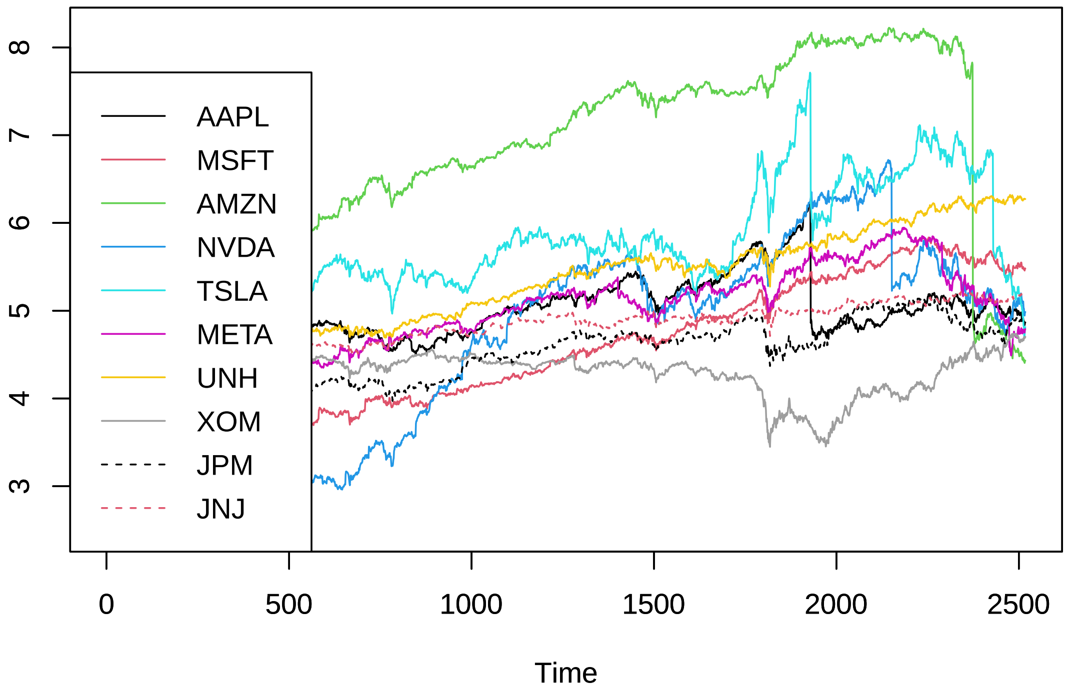
ST326 Assessment Autumn Term 2023
Candidate No: 33732

Predictive Analysis of Stock Market Performance Using Statistical and Financial Methods.
2023-12-08

Question 1
Stock prices are loaded in using the getSymbols function in R. The stocks chosen are as follows Apple (AAPL), Microsoft (MSFT), Amazon (AMZN), NVDIA (NVDA), Tesla (TSLA), Meta (META), United Health Group (UNH), Exxon Mobil (XOM), JPMorgan (JPM), and Johnson and Johnson (JNJ). We obtain their log prices by using the log function. There are no missing values in the log prices for each stock, this can be verified with the any(is.na()) function in R. When we take log returns, we need to eliminate the first observation with na.omit().



Plot of log prices is obtained by using read. bossa.data() function introduced in the lectures that has been adjusted to read stock data (adjustment to scan() function), note that all stocks have closing prices on the same day. Here I used methods as in the lecture notes to produce time series plots for log prices. Note the y-axis is the log price and the x-axis is time(days). In this report, the trend of the daily closing prices of the selected ten companies was analysed. All companies displayed an upward trend, with Tesla exhibiting exceptional growth. The COVID-19 pandemic in 2019 had a negative impact on the prices of these all companies to varying degrees, with Tesla being the most affected and NVDA being minimally impacted. In 2022, a slight economic downturn was observed, resulting in declining prices for most companies. We can see that Amazon and Microsoft experienced a loss but UNH and XOM showed a marginal increase. Overall, the plot provides valuable insights into the performance and volatility of these stocks over the given period, which can be attributed to a variety of factors such as changes in market demand, company announcements, economic indicators, government policies, and global events. By analysing the patterns and trends in the plot, investors and analysts can gain a better understanding of the underlying forces driving the performance of these stocks and make informed investment decisions.

We first convert log prices into returns, here we need to remove the first row as returns are undefined here. Then, we convert it into a matrix using the as.matrix() function in R to obtain the sample covariance matrix through a simple cov() function in R as we are not asked to find the conditional covariance matrix. Then we use SVD to compute the eigen values of the matrix, which will give us the condition number which is defined as

Condition number = |λ| max(A) / |λ| min(A)

Condition numbers

1) 10 years of data: 29.3803
2) 5 years of data: 43.62148
3) 1 year of data: 82.60457

The intuition is as follows as the closer the number of covariates in this case stocks, is to the sample size, the larger the condition number is for the sample covariance matrix. For longer period length, we have more data points for each stock, which give us more stable estimates of the covariances between the stocks. The higher condition numbers can imply that there are potential challenges associated with multicollinearity and instability in the covariance matrix estimates in shorter period lengths. Also, there may have been more noise in the past few years combined with worse market conditions during the shorter period length.

In the context of portfolio optimisation, higher condition numbers could suggest that small changes in returns could lead to large changes in the portfolio weights, indicating a potentially unstable or risky portfolio.

Question 3

(Annualised) Sharpe ratio = √250 × Average daily return/ SD of daily returns

In this report, we take a systematic approach to make meaningful comparisons of Sharpe ratios for different parameter values in the portfolio optimisation problem. We have four tuning parameter (t0, D, l and x), so comparisons can be complex due to multidimensional nature of the parameter space. To simplify our comparisons, we first limit the number of variations by defining realistic and practical ranges for each parameter. We also enforce manageable increments to ensure computational feasibility. Our parameters are given as:

T0: starting point for investment, here we choose T0 to be large enough so that our starting weights accurately reflect the trading period before the start time. A common practice is to use the beginning of a financial year.

D: window length used to compute estimates of the mean return for each stock and the covariances matrix for previous sample returns. There's a trade-off between using a longer window (which may provide more stability in the covariance estimates) and a shorter window (which may be more responsive to recent market conditions).
The choice of window length D needs to reflect our investment strategy and risk tolerance. We also need to choose D so that t – D + 1 is bigger than 0, if not we would be inputting NA values into our optimization problem. Common choices are 1 month, 1 quarter or 1 year.

x: x is a parameter to be varied in our code, it influences the target return input in our quadratic optimization, x can be viewed as the scaling factor for target returns, it is divided by 10 and multiplied with the sum of mean returns for each individual stock. We often start x at a modest level, 0.5 or 1 then is adjusted based on risk-return tradeoff.

I: adjustment interval, one thing to bear in mind here is that in real markets there is a transactional cost involved with updating the portfolio which will effect daily returns. So here we do not want to rebalance our portfolio too often. This is highly dependent on our strategy.

Next, we conduct a sensitivity analysis on each parameter to understand its impact on the investment strategy and performance. We start by varying each parameter individually, we change one parameter while keeping the others constant. The baseline parameters that we use are T0 = 1000 (4 years) , D= 90 (1 quarter), I=90, x=1 (tolerant). We can then compare our performance metrics, the Sharpe ratio and Average sum of absolute changes in portfolio weights(Avs Achg). Through looping methods in R, we are able to define realistic ranges for each parameter to vary to produce a data frame of each combination of tuning parameters and the statistics. An extract of the table is included below.

As we compare Sharpe Ratios we note that different start times result in drastically different outputs, so start time is of our strategy is very important. Next, long windows(>180days) for our rolling statistics can result in Sharpe ratios being reversed, as optimal weights do not reflect market conditions. To add, rebalancing periods do not seem to have a significant as great an impact on Sharpe ratios. As we vary **I** the trading period, the changes to the ratio are small. Lastly, the parameter x also does not seem to have as great an impact on Sharpe ratios as compared to the start time and window length.

AvsAchg= (1/Total No of I day periods) * (Sumt‖wt+1 −wt‖1)
where **w**t is the weight vector used on trading day t.

To calculate the AvsAchg we reference the adjustment points by a looping method, here we must define the loop variable, n from 1: **ceiling(nrow(weights_matrix)-t0)/I** -1 as changes in weights happen at every adjustment point. We add up these absolute changes, count the number I day periods inputted to the function and compute the statistic. The results are included in the table.

In practice a smaller change in absolute average AvsAchg is often considered better for several reasons related to cost, risk management and investment philosophy.

Firstly, each time we rebalance our portfolio we incur costs such as brokerage fees and bid ask spreads. Smaller changes in weights typically mean fewer and less significant transactions, thus lowering these costs. Over time, the savings from reduced transaction costs can compound, significantly impacting the overall returns of the portfolio.

Next, we consider stability and consistency. Smaller incremental shifts in portfolio weights can lead to more stable portfolio performance. Frequent and significant shifts in asset allocation can be unsettling and may suggest a lack of a clear investment strategy. Investors prefer consistency. There are also benefits to performance tracking and simplicity of managing and monitoring the portfolio when fewer changes are made.
In terms of investment philosophy, the smaller Avs Achg implies a long-term strategy, where we aim to minimise reactions to short term market volatility and focus on long term growth. Additionally, it is reflective of a discipline approach of sticking to a predetermined investment strategy.

To find the optimal parameter setting, that will achieve better outputs we need to define our optimisation criteria. We want to find the parameter setting that maximises the Sharpe ratio while minimising Avs Achg, i.e., a better Sharpe ratio and Avs Achg balance. This would mean better risk adjusted returns and ensure efficient portfolio rebalancing. There is a trade-off between a higher Sharpe ratio and a lower Avs Achg. We sorted the table of parameter values in order of descending Sharpe ratios then ascending order of Avs Achg so that the top row on the results data frame would reflect the parameter inputs that satisfies my first optimisation criteria.

The resulting combination is given as t0= 500, D= 60, I= 70 and x= 1.2 as this parameter has a large Sharpe ratio (0.74) but also a small Avs Achg (9.9789). We searched through the results dataframe to find and select this value. Because our optimisation criteria has two constraints, we would have more than one optimal parameter setting.

Our investment strategy is as follows, since **D** (trading window) is used to determine portfolio weights and **I** (rebalance period) indicates how frequently we adjust our portfolio I suggest that the optimal values of **D** and **I** to be the same or very close so that the responsiveness of the trading strategy aligns well with the frequency of capitalising on that information. This will result in good investment decisions. In choosing **I** we also need to consider practical aspects such as transaction cost and the frequency of trading. We test the strategy in various market conditions to see the robustness of our investment strategy. Therefore, **T0** should be flexible to be able to evaluate the strategies performance across different market phases such as bull, bear and high volatility. So, we vary **T0** in backtesting while trying to optimize our other parameters to maximize our performance metrics.

Through this analysis we are able to make conclusions on the optimisation of our parameter inputs . For example, when we take values of **x** greater than 2.5, the Avs Achg becomes bigger implying that we need to adjust our portfolio more aggressively. In practice this would be bad as explained in the section above, so optimally we should have **x** close to 1 so, that our target return is reasonably estimated. Next we consider how **D** and **I** affects the trading strategy, we note that when **D** and **I** close together we are able to produce better Sharpe ratios and smaller Avs Achg. Here **I** has a significant impact on the Avs Achg statistic.

In conclusion our trading strategy is to keep **D** and **I** close, and choose **x** to be close to one. Note that it is critical to adjust our strategy based on changing market conditions and investment goals. The table of initial comparison and statistics given below.

t0	D	I	x	Sharpe_Ratio	AvsAchg
750	90	90	1	-0.7201531	11.6416238
1000	90	90	1	0.65063905	11.127772
1250	90	90	1	0.57034149	9.19411673
1500	90	90	1	-0.5451249	10.6186915
1750	90	90	1	0.3831112	9.85615992
2000	90	90	1	0.28778209	10.6338942
2250	90	90	1	-1.241313	7.79104878
750	60	90	1	-0.7109563	11.958056
750	80	90	1	-0.7139348	12.1313161
750	100	90	1	-0.7171148	11.6178643
750	120	90	1	-0.7319083	11.2949247
1000	60	90	1	0.6590961	9.75858848
1000	80	90	1	0.69384618	10.6593645
1000	100	90	1	0.63721194	10.5300438
1000	120	90	1	0.62452419	10.8157012
1250	60	90	1	0.52939485	8.25065443
1250	80	90	1	-0.4353466	9.83070329
1250	100	90	1	0.52186177	9.64563849
1250	120	90	1	0.52125918	9.56776776
1500	60	90	1	-0.5009175	8.08473275
1500	80	90	1	-0.5466263	9.43251383
1500	100	90	1	-0.5713402	9.1237544
1500	120	90	1	-0.5496087	8.71362598
1750	60	90	1	0.4188276	8.9887738
1750	80	90	1	0.31987283	9.37488219
1750	100	90	1	0.38390115	9.55150623
1750	120	90	1	0.41933555	11.3072811
2000	60	90	1	-0.0295281	9.18254331
2000	80	90	1	0.16885479	9.71876823
2000	100	90	1	-0.0617308	10.1302255
2000	120	90	1	-0.0033226	7.65936913
2250	60	90	1	-1.1979185	4.46990375
2250	80	90	1	-1.2724243	7.49745867
2250	100	90	1	-1.1487127	4.40328336
2250	120	90	1	-1.1465008	9.41161672
750	90	60	1	-0.8057665	8.79524211

750	90	80	1	-0.7492229	11.225717
750	90	100	1	-0.7101921	11.3745094
750	90	120	1	-0.8043751	10.9796316
1000	90	60	1	0.62586238	10.313715
1000	90	80	1	0.61854884	9.7282638
1000	90	100	1	0.67071876	11.6393197
1000	90	120	1	0.6552879	12.3675009
1250	90	60	1	0.55690021	9.31958004
1250	90	80	1	0.42897557	9.35427819
1250	90	100	1	0.41867768	9.87118619
1250	90	120	1	0.49808372	10.2243002
1500	90	60	1	-0.6198957	8.68598084
1500	90	80	1	-0.5780029	9.49412908
1500	90	100	1	-0.7770172	10.5054908
1500	90	120	1	-0.5446072	9.38698551
1750	90	60	1	0.44874253	7.81846588
1750	90	80	1	0.55583454	9.90052233
1750	90	100	1	0.44664123	8.8049271
1750	90	120	1	0.42747422	10.1747416
2000	90	60	1	0.42436373	6.72824954
2000	90	80	1	0.97452303	12.3690575
2000	90	100	1	-0.7896763	10.8519533
2000	90	120	1	0.05504871	6.48314759
2250	90	60	1	-1.1188504	7.61701041
2250	90	80	1	-1.1077283	6.17020763
2250	90	100	1	-1.1206972	8.08700007
2250	90	120	1	-1.0535095	7.03225478
2250	90	90	0.5	-1.262923	8.38412785
2250	90	90	1	-1.241313	7.79104878
2250	90	90	1.5	-1.2209275	7.19796972
2250	90	90	2	-1.2018276	6.60489065
2250	90	90	2.5	-1.1840031	6.01181159