the previous price. This means that the model prediction for the current price, at time t, depends highly on the price at time $t-1$, i.e. previous price.

Observing Fig. 9, the AR(1) model does an accurate job of predicting S&P 500 log-price. The AR(1) model is appropriate since financial time series are martingales. Financial signals can be considered as a random variable for which the conditional expectation of the next value, given all prior values is equal to the present value. In fact, the mean squared error and root mean squared error of the AR(1) predictor is shown in Eq. (22) and Eq. (23), respectively.

$$\text{Mean Absolute Error} = 0.005974 \tag{22}$$

$$\text{Root Mean Squared Error} = 0.009288 \tag{23}$$

Therefore, the AR(1) model is highly accurate. Even though the data is not stationary, the AR(1) model has a very accurate performance. This can be analyzed by considering the correlation between $x[t]$ and $x[t-1]$.

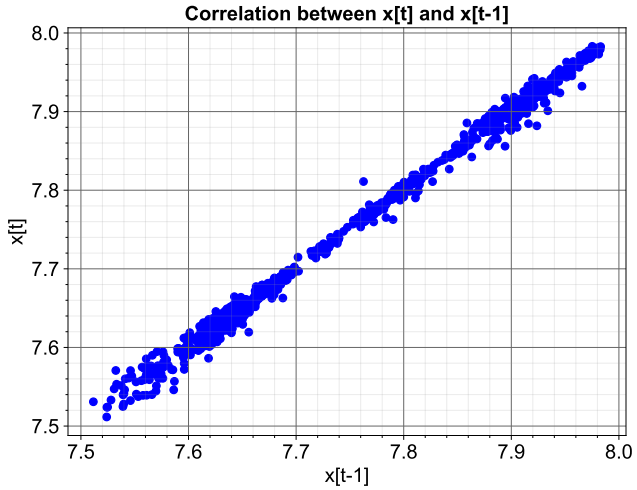


Figure 10: Correlation plot between $x[t]$ and $x[t-1]$

From Fig. 10, it can be seen that there is a very strong positive correlation between the S&P 500 price at time t and the price at time $t-1$. This explains why the AR(1) model performs well on the data despite the latter not being stationary.

In practice, the AR(1) model is not useful in practice since the prediction is a lagged version of the signal. It assumes that today's price is the same as yesterday's price. Moreover, the moving average component is not used which can better model the shock effects within the market such as wars, news, etc.

1.2.3

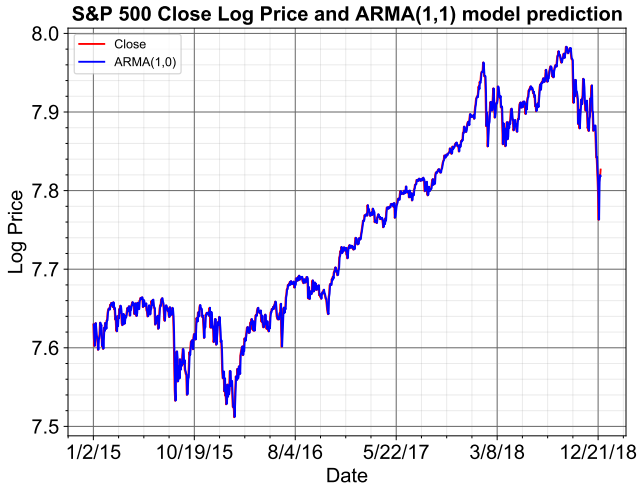


Figure 11: Plot of Natural Logarithm of S&P 500 Close Price (red) along with ARMA(1,1) model prediction (blue)

An ARIMA(1,1,0) model is used to predict the S&P 500 close price as shown in Fig. 11. While there is no moving average component, a differencing order, $d = 1$, is used. Therefore, the ARIMA(1,1,0) model is given by Eq. (24).

$$y[t] = a_1 y[t-1] + \eta[t] \tag{24}$$

Since, $d = 1$, $y[t] = (x[t] - x[t-1])$. Therefore, Eq. (24), can be modified as shown in Eq. (25).

$$x[t] - x[t-1] = a_1(x[t-1] - x[t-2]) + \eta[t] \implies x[t] = x[t-1] + a_1(x[t-1] - x[t-2]) + \eta[t] \tag{25}$$

From Fig. 11, it can be seen that the ARIMA(1,1,0) model has a very high accuracy. In fact, it's root mean squared and mean absolute errors are 0.0086 and 0.00584, respectively. These errors are lower than the AR(1) model which

indicate that ARIMA(1,1,0) is more suitable for prediction. However, the performance of the ARIMA(1,1,0) model is still very similar to the performance of the AR(1) model. This is because the MA part is still not considered which can model unpredictable or inexplicable observations which are independent of past values.

The first order differencing applied removes the non-stationarity in the log-prices. Applying order 1 differencing means that the difference of the natural logarithm of the price at time t and $t-1$ is considered. Therefore, the following is obtained in Eq. (26).

$$\ln\left(p[t]\right) - \ln\left(p[t-1]\right) = \ln\left(\frac{p[t]}{p[t-1]}\right) = \rho[t] \tag{26}$$

From Eq. (26), it can be observed that taking the difference of natural logarithm prices leads to considering the natural logarithm returns. Therefore, the ARIMA(1,1,0) model predicts the next log-return rather than the log-price. This makes the signal stationary since log-returns are stationary as shown in Fig. 5.

Moreover, the model parameters obtained are, $a_1 = -0.008752$ and $\eta[t] = 0.000196$. For the AR(1) model, the parameter values obtained were $\eta[t] = 7.74$ and $a_1 = 0.9974$. $a_1 = -0.008752$ is less than 1 in magnitude for the ARIMA(1,1,0) model and this returns series is stationary and not explosive, unlike the value of a_1 for AR(1) model.

1.2.4

Applying the natural logarithm on the prices of S&P 500 makes the data stationary, as observed in Fig. 5. Taking the log-prices in an ARIMA model with $d = 1$ allows for the log-prices to be transformed to log-returns, $\rho[t]$, by taking the differences of the log-prices. This is shown mathematically in Eq. (27).

$$\rho[t] = \ln\left(\frac{p[t]}{p[t-1]}\right) = \ln\left(p[t]\right) - \ln\left(p[t-1]\right) \tag{27}$$

To reiterate, the ARIMA(p, d, q) with p-AR lags, q-MA lags and d differencing order is given by Eq. (28).

$$y[t] = \sum_{i=1}^{p} a_i y[t-i] + \sum_{i=1}^{q} b_i \eta[t-i] + \eta[t] \tag{28}$$

In Eq. (28), $y[t]$ is the d-th difference of $x[t]$. For instance, the differences for $d = 0, 1, 2$ are shown in Eq. (29).

$$\begin{aligned} &\text{If } d = 0 : y[t] = x[t] \\ &\text{If } d = 1 : y[t] = (x[t] - x[t-1]) \\ &\text{If } d = 2 : y[t] = (x[t] - x[t-1]) - (x[t-1] - x[t-2]) = (x[t] - 2x[t-1] + x[t-2]) \end{aligned} \tag{29}$$

From Eq. (29), for differencing order $d = 1$, the differences of log of prices of $x[t]$ and $x[t-1]$ lead to log-returns, as shown in Eq. (27). Similarly, for $d = 2$, the log-return is given as shown in Eq. (30).

$$\rho[t] = \ln\left(\frac{x[t] \times x[t-2]}{2x[t-1]}\right) \tag{30}$$

Using log-returns as the ones shown in Eq. (27) and Eq. (30) make the data stationary. Using just simple returns is not enough. Additionally, ARIMA models assume that the data is stationary and using log-returns satisfies that criteria. Further benefits such as log-normality of prices, time additivity, mathematical tractability and numerical stability also apply when considering log prices, as discussed in Section 1.1.4.

1.3 Vector Autoregressive (VAR) Models

1.3.1

Vector autoregression (VAR) is a stochastic process that is used to capture the interdependencies among multiple time series. VAR generalizes the simple AR model by allowing for more than one evolving variable. A p-th order VAR, VAR(p) is given by Eq. (31).

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1\mathbf{y}_{t-1} + \mathbf{A}_2\mathbf{y}_{t-2} + \cdots + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{e}_t \tag{31}$$

In Eq. (31), $\mathbf{A}_i$ is a time-invariant $K \times K$ matrix. Eq. (31) can be expanded in matrix form and this is shown in Eq. (32).

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a^1_{1,1} & a^1_{1,2} & \cdots & a^1_{1,k} \\ a^1_{2,1} & a^1_{2,2} & \cdots & a^1_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^1_{k,1} & a^1_{k,2} & \cdots & a^1_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a^p_{1,1} & a^p_{1,2} & \cdots & a^p_{1,k} \\ a^p_{2,1} & a^p_{2,2} & \cdots & a^p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^p_{k,1} & a^p_{k,2} & \cdots & a^p_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix} \tag{32}$$

Eq. (32) can be written in a concise matrix form. The formula for $\mathbf{B}$ is given in Eq. (33).

$$\mathbf{B} = \begin{bmatrix} \mathbf{c} & \mathbf{A}_1 & \mathbf{A}_2 & \ldots & \mathbf{A}_p \end{bmatrix} \tag{33}$$

In Eq. (33), $\mathbf{B} \in \mathbb{R}^{K \times (KP+1)}$. Therefore, Eq. (31) can be rewritten in a matrix form by using $\mathbf{B}$ from Eq. (33), as shown in Eq. (34).

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{c} & \mathbf{A}_1 & \mathbf{A}_2 & \ldots & \mathbf{A}_p \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{y}_{t-1} \\ \mathbf{y}_{t-2} \\ \cdots \\ \mathbf{y}_{t-p} \end{bmatrix} + \mathbf{e}_t \tag{34}$$

Similarly, $\mathbf{y}_{t-1}$ can be written as shown in Eq. (35).

$$\mathbf{y}_{t-1} = \begin{bmatrix} \mathbf{c} & \mathbf{A}_1 & \mathbf{A}_2 & \ldots & \mathbf{A}_p \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{y}_{t-2} \\ \mathbf{y}_{t-3} \\ \cdots \\ \mathbf{y}_{(t-1)-p} \end{bmatrix} + \mathbf{e}_{t-1} \tag{35}$$

The dimensions for $\mathbf{y}_t$ and $\mathbf{y}_{t-1}$ are $\mathbb{R}^{K \times (1)}$. These $\mathbf{y}$ vectors can be stacked in a matrix from $\mathbf{y}_t$ to $\mathbf{y}_{t-(T-1)}$. Therefore, $\mathbf{Y}$ matrix with dimensions $\mathbb{R}^{K \times (T)}$ is given by Eq. (36).

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_t & \mathbf{y}_{t-1} & \cdots & \mathbf{y}_{t-(T-1)} \end{bmatrix} \tag{36}$$

From Eq. (31) and Eq. (32), the expanded version for $\mathbf{e}_t$, where $\mathbf{e_t} \in \mathbb{R}^{K \times 1}$, is given by Eq. (37).

$$\mathbf{e}_t = \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix} \tag{37}$$

The $\mathbf{e}$ vectors for times t to $t-1$ can be stacked vertically in a matrix. Therefore, the matrix for $\mathbf{U}$ where $\mathbf{U} \in \mathbb{R}^{K \times T}$ is obtained as shown in Eq. (38).

$$\mathbf{U} = \begin{bmatrix} \mathbf{e}_t & \mathbf{e}_{t-1} & \cdots & \mathbf{e}_{t-(T-1)} \end{bmatrix} \tag{38}$$

Finally, the column vectors in Eq. (34) and Eq. (35) can be stacked vertically to give $\mathbf{Z}$ where $\mathbf{Z} \in \mathbb{R}^{(KP+1) \times T}$. The matrix $\mathbf{Z}$ is given in Eq. (39).

$$\mathbf{Z} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{y}_{t-1} & \mathbf{y}_{t-2} & \cdots & \mathbf{y}_{(t-1)-(T-1)} \\ \mathbf{y}_{t-2} & \mathbf{y}_{t-3} & \cdots & \mathbf{y}_{(t-2)-(T-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{t-p} & \mathbf{y}_{(t-1)-p} & \cdots & \mathbf{y}_{(t-p)-T} \end{bmatrix} \tag{39}$$

Therefore Eq. (31) can be rewritten in matrix form for times t to $t-(T-1)$ as shown in Eq. (40).

$$\begin{bmatrix} \mathbf{y}_t & \mathbf{y}_{t-1} & \dots & \mathbf{y}_{t-(T-1)} \end{bmatrix} = \begin{bmatrix} \mathbf{c} & \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_p \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{y}_{t-1} & \mathbf{y}_{t-2} & \cdots & \mathbf{y}_{(t-1)-(T-1)} \\ \mathbf{y}_{t-2} & \mathbf{y}_{t-3} & \cdots & \mathbf{y}_{(t-2)-(T-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{t-p} & \mathbf{y}_{(t-1)-p} & \cdots & \mathbf{y}_{(t-p)-T} \end{bmatrix} + \begin{bmatrix} \mathbf{e}_t & \mathbf{e}_{t-1} & \cdots & \mathbf{e}_{t-(T-1)} \end{bmatrix} \quad (40)$$

Eq. (40) can be written in a matrix concise form as shown in Eq. (41) where $\mathbf{Y} \in \mathbb{R}^{K \times T}, \mathbf{B} \in \mathbb{R}^{K \times (KP+1)}, \mathbf{Z} \in \mathbb{R}^{(KP+1) \times T}$ and $\mathbf{U} \in \mathbb{R}^{K \times T}$.

$$\mathbf{Y} = \mathbf{B}\mathbf{Z} + \mathbf{U} \quad (41)$$

1.3.2

Eq. (41) can be rewritten as shown in Eq. (42).

$$\mathbf{U} = \mathbf{Y} - \mathbf{B}\mathbf{Z} \quad (42)$$

To find optimal set of coefficients $\mathbf{B}$, denoted by $\mathbf{B}_{opt}$, the least squares technique can be used. Specifically, the optimization function is to minimize the squared error term which is given by $\mathbf{U}^T\mathbf{U}$. Therefore, $\mathbf{U}^T\mathbf{U}$ is given by Eq. (43).

$$J(\mathbf{B}) = \mathbf{U}^T\mathbf{U} = (\mathbf{Y} - \mathbf{B}\mathbf{Z})^T(\mathbf{Y} - \mathbf{B}\mathbf{Z}) \quad (43)$$

The squared error term $\mathbf{U}^T\mathbf{U}$ in Eq. (25) can be minimized by differentiating it with respect to B. As a result, the following steps are obtained in Eq. (44), Eq. (45) and Eq. (46).

$$\frac{\partial \mathbf{U}^T\mathbf{U}}{\partial \mathbf{B}} = \frac{\partial \mathbf{U}^T\mathbf{U}}{\partial \mathbf{U}^T}\frac{\partial \mathbf{U}^T}{\partial \mathbf{B}} \quad (44)$$

$$\frac{\partial \mathbf{U}^T\mathbf{U}}{\partial \mathbf{B}} = 2(\mathbf{Y} - \mathbf{B}\mathbf{Z})\left(-\mathbf{Z}^T\right) \quad (45)$$

$$\frac{\partial \mathbf{U}^T\mathbf{U}}{\partial \mathbf{B}} = -2\mathbf{Y}\mathbf{Z}^T + 2\mathbf{B}\mathbf{Z}\mathbf{Z}^T \quad (46)$$

The result of Eq. (46) is set to 0 to minimize the coefficients. Therefore, the following results are obtained in Eq. (47) and Eq. (48).

$$-2\mathbf{Y}\mathbf{Z}^T + 2\mathbf{B}_{\text{opt}}\mathbf{Z}\mathbf{Z}^T = 0 \implies 2\mathbf{B}_{\text{opt}}\mathbf{Z}\mathbf{Z}^T = 2\mathbf{Y}\mathbf{Z}^T \quad (47)$$

$$\mathbf{B}_{\text{opt}}\mathbf{Z}\mathbf{Z}^T = \mathbf{Y}\mathbf{Z}^T \implies \mathbf{B}_{\text{opt}} = \mathbf{Y}\mathbf{Z}^T\left(\mathbf{Z}\mathbf{Z}^T\right)^{-1} \quad (48)$$

Therefore, the optimal coefficients, $\mathbf{B}_{opt}$, is given by $\mathbf{Y}\mathbf{Z}^T\left(\mathbf{Z}\mathbf{Z}^T\right)^{-1}$.

1.3.3

A VAR(1) process for the current time instant, t, is shown in Eq. (49).

$$\mathbf{y}_t = \mathbf{A}\mathbf{y}_{t-1} + \mathbf{e}_t \tag{49}$$

At the previous time instant, $t-1$, Eq. (49) can be rewritten as shown in Eq. (50).

$$\mathbf{y}_{t-1} = \mathbf{A}\mathbf{y}_{t-2} + \mathbf{e}_{t-1} \tag{50}$$

Therefore, substituting Eq. (50) in Eq. (49) yields the result shown in Eq. (51).

$$\begin{aligned} \mathbf{y}_t &= \mathbf{A}\mathbf{y}_{t-1} + \mathbf{e}_t \\ &= \mathbf{A}\left(\mathbf{A}\mathbf{y}_{\mathbf{t-2}} + \mathbf{e}_{t-1}\right) + \mathbf{e}_t \\ &= \mathbf{A}^2\mathbf{y}_{\mathbf{t-2}} + \mathbf{A}\mathbf{e}_{t-1} + \mathbf{e}_t \\ &\implies \mathbf{A}^N\mathbf{y}_{\mathbf{t-N}} + \sum_{i=0}^{N-1} \mathbf{A}^i\mathbf{e}_{t-i} \end{aligned} \tag{51}$$

Eigendecomposition can be performed on $\mathbf{A}$ where $\mathbf{Q}$ and Λ represent the eigenvector matrix and diagonal eigenvalue matrix, respectively. Therefore, Eq. (51) can be rewritten as shown in Eq. (52).

$$\mathbf{y}_t = \mathbf{Q}\mathbf{\Lambda}^N\mathbf{Q}^{-1}\mathbf{y}_{t-N} + \sum_{i=0}^{N-1} \mathbf{A}^i\mathbf{e}_{t-i} \tag{52}$$

To ensure $\left\|\mathbf{A}^N\right\| < 1$, it needs to be ensured that $\left\|\mathbf{\Lambda}^N\right\| < 1$. In other words, the eigenvalues of $\mathbf{A}$ need to be less than 1. Else, the power will explode to extremely large values. In other words, if $N \to \infty$, the eigenvalues of $\mathbf{A}$ become large and unstable with $\lambda_i^N > 1$. For stability, $|\lambda_i^N| < 1$ is desired.

1.3.4

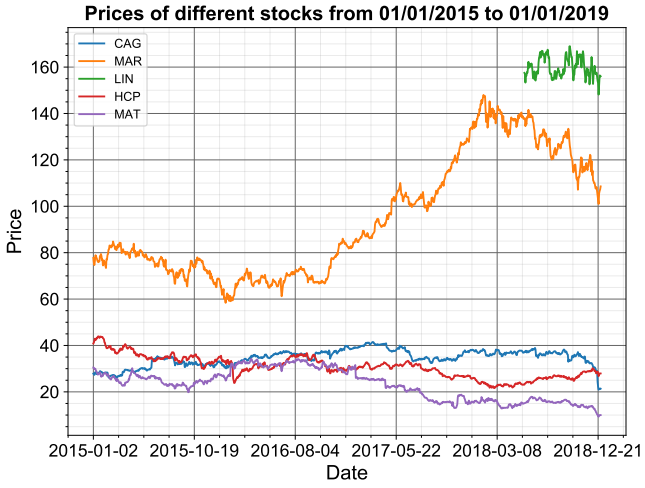


Figure 12: Plot of the Stock Prices of CAG, MAR, LIN, HCP and MAT from 02/01/2015 to 01/01/2019

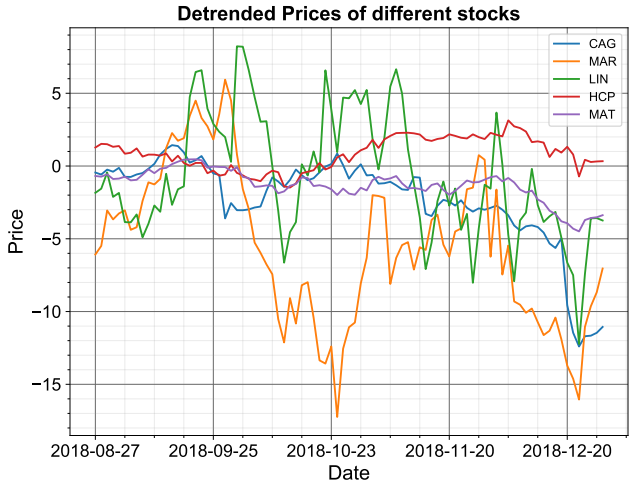


Figure 13: Plot of the Detrended Stock Prices of CAG, MAR, LIN, HCP and MAT

Fig. 12 and Fig. 13 show a plot of the actual and detrended stock prices of the chosen stocks, respectively. The stock prices are detrended using a moving average window of 66 days. Next, a VAR(1) model is applied to the detrended time series and regression matrix **A** is generated and this is shown in Table 1.

	CAG	**MAR**	**LIN**	**HCP**	**MAT**
CAG	0.87279	0.11318	-0.28127	0.01191	0.05878
MAR	-0.06375	0.89582	-0.18482	-0.005	0.02292
LIN	0.00013	-0.11168	0.70402	0.00498	-0.02556
HCP	-0.08478	-0.08383	-0.40142	0.93171	-0.04641
MAT	0.64307	0.09493	2.03304	-0.01288	0.80297

Table 1: Matrix **A** of the VAR(1) model

As previously mentioned, the regression matrix of a VAR(1) model gives insight into how correlated a stock price is to the lagged values of its own stock price and other stock prices. For instance, CAG has a strong correlation of 0.87279 with its own lagged value but a very weak correlation of 0.01191 with the lagged value of HCP. Another interesting example is that the stock price of MAT has a very strong correlation of 2.033 with the previous stock price of LIN. It is important to note that only 1-day lag of stock prices is considered. However, this may be due to the lack of data for LIN. Another example is the strong correlation between the current price of MAT and the previous price of CAG, which has a value of 0.64307. A scatter plot between the current price of MAT and the previous price of CAG is shown in Fig. 14.

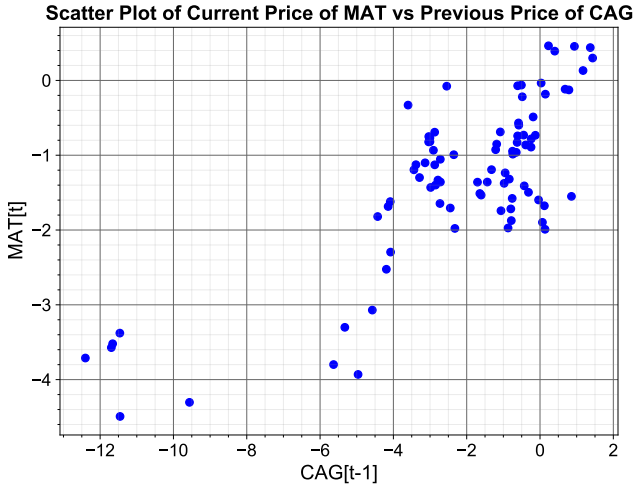


Figure 14: Scatter Plot of the Current Price of MAT vs the Previous Price of CAG

Next, the Eigenvalues of the Matrix **A** are determined and this is shown in Table 2.

Eigenvalues of Matrix A	0.726094	0.726094	1.006360	0.860519	0.911445

Table 2: Eigenvalues of Matrix **A** of the VAR(1) model in Table 1

FromTable 2, one of the eigenvalues of the matrix is 1.006360. This is larger than 1, which indicates the instability of the system. Therefore, it is not advisable to construct a portfolio with these stocks.

However, it is interesting to note the link between the VAR(1) regression matrix and the covariance matrix. As previously mentioned, the VAR(1) regression matrix is a correlation matrix between the current stock price of an entity to the lagged values of the stock prices of itself and other entities. Therefore, adjusting for the standard deviations, the regression matrix **A** in Table 1 can be interpreted as a 'lagged' covariance matrix. Therefore, by determining the eigenvalues, as done in Table 2, and the eigenvectors can help determine the direction and magnitude of highest risk.

In terms of Principal Component Analysis, the eigenvalues of a covariance and correlation matrix represent the 'core' of the PCA. These eigenvectors determine the directions of the new feature space and the eigenvalues determine their magnitude. A large eigenvalue would indicate that the variation between the stocks is high, implying low correlation. For instance, in Table 2, an average of Eigenvalue of 0.846 is obtained. Therefore, it is advisable to construct a portfolio of stocks that are uncorrelated with each other. This would aid in the diversification of the portfolio.

1.3.5

	Minimum Eigenvalue	**Average Eigenvalue**	**Maximum Eigenvalue**
Industrials	0.371246	0.763932	0.991721
Health Care	0.092157	0.62144	0.994153
Information Technology	0.374081	0.809351	0.992738
Communication Services	0.752488	0.926293	0.982263
Consumer Discretionary	0.447563	0.810433	0.99065
Utilities	0.042115	0.599877	0.985648
Financials	0.152575	0.631218	1.00434
Materials	0.137838	0.621833	0.991744
Real Estate	0.763563	0.919369	0.982785
Consumer Staples	0.546458	0.852121	0.991508
Energy	0.825707	0.930601	0.985577

Table 3: Minimum, Average and Maximum Eigenvalues of Regression Matrix **A** constructed by using VAR(1) technique to model portfolios with companies grouped by sector

Section 1.3.4 is repeated, except this is performed on the entire universe of stocks, grouped by the sectors. The regression matrices, **A**, were determined for each industry. Therefore, the maximum, minimum and average eigenvalues for each regression matrix for all industries were determined and this is shown in Table 3. As seen, only Communication Services, Real Estate, Consumer Staples and Energy industries have higher average eigenvalues than the average eigenvalue from Section 1.3.4. It is not advisable to construct portfolios according to the industry since the stocks within a particular industry tend to be highly correlated (very little variations, as witnessed by low eigenvalues in Table 3). This does not allow for diversification of portfolios and stocks across different sectors need to be considered.

The stocks can be diversified by considering the Markowitz Mean-Variance Portfolio Theory. According to Markowitz, the portfolio with weights, $w_1, w_2, ..., w_n$ for n stocks has mean return and variance shown in Eq. (53).

$$\bar{r}_{\mathrm{P}} = \sum_{i=1}^{n} w_i \bar{r}_i \quad , \sigma_{\mathrm{P}}^2 = \sum_{i,j=1}^{n} w_i \sigma_{ij} w_j \tag{53}$$

Therefore, the entire universe of the stocks can be considered at once. This entire universe of stocks in the CSV file is the S&P 500 Portfolio. This universe of stocks had the second lowest standard deviation of 0.0088. The

lowest standard deviation was for Consumer Staples which had 0.008. But the S&P 500 portfolio had a return of 0.0003, double that of Consumer Staples. But from considering an entire universe of stocks, a low standard deviation/risk is obtained by diversifying the portfolio.

2 Bond Pricing

2.1 Examples of Bond Pricing

2.1.1

(a):
For annual, semi-annual compounding, monthly and continuous compounding, the percentage return per annum is given by Eq. (54), Eq. (55), Eq. (56) and Eq. (57), respectively:

$$1100 = 1000(1+r) \implies r = 0.1 = 10\% \tag{54}$$

(b):

$$1100 = 1000\left(1+\frac{r}{2}\right)^2 \implies r = 0.0976 = 9.76\% \tag{55}$$

(c):

$$1100 = 1000\left(1+\frac{r}{12}\right)^{12} \implies r = 0.0957 = 9.57\% \tag{56}$$

(d):

$$1100 = 1000 \times e^r \implies r = 0.0953 = 9.53\% \tag{57}$$

2.1.2

Percentage interest per annum should be 14.91% for continuous compounding which is equivalent to 15% per annum with monthly compounding, as shown in Eq. (58).

$$\left(1+\frac{0.15}{12}\right)^{12} = e^r \implies r = 0.14907 = 14.91\% \tag{58}$$

2.1.3

At the end of 1 year, the total amount after continuous compounding is $\$10000 * e^{0.12} = \$11,274.97$. The total amount at the end of each quarter is shown in Eq. (59).

$$\begin{aligned} 10,000e^{\frac{0.12}{4}} &= 10,304.54 \\ 10,304.54e^{\frac{0.12}{4}} &= 10,618.37 \\ 10,618.37e^{\frac{0.12}{4}} &= 10,941.74 \\ 10,941.74e^{\frac{0.12}{4}} &= 11,274.97 \end{aligned} \tag{59}$$

Therefore, the interests paid at the end of Q1, Q2, Q3 and Q4 are 304.54, 313.83, 323.37 and 333.23, respectively.

2.2 Forward Rates

2.2.1

(a): An investor would be indifferent from the two strategies by the no Arbitrage Principle. If a commitment is made for 2 years at 7%, then the investor will earn \$114.49, at the end of two years. If the investor decides to invest for only one year, they will earn \$105, at the end of one year. The no-Arbitrage Principle states that the investor should get a 9% return between Year 1 and Year 2, i.e the forward rate should be $f_{1,2} = 9\%$. Investing for 2 years would mean that the investor would be less liquid. Additionally, they might also miss out if the interest rates between $f_{1,2}$ go higher. On the other hand, investing only for one year keeps the investor more liquid. However, if the forward rate, $f_{1,2}$, decreases, they will get a less return compared to the 7% strategy.

(b): The 7% strategy involves the investor locking in their funds for two years. This means that the investor will not have access to this particular cash. After two years, they will earn \$114.49, i.e. a return of \$14.49. The 5% strategy may be more suitable to customers who want to keep their assets liquid. After a year, the investors will have access to cash and they can decide to deposit in the bank for another year or invest in something else. If the investor wants to deposit their money and they believe that a year from now, one year interest rates will be greater than 9%, i.e. $f_{1,2} \geq 9\%$, they should definitely go for the 5% strategy. At the end of Year 1, they would have earned \$105. If the interest rate between Year 1 and Year 2 is higher than 9% ($f_{1,2} \geq 9\%$), they can earn more than the 7% strategy. If they are pessimistic about the forward rate $f_{1,2}$ being less than 9% and do not mind being illiquid, they should opt for the 7% strategy.

(c): The forward rate of 9% (Between Year 1 and Year 2) is the market implied rate, assuming that the one-year interest rate is 5% and the two-year interest rate is 7%. Ideally, this is what the investor should earn, if they deposit their funds in the bank for a year, one year from now. In reality, this number could be the same or different, one year into the future. This variation depends the market and economic conditions, one year from now.

(d): The decision of opting between the 5% and 7% strategy would very much depend on the research conducted by the investor. As mentioned, if they are optimistic about the market and believe, a year from now, one-year interest rates will be higher than 9% they should go for the 5% strategy. This would involve investing their \$100 right now, earning \$105 a year from now and hopefully, investing that \$105 at an interest rate higher than 9%. Additionally, if the investor wants to remain liquid, they should opt for the 5% strategy, so that they can withdraw the cash one year from now and consider exploring other avenues or can use that cash for their expenses. Finally, there might be a case where the investor might expect that the market may not perform well, one year from now. Consequently, the investor may believe that in one year, the one-year interest rates may be lower than 9%. In this case, they should opt for the 7% strategy and invest their cash for two years, provided they do not mind being illiquid.

2.3 Duration of a coupon-bearing bond

2.3.1

(a):

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	
Year × Fraction of PV [t × PV(C_t)/PV]	0.0124	0.0236	0.0337	0.0428	0.0510	0.0583	6.5377	

Figure 15: Calculating the duration of the 1% 7-year bonds. The yield to maturity is 5% a year

From Fig. 15, the duration of the Bond is 6.7595 Years which is obtained by summing the $\mathbf{t} \times \mathbf{PV}\left(\mathbf{C_t}\right)/\mathbf{PV}$ term for each year (last row).

(b): The modified duration for Fig. 15 is calculated as shown in Eq. (60).

$$\text{Modified duration } = \frac{\text{duration}}{1 + \text{ yield}} = \frac{6.7595}{1 + 0.05} = 6.4376 \tag{60}$$

Modified duration can determine the percentage change in the price of a bond given a change in yield, as shown in Eq. (61).

$$\begin{aligned} D_M = -\left.\frac{1}{P(\lambda_0)}\frac{\mathrm{d}P(\lambda)}{\mathrm{d}\lambda}\right|_{\lambda=\lambda_0} &\approx -\frac{1}{P}\frac{\Delta P}{\Delta\lambda} \\ \Rightarrow \quad \Delta P &\approx -D_M P \Delta\lambda \end{aligned} \tag{61}$$

(c): Pension plans can be thought of as long duration bonds. Bonds with long durations (time to maturity) are highly susceptible to changes in the interest rates. Therefore, they have high volatility. Using a metric like modified duration can indicate the change in the price when the yield to maturity changes. This is measured by the derivative of the price of the bond, with respect to the yield to maturity. Finally, higher order terms can be considered from the Taylor expansion to protect from the high volatility of the plan. A fixed-income security's price can be expanded as a function of yield around λ_0, as shown in Eq. (62).

$$P(\lambda) = P(\lambda_0) + P'(\lambda_0)(\lambda - \lambda_0) + \frac{1}{2}P''(\lambda_0)(\lambda - \lambda_0)^2 + \cdots \tag{62}$$

More terms can be considered in the Taylor expansion to adjust the Taylor series of the portfolio to the obligation.

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

2.4.1

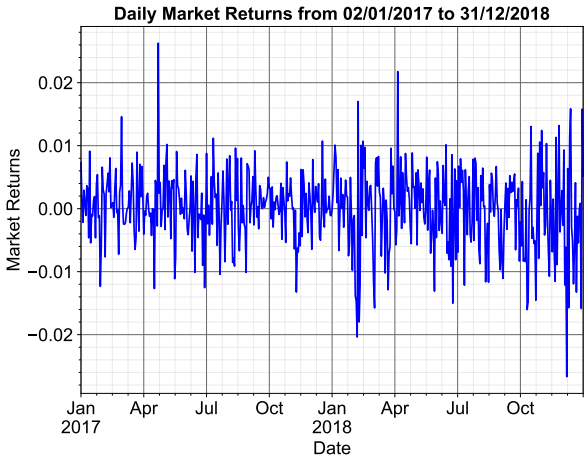


Figure 16: Plot of Daily Market Returns from 02/01/2017 to 31/12/2018

The Capital Asset Pricing Model (CAPM) is a theorem which states that if the market portfolio M is efficient, the expected return $\bar{r_i}$ of any asset i satisfies Eq. (63)

$$\bar{r_i} = r_f + \beta_i(\bar{r_m} - r_f) \tag{63}$$

In Eq. (63), r_f is the risk-free rate, r_m is the expected market return and β_i of asset i is a measure of volatility of a security or portfolio, compared to the market as a whole. Considering returns instead of expected returns, Eq. (63) can be formulated as shown in Eq. (64).

$$r_i = r_f + \beta_i(r_m - r_f) + \epsilon_i \tag{64}$$

In Eq. (64), r_i is the (random) rate of return of asset i and r_m is the market return. From Eq. (63) and Eq. (64), it can be deduced that $E(\epsilon_i) = 0$. ϵ_i is an uncorrelated zero-mean company specific risk random variable. $var(\epsilon_i)$ is termed as the idiosyncratic or specific risk of the company which is uncorrelated with the market and can be reduced by diversification.

The daily market returns, R_{m_t}, at time t, from 02/01/2017 to 31/12/2018 are estimated by taking the average of the daily returns of individual assets. In other words, the daily market return is estimated according to Eq. (65).

$$R_{m_t} = \text{average(company returns)} \tag{65}$$

Therefore, the daily market return for the period is shown in Fig. 16.

2.4.2

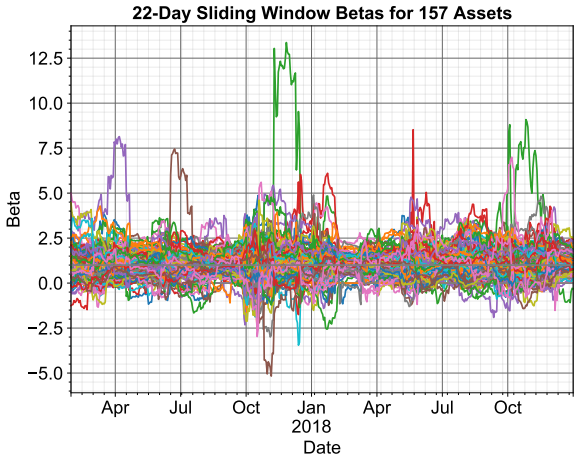


Figure 17: 22-Day Sliding Betas for 157 companies

From the Capital Asset Pricing Model Equation, β_i for an asset i is the sensitivity of the expected excess asset returns for asset i to the expected excess market returns. β_i is given by Eq. (66).

$$\beta_i = \frac{\text{Cov}\left(R_i, R_m\right)}{\text{Var}\left(R_m\right)} = \rho_{i,m}\frac{\sigma_i}{\sigma_m} \tag{66}$$

In Eq. (66), R_i corresponds to the returns of the asset i and R_m is the overall market return. β_i for asset i, at time instant t is given by $\beta_{i,t}$. he 22-day sliding window Betas, β_i, for all 157 assets is shown in Fig. 16 where the daily market return, R_{m_t} was estimated using Eq. (66). The volatility of β_i for asset i can be interpreted as follows:

- $\beta_i > 1$: The asset is more volatile than the market. High-beta stocks are riskier than the market but provide higher return potential.
- $\beta_i = 1$: The asset is as volatile as the market. The asset's price tends to move with the market.
- $0 < \beta_i < 1$: The asset is less volatile than the market.
- $\beta_i = 0$: Regardless of the way the market moves, the value of the asset remains unchanged.
- $\beta_i < 0$: This indicates an inverse relation to the market. If the market returns increase, negative beta asset returns decrease and vice-versa.

2.4.3

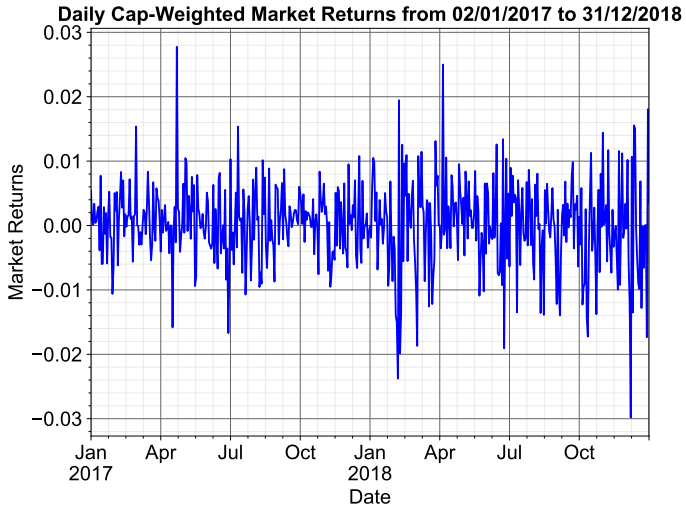


Figure 18: Daily Cap-Weighted Market Returns from 02/01/2017 to 31/12/2018

A capitalization-weighted index is a type of stock market index where individual components of the index are included in amounts that correspond to their total market capitalization (market cap). The cap-weighted market return, R_m, is determined using the formula in Eq. (67).

$$R_m = ret(market) = \sum_i \frac{mcap_i \times ret_i}{\sum_i mcap_i} \tag{67}$$

In Eq. (67), ret_i is returns for asset i and $mcap_i$ is the market capitalization of asset i. $\sum_i mcap_i$ is the weighting coefficient or the sum of individual market capitalizations of all assets. Therefore, $\sum_i mcap_i$ gives the total market capitalization. The market return in Eq. (67), is based the weighted average of individual asset returns based on their respective market capitalization. The daily cap-weighted market return, R_m, from 02/01/2017 to 31/12/2018 is shown in Fig. 18.

2.4.4

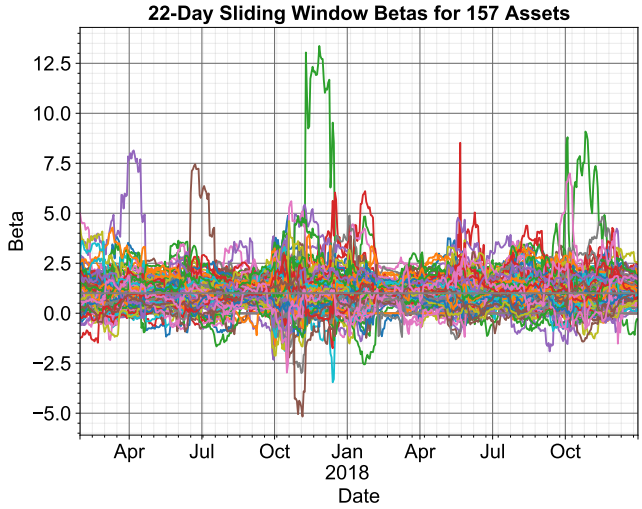


Figure 19: 22-Day Sliding Window Cap-Weighted Betas for 157 companies

The 22-day sliding window Cap-Weighted Betas for all 157 assets is shown in Fig. 19. The cap-weighted beta is also calculated using the formula in Eq. (66). The value of R_m is determined using Eq. (67), i.e. the cap-weighted market returns are used instead of equally weighted market returns.

Assets with a high market capitalization will have a larger cap-weighted beta compared to their equally-weighted beta. An asset, i, with a high market capitalization will significantly influence the cap-weighted market return. Therefore, the correlation between the asset i with a large market capitalization and the cap-weighted market, $\rho_{i,m}$, will also be high compared to the correlation between the asset and the equally-weighted market. This implies a higher cap-weighted beta than the equally-weighted beta.

An example of an asset with the highest market capitalization in the provided dataset can be illustrated. It was determined that 'G_NESNVX' had the highest market capitalization, on average, from 02/01/2017 to 31/12/2018.

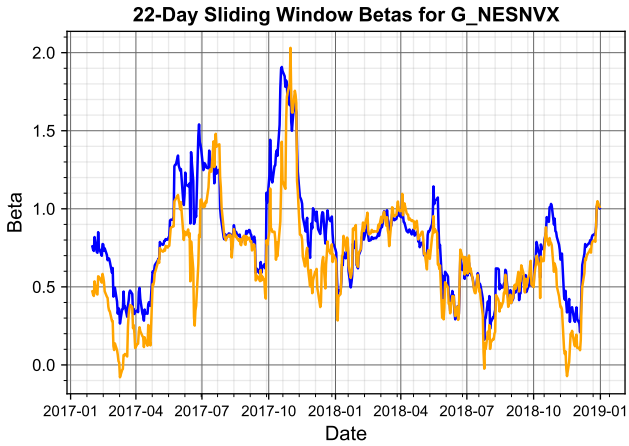


Figure 20: 22-Day Sliding Window Cap-Weighted (blue plot) and Equally-Weighted (orange plot) Betas for 'G_NESNVX'

Fig. 20 shows a plot of 22-Day Sliding Window Cap-Weighted (blue plot) and Equally-Weighted (orange plot) Betas for 'G_NESNVX'. From Fig. 20, it can be observed that on average, cap-weighted betas are larger than equally-weighted betas.

2.4.5

(a:) The Arbitrage Pricing Theory (APT) is a more generalized version of the CAPM Model. In this multi-factor asset pricing model, the asset's returns can be predicted using a linear relationship between the asset's expected return and macroeconomic variables which capture systematic risk. The return for an asset i at time t, r_i, according to a two-factor model is given by Eq. (68).

$$r_i = a + b_{m_i} R_m + b_{s_i} R_s + \varepsilon_i \tag{68}$$

In Eq. (68), the variables are described as follows:

- a: Constant for the asset i
- R_m: Market return at time t
- b_{m_i}: Factor loading or sensitivity of the i^{th} asset to factor m. Factor m is the market return at time t, R_m.
- b_{s_i}: Factor loading or sensitivity of the i^{th} asset to factor s. b_{s_i} is the exposure to size, i.e. $b_{s_i} = \ln(size)$. Factor s is R_s at time t.
- ϵ_i: The stock's idiosyncratic random shock with zero mean. These idiosyncratic risks are uncorrelated with the market and can be reduced by diversification. This is also called the residual of the regression.

Moreover, in Eq. (68), r_i, b_{m_i} and $b_{s_i} \in \mathbb{R}^{157\times 1}$. The parameters, a, R_m and R_s are estimated and this is shown in Fig. 21. These parameters are determined using Linear Regression which is performed using Ordinary Least Squares (OLS) method. ϵ_i is the residual of the regression which should be minimized according to the OLS method. Therefore, given the factor loadings, b_{m_i} and b_{s_i}, the parameters, a, R_m and R_s can be estimated.

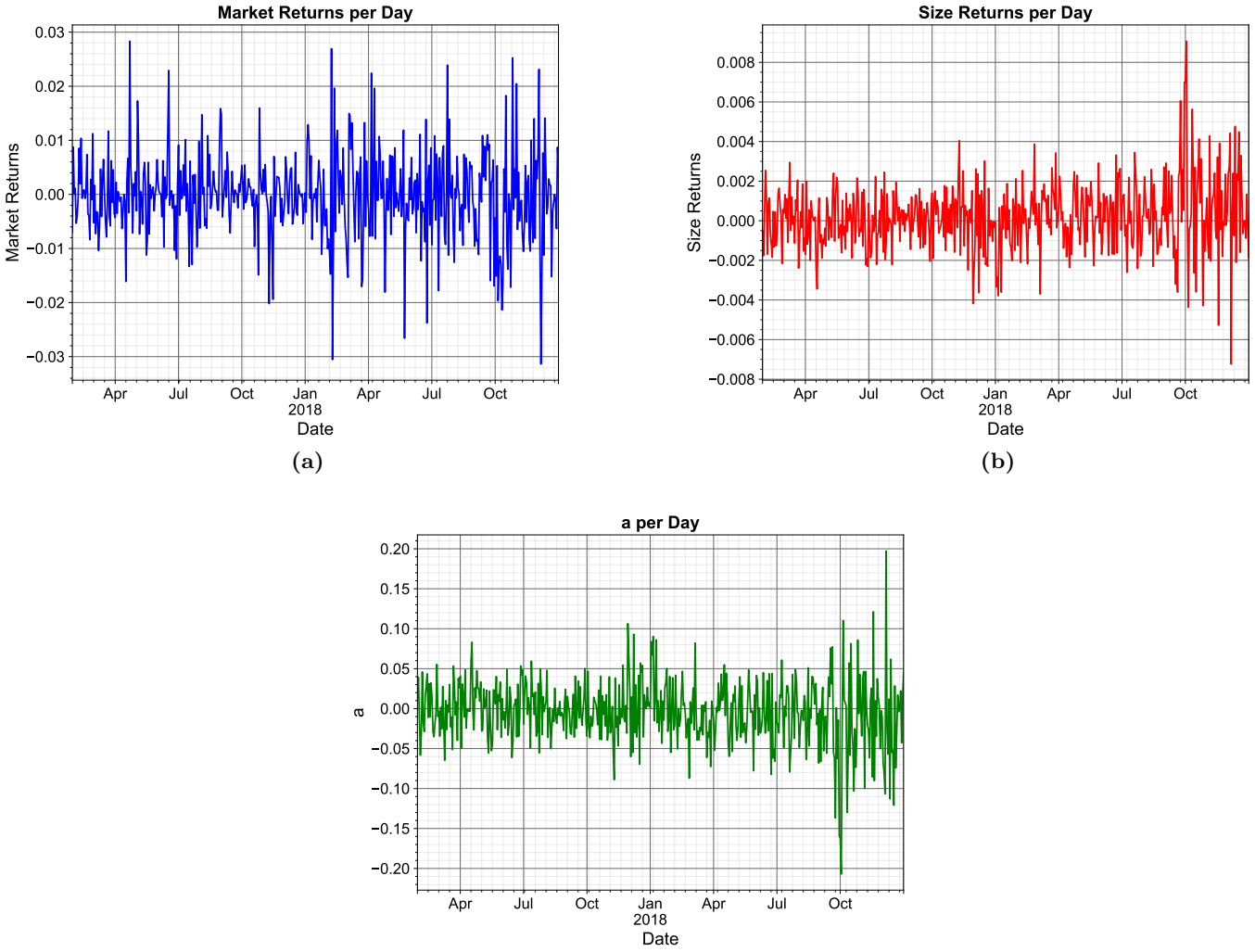


Figure 21: Plot of the Daily Market (a) and Size (b) Returns along with a (c)

(b): The market returns have a very high daily average standard deviation of 0.008 since at the aggregate level, the day-to-day variations of the market are significant. In fact, the parameter a had the daily highest average standard deviation of 0.06. Finally, the size returns have the lowest daily average standard deviation of 0.0002. An increase in the standard deviation (variation) of the size returns is also observed around October 2018, as shown in Fig. 21b. Finally, R_s has the least magnitude of 0.003 since it does not affect the return per company. This indicates that the company's size is not a significant factor in determining the stock return of a company. On the other hand, a has the highest average magnitude of 0.06, which indicates that the returns are highly dependent on this factor.

(c): The correlation through time for every company, is determined between r_i and ϵ_i. This is done to transform into a temporal domain. Therefore, the result is shown in Fig. 22.

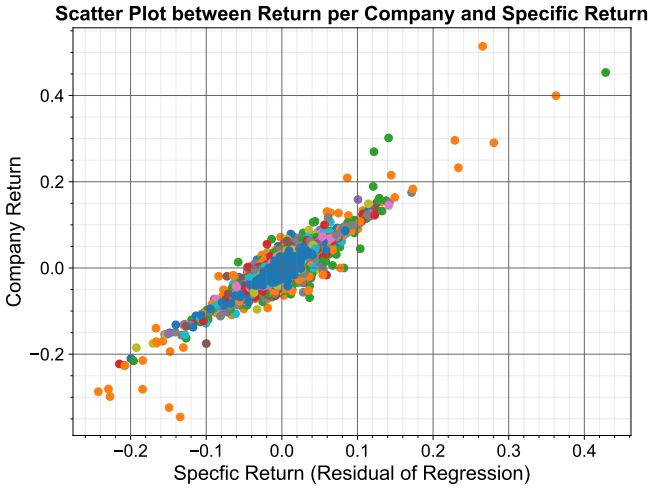


Figure 22: Scatter Plot between Return Per Company and Specific Return to determine the correlation through time for every company

(d):

	Rm	**Rs**
Rm	1.996404e-06	-3.163139e-07
Rs	-3.163139e-07	2.019414e-07

Table 4: Covariance Matrix of Matrix **R**

As observed in Table 4, the matrix is a 2×2 matrix since a 2-Factor model is being considered. The determinant of the matrix above is $3.0294671 \times 10^{-13}$. Additionally, the first element of the matrix is positive. Therefore, the symmetric matrix in Table 4 is positive definite and has positive eigenvalues, which indicate the stability of the system. Additionally, the magnitude of each element in the matrix is less than 1, which further indicates the stability of the matrix. The advantage of this representation is that this matrix has mathematical stability and low dimensionality. However, the magnitude of each element in the covariance matrix is extremely low. When large windows are used, the magnitude of the elements of the covariance matrix decreases to a significant extent. This may suggest the use of smaller window sizes, which may be more relevant for analysis since more recent events tend to affect the current market more.

(e):

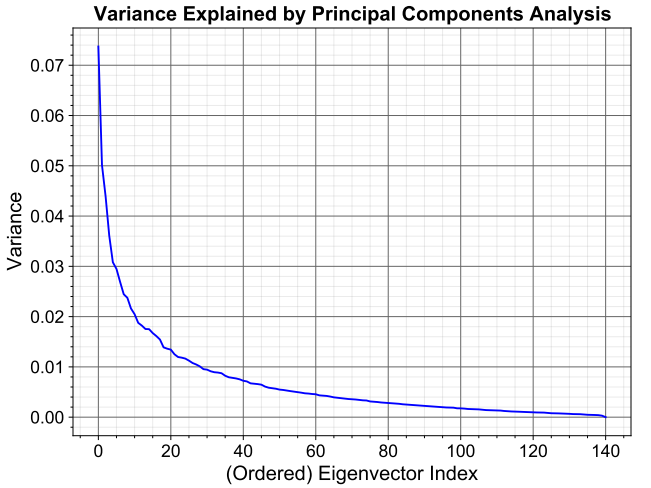


Figure 23: PCA performed on the Covariance Matrix, $cov(\mathbf{E})$ resulting in the Variance explained by the Principal Component

For every company, the residual regression, i.e. the specific return, $\varepsilon_{i,t}$ was determined in Section 2.4.5a. Therefore, a 157×157 covariance matrix was determined. PCA, a dimensionality reduction technique, was performed on this covariance matrix and the percentage of variance explained by each principal component in the matrix $\mathbf{E}$ is shown in Fig. 23. The eigenvectors of the PCA analysis represent the directions of the variations and the eigenvalues represent the magnitude of the directions. The percentage of variance explained by the first component is 7.41%. This means that 7.41% of the market volatility depends only on the first principal component.

3 Portfolio Optimization

3.1 Adaptive minimum-variance portfolio optimization

3.1.1

The optimal weights $\mathbf{w}_{opt}$ need to be determined to construct the minimum variance portfolio. The optimization problem is stated as shown in Eq. (69).

$$\begin{aligned} \min_{\mathbf{w}} \quad & J(\mathbf{w}, \mathbf{C}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{C}\mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^T\mathbf{1} = 1 \end{aligned} \tag{69}$$

In Eq. (69), $\mathbf{w}^T\mathbf{C}\mathbf{w}$ is the variance of the portfolio. Specifically, a portfolio consists of several assets. The weightings of each asset within the portfolio is given by Eq. (70)

$$\mathbf{w} = [w_1, \ldots, w_M]^T \tag{70}$$

On the other hand, $\mathbf{C}$ is the covariance matrix of all assets, which summarizes the risk structure of the system. The covariance matrix, $\mathbf{C}$, is given by Eq. (71).

$$\mathbf{C} = E\left\{(\mathbf{r}[t] - \boldsymbol{\mu})(\mathbf{r}[t] - \boldsymbol{\mu})^T\right\} \in \mathbb{R}^{M \times M} \tag{71}$$

Finally in Eq. (69), the constraint of the optimization problem is that $\mathbf{w}^T\mathbf{1} = 1$ where $\mathbf{1}$ is a vector of ones. This means that all portfolio weights sum to unity. Therefore, to solve the optimization problem in Eq. (69), the Lagrangian can be formed as shown in Eq. (72).

$$\min_{\mathbf{w},\lambda} J'(\mathbf{w}, \lambda, \mathbf{C}) = L = \frac{1}{2}\mathbf{w}^T\mathbf{C}\mathbf{w} - \lambda\left(\mathbf{w}^T\mathbf{1} - 1\right) \tag{72}$$

To determine the optimal weights of the portfolio, the Lagrangian in Eq. (72) has to be minimized. To minimize the Lagrangian in Eq. (72), it has to be differentiated with respect to $\mathbf{w}$ and λ. Therefore, the Lagrangian is differentiated with respect to $\mathbf{w}$ and is shown in Eq. (73).

$$\frac{dL}{d\mathbf{w}} = \mathbf{C}\mathbf{w} - \lambda\mathbf{1} = 0 \implies \mathbf{w} = \lambda\mathbf{C}^{-1}\mathbf{1} \tag{73}$$

Similarly, the Lagrangian in Eq. (72) is differentiated with respect to λ and is shown in Eq. (74)

$$\frac{dL}{d\lambda} = \mathbf{w}^T\mathbf{1} - 1 = 0 \implies \mathbf{w}^T\mathbf{1} = 1 \tag{74}$$

Eq. (73) and Eq. (74) are also referred to as the optimality conditions. The optimal weight vector $\mathbf{w}_{opt}$ can be determined by first substituting Eq. (73) in Eq. (74) and this is shown in Eq. (75)

$$(\lambda\mathbf{C}^{-1}\mathbf{1})^T\mathbf{1} = 1 \implies \lambda\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1} = 1 \implies \lambda = \frac{1}{\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1}} \tag{75}$$

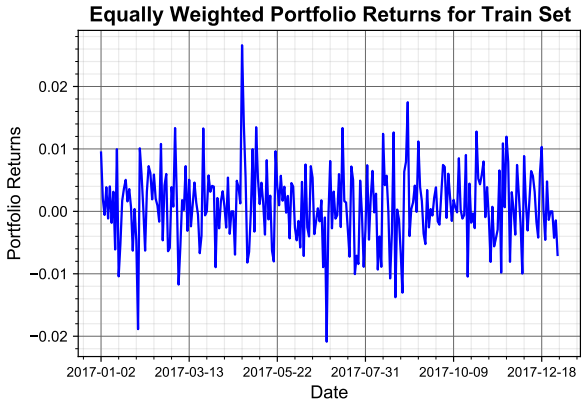
The result of λ in Eq. (75) can be substituted back in Eq. (73) and this is shown in Eq. (76). Therefore, the optimal weights of the portfolio with minimum variance, $\mathbf{w}_{opt}$ is shown in Eq. (76).

$$\mathbf{w}_{opt} = \frac{\mathbf{C}^{-1}\mathbf{1}}{\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1}} \tag{76}$$

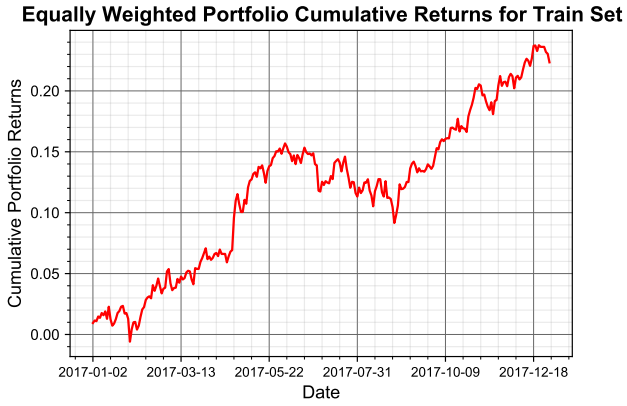
The theoretical variance of the optimal portfolio, σ_p^2, can be determined by substituting the optimal weights derived in Eq. (76) in $\mathbf{w}_{opt}\mathbf{C}\mathbf{w}_{opt}$. This is illustrated in Eq. (77).

$$\sigma_p^2 = \mathbf{w}_{opt}^T\mathbf{C}\mathbf{w}_{opt} = \left(\frac{\mathbf{C}^{-1}\mathbf{1}}{\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1}}\right)^T \mathbf{C}\frac{\mathbf{C}^{-1}\mathbf{1}}{\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1}} = \frac{(\mathbf{C}^{-1}\mathbf{1})^T\mathbf{1}}{(\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1})^2} = \frac{\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1}}{(\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1})^2} = \frac{1}{\mathbf{1}^T(\mathbf{C}^{-1})^T\mathbf{1}} \tag{77}$$

3.1.2

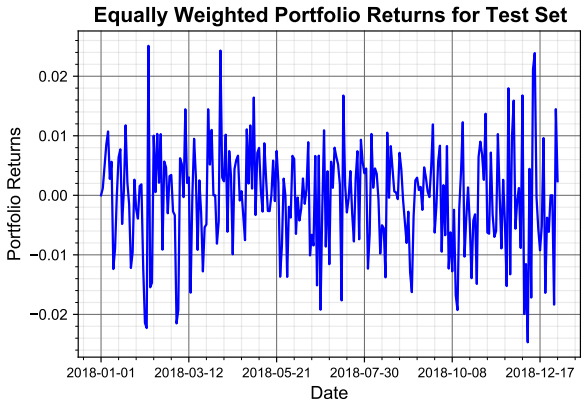


(a) Portfolio Returns

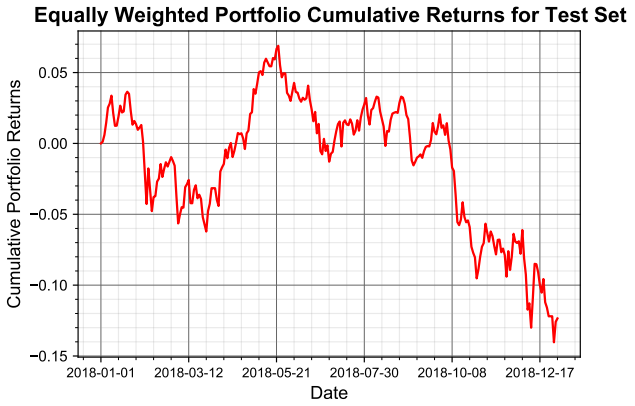


(b) Cumulative Returns

Figure 24: Daily Portfolio Returns (left) and Cumulative Returns (right) for an Equally-Weighted Portfolio on the Training Set

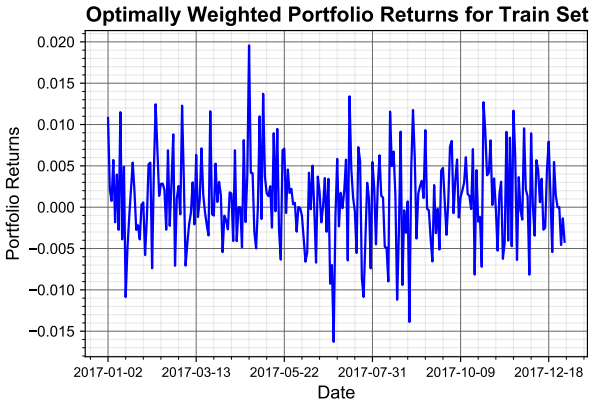


(a) Portfolio Returns

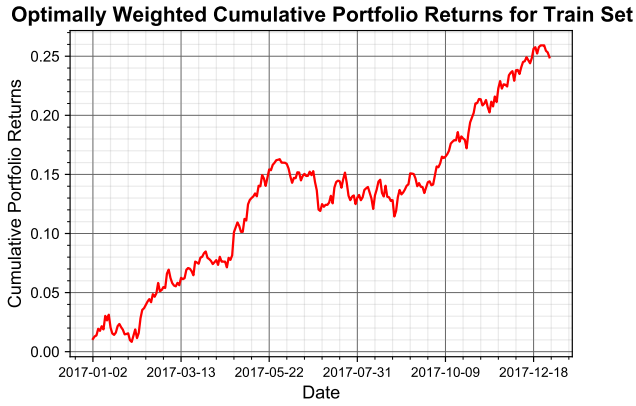


(b) Cumulative Returns

Figure 25: Daily Portfolio Returns (left) and Cumulative Returns (right) for an Equally-Weighted Portfolio on the Test Set

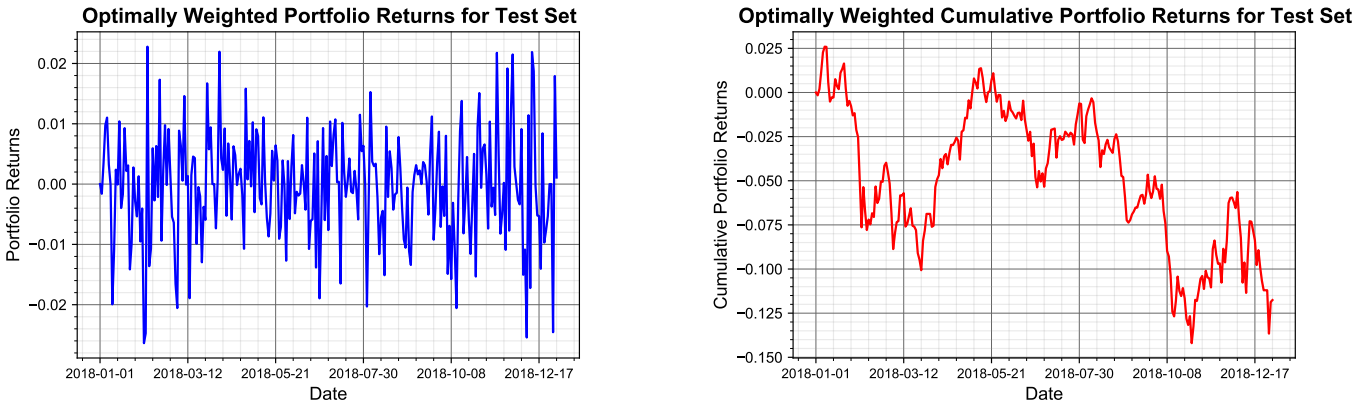


(a) Portfolio Returns



(b) Cumulative Returns

Figure 26: Daily Portfolio Returns (left) and Cumulative Returns (right) for an Optimally-Weighted Portfolio on the Train Set



(a) Portfolio Returns

(b) Cumulative Returns

Figure 27: Daily Portfolio Returns (left) and Cumulative Returns (right) for an Optimally-Weighted Portfolio on the Test Set

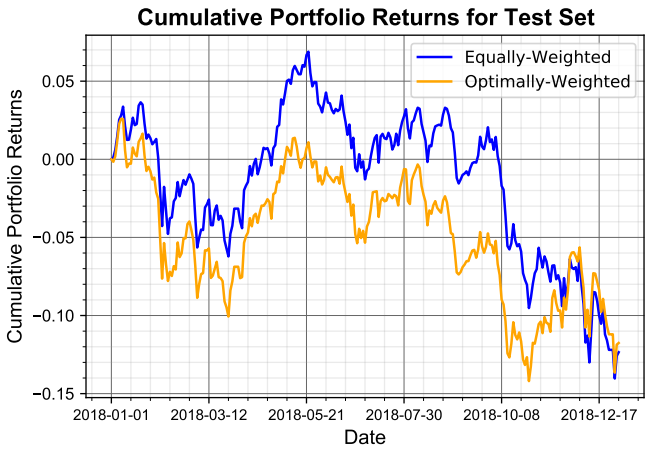


Figure 28: Comparison of the Cumulative Portfolio Returns for an Equally-Weighted Portfolio (blue plot) and Optimally-Weighted Portfolio (orange plot) for the Test Set

A portfolio with equally-weighted assets is computed on the training set. Therefore, the portfolio returns and cumulative returns for an equally-weighted portfolio for the training and test set are shown in Fig. 24 and Fig. 25, respectively.

The minimum variance portfolio weights or optimal weights for the portfolio are computed on the training set using the Lagrangian Optimization problem in Eq. (69). Therefore, the minimum variance or optimal weights for the portfolio are computed on the training set from 02/01/2017 to 31/12/2017 using Eq. (76). Therefore, the optimally weighted portfolio returns and cumulative returns are shown in Fig. 26.

The optimal weights derived from the training set are then applied to the test set. Therefore, the portfolio returns and cumulative returns for the minimum variance (optimally-weighted) portfolio for the test set from 02/01/2018 to 31/12/2018 is shown in Fig. 27.

Therefore, the cumulative portfolio returns for an equally-weighted portfolio and optimally-weighted portfolio for the test set are shown together in Fig. 28, for comparison.

	Equally-Weighted	**Optimally-Weighted**
Mean of Portfolio Returns	0.00085955	0.00095793
Variance	3.75×10^{-5}	2.86×10^{-5}
Cumulative Return	0.223484	0.249062

Table 5: Characteristics of Equally-Weighted and Minimum Variance (Optimally-Weighted) Portfolios on the Training Set

	Equally-Weighted	Optimally-Weighted
Mean of Portfolio Returns	-0.00047319	-0.0004505
Variance	7.93×10^{-5}	8.19×10^{-5}
Cumulative Return	-0.123503	-0.117582

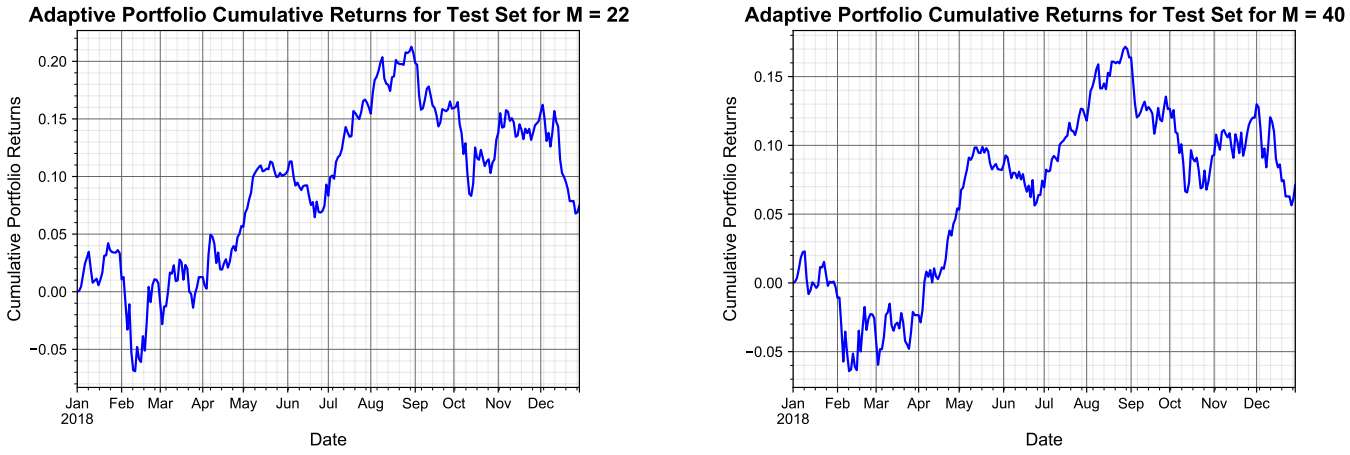
Table 6: Characteristics of Equally-Weighted and Minimum Variance (Optimally-Weighted) Portfolios on the Test Set

From Table 5, the minimum variance portfolio based on optimal weights has a higher mean and cumulative return than the equally-weighted portfolio for the training set. Moreover, the optimally-weighted portfolio's variance is also lower than the variance of the equally-weighted portfolio. Therefore, the optimally-weighted portfolio performs better than the equally-weighted portfolio, on the training set.

The optimally-weighted portfolio also performs better than the equally-weighted portfolio for the test set. The optimally weighted portfolio has a higher mean of portfolio returns and higher cumulative return, compared to the equally-weighted portfolio. But at the same time, the optimally-weighted portfolio also has a higher variance than the equally-weighted portfolio on the test set.

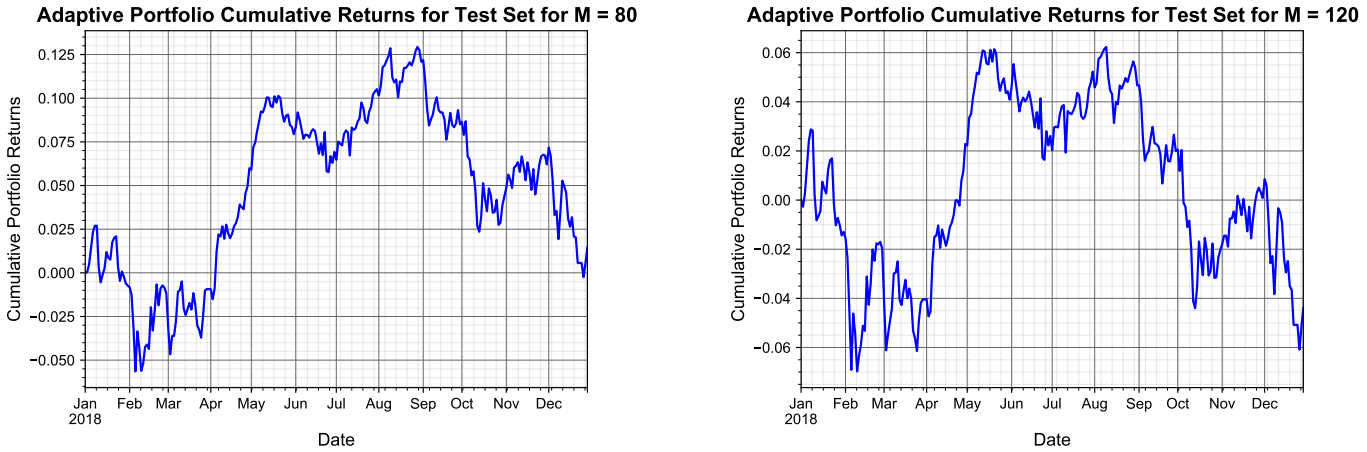
Finally, the theoretical portfolio variance for the optimally-weighted portfolio was determined using Eq. (77) which is equal to 2.86×10^{-5}. This theoretical variance exactly matches the optimally-weighted portfolio's variance for the training set, as shown in Table 5. This is because the optimal weights are determined using the training set.

3.1.3



(a) M = 22 **(b)** M = 40

Figure 29: Cumulative Returns for Adaptive Time-Varying Minimum Variance Portfolios with $M = 22$ (left) and $M = 40$ (right) for the Test Set



(a) M = 80 **(b)** M = 100

Figure 30: Cumulative Returns for Adaptive Time-Varying Minimum Variance Portfolios with $M = 80$ (left) and $M = 120$ (right) for the Test Set

	Eq. Weighted	**Opt. Weighted**	**Adaptive Minimum Variance Portfolios**			
			M = 22	M = 40	M = 80	M = 100
Mean	-0.0004732	-0.0004505	0.0002865	0.0002730	5.46×10^{-5}	-0.0001686
Variance	7.93×10^{-5}	8.19×10^{-5}	1.08×10^{-4}	7.99×10^{-5}	7.12×10^{-5}	6.99×10^{-5}
Cumulative Ret.	-0.1235031	-0.1175824	0.0747735	0.0712478	0.0142627	-0.0440088

Table 7: Characteristics of Equally-Weighted, Minimum Variance (Optimally-Weighted) and Adaptive Minimum Variance Portfolios on the Test Set

Comparing the Performances:

Fig. 29 and Fig. 30 show the plots of cumulative portfolio returns for Adaptive Time-Varying Minimum Variance Portfolios for different length of windows, i.e. M for the test set. The characteristics of the adaptive portfolio such as daily mean of portfolio returns, variance and cumulative returns for different M, on the test set, are shown in Table 7. These are also compared alongside the characteristics of equally-weighted and optimally-weighted portfolios on the test set, from Table 6.

From Table 7, the Adaptive Time-Varying Minimum Variance Portfolios for any M has a higher mean of daily portfolio returns and cumulative returns than the equally-weighted and optimally-weighted portfolios for the test set from 02/01/2018 to 31/12/2018. Specifically, the adaptive minimum variance portfolio performs the best for $M = 22$ since it has the highest mean. But at the same time, it also has the highest variance. Additionally, using a larger window, M, will lead to decreasing cumulative returns with very little changes in the variance.

Effect of Recursive Update of the Variables:

The minimum variance optimization technique from Section 3.1.2 is used in a recursive manner to determine the optimal weights wherein the weights are updated based on a sliding window of M days. This way, the optimization problem considers the recent changes within the markets and accordingly, modifies the optimal weights.

However, as seen from Table 7, the variances for different window lengths for an adaptive minimum variance portfolio is high. This is because the weights calculated at each instant is uncorrelated with each other and hence, the adaptive system is not learning from the previous errors. A better method would be to use gradient descent to repeatedly reduce the error and obtain the optimal weights.

Alternative Method to Compute the Sample Covariance Matrix:

Instead of using the sample covariance estimator to estimate the covariance matrix, the Minimum Covariance Determinant (MCD) can also be used to compute the covariance matrix. The algorithm for the Raw MCD estimator involves the following steps:

1. **Step 1:** This involves choosing H samples form N data samples, such that, $(N/2 \leq H \leq N)$ and this should satisfy the minimal determinant of the sample covariance matrix of the H samples.

2. **Step 2:** It is also important to further consider two variables. μ_{MCD} is the sample mean of H samples and Σ_{MCD} is proportional to the sample covariance matrix of the H sakmples by a consistency factor. Affine Equivariance refers to the invariance with respect to a change of affine. Therefore, the affine equivariances are given in Eq. (78) and Eq. (79).

$$\boldsymbol{\mu}_{MCD}(\mathbf{XA}+\mathbf{b}) = \boldsymbol{\mu}_{MCD}(\mathbf{X})\mathbf{A}+\mathbf{b} \tag{78}$$

$$\boldsymbol{\Sigma}_{MCD}(\mathbf{XA}+\mathbf{b}) = \mathbf{A}^T\boldsymbol{\Sigma}_{MCD}(\mathbf{X})\mathbf{A} \tag{79}$$

3. **Step 3:** The N-samples are reordered from small to large, in terms of the Mahalanobis distance, i.e. $(\mathbf{x}_n - \boldsymbol{\mu}_{MCD})^T \boldsymbol{\Sigma}_{MCD}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_{MCD})$.

4. **Step 4:** Steps 1-3 are repeated until convergence.

The MCD determinant is a robust covariance matrix estimation technique. However, this robustness does not hold in extreme cases which are large losses that are of great concern. On the other hand, the sample covariance matrix has a very high sample risk since the model is just based on samples. **Finally, using Exponentially Weighted Averages, which is based on adaptive models, will also behave well on the data.** In the Exponentially Weighted Averages technique, more weightage can be given to recent days such that it can have a stronger influence on the estimation of the covariance matrix.

4 Robust Statistics and Non-Linear Methods

4.1 Data Import and Exploratory Data Analysis

4.1.1

	Open	High	Low	Close	Adj. Close	Volume
Mean	187.687	189.562	185.824	187.712	186.174	3.270e+07
Median	186.290	187.400	184.940	186.120	184.352	2.918e+07
Standard Deviation	22.146	22.282	22.009	22.161	21.905	1.418e+07
Median Absolute Deviation	15.890	15.610	15.920	15.940	15.476	7.574e+06
Interquartile Range	36.000	36.340	36.060	36.755	35.685	1.631e+07

Table 8: Descriptive Statistics for Apple (AAPL) from 16/03/2018 to 11/03/2019

	Open	High	Low	Close	Adj. Close	Volume
Mean	138.454	139.492	137.329	138.363	134.903	5.199e+06
Median	142.810	143.990	142.060	142.710	138.566	4.238e+06
Standard Deviation	12.114	11.913	12.205	12.028	10.672	3.329e+06
Median Absolute Deviation	5.270	5.310	5.190	5.230	4.494	9.207e+05
Interquartile Range	15.380	14.720	16.340	15.505	14.104	1.953e+06

Table 9: Descriptive Statistics for IBM (IBM) from 16/03/2018 to 11/03/2019

	Open	High	Low	Close	Adj. Close	Volume
Mean	108.708	109.652	107.683	108.607	107.263	1.470e07
Median	109.180	110.530	107.790	109.020	107.219	1.363e+07
Standard Deviation	5.359	5.203	5.433	5.300	4.833	5.350e+06
Median Absolute Deviation	4.470	4.310	4.240	4.350	3.450	3.035e+06
Interquartile Range	8.810	8.845	8.845	8.835	7.222	6.233e+06

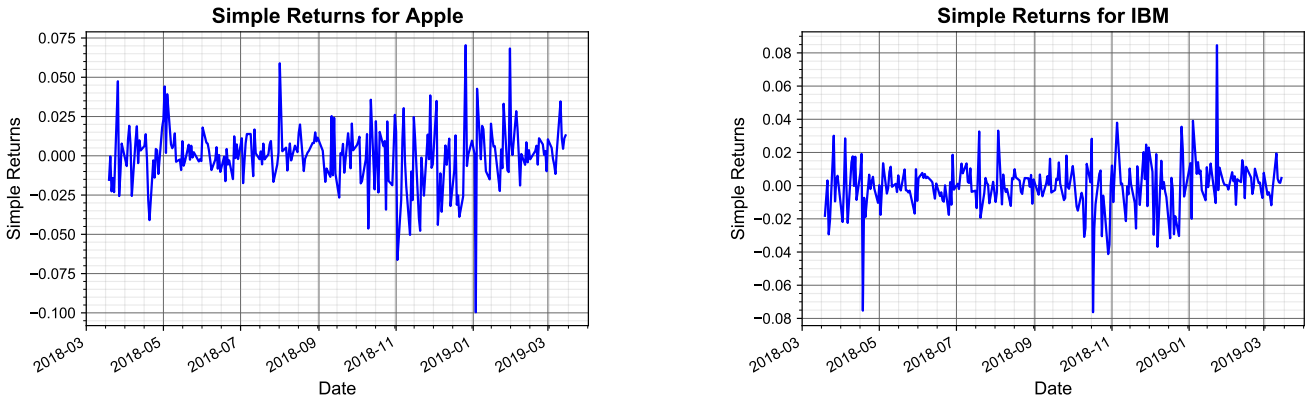
Table 10: Descriptive Statistics for J.P. Morgan (JPM) from 16/03/2018 to 11/03/2019

	Open	High	Low	Close	Adj. Close	Volume
Mean	25001.257	25142.042	24846.002	24999.154	24999.154	3.329e+08
Median	25025.580	25124.100	24883.039	25044.289	25044.289	3.138+08
Standard Deviation	858.835	815.204	903.302	859.132	859.132	9.408e+07
Median Absolute Deviation	543.541	537.619	601.568	590.721	590.721	5.046e+07
Interquartile Range	1109.435	1077.816	1204.419	1158.155	1158.155	1.089e+08

Table 11: Descriptive Statistics for the Dow Jones Index (DJI) from 16/03/2018 to 11/03/2019

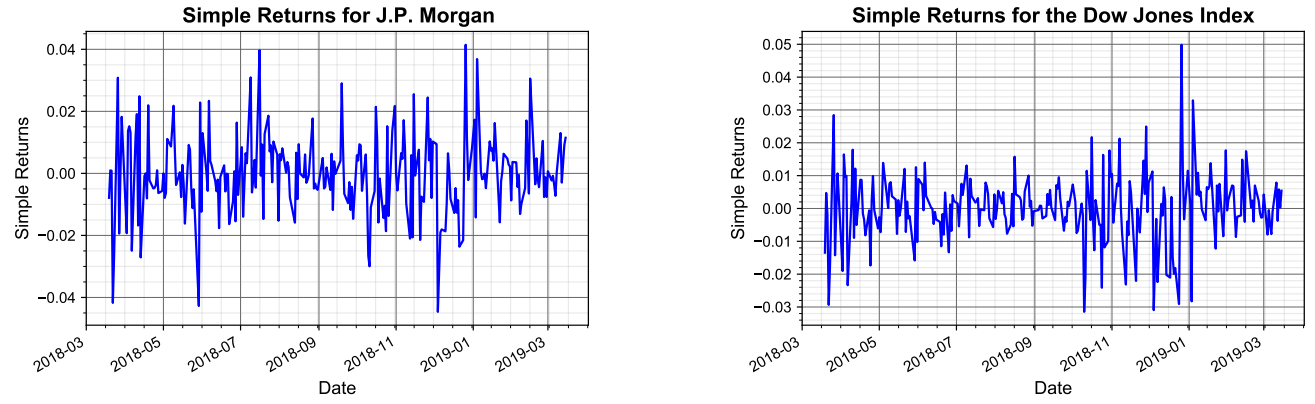
Table 8, Table 9, Table 10 and Table 11 show the descriptive statistics for Apple, IBM, J.P. Morgan and the Dow Jones Index, respectively, from 16/03/2018 to 11/03/2019. The descriptive statistics chosen are Mean, Median, Standard Deviation, Median Absolute Deviation (MAD) and Interquartile Range (IQR). The Median Absolute Deviation (MAD) is a robust measure of how spread out the data is. MAD is calculated using Eq. (80) where X_i is the i^{th} value in the dataset and $\tilde{X}$ is the median of the dataset.

$$\text{MAD} = \text{Median}\left(\left|X_i - \tilde{X}\right|\right) \tag{80}$$



(a) Apple **(b)** IBM

Figure 31: Daily Simple Returns for Apple (a) and IBM (b) using the adjusted closing prices from 16/03/2018 to 11/03/2019

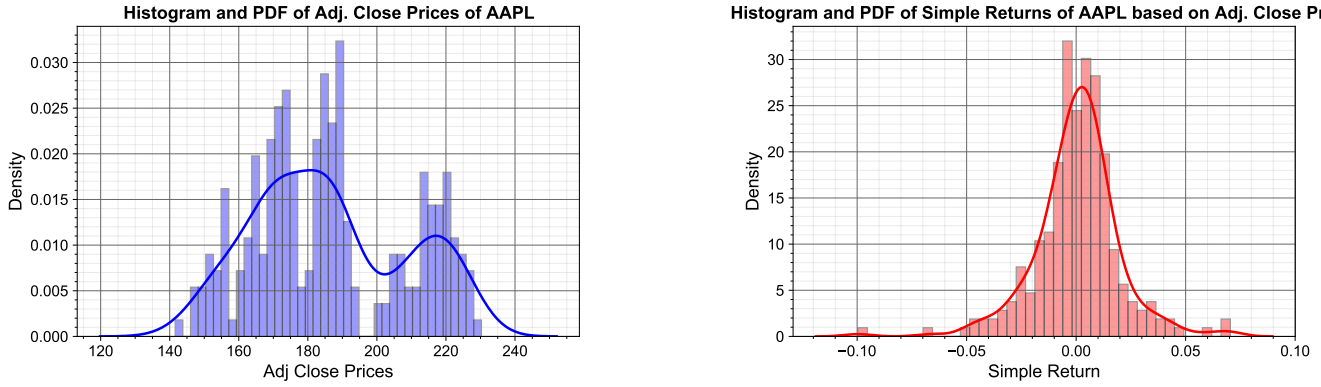


(a) J.P. Morgan **(b)** The Dow Jones Index

Figure 32: Daily Simple Returns for J.P. Morgan (a) and The Dow Jones Index (b) using the adjusted closing prices from 16/03/2018 to 11/03/2019

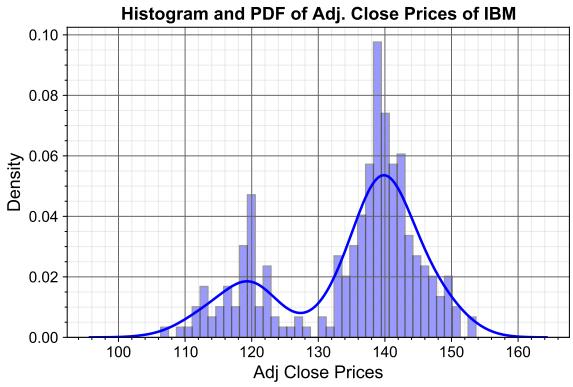
Finally, the daily simple returns for Apple, IBM, J.P. Morgan and the Dow Jones Index using the adjusted closing prices from 16/03/2018 to 11/03/2019 are shown in Fig. 31 and Fig. 32.

4.1.2

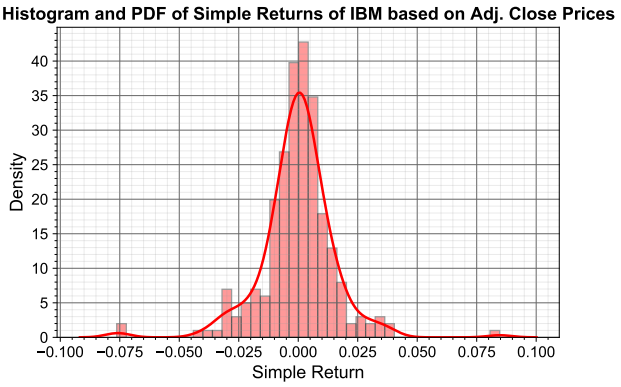


(a) Adj. Close Prices **(b)** Simple Returns based on Adj. Close Prices

Figure 33: Histogram and PDF of Adj. Close Prices (a) and Daily Simple Returns based on Adj. Close Prices (b) of Apple

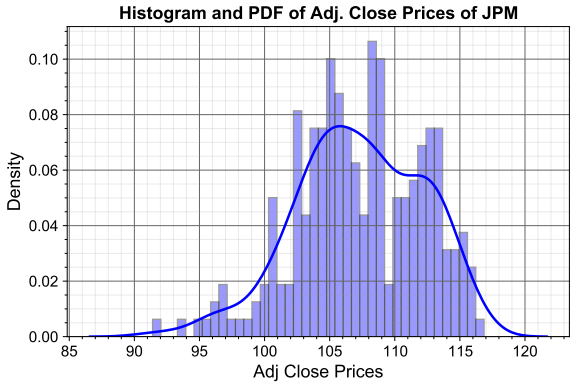


(a) Adj. Close Prices

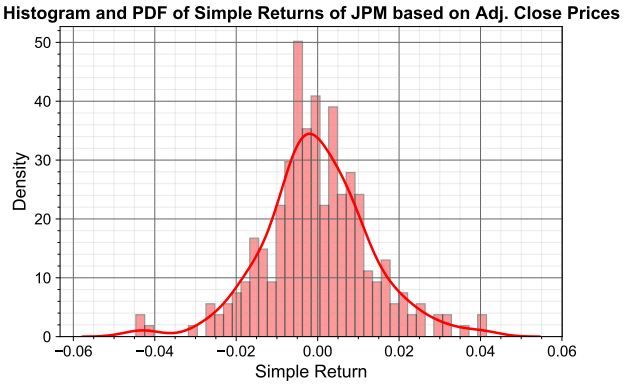


(b) Simple Returns based on Adj. Close Prices

Figure 34: Histogram and PDF of Adj. Close Prices (a) and Daily Simple Returns based on Adj. Close Prices (b) of IBM

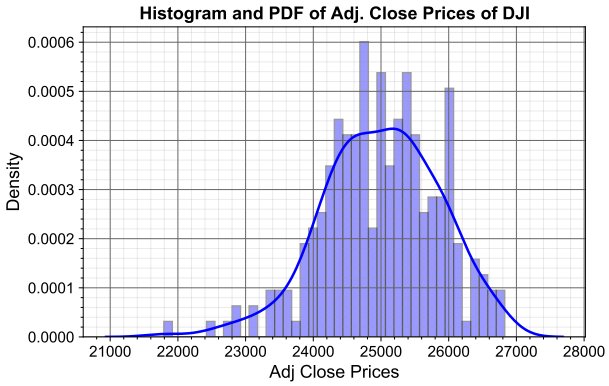


(a) Adj. Close Prices

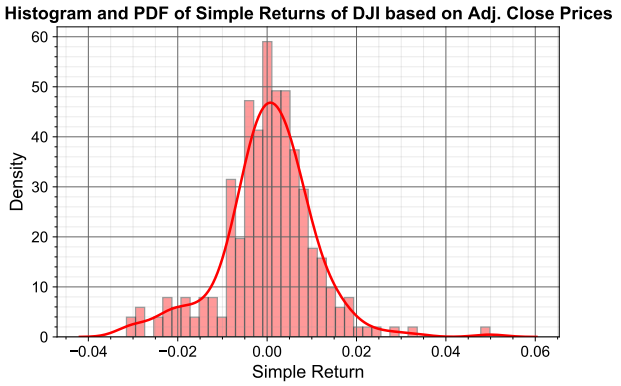


(b) Simple Returns based on Adj. Close Prices

Figure 35: Histogram and PDF of Adj. Close Prices (a) and Daily Simple Returns based on Adj. Close Prices (b) of J.P. Morgan



(a) Adj. Close Prices



(b) Simple Returns based on Adj. Close Prices

Figure 36: Histogram and PDF of Adj. Close Prices (a) and Daily Simple Returns based on Adj. Close Prices (b) of the Dow Jones Index

The histogram and Probability Density Function (PDF) of adjusted close prices and daily simple returns based on adjusted close prices are shown for Apple, IBM, J.P. Morgan and the Dow Jones index in Fig. 33, Fig. 34 and Fig. 35 and Fig. 36, respectively.

The histogram and the PDF were determined using the `distplot` function in the `seaborn` library. The `distplot` function can not only plot histograms for the data, using the `hist` function but also plot the estimated PDF of the data using the `kdeplot` function. The `kdeplot` function is a Kernel Density Estimate (KDE) plot

method for visualizing the distribution of observations in a dataset, analagous to a histogram. KDE represents the data using a continuous probability density curve in one or more dimensions. An example of plotting the histogram and the PDF of adjusted close prices of Apple is shown in Listing 1.

```
sns.distplot(apple_data['Adj Close'], hist=True, kde=True, bins=40, color = 'blue', hist_kws={'
    edgecolor':'black'}, kde_kws={'linewidth': 2})
```

Listing 1: Determining the Histogram and PDF of Apple's Adjusted Close Price using the `distplot` function

From Fig. 33, Fig. 34, Fig. 35 and Fig. 36, it can be seen that the PDF of the daily simple returns based on adjusted close prices for the 3 stocks and the index are centered around 0. On the other hand, the PDF of the adjusted close prices itself is not centered around 0. The daily simple returns based on adjusted close prices follow a more normal or Gaussian distribution with 0 mean than the adjusted close prices. This suggests the use of returns over raw prices for statistical analysis. Moreover, the PDFs of the adjusted close price of Apple and IBM have 2 distinguishable peaks, which can be seen as the sum of two normal distributions.

4.1.3

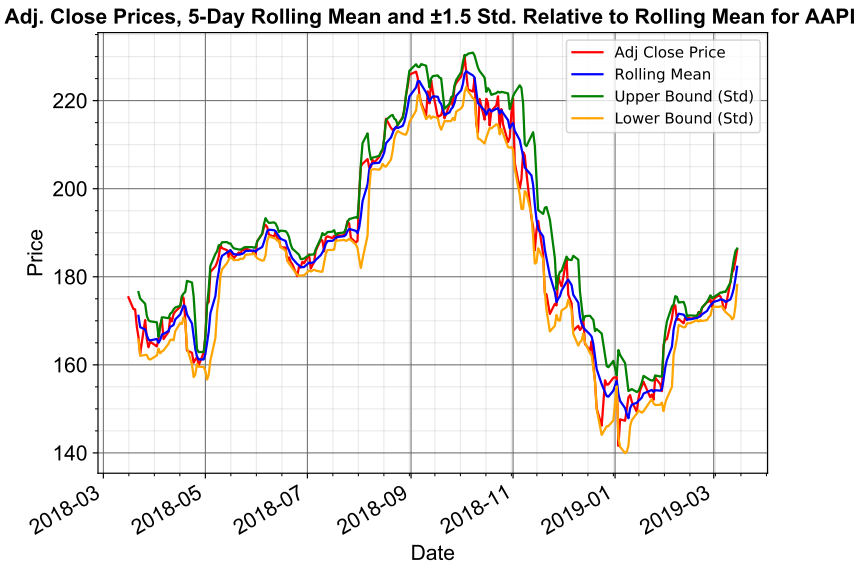


Figure 37: Adjusted Close Price (Red), 5-Day Rolling Mean of Adjusted Close Price (Blue), $\pm$ 1.5 $\times$ Standard Deviations Relative to the Rolling Mean (Green, Orange) for Apple from 16/03/2018 to 11/03/2019

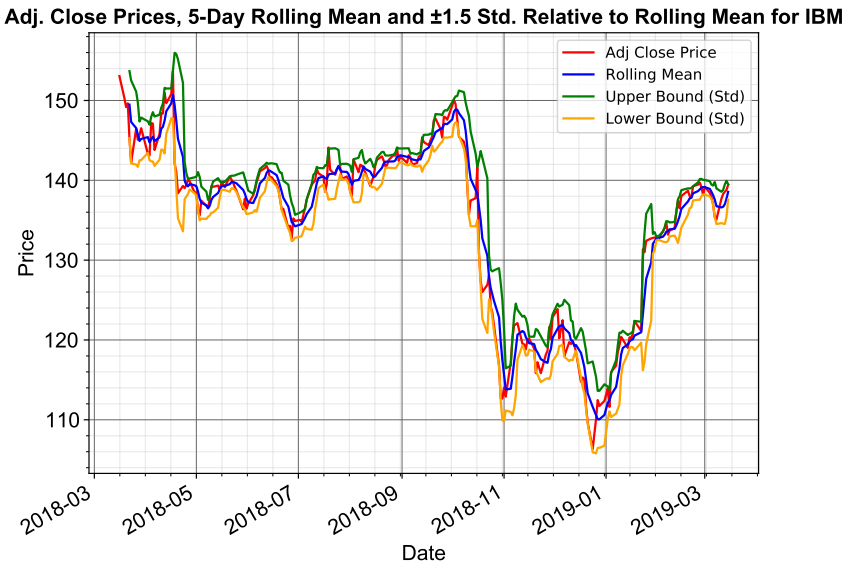


Figure 38: Adjusted Close Price (Red), 5-Day Rolling Mean of Adjusted Close Price (Blue), ± 1.5 × Standard Deviations Relative to the Rolling Mean (Green, Orange) for IBM from 16/03/2018 to 11/03/2019

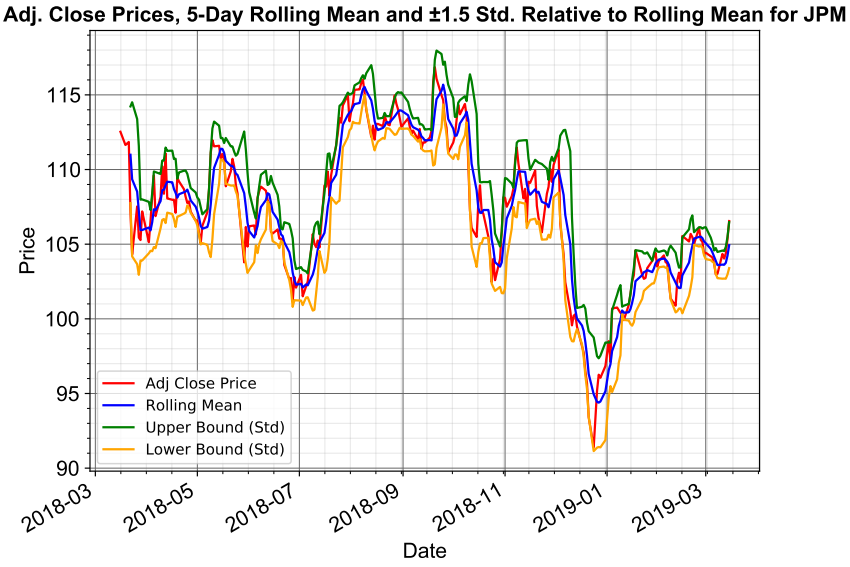


Figure 39: Adjusted Close Price (Red), 5-Day Rolling Mean of Adjusted Close Price (Blue), ± 1.5 × Standard Deviations Relative to the Rolling Mean (Green, Orange) for J.P. Morgan from 16/03/2018 to 11/03/2019

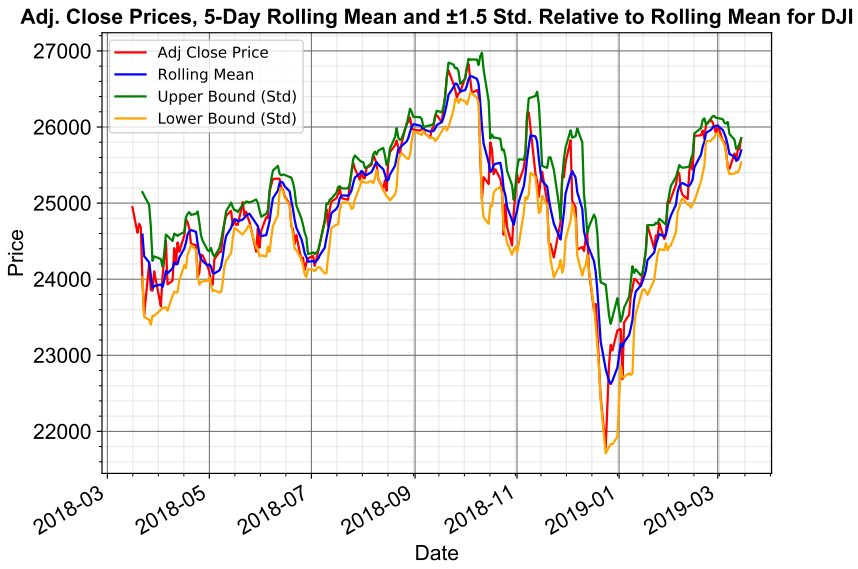


Figure 40: Adjusted Close Price (Red), 5-Day Rolling Mean of Adjusted Close Price (Blue), ± 1.5 × Standard Deviations Relative to the Rolling Mean (Green, Orange) for the Dow Jones Index from 16/03/2018 to 11/03/2019

In the first technique, all prices outside of the Rolling Price Mean ± 1.5 × Standard Deviation of the Rolling Price region are classified as anomalies. Specifically, this technique involves creating an upper bound and a lower bound. The upper and lower bound plots in Fig. 37, Fig. 38, Fig. 39 and Fig. 40 are determined for Apple, IBM, J.P. Morgan and the Dow Jones Index, respectively. These upper and lower bound plots are determined using Eq. (81) and Eq. (82), respectively.

$$\text{Upper Bound} = \text{5-Day Rolling Mean} + (1.5 \times \text{5-Day Rolling Standard Deviation}) \tag{81}$$

$$\text{Lower Bound} = \text{5-Day Rolling Mean} - (1.5 \times \text{5-Day Rolling Standard Deviation}) \tag{82}$$

A window of 5-days is used to compute the rolling mean and the rolling standard deviation. The second technique to determine the anomalies is also explored.

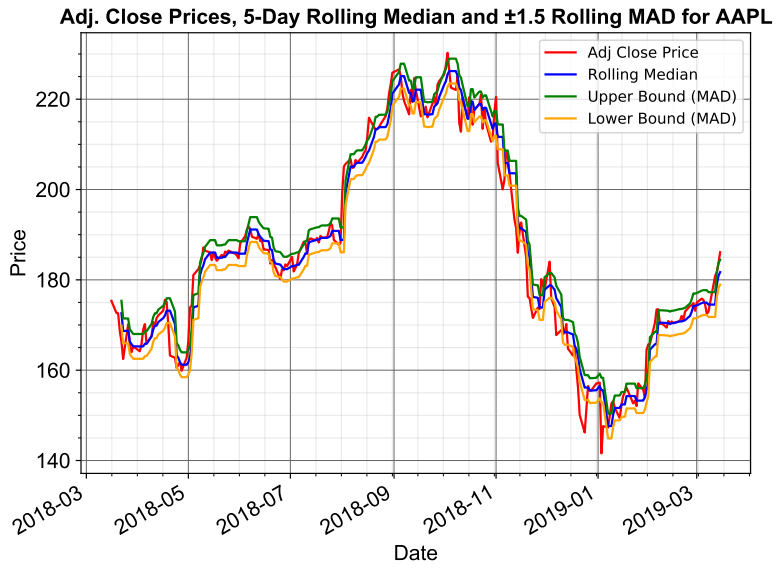


Figure 41: Adjusted Close Price (Red), 5-Day Rolling Median of Adjusted Close Price (Blue), ± 1.5 × Median Absolute Deviations Relative to the Rolling Median (Green, Orange) for Apple from 16/03/2018 to 11/03/2019

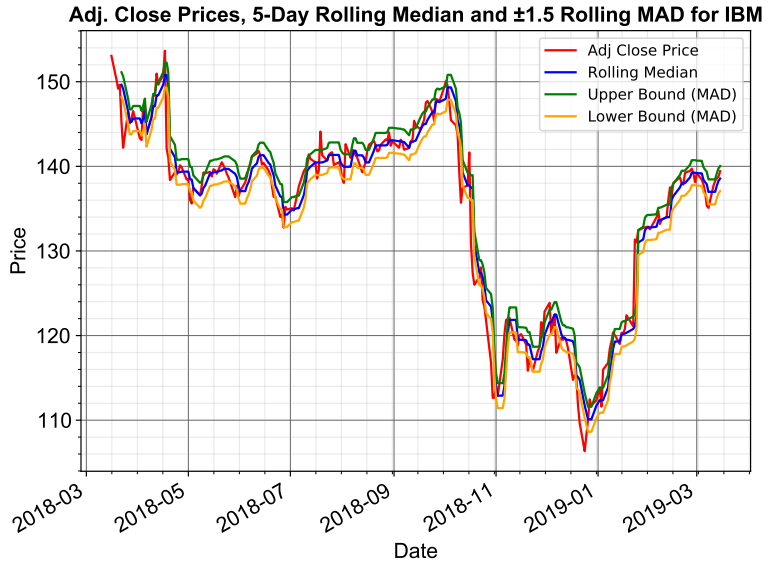


Figure 42: Adjusted Close Price (Red), 5-Day Rolling Median of Adjusted Close Price (Blue), ± 1.5 × Median Absolute Deviations Relative to the Rolling Median (Green, Orange) for IBM from 16/03/2018 to 11/03/2019

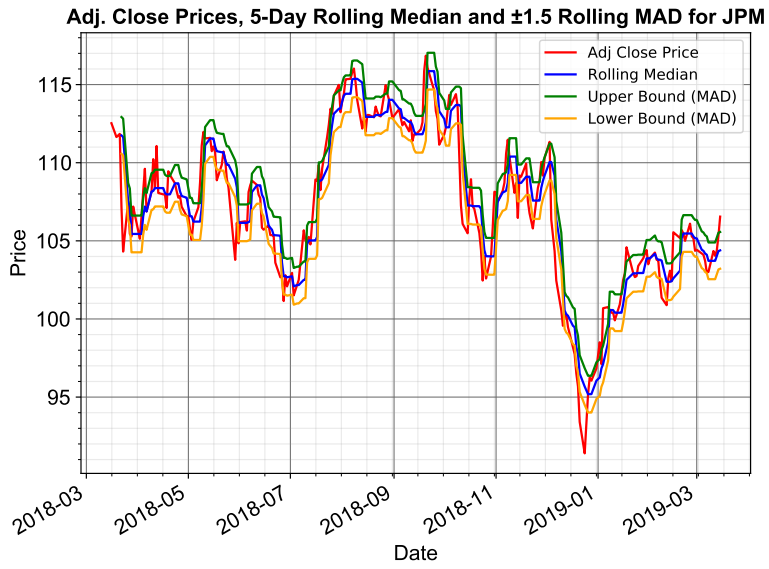


Figure 43: Adjusted Close Price (Red), 5-Day Rolling Median of Adjusted Close Price (Blue), ± 1.5 × Median Absolute Deviations Relative to the Rolling Median (Green, Orange) for J.P. Morgan from 16/03/2018 to 11/03/2019

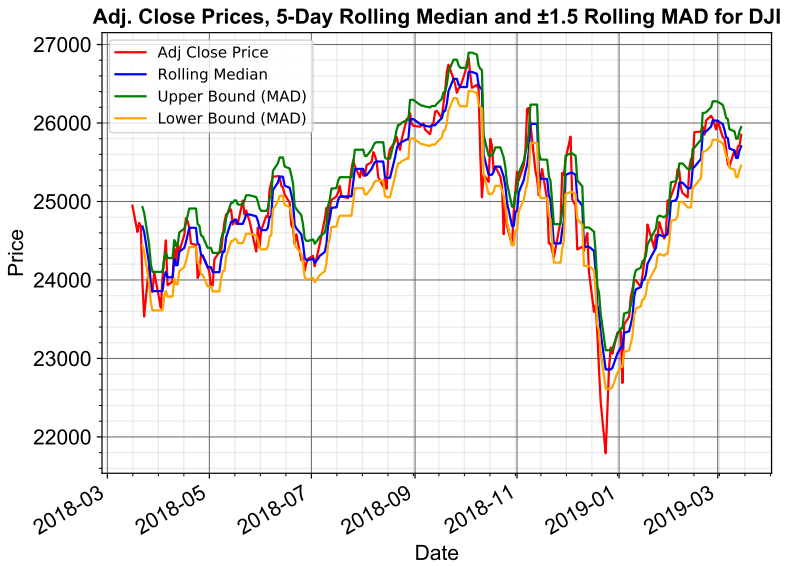


Figure 44: Adjusted Close Price (Red), 5-Day Rolling Median of Adjusted Close Price (Blue), ± 1.5 × Median Absolute Deviations Relative to the Rolling Median (Green, Orange) for the Dow Jones Index from 16/03/2018 to 11/03/2019

The second technique to determine the anomalies involves considering the median and the median absolute deviation (MAD) of the financial data. In this technique, all prices outside of the Rolling Price Median ± 1.5 × Rolling Median Absolute Deviation (MAD) region are classified as anomalies. This technique also involves creating an upper bound and a lower bound. The upper and lower bound plots in Fig. 41, Fig. 42, Fig. 43 and Fig. 44 are determined for Apple, IBM, J.P. Morgan and the Dow Jones Index. These upper and lower bound plots are determined using Eq. (83) and Eq. (84).

$$\text{Upper Bound} = \text{5-Day Rolling Median} + (1.5 \times \text{5-Day Rolling Median Absolute Deviation}) \tag{83}$$

$$\text{Lower Bound} = \text{5-Day Rolling Median} - (1.5 \times \text{5-Day Rolling Median Absolute Deviation}) \tag{84}$$

The 5-Day Rolling Median Absolute Deviation is determined using using Eq. (80). For instance, the 5-Day Rolling Median Absolute Deviation for Apple is shown in Listing 2.

```
rolling_mad = abs(apple_data['Adj Close'] -  apple_data['Adj Close'].rolling(5).median()).
    median()
```

Listing 2: Determining the Median Absolute Deviation of Apple

In Listing 2, `rolling_mad` is the Rolling Median Absolute Deviation variable. `apple_data` is the Pandas dataframe of Apple consisting of Open, High, Low and Adjusted Prices. The 5-day rolling window median of the Adjusted Prices of Apple is calculated which is then subtracted from each of the Adjusted Prices. The median of the absolute of value of the aforementioned result gives the 5-Day Rolling Median Absolute Deviation.

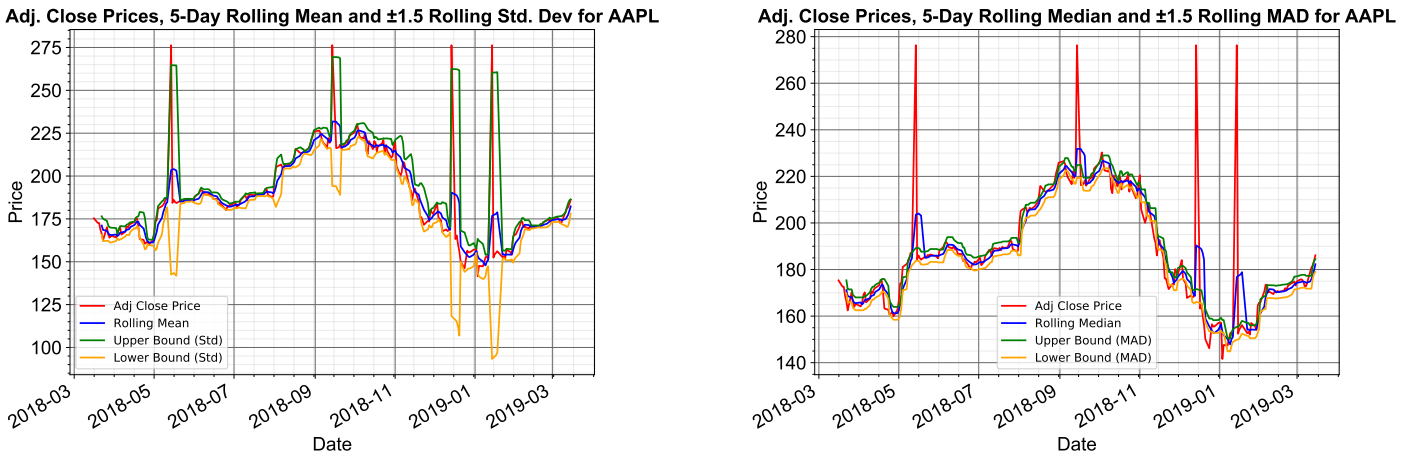
Technique 1, i.e. the mean and standard deviation method, shown in Fig. 37, Fig. 38, Fig. 39 and Fig. 40, are deeply affected by the presence of outliers and therefore, the technique is not robust. In other words, Technique 1 is more susceptible to changes. This is also discussed in Section 4.1.4.

Entity	**Anomalies**	
	Mean and Std. Dev Technique	**Median and MAD Technique**
Apple	30	90
IBM	31	89
J.P. Morgan	33	92
The Dow Jones Index	30	80

Table 12: Anomaly Count for Apple, IBM, J.P. Morgan and the Dow Jones Index using the upper and lower bound created with the two techniques

Moreover, from Table 12, all of the entities have a higher anomaly count using the Median and Median Absolute Deviation technique (Technique 2) compared to using the Mean and Standard Deviation technique (Technique 1). This shows that Technique 2 is more robust than Technique 1.

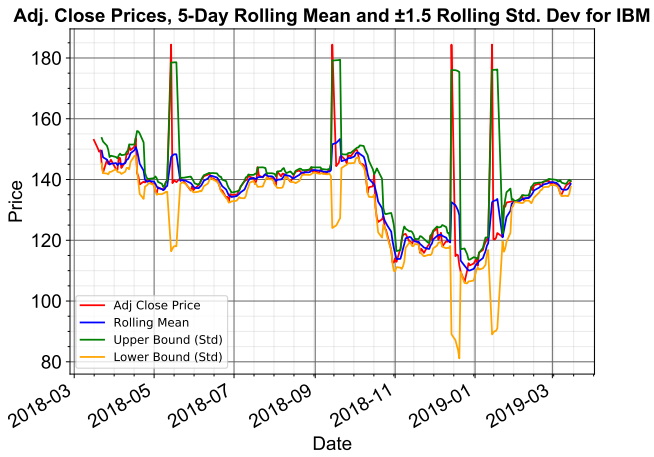
4.1.4



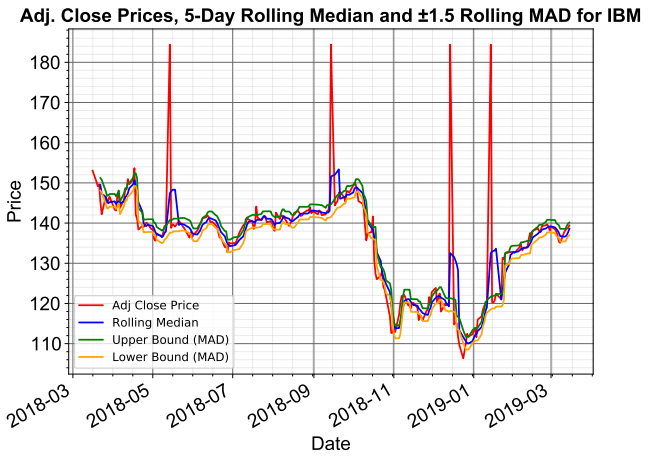
(a) Mean and Standard Deviation Technique

(b) Median and Median Absolute Deviation Technique

Figure 45: 5-Day Rolling Mean, Rolling Mean ±1.5 × Rolling Standard Deviations (a) and 5-Day Rolling Median, Rolling Median ±1.5 × Rolling Median Absolute Deviations (b) for Apple from 16/03/2018 to 11/03/2019

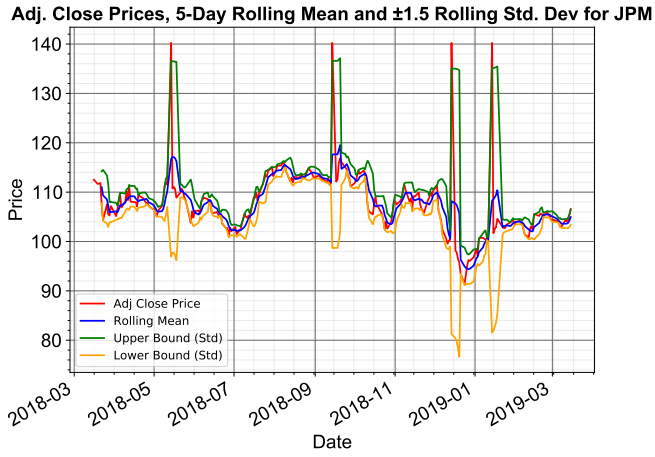


(a) Mean and Standard Deviation Technique

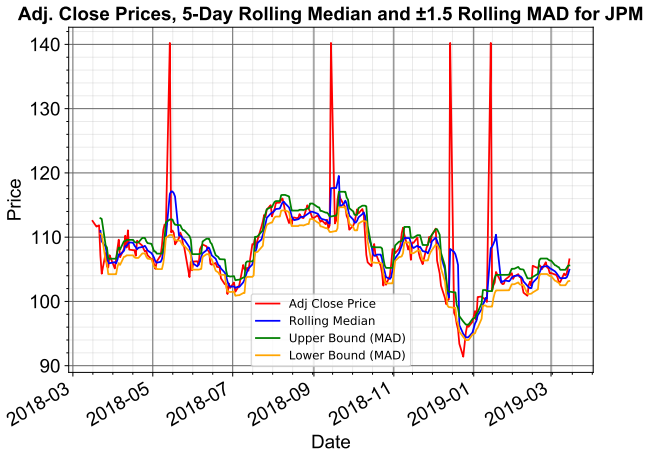


(b) Median and Median Absolute Deviation Technique

Figure 46: 5-Day Rolling Mean, Rolling Mean ±1.5 × Rolling Standard Deviations (a) and 5-Day Rolling Median, Rolling Median ±1.5 × Rolling Median Absolute Deviations (b) for IBM from 16/03/2018 to 11/03/2019

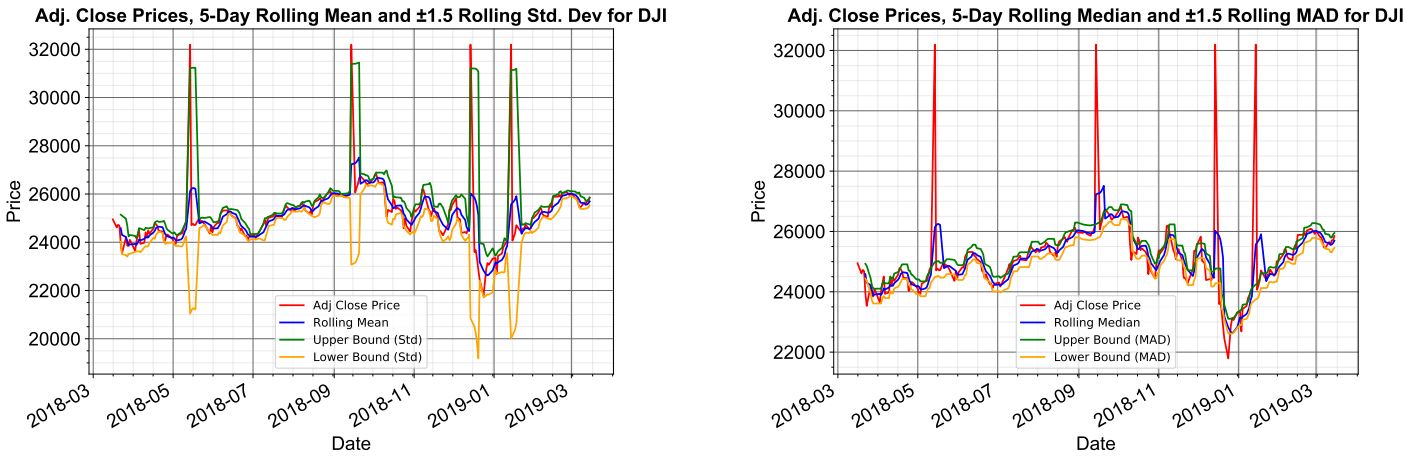


(a) Mean and Standard Deviation Technique



(b) Median and Median Absolute Deviation Technique

Figure 47: 5-Day Rolling Mean, Rolling Mean ±1.5 × Rolling Standard Deviations (a) and 5-Day Rolling Median, Rolling Median ±1.5 × Rolling Median Absolute Deviations (b) for J.P. Morgan from 16/03/2018 to 11/03/2019



(a) Mean and Standard Deviation Technique

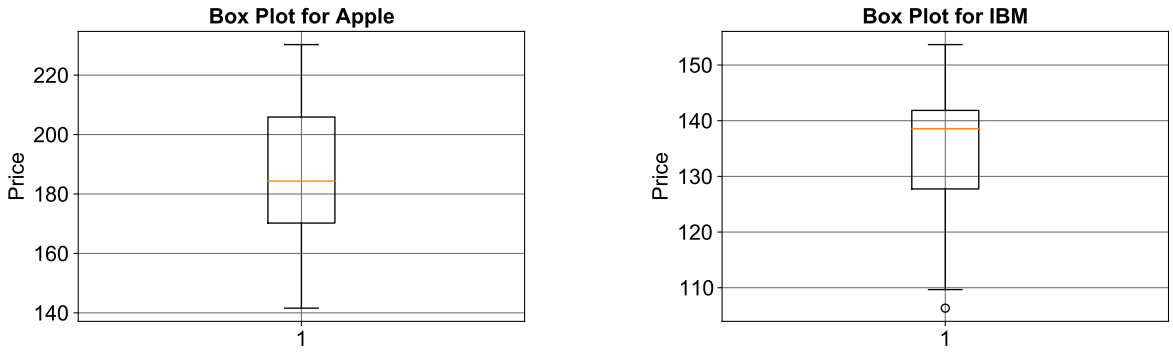
(b) Median and Median Absolute Deviation Technique

Figure 48: 5-Day Rolling Mean, Rolling Mean ±1.5 × Rolling Standard Deviations (a) and 5-Day Rolling Median, Rolling Median ±1.5 × Rolling Median Absolute Deviations (b) for the Dow Jones Index from 16/03/2018 to 11/03/2019

In this section, outlier points are introduced on 4 dates (2018-05-14, 2018-09-14, 2018-12-14, 2019-01-14) with a value equal to 1.2 times the maximum value of the column. From Fig. 45, Fig. 46, Fig. 47 and Fig. 48, the upper and lower bound plots for the Mean and Standard Deviation Technique (Technique 1) are highly susceptible to changes within the financial data. Specifically, at the aforementioned dates, the upper and lower bound plots for Technique 1 are affected for all the entities and significant spikes are noticed.

On the other hand, the upper and lower bound plots for the Median and Median Absolute Deviation Technique (Technique 2) are not highly susceptible to changes within the financial data. In fact, at the outliers on the given dates, there are no spikes observed in the upper and lower bound plots for all of the entities. The upper and lower bound plots for Technique 2 for all of the entities preserve their shape, in response to the outliers. These upper and lower bound plots for Technique 2 are very similar to the ones shown in Section 4.1.3 for all entities. Hence, Technique 2 is more robust to outliers.

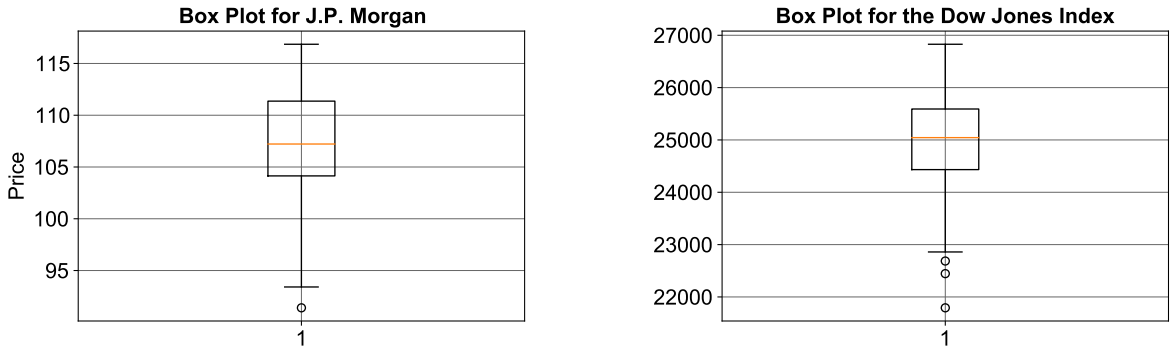
4.1.5



(a) Apple

(b) IBM

Figure 49: Box and Whisker Plots of the Adjusted Close Prices of Apple (a) and IBM (b)



(a) J.P. Morgan **(b)** The Dow Jones Index

Figure 50: Box and Whisker Plots of the Adjusted Close Prices of J.P. Morgan (a) and The Dow Jones Index (b)

Box and Whisker plots are convenient ways of visually displaying the data distribution through their quartiles. The horizontal line inside the box indicates the median of the dataset. The length (vertical height) of the box indicates the interquartile range (IQR). In this interquartile range, 50% of the data is found. This box ranges from Q1, where 25% of the data is found, to Q3, where 75% of the data is found.

The lines extending from the boxes are called the 'whiskers', which indicate the variability outside the upper and lower quartiles. The upper end of the whisker (maximum) is given by $Q3 + 1.5(IQR)$ whereas the lower end of the whisker (minimum) is given by $Q1 - 1.5(IQR)$. Finally, outliers are plotted as individual dots that are in line with whiskers. The anatomy of the Box and Whisker Plot is summarized in Fig. 51.

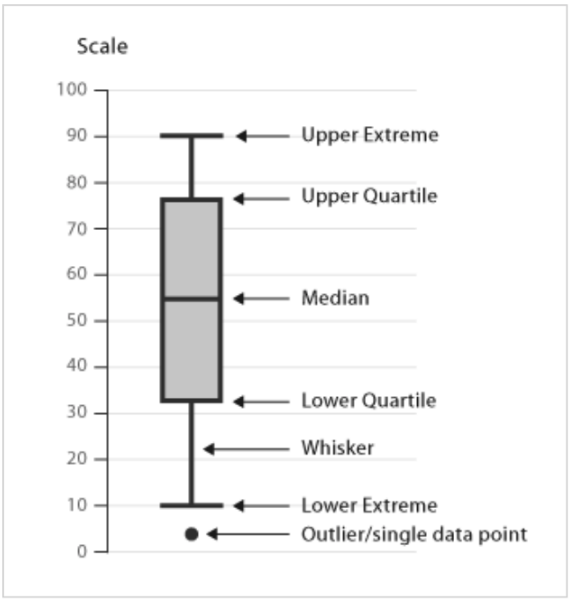


Figure 51: Box and Whisker Plot Anatomy

The Box and Whisker Plots of the Adjusted Close Prices of Apple, IBM, J.P. Morgan and The Dow Jones Index are shown in Fig. 49 and Fig. 50. These plots can also be used to determine the outliers. From Fig. 49, while the Apple's Adjusted Close Price has no outliers, it has a high variability, as seen by the vertical height of the box (high IQR). Apple's adjusted close prices have signficantly changed from 16/03/2018 to 11/03/2019 and this is indicated by the large height of the IQR box. Moreover, the median of the adjusted close price of Apple is closer to the minimum than the maximum.

On the other hand, IBM's Adjusted Close Price has one outlier but has low variability. It's median is very close to the maximum but it's bottom whisker is much larger than the top whisker. This can be explained by the fact that IBM's adjusted close prices were stable at high prices from 03/2018 to 10/2018. However, there was a significant dip in the prices for IBM between 10/2018 to 11/2018. Hence, the distribution is not symmetric and therefore, the price distribution will not be Gaussian.

Similarly, from Fig. 50, the J.P. Morgan's Adjusted Close Price has low variability but it has one outlier, close to the lower end of the whisker (minimum). At the same time, the bottom whisker is much larger than the upper whisker.

In contrast, from Fig. 48, the Dow Jones Index has high variability and multiple outliers, close to the minimum. However, the median is very close to Q2, indicating the symmetric nature of the distribution. Hence, the Dow Jones Index's adjusted close price will be normally distributed, as also shown in Fig. 36.

4.2 Robust Estimators

4.2.1

```
1
2  def med(df):
3      df_sorted = df.sort_values()
4      middle_index = round(len(df_sorted) / 2)
5      return df_sorted[middle_index]
6
7  def interquartile_range(df):
8      df_sorted = df.sort_values()
9      quarter_index = round(len(df_sorted) / 4)
10     three_quarter_index = round(3 * len(df_sorted) / 4)
11     Q25 = df_sorted[quarter_index]
12     Q75 = df_sorted[three_quarter_index]
13     return Q75 - Q25
14
15 def median_absolute_deviation(df):
16     initial_median = med(df)
17     absolute_deviations = abs(df - initial_median)
18     return med(absolute_deviations)
```

Listing 3: Python functions for Median, Interquartile Range (IQR) and Median Absolute Deviation (MAD)

Listing 3 shows the Python functions for Median (`med`), Interquartile Range (`interquartile_range`) and Median Absolute Deviation (`median_absolute_deviation`). Each of these functions take a **`pandas.Series`** as an input and return the estimator value as an output.

4.2.2

Computational Efficiency of the Median Function (`med`):

The `pandas.Series` is sorted in ascending order using the `sort_values` method. This method, by default, uses the Quick Sort algorithm which is a type of a divide and conquer algorithm. The time complexity of the quick sort algorithm, where n is the length of the `pandas.Series`, is given as follows:

- **Best-Case Time Complexity:** $O(n\log(n))$
- **Worst-Case Time Complexity:** $O(n^2)$
- **Average Time Complexity:** $O(n\log(n))$

The time complexity of the `len` function in Python is $O(1)$. Additionally, the time complexity to access the middle value of the sorted `pandas.Series` is $O(1)$. Therefore, the total time complexity of the Median function is $O(n\log(n)+1+1) \approx O(n\log(n))$. Finally, the space complexities of storing n variables is $O(n)$ and that of the quick sort algorithm is $O(log(n))$. Therefore, the total space complexity of the function is $O(n+log(n)) \approx O(n)$

Computational Efficiency of the Interquartile Range Function (`interquartile_range`):

The `interquartile_range` function also sorts the `pandas.Series` at the beginning using the quick sort algorithm, which has an average time complexity of $O(n\log(n))$. Additionally, the Python `len` function is used twice and the sorted `pandas.Series` is accessed twice to determine the upper and lower quartile. Therefore,

the total time complexity is $O(n\log(n)+1+1+1+1)$ or $O(n\log(n))$. The space complexity of the `interquartile_range` function is $O(log(n)+n) \approx O(n)$ Therefore, the `interquartile_range` function is slightly more expensive to compute than the `med` function.

Computational Efficiency of the Median Absolute Deviation Function (`median_absolute_deviation`):

In the `median_absolute_deviation` function, the `med` function to calculate the Median is called twice. Additionally, the absolute deviations computed have a complexity of $O(n)$ Therefore, the total time complexity of the operation is $O(2n\log(n)+4+n) \approx O(2n\log(n)+n) \approx O(n\log(n))$. Finally, the space complexity of this function can also be approximated as $O(n)$. Therefore, all the functions in Listing 3 have the same time and space complexities, for large n.

4.2.3

The breakdown point is a measure of the estimator's robustness, wherein it gives the proportion of incorrect observations an estimator can handle before giving an incorrect result. Additionally, the breakdown point or the asymptotic breakdown point is the finite sample breakdown point where the number of samples, n, goes to ∞. For instance, the sample mean has a finite sample breakdown point of $1/n$. Therefore, its breakdown point is 0, which indicates that the sample mean is not a robust estimator.

The concept of breakdown point can be formulated more mathematically. For a sample of size, n, and each $j = 1, \ldots, n$, the order statistic $T = X_j$ has a breakdown point of $\varepsilon^*(T) = \frac{1}{n}\min(j-1, n-j)$ [1].

The higher the breakdown point of an estimator, the more robust the estimator is. An estimator with a high breakdown point is called a resistant statistic. The asymptotic breakdown point or the breakdown point of the estimators are given as follows:

1. **Median:** If there are odd number of samples, n, such that $n = 2k+1$ for an integer k, then the sample median X_{k+1} has a finite breakdown point which can be calculated as shown in Eq. (85)

$$\text{Finite Breakdown Point} = \frac{1}{2k+1}\min(k+1-1, 2k+1-k-1) = \frac{k}{2k+1} \tag{85}$$

   Substituting for k as $k = \frac{n-1}{2}$, gives the following result in Eq. (86).

$$\text{Finite Breakdown Point} = \frac{\frac{n-1}{2}}{n-1+1} = \frac{1}{2} - \frac{1}{2n} \tag{86}$$

   Therefore, the asymptotic breakdown point or just the breakdown point can be calculated by taking $n \to \infty$ in Eq. (86). Therefore, the breakdown point for $n = 2k+1$ for the sample median is calculated as shown in Eq. (87).

$$\text{Asymptotic Breakdown Point} = \text{Breakdown Point} = \frac{1}{2} \tag{87}$$

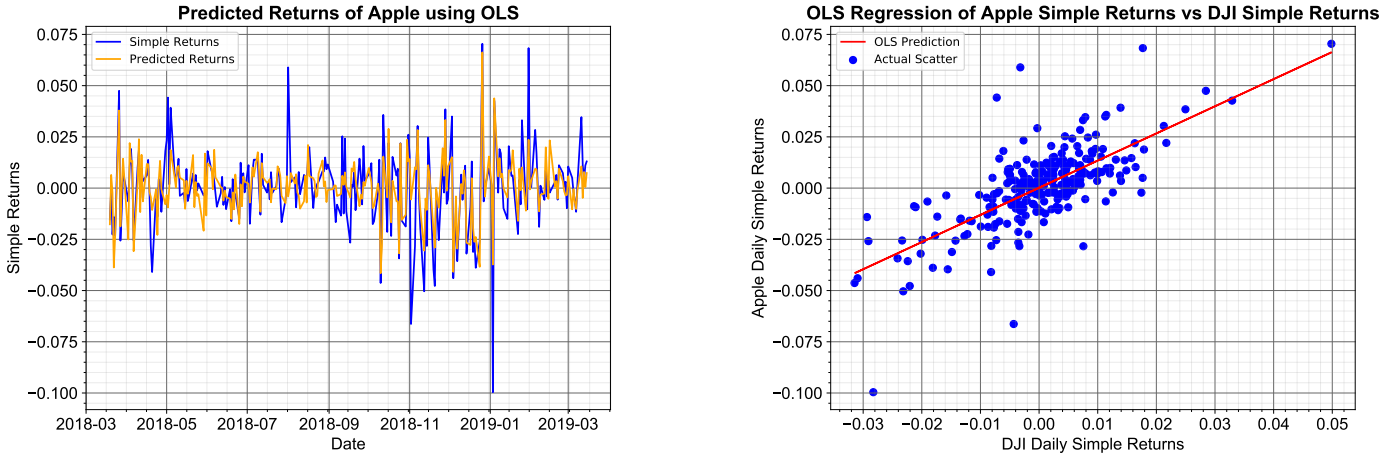
   If the number of samples is even, i.e. $n = 2k$, for an integer k, then the two endpoints of the interval of medians, X_k and X_{k+1}, each have a breakdown point of $\frac{1}{2} - \frac{1}{n}$. Therefore, any sample median has an asymptotic breakdown point or a breakdown point of $\frac{1}{2}$ [1].

2. **Interquartile Range (IQR):** The Interquartile Range estimator has a breakdown point of 0.25 [1]. Therefore, it is not as robust as the median estimator.

3. **Median Absolute Deviation (MAD):** This estimator has a breakdown point of 0.5. Since the Median operation is applied twice to estimate MAD, the breakdown point of MAD is limited by the breakdown point of the median. The median is computed and subtracted from each value of the dataset. The median of the absolute value of the aforementioned result is computed again.

4.3 Robust and OLS Regression

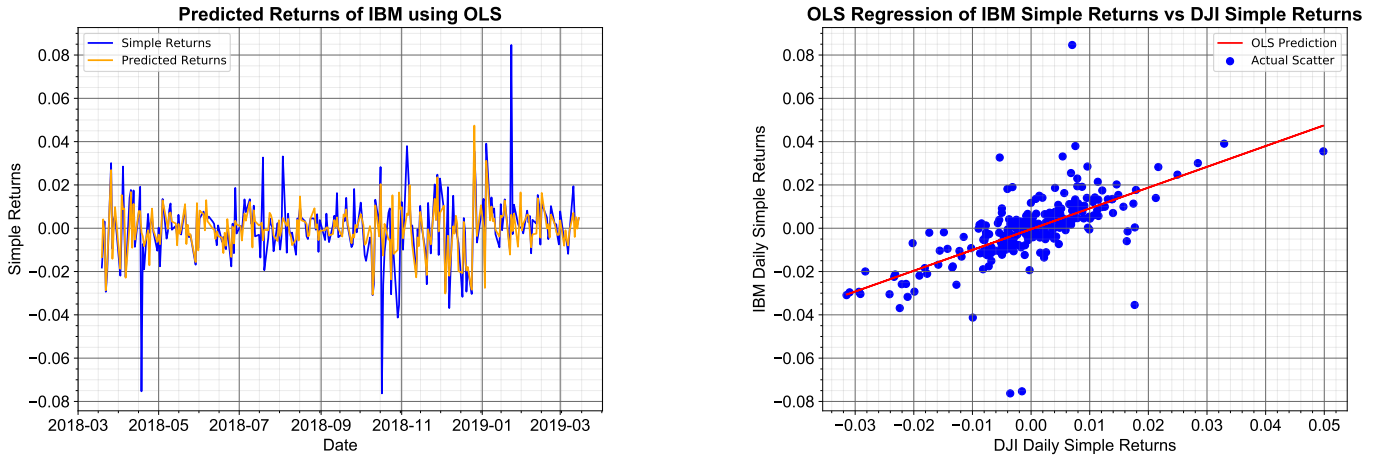
4.3.1



(a) Actual Returns and Predicted Returns

(b) Actual Scatter and OLS Regression

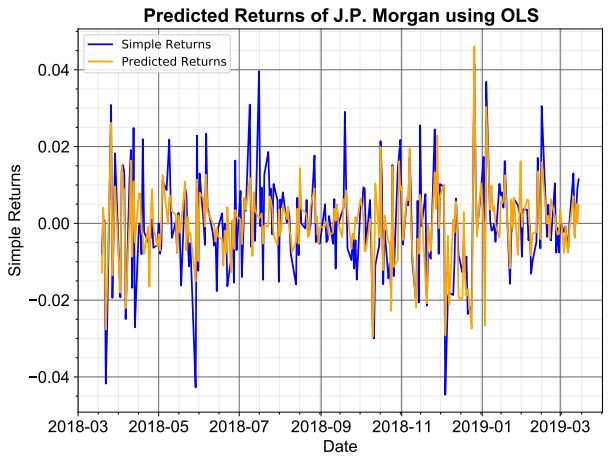
Figure 52: Actual Daily Simple Returns vs Predicted Daily Simple Returns of Apple (a) and Actual Scatter and OLS Regression of Daily Simple Returns of Apple vs Daily Simple Returns of DJI (b)



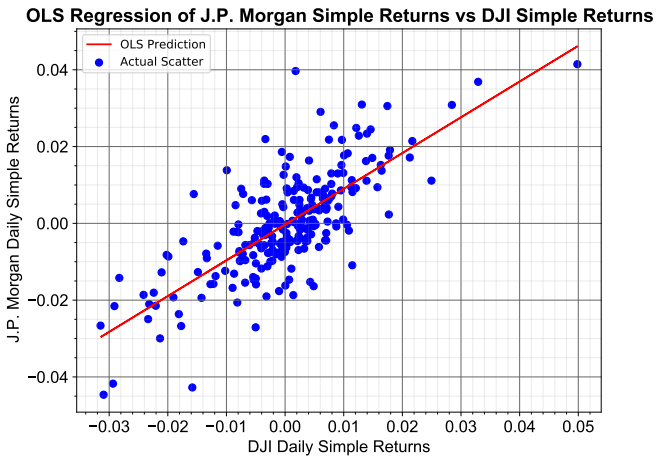
(a) Actual Returns and Predicted Returns

(b) Actual Scatter and OLS Regression

Figure 53: Actual Daily Simple Returns vs Predicted Daily Simple Returns of IBM (a) and Actual Scatter and OLS Regression of Daily Simple Returns of IBM vs Daily Simple Returns of DJI (b)



(a) Actual Returns and Predicted Returns



(b) Actual Scatter and OLS Regression

Figure 54: Actual Daily Simple Returns vs Predicted Daily Simple Returns of J.P. Morgan (a) and Actual Scatter and OLS Regression of Daily Simple Returns of J.P. Morgan vs Daily Simple Returns of DJI (b)

Linear regression in one variable, is a technique to determine a linear relationship between input x and output y. Given that there are m training examples, the outputs y for each training example can be stacked into a column vector to give Y with dimensions $Y \in \mathbb{R}^{(m,1)}$. Similarly, a matrix X can be formed by vertically stacking the inputs x, along with the first column as ones which gives $X \in \mathbb{R}^{(m,2)}$. Therefore, the feature matrix, $\theta \in \mathbb{R}^{(2,1)}$, of the linear regression is given by Eq. (88).

$$\theta = \left(X^T X\right)^{-1} X^T Y \tag{88}$$

Another way to mathematically formulate the Linear Regression or the Ordinary Least Squares (OLS) problem is to minimize the squared error of the residual, ϵ. The regression problem can be formulated as shown in Eq. (89).

$$Y = X\theta + \epsilon \tag{89}$$

Therefore, using Eq. (89), the OLS problem is to minimize the squared error of the residual which is given by Eq. (90).

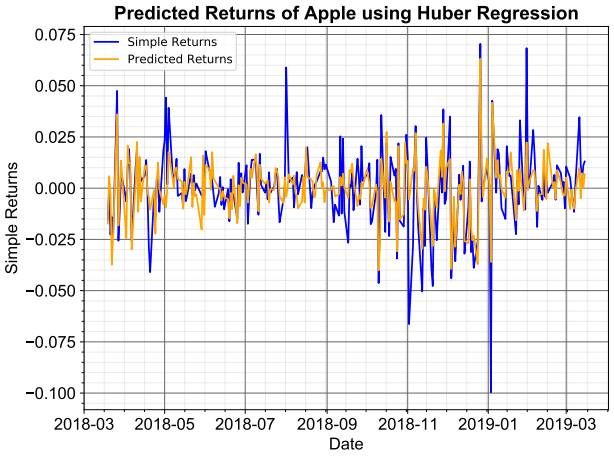
$$||\epsilon||^2 = ||Y - X\theta||^2 \tag{90}$$

Linear Regression or Ordinary Least Squares (OLS) regression method is used to regress each stock's daily simple returns against the Dow Jones' daily simple returns. The predicted daily simple returns along with OLS regression for Apple, IBM and J.P. Morgan are shown in Fig. 52, Fig. 53 and Fig. 54, respectively. The intercepts and the coefficients of the linear regression (OLS) performed for the daily returns of each stock against the daily returns of the Dow Jones Index is shown in Table 13.

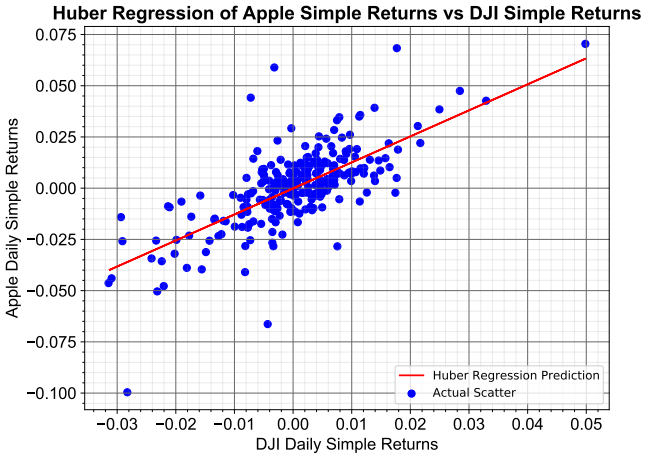
	Apple	**IBM**	**J.P. Morgan**
Coefficient	1.3255798	0.9600925	0.9314082
Intercept	0.0001647	-0.0004406	-0.0003163

Table 13: Coefficients and Intercepts generated by regressing the daily simple returns of each stock (Apple, IBM, J.P.Morgan) individually against the daily simple returns of the Dow Jones Index using OLS regression

4.3.2

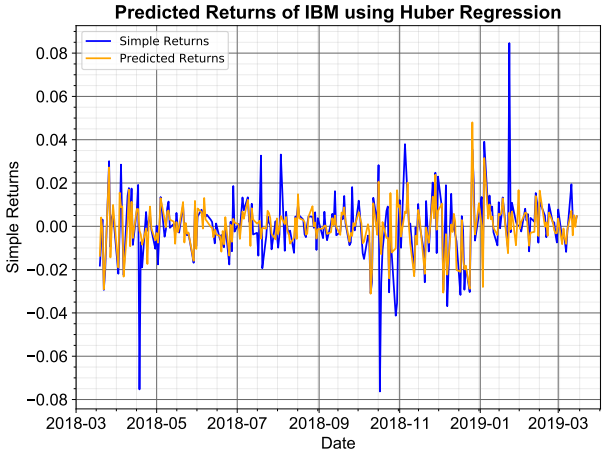


(a) Actual Returns and Predicted Returns

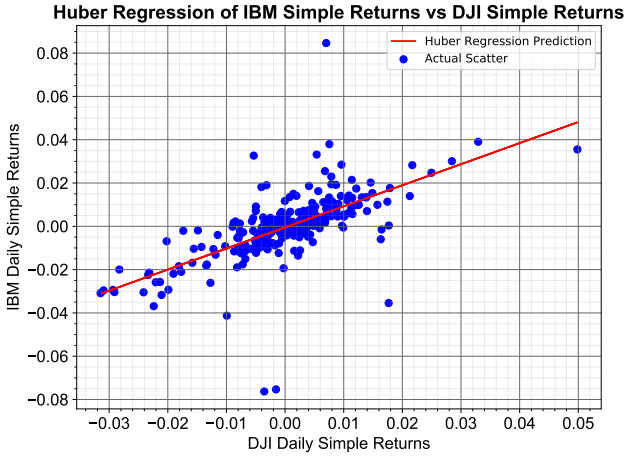


(b) Actual Scatter and Huber Regression

Figure 55: Actual Daily Simple Returns vs Predicted Daily Simple Returns of Apple (a) and Actual Scatter and Huber Regression of Daily Simple Returns of Apple vs Daily Simple Returns of DJI (b)

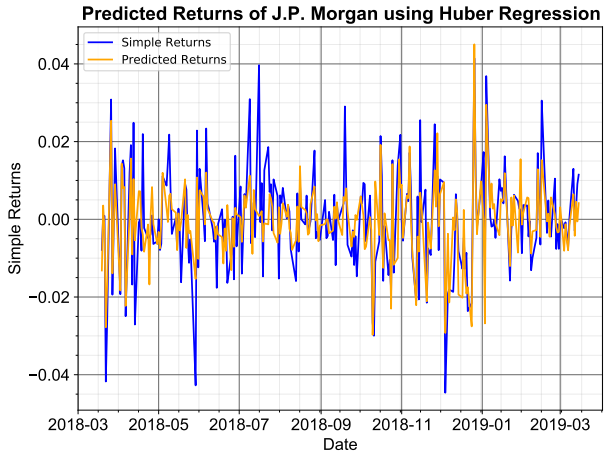


(a) Actual Returns and Predicted Returns

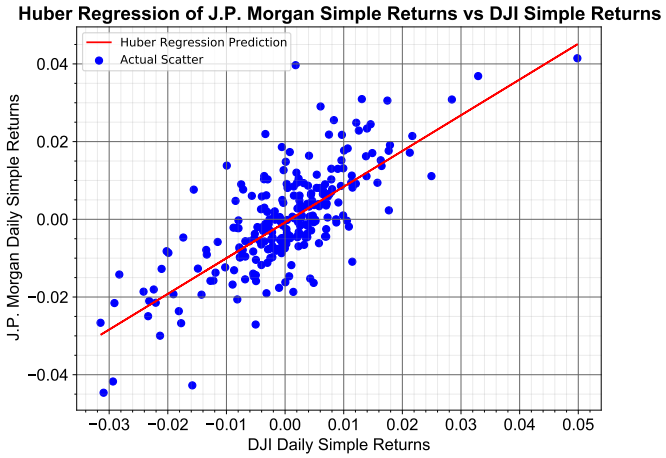


(b) Actual Scatter and Huber Regression

Figure 56: Actual Daily Simple Returns vs Predicted Daily Simple Returns of IBM (a) and Actual Scatter and Huber Regression of Daily Simple Returns of IBM vs Daily Simple Returns of DJI (b)



(a) Actual Returns and Predicted Returns



(b) Actual Scatter and Huber Regression

Figure 57: Actual Daily Simple Returns vs Predicted Daily Simple Returns of J.P. Morgan (a) and Actual Scatter and Huber Regression of Daily Simple Returns of J.P. Morgan vs Daily Simple Returns of DJI (b)

A type of regression technique that is robust to outliers is Huber Regression. Using this regression technique, the squared loss for samples is optimized where $\frac{|Y-X\theta|}{\sigma} < \epsilon$ and the absolute loss for the samples where $\frac{|Y-X\theta|}{\sigma} > \epsilon$. θ and σ are parameters that need to be optimized. The default value of epsilon is used which is 1.35.

The Huber Regressor is identical to the OLS squared error penalty but for large residuals, $\epsilon > 1.35$, the penalty is lower and increases linearly instead of quadratically. Hence, the regressor is more robust against the impact of outliers. The Huber Regressor concept can be formulated mathematically as shown in Eq. (91)

$$\underset{\beta}{\text{minimize}} \quad \phi\left(Y - X\theta\right) \tag{91}$$

In Eq. (91), the loss function, ϕ can be written as shown in Eq. (92).

$$\phi(u) = \begin{cases} u^2 & \text{if } |u| \leq \epsilon \\ 2\epsilon u - \epsilon^2 & \text{if } |u| > \epsilon \end{cases} \tag{92}$$

The Huber Regressor method is used to regress each stock's daily simple returns against the Dow Jones' daily simple returns. The predicted daily simple returns along with the Huber Regression for Apple, IBM and J.P. Morgan are shown in Fig. 55, Fig. 56 and Fig. 57, respectively. Finally, the intercepts and the coefficients of the Huber Regression performed for the daily returns of each stock against the daily returns of the Dow Jones Index is shown in Table 14.

	Apple	**IBM**	**J.P. Morgan**
Coefficient	1.2702124	0.9735621	0.9196621
Intercept	-0.0001304	-0.0005094	-0.0008009

Table 14: Coefficients and Intercepts generated by regressing the daily simple returns each stock (Apple, IBM, J.P.Morgan) individually against the daily simple returns of the Dow Jones Index using the Huber regression

4.3.3

Comparing the Performance:

	OLS Regression R^2 Score	**Huber Regression R^2 Score**
Apple	0.516518	0.515365
IBM	0.417773	0.417672
J.P. Morgan	0.555864	0.554386

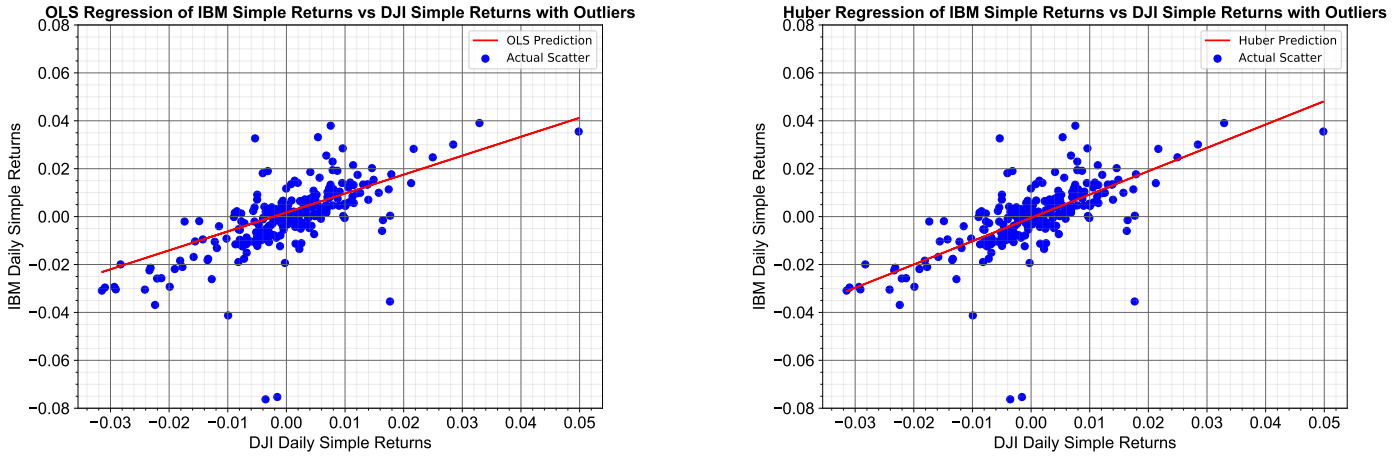
Table 15: R^2 scores of the the OLS Regression and Huber Regression Methods used for daily returns of Apple, IBM and J.P. Morgan against the daily returns of the Dow Jones Index

R^2 or R-Squared is a statistic that provides information on the goodness of fit of a model. Also referred to as the coefficient of determination, it is a measure of how close regression predictions are to the real data points. Using Python, the best possible value of R^2 is 1, which indicates that the regression model fits the data, perfectly. From Table 15, the OLS Regression and Huber Regression have almost the same R^2 scores for all the stocks.

Comparing the Performance with Outliers:

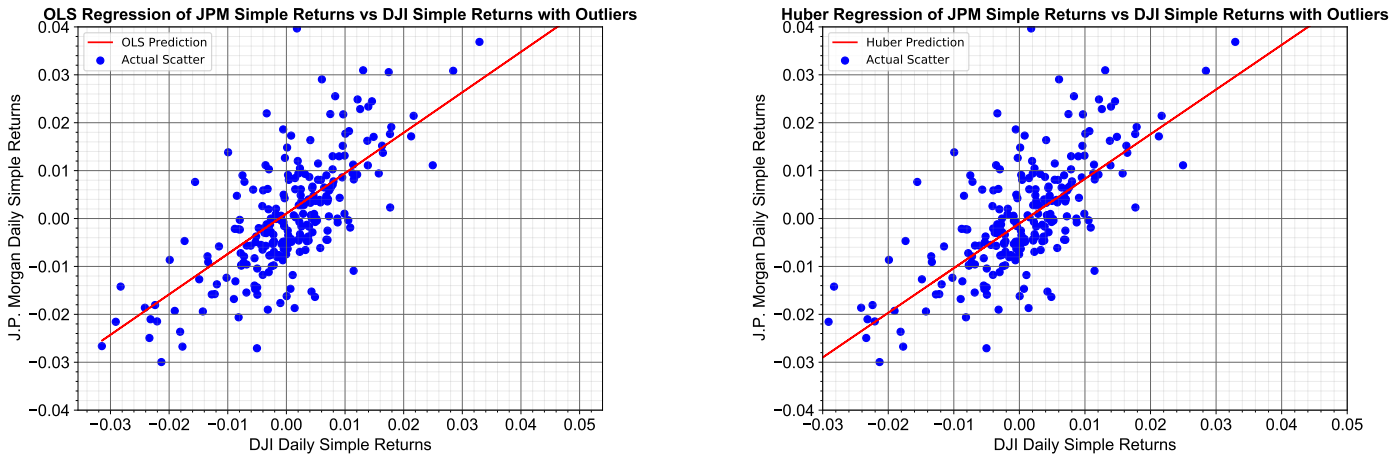
(a) OLS Regression **(b)** Huber Regression

Figure 58: Actual Scatter and OLS (a) and Huber Regression (b) of Daily Simple Returns of Apple vs Daily Simple Returns of DJI with Outliers introduced in the Apple Data



(a) OLS Regression **(b)** Huber Regression

Figure 59: Actual Scatter and OLS (a) and Huber Regression (b) of Daily Simple Returns of IBM vs Daily Simple Returns of DJI with Outliers introduced in the IBM Data



(a) OLS Regression

(b) Huber Regression

Figure 60: Actual Scatter and OLS (a) and Huber Regression (b) of Daily Simple Returns of J.P. Morgan vs Daily Simple Returns of DJI with Outliers introduced in the JPM Data

For this part, the outliers are introduced in the adjusted close prices of Apple, IBM and J.P. Morgan. Specifically, the outliers for each dataset are introduced at the dates, [2018-05-14, 2018-09-14, 2018-12-14, 2019-01-14], with a value 1.2 $\times$ the maximum value of the respective adjusted close price column. Then, the daily simple returns of each stock is calculated.

The daily simple returns of Apple, IBM and J.P. Morgan with outliers is OLS and Huber regressed against the daily simple returns of the Dow Jones Index and the plots are shown in Fig. 58, Fig. 59 and Fig. 60, respectively. Therefore, the corresponding coefficients and intercepts of plots for both regression methods is shown in Table 16.

	OLS Regression			**Huber Regression**		
	Apple	**IBM**	**J.P. Morgan**	**Apple**	**IBM**	**J.P. Morgan**
Coefficient	1.0898122	0.7904433	0.8435895	1.2694706	0.9738519	0.9320168
Intercept	0.0035495	0.0017163	0.0010421	7.5229486e-06	-0.0005072	-0.0010399

Table 16: Coefficients and Intercepts generated by regressing the daily simple returns of each stock (Apple, IBM, J.P. Morgan) individually against the daily simple returns of the Dow Jones Index using OLS and Huber Regression with outliers introduced in the stock data

From Table 16, the coefficients and intercepts of the Huber Regression for each stock with outliers has not changed to a large extent, when compared to Table 14. In contrast, the coeffients and intercepts of the OLS Regression for each stock with outliers has changed to a large extent when compared to Table 13. The percentage change in the coefficients due to the introduction of outliers for each stock and regression method is shown in Table 17.

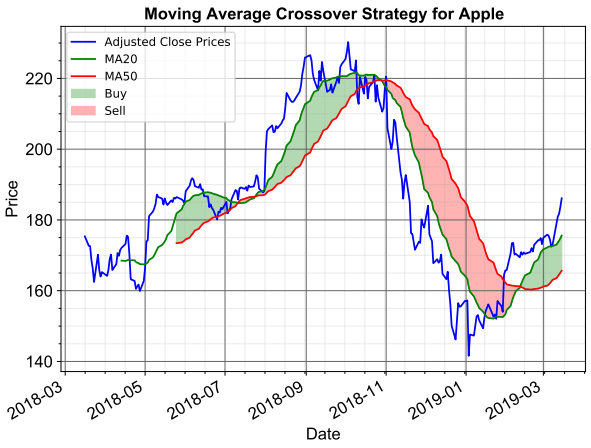
	OLS Regression			**Huber Regression**		
	Apple	**IBM**	**J.P. Morgan**	**Apple**	**IBM**	**J.P. Morgan**
Coefficient	17.6%	17.7%	9.4%	0.058%	0.03%	1.34%

Table 17: Percentage change in the coefficient values for the OLS and Huber Regression Methods when the outliers are introduced in the adjusted close prices

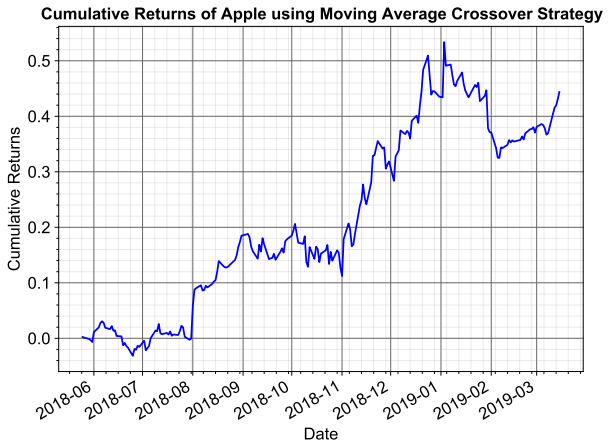
From Table 17, when outliers are introduced, the OLS regression coefficients have a higher percentage change for each stock compared to Huber Regression. Therefore, Huber Regression is more robust to outliers compared to OLS regression. The percentage change in intercepts is not considered since these values for both OLS and Huber Regression are very close to 0.

4.4 Robust Trading Strategies

4.4.1

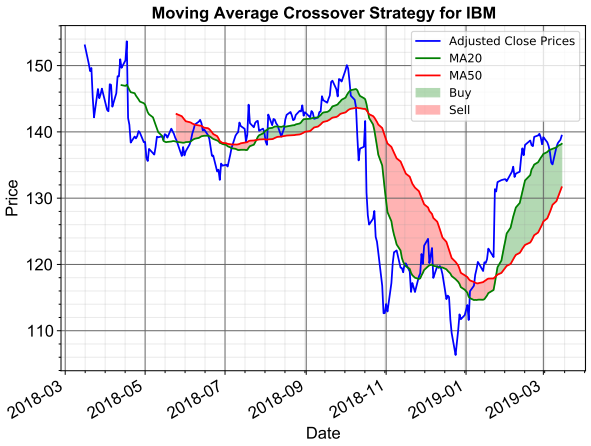


(a) MA Crossover Strategy for Apple

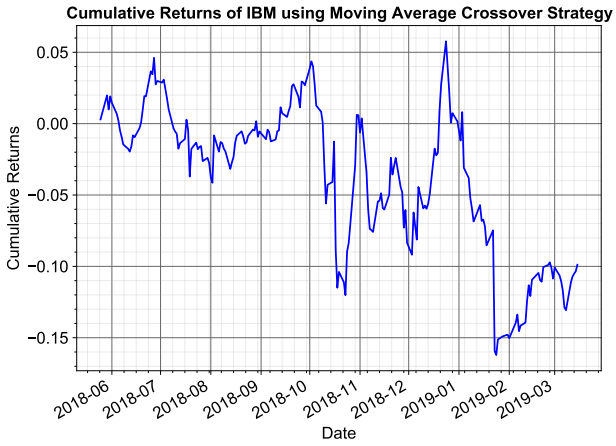


(b) Cumulative Returns

Figure 61: Moving Average Cross Over Strategy implemented on Apple's Adjusted Close Prices (left) and Cumulative Returns of Apple using the strategy (right)

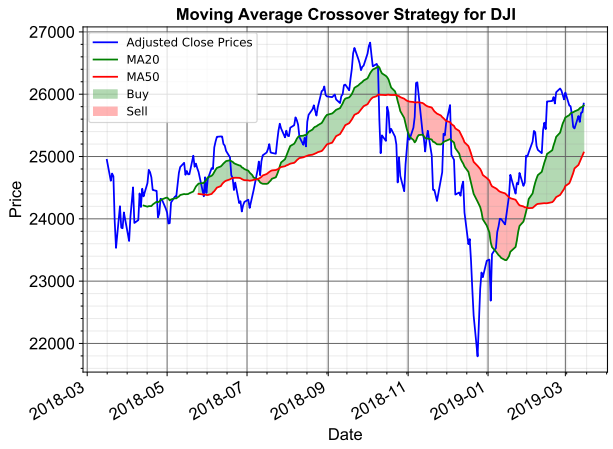


(a) MA Crossover Strategy for IBM

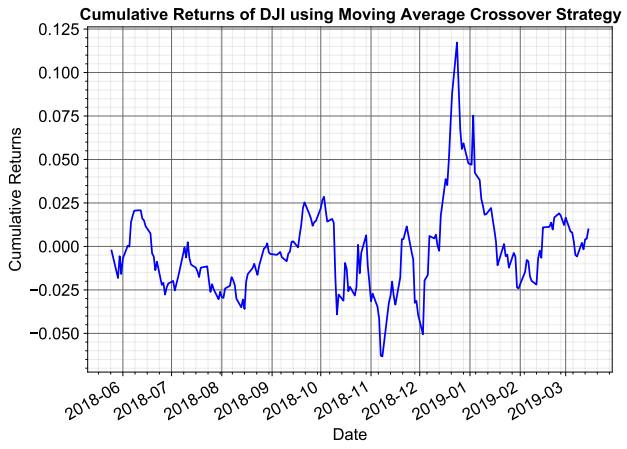


(b) Cumulative Returns

Figure 62: Moving Average Cross Over Strategy implemented on IBM's Adjusted Close Prices (left) and Cumulative Returns of IBM using the strategy (right)

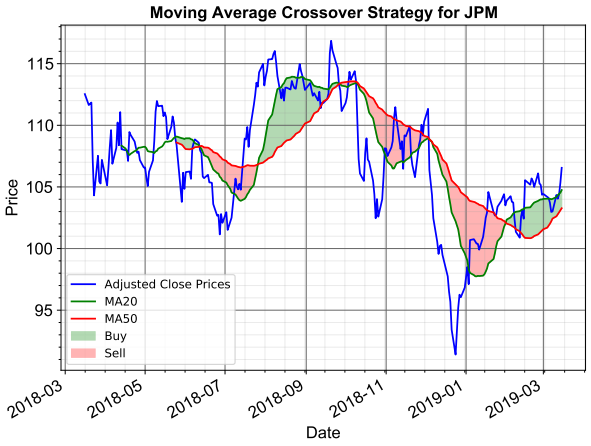


(a) MA Crossover Strategy for DJI

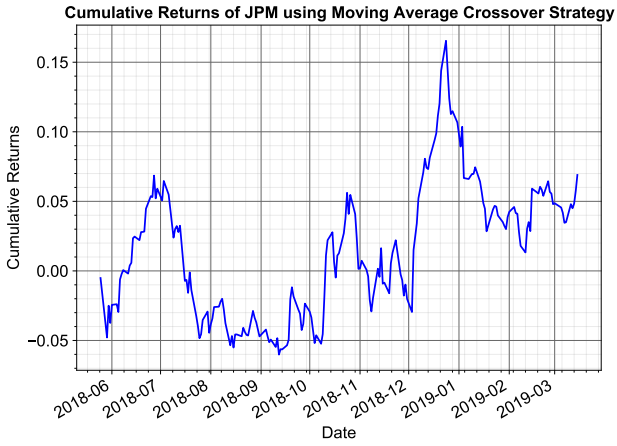


(b) Cumulative Returns

Figure 63: Moving Average Cross Over Strategy implemented on DJI's Adjusted Close Prices (left) and Cumulative Returns of DJI using the strategy (right)



(a) MA Crossover Strategy for J.P. Morgan

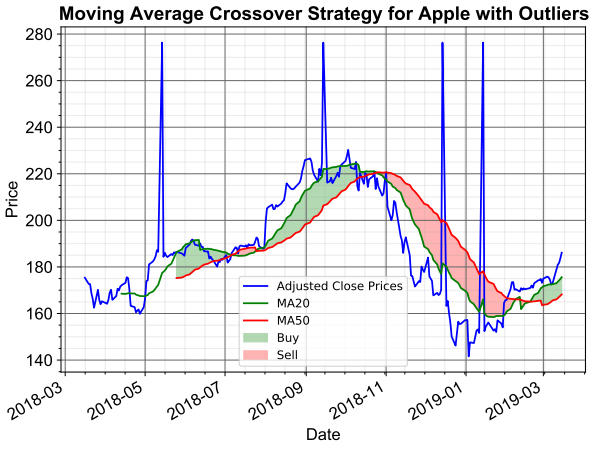


(b) Cumulative Returns

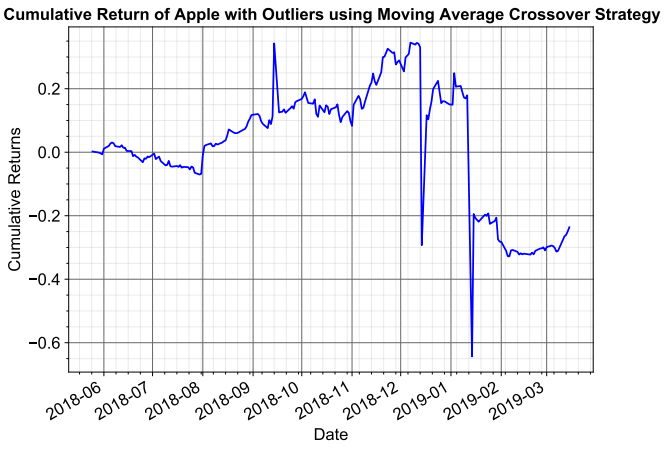
Figure 64: Moving Average Cross Over Strategy implemented on J.P. Morgan's Adjusted Close Prices (left) and Cumulative Returns of J.P. Morgan using the strategy (right)

The Moving Average Crossover Strategy (shown with the corresponding cumulative returns) is implemented for the adjusted close prices of Apple, IBM, the Dow Jones Index and J.P. Morgan in Fig. 61, Fig. 62, Fig. 63 and Fig. 64, respectively. Next, the adjusted close prices are corrupted with outliers. Specifically, the outliers for each stock and the DJI are introduced on 2018-05-14, 2018-09-14, 2018-12-14 and 2019-01-14 with a value equal to 1.2 $\times$ the maximum value of the corresponding column.

The Moving Average Crossover Strategy (shown with the corresponding cumulative returns) is implemented for adjusted close prices with outliers for Apple, IBM, the Dow Jones Index and J.P. Morgan in Fig. 65, Fig. 66, Fig. 67 and Fig. 68, respectively.

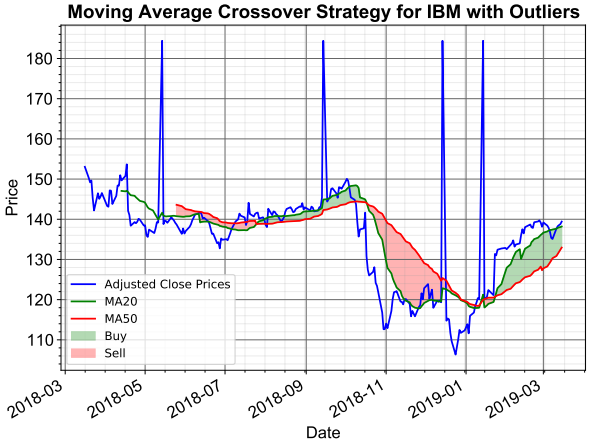


(a) MA Crossover Strategy for Apple with Outliers

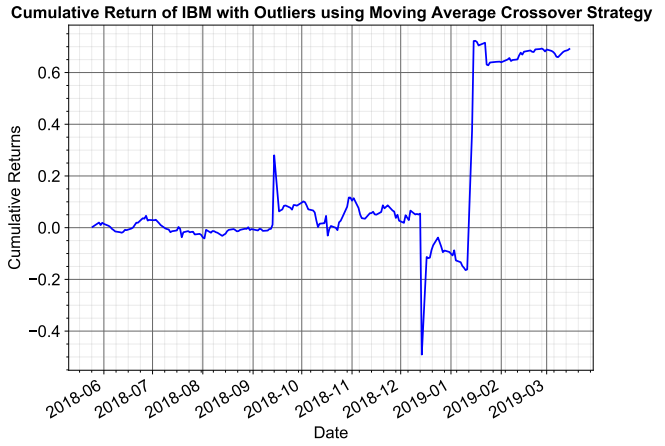


(b) Cumulative Returns with Outliers

Figure 65: Moving Average Cross Over Strategy implemented on Apple's Adjusted Close Prices with outliers (left) and Cumulative Returns of Apple using the strategy (right)

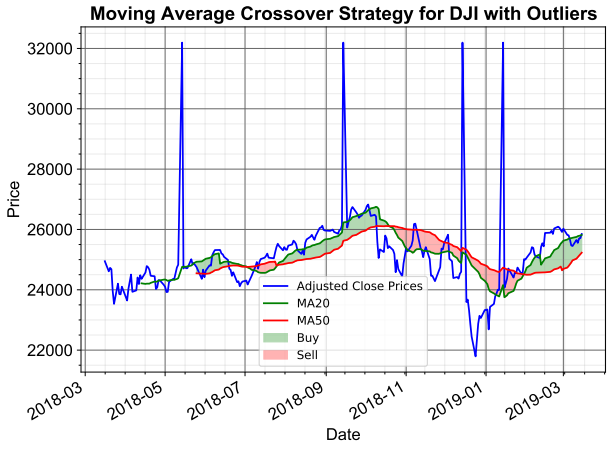


(a) MA Crossover Strategy for IBM with Outliers

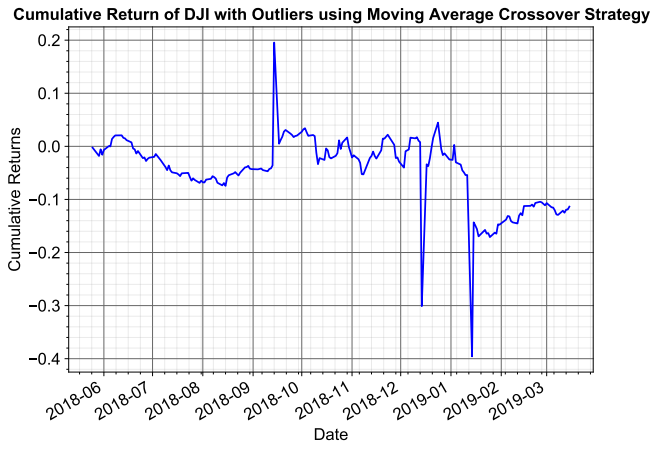


(b) Cumulative Returns with Outliers

Figure 66: Moving Average Cross Over Strategy implemented on IBM's Adjusted Close Prices with outliers (left) and Cumulative Returns of IBM using the strategy (right)

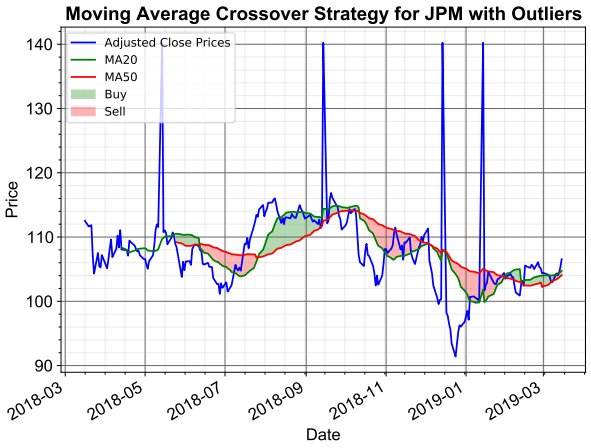


(a) MA Crossover Strategy for DJI with Outliers

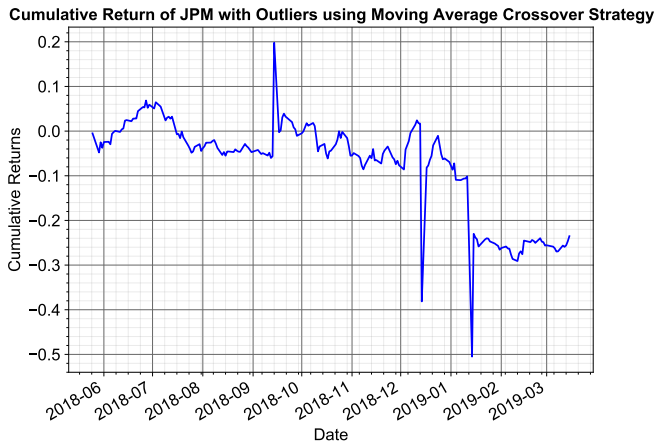


(b) Cumulative Returns with Outliers

Figure 67: Moving Average Cross Over Strategy implemented on DJI's Adjusted Close Prices with outliers (left) and Cumulative Returns of DJI using the strategy (right)



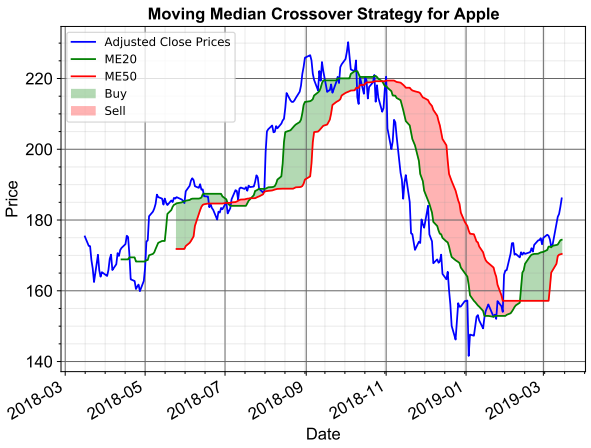
(a) MA Crossover Strategy for J.P. Morgan with Outliers



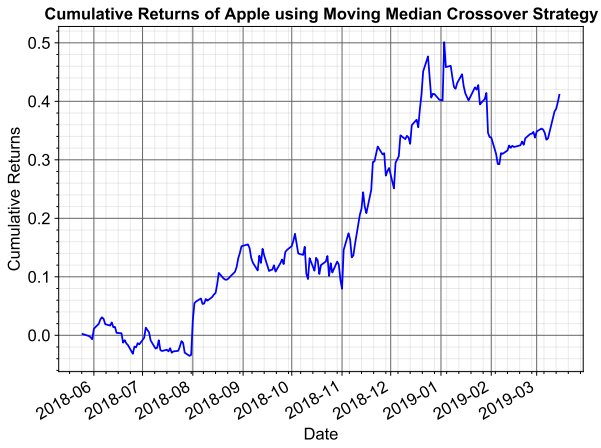
(b) Cumulative Returns with Outliers

Figure 68: Moving Average Cross Over Strategy implemented on J.P. Morgan's Adjusted Close Prices with outliers (left) and Cumulative Returns of J.P. Morgan using the strategy (right)

4.4.2

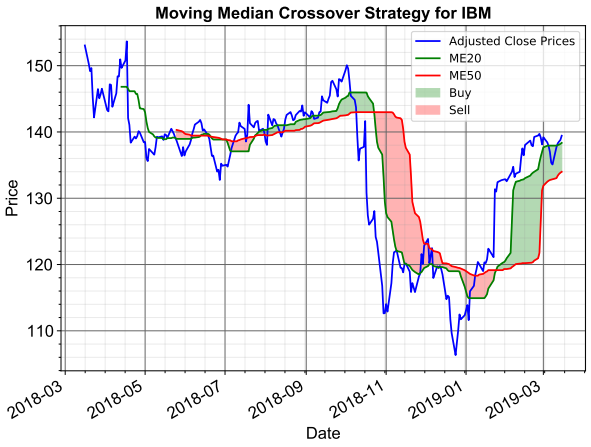


(a) MA Crossover Strategy for Apple

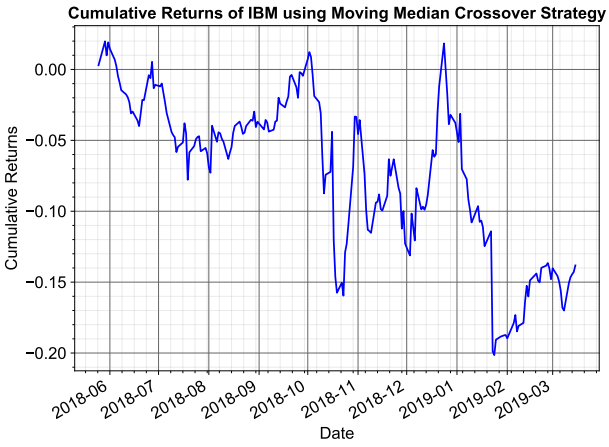


(b) Cumulative Returns

Figure 69: Moving Median Cross Over Strategy implemented on Apple's Adjusted Close Prices (left) and Cumulative Returns of Apple using the strategy (right)

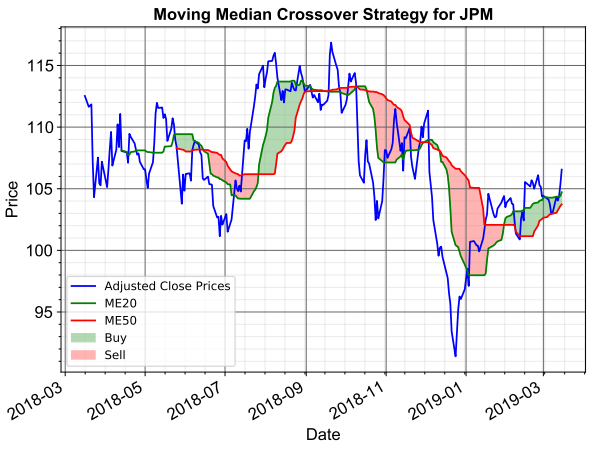


(a) MA Crossover Strategy for IBM

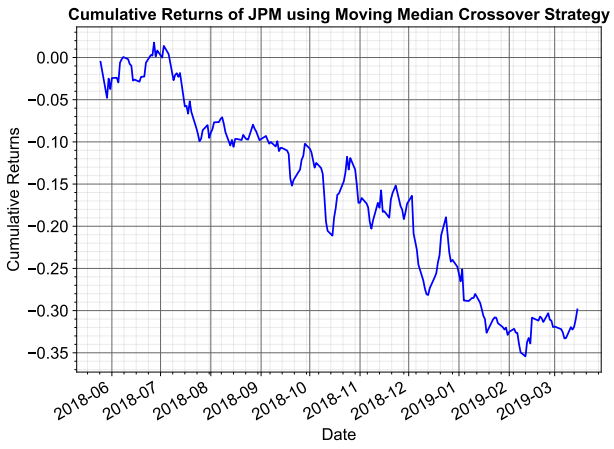


(b) Cumulative Returns

Figure 70: Moving Median Cross Over Strategy implemented on IBM's Adjusted Close Prices (left) and Cumulative Returns of IBM using the strategy (right)

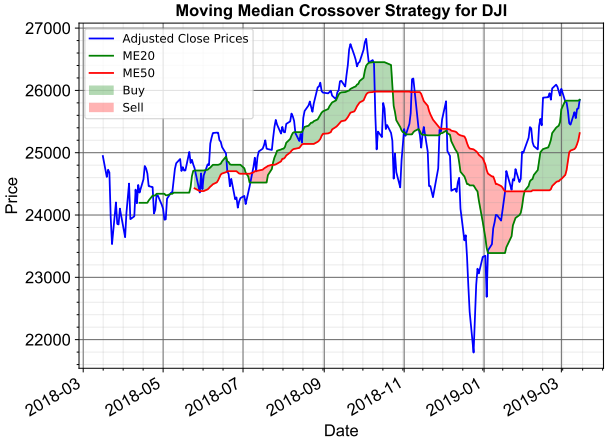


(a) MA Crossover Strategy for DJI

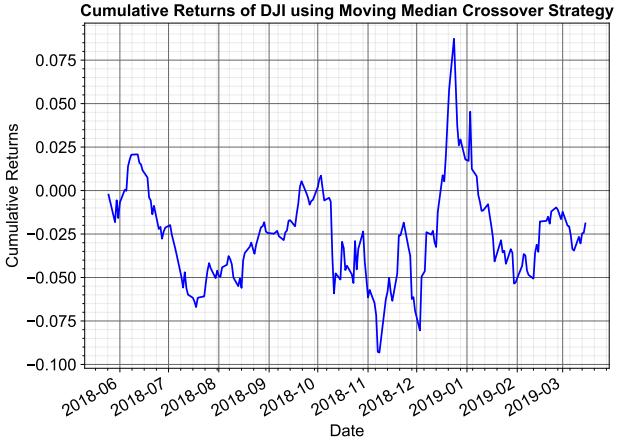


(b) Cumulative Returns

Figure 71: Moving Median Cross Over Strategy implemented on DJI's Adjusted Close Prices (left) and Cumulative Returns of DJI using the strategy (right)



(a) MA Crossover Strategy for J.P. Morgan

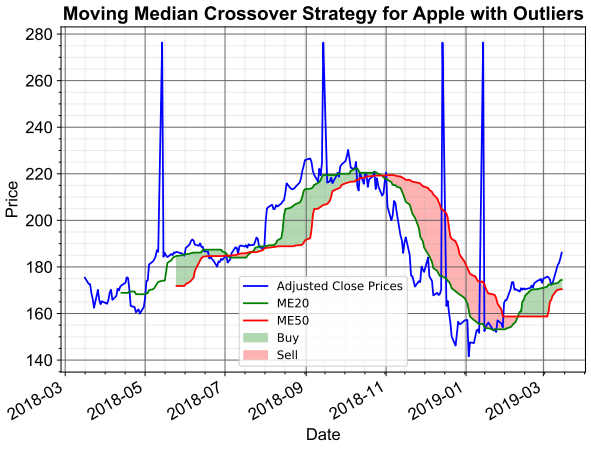


(b) Cumulative Returns

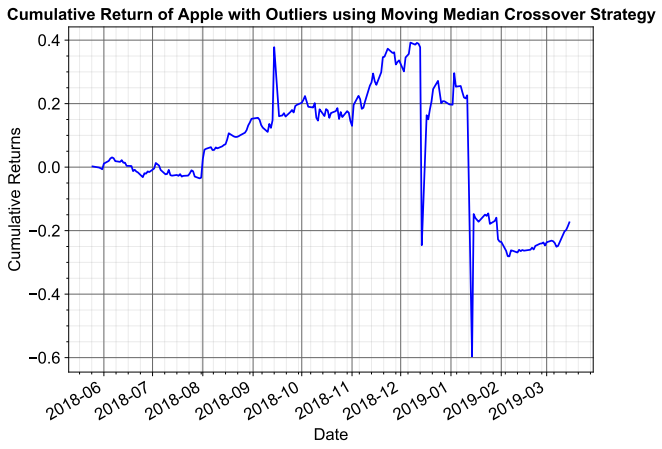
Figure 72: Moving Median Cross Over Strategy implemented on J.P. Morgan's Adjusted Close Prices (left) and Cumulative Returns of J.P. Morgan using the strategy (right)

A Robust version of the Moving Average Crossover Strategy is the Moving Median Crossover strategy which uses the rolling median. The Moving Median Average Crossover Strategy (shown with the corresponding cumulative returns) is implemented for the adjusted close prices of Apple, IBM, the Dow Jones Index and J.P. Morgan in Fig. 69, Fig. 70, Fig. 71 and Fig. 72, respectively. Next, the adjusted close prices are corrupted with outliers. Specifically, the outliers for each stock and the DJI are introduced on 2018-05-14, 2018-09-14, 2018-12-14 and 2019-01-14 with a value equal to 1.2 $\times$ the maximum value of the corresponding column.

The Moving Median Crossover Strategy (shown with the corresponding cumulative returns) is implemented for adjusted close prices with outliers for Apple, IBM, the Dow Jones Index and J.P. Morgan in Fig. 73, Fig. 74, Fig. 75 and Fig. 76, respectively.

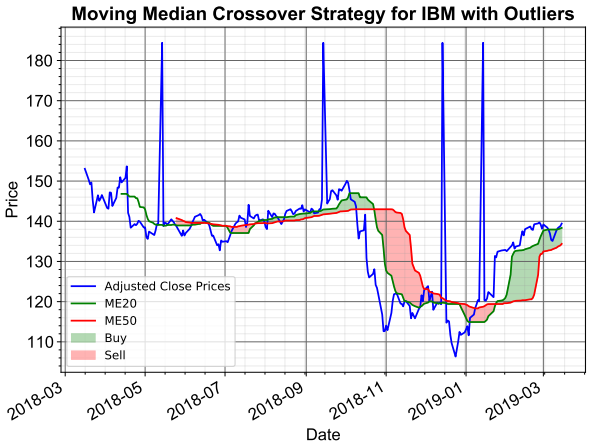


(a) MA Crossover Strategy for Apple with Outliers

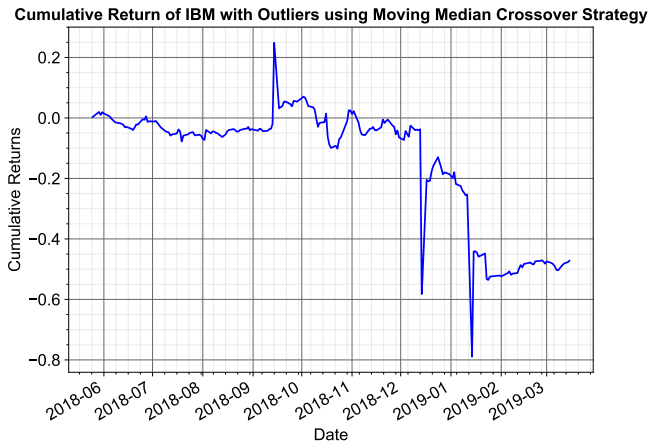


(b) Cumulative Returns with Outliers

Figure 73: Moving Median Cross Over Strategy implemented on Apple's Adjusted Close Prices with outliers (left) and Cumulative Returns of Apple using the strategy (right)

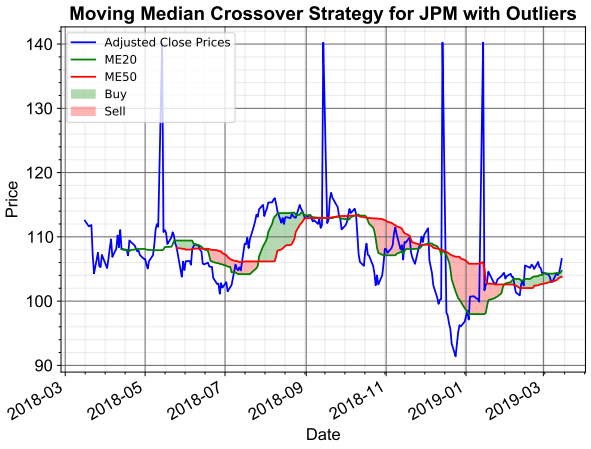


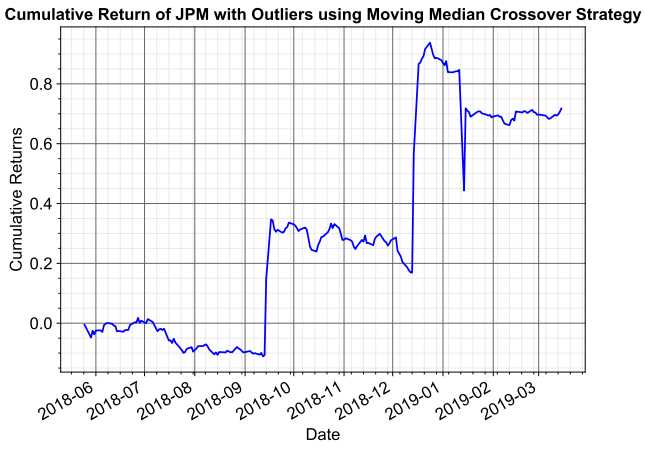
(a) MA Crossover Strategy for IBM with Outliers



(b) Cumulative Returns with Outliers

Figure 74: Median Median Cross Over Strategy implemented on IBM's Adjusted Close Prices with outliers (left) and Cumulative Returns of IBM using the strategy (right)

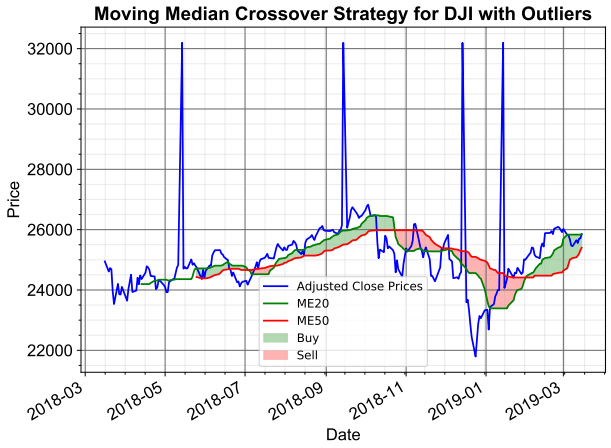


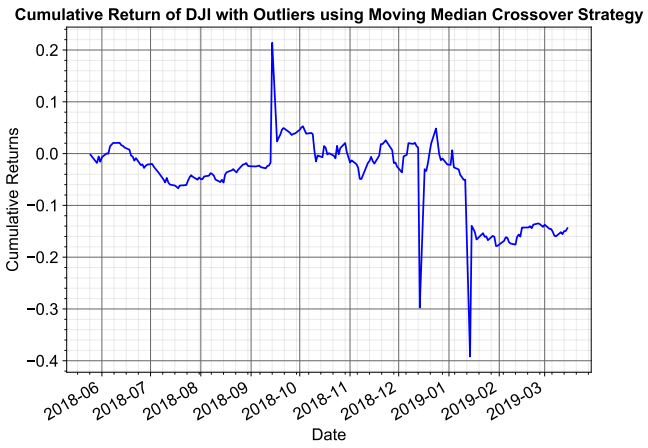


(a) MA Crossover Strategy for DJI with Outliers

(b) Cumulative Returns with Outliers

Figure 75: Median Median Cross Over Strategy implemented on DJI's Adjusted Close Prices with outliers (left) and Cumulative Returns of DJI using the strategy (right)





(a) MA Crossover Strategy for J.P. Morgan with Outliers

(b) Cumulative Returns with Outliers

Figure 76: Median Median Cross Over Strategy implemented on J.P. Morgan's Adjusted Close Prices with outliers (left) and Cumulative Returns of J.P. Morgan using the strategy (right)

From Fig. 73, Fig. 74, Fig. 75 and Fig. 76, the introduction of outliers does not impact the moving median crossover strategy. The aforementioned plots are very similar to their respective plots without outliers in Fig. 69, Fig. 70, Fig. 71 and Fig. 72, respectively. But as seen from Fig. 65, Fig. 66, Fig. 67 and Fig. 68, the moving average crossover strategy changes to a large extent.

Moreover, from Fig. 73, Fig. 74, Fig. 75 and Fig. 76, Apple, IBM and DJI produce negative cumulative returns while JPM produces a positive cumulative return in the presence of outliers . On the other hand, Fig. 65, Fig. 66, Fig. 67 and Fig. 68, IBM produces a positive cumulative return while Apple, DJI and JPM produce negative cumulative returns using the Moving Average Crossover Strategy with outliers. Finally, while the Moving Median Crossover Strategy is a robust technique to mitigate outliers, its does not have a low response time to profit from sudden positive spikes in the prices.

5 Graphs in Finance

5.1

Symbol	Security	Headquarters Location	Date first added
APC	Anadarko Petroleum Corp	The Woodlands, Texas	1997-07-28
APA	Apache Corporation	Houston, Texas	1997-07-28
COG	Cabot Oil & Gas	Houston, Texas	2008-06-23
CXO	Concho Resources	Midland, Texas	2016-02-22
COP	ConocoPhillips	Houston, Texas	
FANG	Diamondback Energy Inc	Midland, Texas	2018-12-03
EOG	EOG Resources	Houston, Texas	2000-11-02
MRO	Marathon Oil Corp.	Houston, Texas	1991-05-01
NFX	Newfield Exploration Co	Houston, Texas	2010-12-20
NBL	Noble Energy Inc	Houston, Texas	2007-10-08

Table 18: Details of Chosen 10 Energy Companies within the Oil and Gas Exploration and Production Sub-Industry with Headquarters in Texas

The 10 chosen companies for this section are companies within the Energy Industry. Specifically, these 10 chosen companies are within the Oil & Gas Exploration and Production Sub-Industry with headquarters in the state of Texas, United States. An Exploration & Production (E&P) company is involved in the early stages of energy production, which includes searching and extracting oil and gas. The list of companies chosen along with the details about Headquarters and Symbols is shown in Table 18.

While Chevron and Exxonmobil have a significant share in the E&P market they are classified under the 'Integrated Oil & Gas' Sub-Industry. Chevron and Exxonmobil are actively involved not just in the E&P activities but also in at least one of refining, marketing and transportation or chemicals. For this exercise, companies that are solely involved in the E&P business with Headquarters in Texas are chosen.

The Oil & Gas Industry is one of the largest industries in the U.S. and supports 10.3 million jobs whilst contributing to nearly 8% of the country's Gross Domestic Product (GDP). Moreover, around 69% of energy consumption in the U.S. was from Petroleum and Natural Gas, in 2019. Therefore, it is interesting to explore the relationship between the different E&P companies in the country. Moreover, companies with headquarters in Texas were chosen since the state is the largest producer of Oil and Natural Gas in the U.S. Finally, the prices of the 10 stocks and their Natural Log-Returns are shown in Fig. 77 and Fig. 78, respectively.

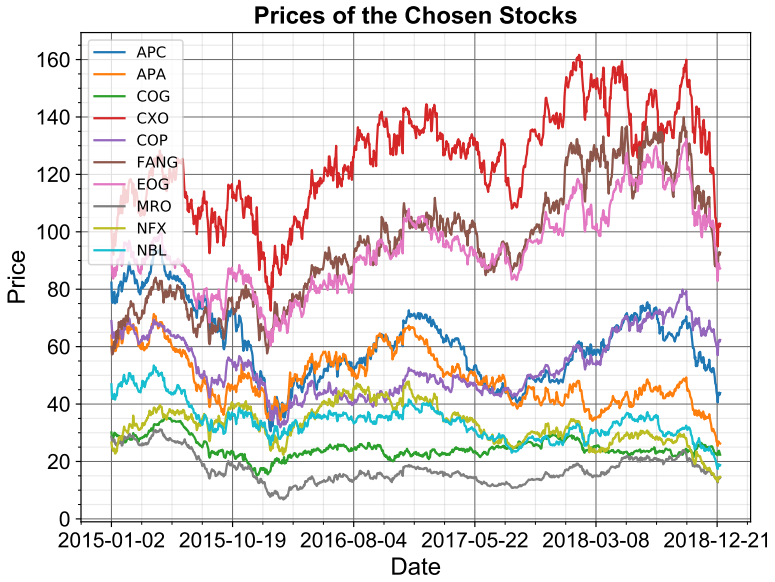


Figure 77: Prices of the chosen stocks in Table 18 from 02/01/2015 to 31/12/2018

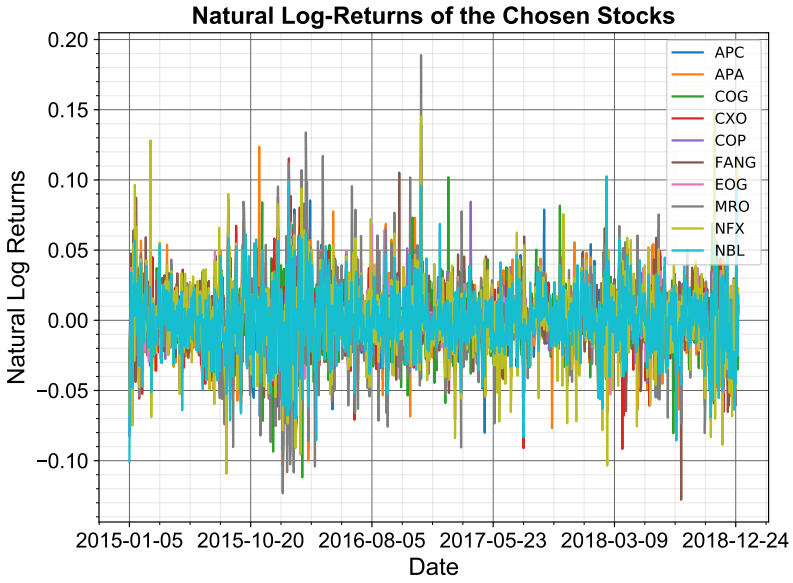


Figure 78: Natural Log-Returns of the chosen stocks in Table 18 from 02/01/2015 to 31/12/2018

5.2

	APC	APA	COG	CXO	COP	FANG	EOG	MRO	NFX	NBL
APC	1.0	0.70337	0.44246	0.69483	0.73922	0.63345	0.75636	0.76204	0.69567	0.74332
APA	0.70337	1.0	0.48121	0.69441	0.70989	0.6254	0.74448	0.72969	0.65277	0.70962
COG	0.44246	0.48121	1.0	0.45071	0.43973	0.4472	0.49218	0.4681	0.45239	0.45168
CXO	0.69483	0.69441	0.45071	1.0	0.70027	0.76856	0.79999	0.68704	0.69897	0.70545
COP	0.73922	0.70989	0.43973	0.70027	1.0	0.65477	0.77712	0.8003	0.68107	0.71172
FANG	0.63345	0.6254	0.4472	0.76856	0.65477	1.0	0.74562	0.62761	0.68227	0.63554
EOG	0.75636	0.74448	0.49218	0.79999	0.77712	0.74562	1.0	0.74374	0.71487	0.74901
MRO	0.76204	0.72969	0.4681	0.68704	0.8003	0.62761	0.74374	1.0	0.68597	0.72016
NFX	0.69567	0.65277	0.45239	0.69897	0.68107	0.68227	0.71487	0.68597	1.0	0.68366
NBL	0.74332	0.70962	0.45168	0.70545	0.71172	0.63554	0.74901	0.72016	0.68366	1.0

Table 19: Correlation Matrix generated using Natural Log-Returns of the 10 chosen Stocks in Table 18

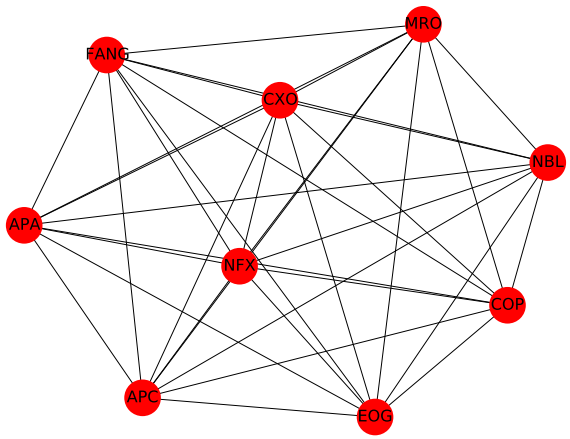


Figure 79: Correlation Log-Returns Graph for the Correlation Matrix in Table 19 with Threshold = 0.5

Table 18 shows the correlation matrix is generated using the Natural Log-Returns, from 02/01/2015 to 31/12/2018, of the chosen 10 stocks in Table 18. Consequently, the correlation log-returns graph between the stocks is constructed according to the correlation matrix in Table 19. In the correlation matrix in Table 19, the correlation values can range from -1 to 1. Values of 1, 0, -1 indicate high correlation, no correlation and negative correlation,

respectively. As expected, the diagonal of the matrix in Table 19 is ones, since the log-returns of a stock is highly correlated with itself. It is also interesting to note that the correlation matrix is symmetric. Finally, the correlation log-returns graph constructed according to the correlation matrix in Table 19 with a threshold of 0.5 is shown in Fig. 79.

Considering the correlation log-returns graph in Fig. 79, the graph topology can be analyzed. There are 10 nodes (red-filled circles) in Fig. 79, each corresponding to one of the 10 companies. These nodes are spatially placed according to its correlation value with other nodes. The lines connecting different nodes in a graph are called graph edges. These graph edges between different two different nodes are drawn, only if the correlation between the corresponding log-returns of the two stocks is greater than 0.5. In other words, the correlation threshold for connection between two different nodes is set to 0.5. The correlation threshold for connection between two different nodes can also be set to higher values such as 0.6 and 0.7, as shown in Fig. 80 and Fig. 81, respectively. Finally, the self-connecting graph edges for each node can also be drawn but these are omitted since the correlation of the log-returns of a stock with itself is 1, as seen in Table 19.

The correlation matrix plays an important role in the construction of the graphs. It indicates how different nodes should be connected to one another. If a stock is highly correlated with multiple stocks, this node can be spatially placed such that it is closer to multiple nodes. This can be better visualized in Fig. 81, where the correlation threshold is 0.7. For instance, Noble Energy Inc. (NBL) has log-returns that have a high correlation of > 0.7 with 6 other companies, as seen in Table 19. Therefore, the NFX node in Fig. 81 is placed such that it is close and connected to 6 other nodes. On the other hand, Newfield Exploration Company (NFX) has log-returns that only have a high correlation of > 0.7 with the log-returns of 1 other company, i.e. EOG Resources Inc. (EOG). Therefore, the NFX node is placed further apart.

Finally, the log-returns of Cabot Oil and Gas Exploration Inc. (COG) are do not have a high correlation > 0.5 with any other stock. Therefore, the COG node does not appear in any of the graphs in Fig. 79, Fig. 80 and Fig. 81.

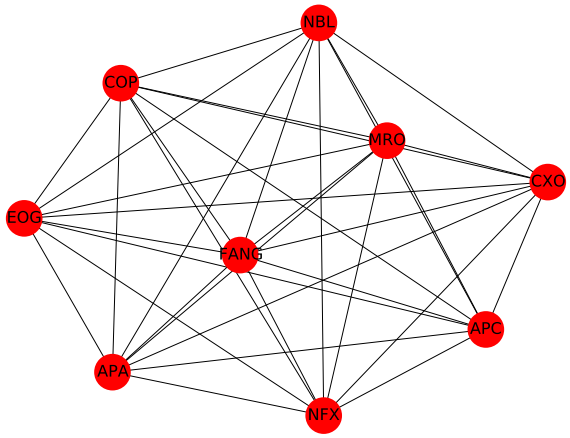


Figure 80: Correlation Returns Graph for the Correlation Matrix in Table 19 with Threshold = 0.6

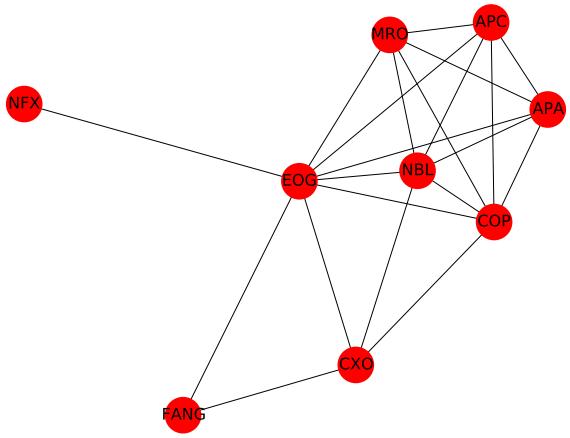


Figure 81: Correlation Returns Graph for the Correlation Matrix in Table 19 with Threshold = 0.7

5.3

The topology of the graph is dictated by the nature of the data. As previously explored in Section 5.2, the correlation matrix affects the topology of the graph and how the connections between different vertices are made. The impact of using raw prices instead of log-returns to construct the correlation matrix and the Graph will be explored in Section 5.5. Additionally, data similarity also dictates the underlying graph topology. Several approaches such as precision matrix can be used to determine the data similarity apart from using the correlation matrix.

Reordering of Graph Vertices (Nodes) does not affect the result. In fact, using Python's `NetworkX` generates a different order of Graph vertices, each time the graph is plotted using the correlation matrix. The results are not different as long as the connection between the nodes remain the same and the highly connected vertices are placed together.

	APC	**APA**	**COG**	**CXO**	**COP**	**FANG**	**EOG**	**MRO**	**NFX**	**NBL**
APC	1.0	0.70337	0.44246	0.69483	0.73922	0.63345	0.75636	0.76204	0.69567	0.74332
APA	0.70337	1.0	0.48121	0.69441	0.70989	0.6254	0.74448	0.72969	0.65277	0.70962
COG	0.44246	0.48121	1.0	0.45071	0.43973	0.4472	0.49218	0.4681	0.45239	0.45168
CXO	0.69483	0.69441	0.45071	1.0	0.70027	0.76856	0.79999	0.68704	0.69897	0.70545
COP	0.73922	0.70989	0.43973	0.70027	1.0	0.65477	0.77712	0.8003	0.68107	0.71172
FANG	0.63345	0.6254	0.4472	0.76856	0.65477	1.0	0.74562	0.62761	0.68227	0.63554
EOG	0.75636	0.74448	0.49218	0.79999	0.77712	0.74562	1.0	0.74374	0.71487	0.74901
MRO	0.76204	0.72969	0.4681	0.68704	0.8003	0.62761	0.74374	1.0	0.68597	0.72016
NFX	0.69567	0.65277	0.45239	0.69897	0.68107	0.68227	0.71487	0.68597	1.0	0.68366
NBL	0.74332	0.70962	0.45168	0.70545	0.71172	0.63554	0.74901	0.72016	0.68366	1.0

Table 20: Correlation Matrix of the 10 chosen stocks in Table 18 when the daily log-returns are shuffled

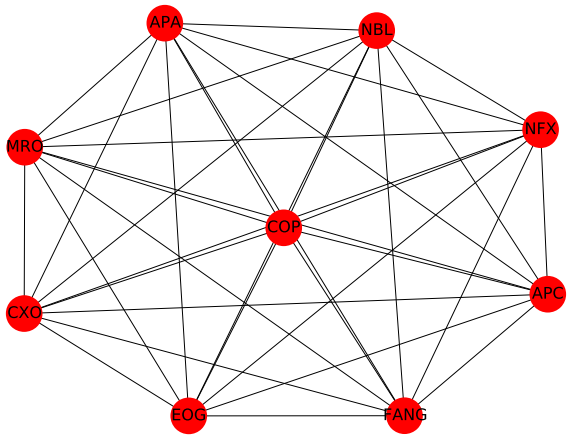


Figure 82: Correlation Returns Graph for the Correlation Matrix in Table 20 with Threshold = 0.5

Finally, the log-returns time-series is shuffled and the corresponding correlation matrix between the different stocks is shown in Table 20. It must be emphasized that only the returns series was shuffled and not the original time series, consisting of prices. From Table 20, it can be noted that the correlation matrix of the shuffled returns time series is identical to the correlation matrix obtained for the original returns time series in Table 19

The matrices in Table 20 and Table 19 are identical since the correlation matrix is unaffected by the re-ordering of the values given the correlation is just a measure of similarity between the two time series. The graph topologies in Fig. 79 and Fig. 82 corresponding to the matrices in Table 19 and Table 20, respectively, do not look exactly the same. This is because, each time, the Python `NetworkX` library generates a different looking graph. But the connection between different nodes and the arrangement is exactly the same. For instance, comparing Fig. 81 and Fig. 84, the NFX node is still further away from the graph, given its log-returns do not have a high correlation of > 0.7. At the same time, the NBL node in both Fig. 81 and Fig. 84 is placed such that it is connected to 6 other nodes. This is because NBL's log-returns have a high correlation of > 0.7, with 6 other companies.

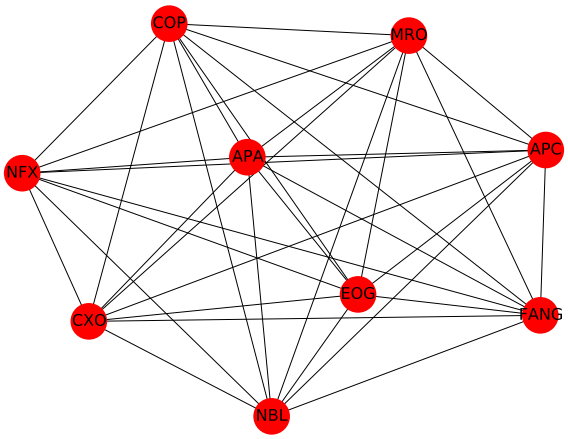


Figure 83: Correlation Returns Graph for the Correlation Matrix in Table 20 with Threshold = 0.6

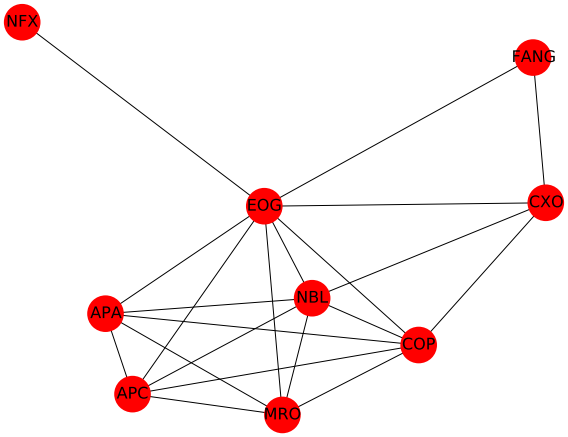


Figure 84: Correlation Returns Graph for the Correlation Matrix in Table 20 with Threshold = 0.7

5.4

	APC	APA	COG	CXO	COP	FANG	EOG	MRO	NFX	NBL
APC	0.0	0.45878	0.49552	0.45685	0.41755	0.47747	0.40559	0.52152	0.52935	0.43032
APA	0.45878	0.0	0.51932	0.48086	0.46501	0.49731	0.44584	0.56419	0.55708	0.47855
COG	0.49552	0.51932	0.0	0.50391	0.46212	0.51105	0.46767	0.62064	0.60237	0.50991
CXO	0.45685	0.48086	0.50391	0.0	0.44589	0.43919	0.39134	0.58	0.53745	0.44904
COP	0.41755	0.46501	0.46212	0.44589	0.0	0.45954	0.36068	0.52654	0.54651	0.42758
FANG	0.47747	0.49731	0.51105	0.43919	0.45954	0.0	0.41249	0.60287	0.53865	0.47472
EOG	0.40559	0.44584	0.46767	0.39134	0.36068	0.41249	0.0	0.58559	0.53915	0.40693
MRO	0.52152	0.56419	0.62064	0.58	0.52654	0.60287	0.58559	0.0	0.6115	0.55313
NFX	0.52935	0.55708	0.60237	0.53745	0.54651	0.53865	0.53915	0.6115	0.0	0.529
NBL	0.43032	0.47855	0.50991	0.44904	0.42758	0.47472	0.40693	0.55313	0.529	0.0

Table 21: Matrix generated using the Dynamic Time Warping Algorithm applied to the Natural Log-Returns of the chosen stocks in Table 18

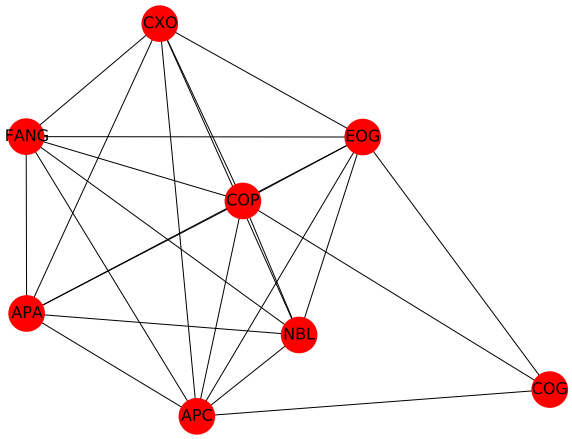


Figure 85: Graph for the Dynamic Time Warping Matrix in Table 21 with Threshold = 0.5

Dynamic Time Warping (DTW) is an algorithm in Time Series analysis that measures the similarity between two temporal sequences. While Dynamic Time Warping can be used to compare sequences of different lengths and has applications in speech recognition, it is also highly applicable for Financial Time Series data. For two symbols x and y, the distance between the two symbols $d(x, y)$ is given by $d(x, y) = |x - y|$. The pseudocode for the DTW algorithm is shown in Listing 4.

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
    DTW := array [0..n, 0..m]

    for i := 0 to n
        for j := 0 to m
            DTW[i, j] := infinity
    DTW[0, 0] := 0

    for i := 1 to n
        for j := 1 to m
            cost := d(s[i], t[j])
            DTW[i, j] := cost + minimum(DTW[i-1, j  ],  // insertion
                                        DTW[i  , j-1],  // deletion
                                        DTW[i-1, j-1])  // match

    return DTW[n, m]
}
```

Listing 4: Psuedocode for the Dynamic Time Warping (DTW) Algorithm

The Dynamic Time Warping algorithm is applied to the natural log-returns of the 10 chosen stocks in Table 18. Consequently, the matrix is generated in Table 21. The dynamic time warping results between two stocks vary from 0 to 1. The lower (higher) the result, the closer (further) is the distance between the natural log-returns of the two stocks. As expected, the diagonal of the matrix in Table 21 consists of zeroes since the distance between the log-returns of the stock with itself is just 0.

The graph generated for the matrix in Table 21 with a threshold 0.5 is shown in Fig. 85. The graph in Fig. 79 is generated for a threshold of > 0.5. In other words, the graph edges are added between two different stocks if the correlation of their natural log-returns is > 0.5. In the case of DTW, the natural log-returns of the two different stocks are closer if the distance between them is low. Therefore, the threshold for the graph in Fig. 85 is < 0.5.

The differences between Fig. 79 and Fig. 85 can be noted. For instance, the Cabot Oil and Gas Corporation (COG) vertex does not appear in Fig. 79 since it's log-returns have a correlation of < 0.5 compared to that of the other stocks. In fact, the COG vertex is not included in any of the graphs based on correlation of log-returns. However, COG appears on Fig. 85 since it has log-return distances of < 0.5 with COP, EOG and APC. Similarly, while NFX has graph edges to several other nodes in Fig. 79, it does not even have a vertex in Fig. 85.

	APC	APA	COG	CXO	COP	FANG	EOG	MRO	NFX	NBL
APC	0.0	0.46623	0.49757	0.46942	0.43692	0.49012	0.40127	0.53201	0.52925	0.44876
APA	0.46623	0.0	0.52532	0.47726	0.45813	0.52241	0.44772	0.54369	0.54966	0.4505
COG	0.49757	0.52532	0.0	0.49429	0.46217	0.5107	0.4498	0.60445	0.57456	0.50432
CXO	0.46942	0.47726	0.49429	0.0	0.43618	0.42953	0.3904	0.58026	0.53342	0.45224
COP	0.43692	0.45813	0.46217	0.43618	0.0	0.44579	0.35675	0.52541	0.55162	0.4357
FANG	0.49012	0.52241	0.5107	0.42953	0.44579	0.0	0.4159	0.60125	0.54527	0.48537
EOG	0.40127	0.44772	0.4498	0.3904	0.35675	0.4159	0.0	0.57922	0.52925	0.415
MRO	0.53201	0.54369	0.60445	0.58026	0.52541	0.60125	0.57922	0.0	0.6042	0.56545
NFX	0.52925	0.54966	0.57456	0.53342	0.55162	0.54527	0.52925	0.6042	0.0	0.53324
NBL	0.44876	0.4505	0.50432	0.45224	0.4357	0.48537	0.415	0.56545	0.53324	0.0

Table 22: Matrix generated using the Dynamic Time Warping Algorithm applied to the shuffled Natural Log-Returns of the chosen stocks in Table 18

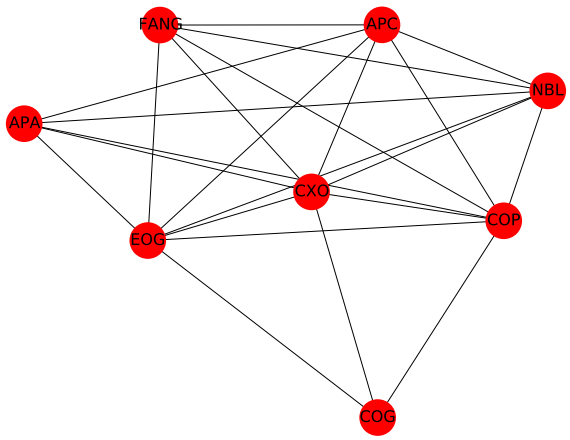


Figure 86: Graph for the Dynamic Time Warping Matrix in Table 22 with Threshold = 0.5

Finally, the natural log-returns time series is shuffled and the matrix in Table 22 is generated using the Dynamic Time Warping Algorithm. The values in the matrix in Table 22 are not exactly the same as the ones in Table 21. But at the same time, the values in Table 22 have a very low deviation from the values in Table 21, i.e. a very low percentage error. The resulting graph topology generated using the matrix in Table 21 is shown in Fig. 86 which has exactly the same graph edge connections as in Fig. 85.

5.5

	APC	APA	COG	CXO	COP	FANG	EOG	MRO	NFX	NBL
APC	1.0	0.58023	0.52077	0.1663	0.64164	-0.02836	0.32063	0.92658	0.27483	0.81258
APA	0.58023	1.0	0.3742	-0.01628	-0.01321	-0.28715	-0.07505	0.48564	0.7248	0.83699
COG	0.52077	0.3742	1.0	0.17332	0.36722	-0.04388	0.15192	0.64009	0.06566	0.44832
CXO	0.1663	-0.01628	0.17332	1.0	0.41045	0.9062	0.86394	0.17003	0.0271	-0.09902
COP	0.64164	-0.01321	0.36722	0.41045	1.0	0.43414	0.71488	0.78638	-0.3552	0.26159
FANG	-0.02836	-0.28715	-0.04388	0.9062	0.43414	1.0	0.89389	0.01113	-0.21371	-0.32849
EOG	0.32063	-0.07505	0.15192	0.86394	0.71488	0.89389	1.0	0.38299	-0.20192	-0.04678
MRO	0.92658	0.48564	0.64009	0.17003	0.78638	0.01113	0.38299	1.0	0.08773	0.73748
NFX	0.27483	0.7248	0.06566	0.0271	-0.3552	-0.21371	-0.20192	0.08773	1.0	0.5471
NBL	0.81258	0.83699	0.44832	-0.09902	0.26159	-0.32849	-0.04678	0.73748	0.5471	1.0

Table 23: Correlation Matrix of the 10 Chosen Stocks in Table 18 based on Raw Prices

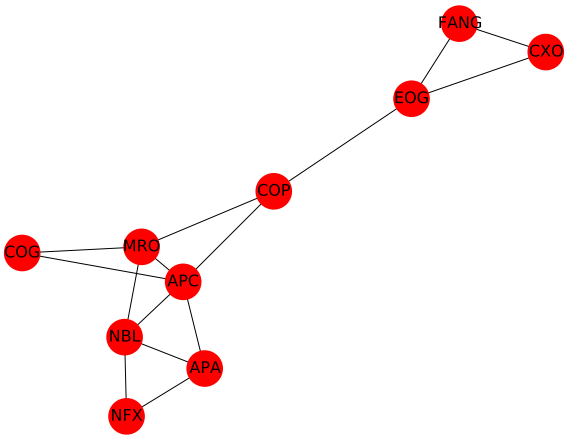


Figure 87: Graph for the Correlation Matrix in Table 23 with Threshold = 0.5

The correlation matrix generated using raw prices along with the corresponding graph topology is shown in Fig. 87 and Table 23, respectively. When comparing Table 19 and Table 23, it can be deduced that raw prices do not capture the relationships between the different stocks. Using log-returns, not only the data is stationary but also the relationship between the different stocks can be determined. For instance, using the log-returns of the stocks to determine the graph shows that all 9 stocks are highly correlated with each other, as shown in Fig. 79. However, this is not the case when raw prices are used as shown in Fig. 87.

Additionally, reordering of the time series does not affect the correlation matrix generated in Table 23. Instead of returns, the raw prices are used and the actual order of the prices does not matter, when determining the correlation between two stocks.

Finally, the DTW algorithm does not have the values bounded from 0 to 1 in the matrix even if normalized prices are used. Therefore, it becomes difficult to set the threshold and an alternative method is to be used, if raw prices are considered.

References

[1] 3.44 robustness, breakdown points, and 1-dimensional location m-estimators. **april** 2003. URL: http://www2.myoops.org/twocw/mit/NR/rdonlyres/Mathematics/18-466Mathematical-StatisticsSpring2003/A0F1067F-3FB1-4B90-A07C-B5EE182996B2/0/c3s44.pdf.