## **Introduction**

Time series data can be found in numerous industries, including finance, economics, ecology, neuroscience, and physics [1]. Time series analysis has countless applications including financial planning, personal and corporate budgeting, supply chain management, energy consumption, and retail scheduling. This is used to determine patterns in data over time from historical time-dependent data. The data can then be used to predict or forecast future behaviors [2]. The data is often created from a set of fixed frequency data points where the observations occur at regular time intervals. However, any data that is created from observed or measured observations at many points in time qualifies as time series data. This type of data can also be irregular if there is not a fixed unit of time between each data point [1].

Stock market data is a specific application of time series data that includes the price value for stocks in a specific company measured over time. This data is useful to track the market value for public companies over time and can be used to attempt to predict future stock prices or analyze trends. With the available open source software, open data sources, and open online trading platforms, algorithmic trading (or “algo” trading) has come to stage with wide and lasting impacts in recent years.

This case study is an extension of the work done by **Wes McKinney** for time series analysis (github.com/wesm/pydata-book/tree/1st-edition). The code was adopted by Robert Slater and expanded by our team for this analysis. Specifically, we take a look at the signal frontier analysis and expand the work to test various lookback and holding periods of a cross-sectional momentum investing strategy.

Our simplified cross-sectional momentum portfolio was constructed with stocks from major US technology companies: 'NVDA' (Nvidia), 'GOOG' (Alphabet Inc.), 'AMZN' (Amazon), 'AAPL' (Apple), 'IBM' (IBM), 'MSFT' (Microsoft), and 'INTC' (Intel). These data were extracted from Yahoo Finance from 03/01/2016 to 03/01/2019. We will compare these values to the market value for 'SPY' (SPDR S&P 500 Trust) index and further explore if intermediate momentum strategies outperform long and short-term strategies for this portfolio.

## **Background**

Stock trading activity now has been dominated by computers that make buying and selling decisions with nano-second precision in network time protocol. Technical analysis is used to forecast the direction of stock prices through the study of past market data, primarily stock price and value. This method often stands in contradiction to much of modern portfolio theory which assemble a portfolio of assets such that the expected return is maximized for a given level of risk.

As simple technical analysis indicators, **momentum** (MTM) and **rate of change** (ROC) are to show the difference between today’s closing price and the close N days ago. Momentum investing, as a short-term strategy, aims to purchase stocks that have been showing an upward price trend or short sell stocks that have been showing a downward trend. The main rationale behind momentum investing is there is a great probability that the trend will continue once it is established.

Under alternative implementation strategies, momentum investing can be classified into cross-sectional or time-series momentum. Cross-sectional momentum is a technique to sort stocks based on some measures of past return, while time-series momentum assigns stocks on the basis of their absolute performance.

The lookback period and the holding period are two critical parameters to momentum investing strategy. Both can help draw a line between two points on a time series and calculate a return:

Where represents the daily return for a given business day and the difference of daily returns for and is defined by the lookback period. This lookback period can range from daily to yearly. For example, one strategy uses a lookback period of 12 months and a holding period of 1 month.

The lookback period and associated returns for a portfolio allow for the weighting of each investment in a portfolio. For example, given a lookback period of 30 days, if 2 out of 7 stocks are trending downward (have a negative return), these stocks would be down weighted in the portfolio. If the remaining stocks have an increasing trend, these stocks would be up weighted in the portfolio.

Based on the signal provided by a given lookback period, a momentum investor retains a portfolio mix of up-weighted and down-weighted stocks over a given "holding period," capturing cumulative returns over the holding period. This holding period can also take on any frequency, however, it should be consistent with the lookback period frequency in order to apply momentum weights appropriately. The ultimate goal is to maximize returns based on historical trends.

Momentum investment strategies can be broken down into three different sub-strategies: short-term, intermediate term and long-term momentum. Short-term momentum strategies utilize 1 month or less of a lookback period, while long-term strategies typically utilize 3 to 5 years for a lookback period. Both long and short-term strategies have been empirically shown to experience significant reversion. However, intermediate term strategies, which use a lookback period of 3 to 12 months show greater success, often resulting in no reversals.

**Sharpe ratio** is an acceptable metric to examine the performance of an investment by adjusting for its risk. The ratio measures the excessive return per unit of deviation in a trading strategy.

where is the expected return of portfolio return, is the risk-free rate, and is the portfolio standard deviation of the portfolio’s excess return. The Sharpe ratio has become the most widely used method for calculating the risk-adjust portfolio. Typically, Sharpe Ratio greater than 1 is considered acceptable to good by investor. A ratio greater than 2 is rated very good, and a ratio of 3 or higher is considered excellent.

The stocks of interest in this study, 'NVDA', 'GOOG', 'AMZN', 'AAPL', 'IBM', 'MSFT', and 'INTC', all represent popular US technology companies. We will compare the recent market values of these companies using time series data obtained from Yahoo’s stock data and explore if intermediate momentum strategies outperform long and short-term strategies for this portfolio.

## **Method**

In this study, we will investigate various combinations of lookback and holding periods and calculate corresponding Shape ratios in order to find the optimal momentum investing strategy for our US technology portfolio stocks. The historical Yahoo! finance stock price were extracted from 03/01/2016 to 03/01/2019. The recent 5-day adjusted close price for our portfolio stocks are shown in **Table 1**. We can see all the stock prices have less fluctuations in last 5 days. The stock prices for GOOG and AMZN are much higher than other chosen stocks.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 1: Recent adjusted close price for portfolio stocks from Yahoo Finance API** | | | | | | | | |
| Date | NVDA | GOOG | AMZN | AAPL | IBM | MSFT | INTC | SPY |
| **2019-02-25** | 158.53 | 1109.4 | 1633 | 174.23 | 139.46 | 111.59 | 53.1 | 279.52 |
| **2019-02-26** | 156.94 | 1115.13 | 1636.4 | 174.33 | 139.72 | 112.36 | 53.23 | 279.32 |
| **2019-02-27** | 155.25 | 1116.05 | 1641.09 | 174.87 | 139.17 | 112.17 | 53.24 | 279.2 |
| **2019-02-28** | 154.26 | 1119.92 | 1639.83 | 173.15 | 138.13 | 112.03 | 52.96 | 278.68 |
| **2019-03-01** | 156.45 | 1140.99 | 1671.73 | 174.97 | 139.2 | 112.53 | 53.3 | 280.42 |

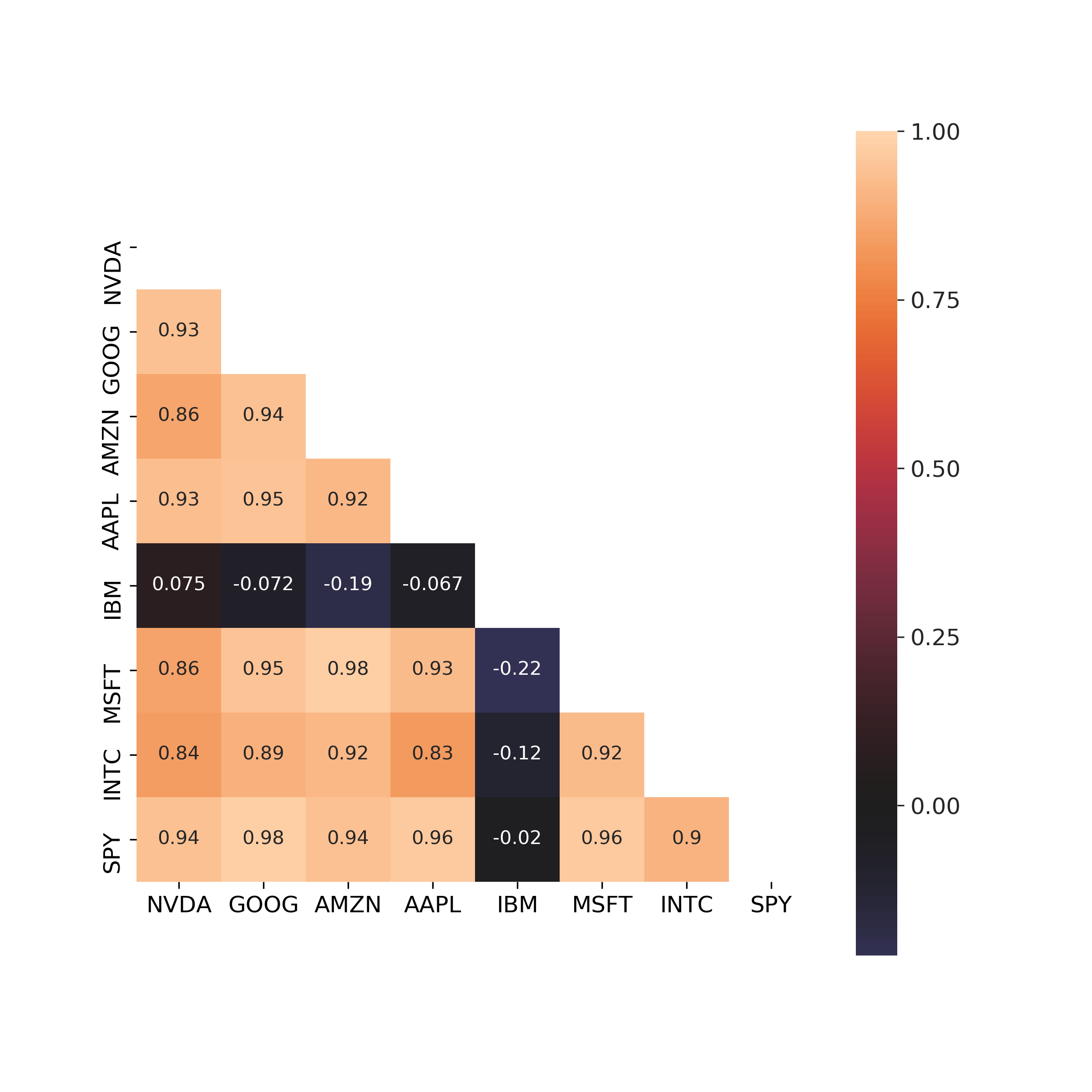
With the daily-adjusted close prices with that of the S&P 500 stock market index after removing days where the market did not report in the Yahoo data feed, **Table 2** describes the statistics of the close prices, which can be compared to the S&P 500 index. Amazon is the highest in overall values and variations.

**Table 2. Summary statistics of each stock portfolio compared to S&P 500**

|  | NVDA | GOOG | AMZN | AAPL | IBM | MSFT | INTC | SPY |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **mean** | 150.75 | 946.12 | 1159.20 | 148.18 | 140.76 | 76.95 | 39.59 | 240.46 |
| **std** | 75.25 | 159.32 | 416.62 | 36.94 | 10.88 | 21.05 | 7.85 | 29.31 |
| **min** | 31.25 | 668.26 | 552.08 | 86.31 | 106.33 | 45.77 | 27.51 | 186.70 |
| **25%** | 91.41 | 786.14 | 789.80 | 111.21 | 135.34 | 57.99 | 33.36 | 210.98 |
| **50%** | 151.44 | 952.77 | 991.60 | 151.81 | 141.18 | 71.62 | 35.30 | 240.22 |
| **75%** | 221.11 | 1079.23 | 1581.51 | 172.15 | 146.07 | 97.11 | 46.97 | 267.17 |
| **max** | 288.77 | 1268.33 | 2039.51 | 230.28 | 166.96 | 114.62 | 56.03 | 290.56 |

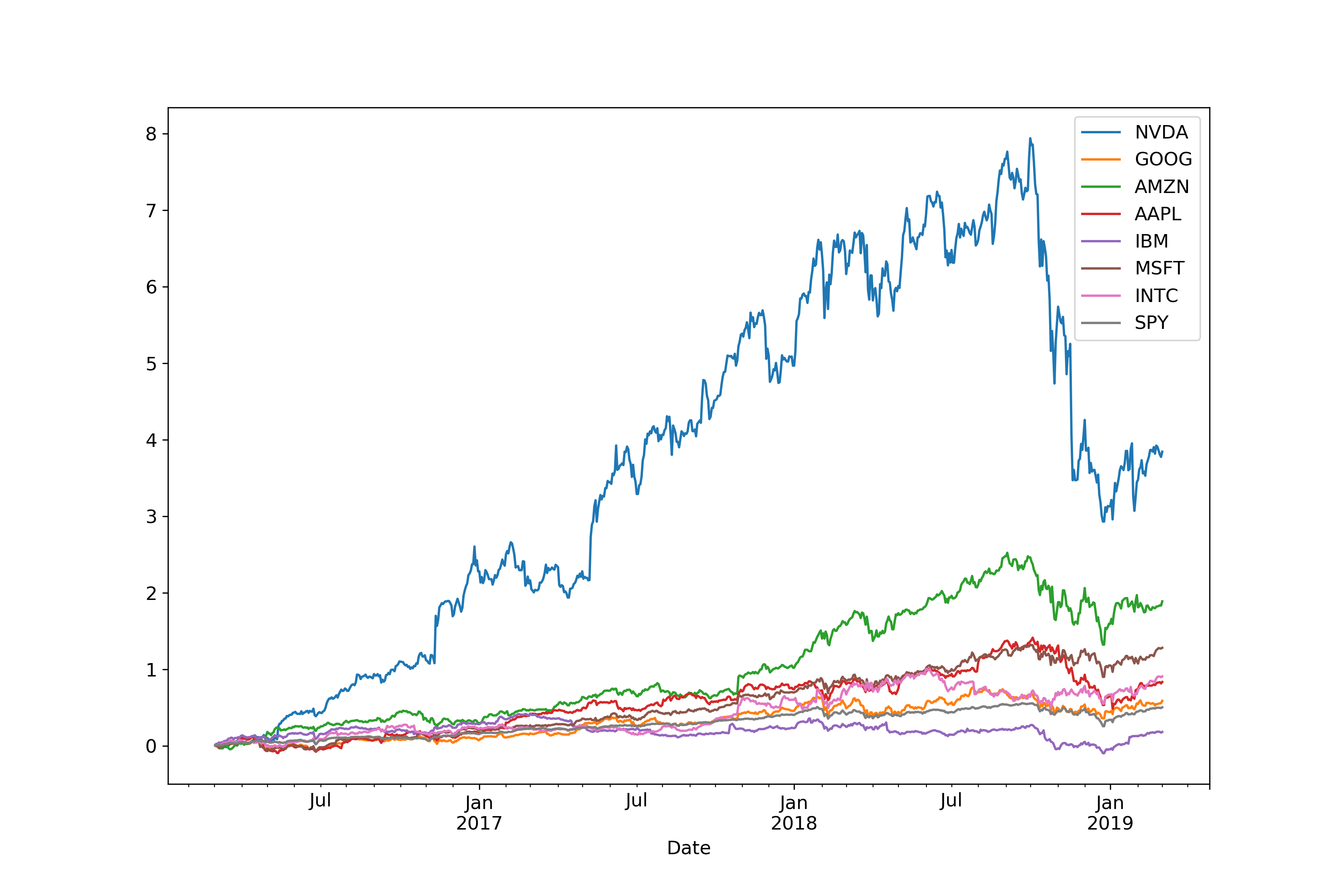
In addition, we can examine the correlations between the stock values, which can be seen in **Figure 1**. This reveals that almost all stocks, only except IBM, in this portfolio are highly correlated with S&P 500 index and the correlation efficiencies are around 90%. The IBM stock performance behaves differently when compared with other stocks in this portfolio.

**Figure 1. Heat map of correlations between the stocks**



Transforming the stock values to business day frequencies, we can then display the cumulative returns over the frequency of business days. This percentage change of the stock value from 03/01/2016 to 03/01/2019 has been displayed in **Figure 2**, revealing AMZN, MSFT, INTC stocks in our Technology portfolio significantly outpace the market. In addition, NVDA and IBM have been falling behind the market since last October. Specifically, NVDA stock had plunged by 50% since 10/2018 as trigger by the worried trade war between China and the US and the lack of demand [3].

For establishing momentum strategy, we have to calculate momentum weights and holding period returns. We calculate the momentum for each stock in the portfolio by calculating the percentage change over the frequency of business days defined by the lookback period. This lookback period is established at the beginning of the dataset. For instance, if we use a 10-day lookback period and a timeframe beginning on 03/01/2018, 03/15/2018 would be the first valid business date in which we could calculate momentum or percentage change. These percentages are ranked in ascending order, demeaned and then standardized to obtain a tabular output of portfolio weights for each relevant business day based on a given lookback period. These weights follow momentum investing guidelines: stocks with the greatest momentum in the portfolio are weighted the heaviest.

**Figure 2. Plot of the percentage change of stock values compared to S&P 500**

For each lookback period, an accompanying holding period is also defined, and cumulative returns are calculated for each stock in the portfolio based on this holding period:

Where the cumulative return for a given stock in the portfolio is . These returns are aggregated into holding period length bins, timestamped left-inclusive. For instance, if we held a portfolio position (mix of stocks) for 10 business days, portfolio returns would be aggregated in 10 business day increments with cumulative returns for each stock.

Given our momentum investing strategy of buying on uptrends and selling off on downtrends that we previously defined, these returns need to be weighted. This is where our lookback period has direct effect on the size of each investment in our portfolio. Our lookback period is responsible for creating the momentum weights of our portfolio, which will be multiplied by the returns for each of the relevant stocks in the portfolio. However, given our returns are aggregated by the holding period chosen, the holding period also determines the aggregation or resampling method for the portfolio weights to ensure dates are aligned properly. Returns and weights are aligned based on the holding period timeframe of 10 business days. As previously stated, stocks are ranked in ascending order based on their given momentum calculations. These ranks are then standardized in order to create weights for the portfolio.

We multiply the returns and weights for each holding period over our timeframe of 03/01/2016 to 03/01/2019, sum along the rows to get portfolio returns for each holding period, and then calculate a simplified Sharpe ratio by taking the mean of the returns divided by the standard deviation of the returns to score the quality of a given lookback, holding period combination. In addition, we set up a back-testing function to calculate the portfolio, iterating over many different lookback / holding period combinations. Here, we use Sharpe ratio to examine a range of 10 to 360 periods for both lookback and holding periods.

## **Results**

The Sharpe ratios for each combination of lookback and holding period are used to rank momentum investing strategies. The parameters for lookback and holding periods with 10 day increments up to last 360 days were iterated over to find the optimal range of combinations for the two parameters.

**Sharpe ratios for each combination of lookback and holding period**

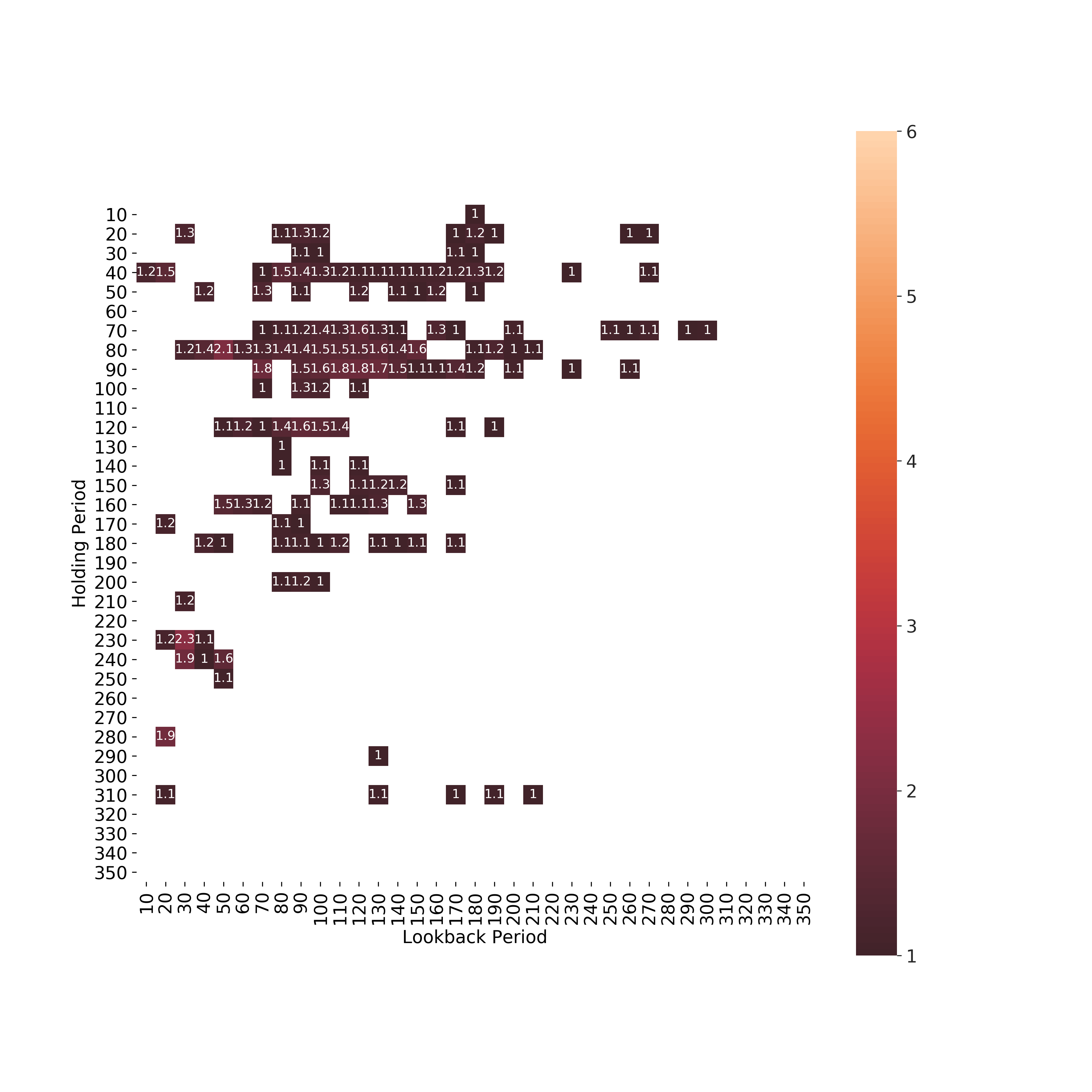
The test results summarized in heat map visualization (**Figure 3**) highlight higher Sharpe ratios in brighter colors. As we mentioned in background section, Sharpe Ratio greater than 1 is considered acceptable to good by investor. A ratio greater than 2 is rated very good, and a ratio of 3 or higher is considered excellent. In **Figure 3**, only the Sharpe ratio >1 value is shown to find the optimal clusters in different lookback and holding period combinations.

The parameter grids in **Figure 3** reveals the good returns for the US technology portfolio consistently occur in look back period during 70 to 180 days, and the holding period during 40 to 90 days. For example, the 120-day lookback period has best holding periods from 60 to 90 days with Sharpe ratio of 1.5 to 1.8. More than 180-day holding periods are not recommended though there are special cases that the Sharpe ratios for look back period at 30 days and holding period at 230 days can give Sharpe ratio value at 2.3. In addition, more than 180-day lookback period could not find favorable holding period.

Our Momentum investment strategies define short-term momentum from 1 month or less, intermediate term from 3 to 12 months, and long-term typically more than 1 year of a lookback period. Given our US technology portfolio, lookback periods between 2 and 6 months tend to perform better. These results favor overall intermediate term strategy.

In order to confirm intermediate term investment strategy, perform best in our US technology portfolio, we use a Kruskal-Wallis test to compare short term and intermediate term since the long-term strategy is obviously not the option. Statistical results indicate a significant difference (p<0.001) between these two groups either with strict definition of intermediate-term (90 to 360 days) or shift and narrow down to near intermediate-term (30 to 180 days).

**Figure 3: Heatmap of Sharpe ratios for different lookback and holding period combinations**

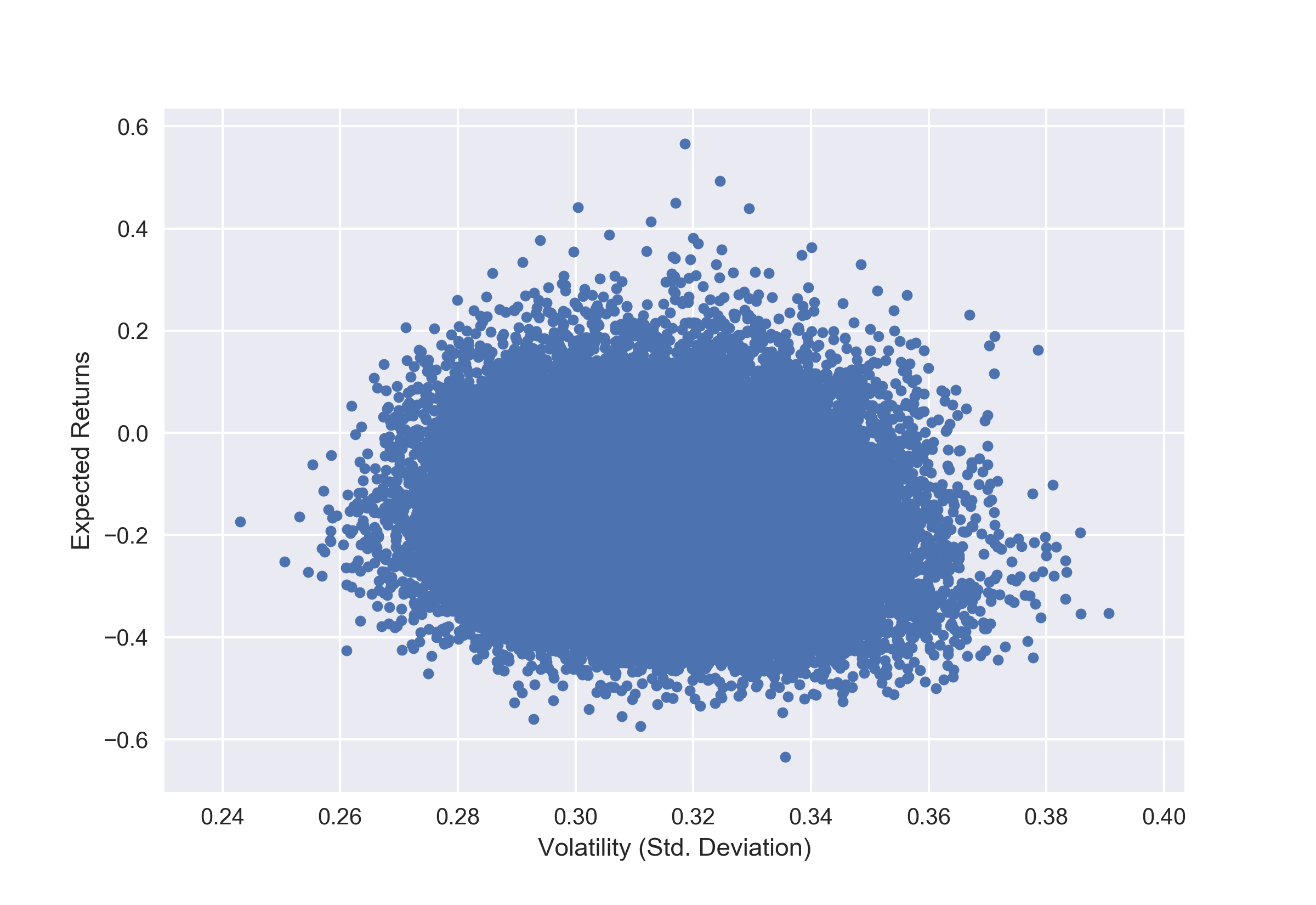
****

This US technology portfolio’s performance is also compared with the S&P 500 index with a Sharpe ratio of 1.4566 during the same period, indicating intermediate-term investment outperform S&P 500 in 12-month period.

**Volatility *vs* return of in US technology portfolio**

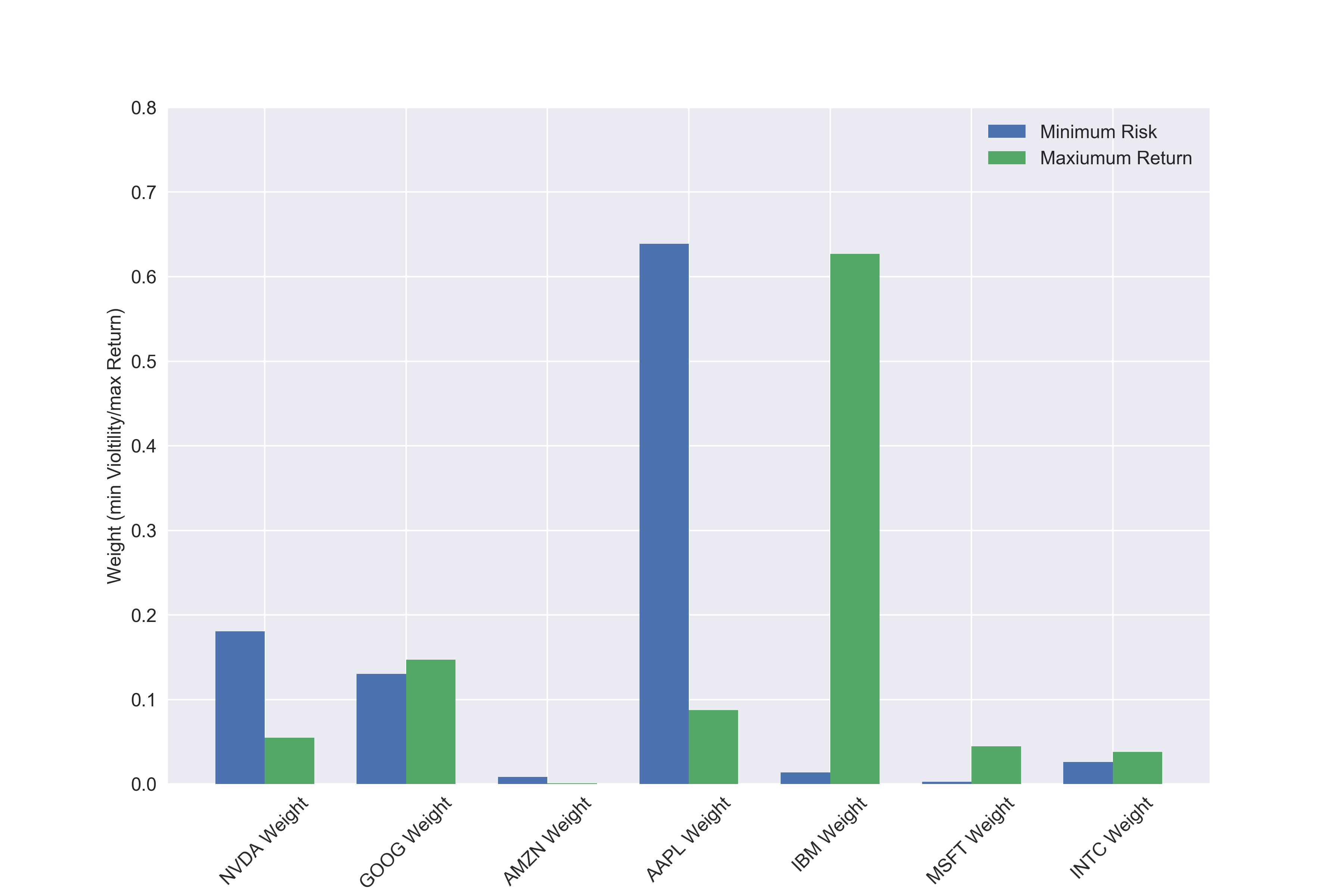
The higher the volatility, the risker the security. The volatility can either be measure by using the standard deviation or variance between returns form the same security or market index. As shown in **Figure 4**, the volatility is between 26% to 36%, with expected return between 40% loss to 20% gain in this period.

**Figure 4** is to show the volatility of a simulated portfolio vs the return in US technology portfolio. The left side of the cluster is semi arc shaped, this arc is called the efficient frontier. This frontier is used to determine the maximum return one can achieve for a certain, ideally lowest, level of risk. This would be used to select the optimal portfolio determined by your desired return and risk levels. The shape of the cluster is formed due to the tradeoff between risk and return. As on increases the riskiness of their portfolio they should expected to see a larger return as a reward for the extra risk they have taken. In theory one should always be selecting a portfolio on the outer edge of the cluster. This is because the outer edge portfolios are the ones that have the highest expected return for a certain level of risk.

**Figure 4. Efficient frontier of simulated portfolios**

**Stock weights for minimum risk and maximum return**

The expected return of a portfolio is calculated from the expected return and weight of each asset in a portfolio. **Figure 5** displays the weights for each stock in the minimum risk and maximum return portfolios. Three stocks are the drivers behind the return vs risk trade off. IBM is very heavily weighted in the maximum return portfolio, making up around 60% of the portfolio. However, in the minimum risk portfolio IBM has a weighting of less than 5%. This shows that they provide good returns but are extremely volatile and thus a very risky investment. The opposite of these two stocks is APPL. This stock makes up nearly 65% of the minimum risk portfolio buts only 10% of the maximum return portfolio. This gives evidence that the price of this stock remains relatively steady and does not produce very large returns.

**Figure 5. Weights for lowest volatility and highest return portfolios**

## **Conclusions**

For better customer investment protection, the financial advisors have the obligations to discuss the investment prospects before implementing any transactions. The chosen portfolio should be prepared with introducing transparent needs and specific modifications to meet customer’s best concerns and interests.

Momentum investment strategy can offer the framework to allocate funds with selected stocks, minimizing the risks and maximizing the returns. The efficient frontier is a very useful tool in determining how to weight a stock portfolio. One key factor is the portfolio team’s decision on risk vs return trade off. This will have a huge impact on which portfolio, and in turn, weights they decide to choose. As seen above portfolios with different risk vs return tradeoffs have drastically different weights when it comes to the stocks in them. Once the level of risk is determined the efficient frontier can be used to select the portfolio that will generate the maximum expected return for such level of risk.

In our US technology stock portfolio, the intermediate term momentum investment strategy is the best option when compared with short-term and long-term investment strategies. This strategy can minimize risk and maximize adjusted returns for high growth. Additionally, it would be even beneficial when adjusting the lookback period to be from 30 to 180 days. In the future study, we will continue to optimize the momentum portfolio construction and implement other trading strategies for stock investments.

## **Future Work**

Using Signal Frontier Analysis, it is possible to analyze a basket of securities and find the optimal holding and lookback periods, which yields the highest Sharpe Ratio. This time series stock market data analysis can be very useful to predict prices, analyze trends, and understand the value of a company over time. Analyses such as this one can provide insight into risk, tradeoff, and overall worth of companies.

Optimize cross-sectional momentum portfolio construction and a grid of model parameterizations. This analysis can be expanded upon and used in the future for the stocks presented in this report and for others. Additional stocks can be evaluated to determine the weight portfolio for various stock options. The time frame used in the analysis of risk and returns can also be adjusted for future models.

Backtesting [4] is the process of applying a trading strategy or analytical method to historical data to see how accurately the strategy or method would have predicted actual results. It is a common practice to test various algorithms, portfolios, or used as a comparison to study a portfolio's overall risk-return characteristics when a new asset or class has been added to it. It can be used to help a portfolio manager determine whether their additional allocation of assets has improved the risk-return dynamics.

In the future study, we will optimize momentum portfolio construction and implement the backtesting trading strategy to minimize investment risks and maximize the expected returns.

**Ethical considerations**

Working with stock market data has many ethical considerations due to the potential monetary gain and implications it holds. The use of this data and this analysis should only be applied to ethical stock market engagement or personal knowledge. The use of this data or analysis for short selling, “pump and dumps,” or insider trading may not align with the ethical values that should be maintained by data scientists. Short selling refers to the selling of stocks that the seller does not own. After this stock is sold at a high price, the seller will then buy back the stock at a low price. This strategy has negative impacts on the market and could even lead to market crashes. The second unethical activity is when a user engages in “pump and dump” schemes. This is where inaccurate information is “pumped” into the public to artificially inflate the value of a stock. When the value of the stock is high due to the hype, sellers will then sell all of their shares. Other users are then left with stocks that are not as valuable as they were told. Finally, users should avoid participating in insider trading. This is when stock transactions occur based on insider knowledge about a company’s future [5]. However, this analysis details ways that stock market data can be ethically analyzed to make informed decisions in the stock market.

**References**

1. McKinney, W. (2013). *Python for Data Analysis*. Sebastopol, CA: O’Reilly Media (Chapter 11).
2. The untapped potential of time series data mining. (2016, May 21). Retrieved February 25, 2019, from <http://analytics-magazine.org/the-untapped-potential-of-time-series-data-mining/>
3. Nvidia’s stock faces even steeper declines. <https://seekingalpha.com/article/4224502-nvidias-stock-faces-even-steeper-declines>
4. What is backtesting a trading strategy? [www.quantinsti.com/blog/backtesting](http://www.quantinsti.com/blog/backtesting)
5. Highland, J. (2017, November 21). Ethical Decisions in the Stock Market. Retrieved March 2, 2019, from <https://smallbusiness.chron.com/ethical-decisions-stock-market-1606.html>

## **Appendix - Python code**

## Python codes for Case study 4 (Unit 8): Signal Frontier Analysis

import datetime as dt

import pandas as pd

from pandas import Series, DataFrame

from pandas\_datareader import data as web

import pandas\_datareader as pdr

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

plt.rc('figure', figsize=(12, 6))

import warnings

warnings.filterwarnings("ignore", category=RuntimeWarning)

## set date range

start\_dt = dt.datetime(2016, 3, 1)

end\_dt = dt.datetime(2019, 3, 1)

## use S&P 500 Index (SPY) to check for market correlations, comparisons for returns

def get\_portfolio():

port = pd.DataFrame()

names = ['NVDA', 'GOOG', 'AMZN', 'AAPL', 'IBM', 'MSFT', 'INTC', 'SPY']

for stock in names:

while True:

try:

port[stock] = web.get\_data\_yahoo(stock, start\_dt, end\_dt)['Adj Close']

break

except:

print('Unable to read stock: {0}, trying again'.format(stock))

return port

px = get\_portfolio()

px = px.loc[~(px==0).all(axis=1)] # strip out days with no trading data for all stocks

## daily adjusted close prices for pizza portfolio + S&P

lastWeekClose = px.tail()

lastWeekClose.to\_excel('lastWeekClose.xlsx')

## get standard statistics for portfolio compared to S&P

stat = px.describe()

stat.to\_excel('stat.xlsx')

## EDA for linear relationships, including SPY

corr = px.corr()

mask = np.zeros\_like(corr)

mask[np.triu\_indices\_from(mask)] = True

f, ax = plt.subplots(figsize=(8, 8))

with sns.axes\_style("darkgrid"):

ax = sns.heatmap(corr, mask=mask, vmax=1, center=0, square=True, annot=True)

plt.savefig('corr.png', dpi=300)

## transform to business day frequency and calculate percentage change

## show cumulative returns over frequency of business days

px = px.asfreq('B').fillna(method='pad') # pad == ffill

rets = px.pct\_change()

((1+rets).cumprod()-1).plot()

plt.rc('figure', figsize=(12, 6))

plt.rc('xtick', labelsize=14)

plt.rc('ytick', labelsize=14)

plt.rc('axes', labelsize=14)

plt.legend(fontsize=12)

plt.savefig('cumulativeReturn.png', dpi=300)

## compute momentum over a lookback and rank in ASCENDING order and standardize to get portfolio weights

def calc\_mom(price, lookback, lag):

'''Calculates pct change based on user input shift and lookback period, ranks, then standardizes ranks'''

mom\_ret = price.shift(lag).pct\_change(lookback)

ranks = mom\_ret.rank(axis=1, ascending=True)

demeaned = ranks.subtract(ranks.mean(axis=1), axis=0)

return demeaned.divide(demeaned.std(axis=1), axis=0)

compound = lambda x : (1+x).prod()-1 # cumulative returns

daily\_sr = lambda x : x.mean() / x.std() # daily sharpe ratio

## Compute portfolio weights using rank-standardized momentum portfolio

def strat\_sr(prices, lb, hold):

freq = '%dB' % hold # hold for how many business day

port = calc\_mom(prices, lb, lag=1)

daily\_rets = prices.pct\_change()

# Compute portfolio returns: aggregate, multiply and take Sharpe ratio

port = port.shift(1).resample(freq).first()

returns = daily\_rets.resample(freq).apply(compound)

port\_rets = (port \* returns).sum(axis=1)

return daily\_sr(port\_rets) \* np.sqrt(252 / hold)

## sharpe ratio comparison setup

## iterate over combinations of lookback and holding periods from 10 days to 360 days

from collections import defaultdict

lookbacks = range(10, 360, 10)

holdings = range(10, 360, 10)

dd = defaultdict(dict)

for lb in lookbacks:

for hold in holdings:

# dont include S&P in analysis of fast food portfolio, index out

dd[lb][hold] = strat\_sr(px.iloc[:, :-1], lb, hold)

ddf = pd.DataFrame(dd)

ddf.index.name = 'Holding Period'

ddf.columns.name = 'Lookback Period'

## set up kruskal wallis non-parametric test for lookback periods

## formalize if differences between short and intermediate term strategies

## considering 90 business days to 180 business days intermediate (grouped)

from scipy.stats import mstats

short = ddf.iloc[:,:4].values.reshape(-1) # flatten vals to 1D array

inter = ddf.iloc[:,4:].values.reshape(-1)

H, pval = mstats.kruskalwallis(short, inter) # non-parametric ANOVA

print('Median of short term, intermediate term Sharpe Ratios for Lookback Period')

print('\nShort:', round(np.median(short),3), '\nIntermediate:',round(np.median(inter),3))

print('\nNon-parametric ANOVA results for short term vs intermediate term groupings')

print(H, pval)

## considering strict definition of short (1 month) and intermediate (6-12 months) periods

## adding a "between" category for analysis purposes

short = ddf.loc[:,:90].values.reshape(-1) # flatten vals to 1D array

between = ddf.loc[:,90:180].values.reshape(-1)

inter = ddf.loc[:,180:].values.reshape(-1)

H, pval = mstats.kruskalwallis(between, inter) # non-parametric ANOVA

print('\n\nStrict Definition: Median of short term, between and intermediate term Sharpe ratios for Lookback Period')

print('Short:', round(np.median(short),3),

'\nBetween:', round(np.median(between),3),

'\nInter:', round(np.median(inter),3))

print('\nNon-parametric ANOVA results for: "between" vs intermediate term groupings')

print('H:', H, 'p:', pval)

## set up kruskal wallis non-parametric test for holding periods

## formalize if differences between short and intermediate term strategies

# considering 30 business days to 180 business days intermediate (grouped)

short = ddf.iloc[:4,:].values.reshape(-1) # flatten vals to 1D array

inter = ddf.iloc[4:,:].values.reshape(-1)

H, pval = mstats.kruskalwallis(short, inter) # non-parametric ANOVA

print('Median of short term, intermediate term Sharpe Ratios for Holding Period')

print('\nShort:', round(np.median(short),3), '\nIntermediate:',round(np.median(inter),3))

print('\nNon-parametric ANOVA results for short term vs intermediate term groupings')

print(H, pval)

# considering strict definition of short (1 month) and intermediate (6-12 months) periods

# adding a "between" category for analysis purposes

short = ddf.loc[:30,:].values.reshape(-1) # flatten vals to 1D array

between = ddf.loc[30:180,:].values.reshape(-1)

inter = ddf.loc[180:,:].values.reshape(-1)

H, pval = mstats.kruskalwallis(between, inter) # non-parametric ANOVA

print('\n\nStrict Definition: Median of short term, between and intermediate term Sharpe ratios for Holding Period')

print('Short:', round(np.median(short),3),

'\nBetween:', round(np.median(between),3),

'\nInter:', round(np.median(inter),3))

print('\nNon-parametric ANOVA results for: "between" and intermediate term groupings')

print('H:', H, 'p:', pval)

# heatmap from book

def heatmap(df, cmap = plt.cm.gray\_r):

fig = plt.figure(figsize=(10,10))

ax = fig.add\_subplot(111)

axim = ax.imshow(df.values, cmap = cmap, interpolation='nearest')

ax.set\_xlabel(df.columns.name)

ax.set\_xticks(np.arange(len(df.columns)))

ax.set\_xticklabels(list(df.columns))

ax.set\_ylabel(df.index.name)

ax.set\_yticks(np.arange(len(df.index)))

ax.set\_yticklabels(list(df.index))

plt.colorbar(axim)

heatmap(ddf)

plt.savefig('heatmap\_book.png', dpi=300)

## better heatmap, clusters around intermediate term lookbacks

df = ddf[ddf>1] # show only "good" or above Sharpe ratios

f, ax = plt.subplots(figsize=(12, 12))

with sns.axes\_style("darkgrid"):

ax = sns.heatmap(df, mask=None, vmax=6, center=0, square=True, annot=True)

plt.rc('figure', figsize=(12, 8))

plt.rc('xtick', labelsize=12)

plt.rc('ytick', labelsize=12)

plt.rc('axes', labelsize=12)

plt.savefig('heatmap\_2.png', dpi=300)

## Calculate S&P cumulative return for same analysis period

((1+px['SPY'].pct\_change()).cumprod()-1).iloc[-1]

## Calculate portfolio cumulative return approximation for same analysis period

## w/equal weight (if we held from day 1 and did not trade momentum)

wts = [.14275, .14275, .14275, .14275, .14275, .14275, .14275]

(((1+px.iloc[:,:-1].pct\_change()\*wts).cumprod()-1).iloc[-1]).sum() # considering equal weighting

## Annualized Check (should be roughly similar to returns above)¶

# annualized returns for S&P 500

mean\_daily\_returns = px.iloc[:,-1].pct\_change().mean()

portfolio\_return = round(np.sum(mean\_daily\_returns) \* 370, 4)

print('S&P return ',portfolio\_return)

# annualized returns for our portfolio considering equal weights

wts = ([.125, .125, .125, .125, .125, .125, .125]) # 0.14275

mean\_daily\_returns = px.iloc[:,:-1].pct\_change().mean()

portfolio\_return = round(np.sum(mean\_daily\_returns \* wts) \* 360, 4)

print('Equal weighting return ', portfolio\_return)

## use Sharpe ratio to take good scenario from momentum investing results

print('Sharpe for one scenario of momentum: ',strat\_sr(px.iloc[:,:-1], 90, 100))

## get S&P sharpe for same periods to compare

SP\_holding = (px['SPY'].pct\_change()).resample('100B').apply(compound)

print('S&P 500 Sharpe: ',SP\_holding.mean()/SP\_holding.std()\*np.sqrt(252 / 100))

## Use VIX index

vix = pdr.DataReader('VIXCLS', 'fred').dropna().squeeze()

lower, upper = 16.5, 19.5

# Each term inside parentheses is [False, True, ...]

# Both terms must be True element-wise for a trigger to occur

blue = (vix < upper) & (vix.shift() >= upper)

yellow = (vix < lower) & (vix.shift() >= lower)

green = (vix > upper) & (vix.shift() <= upper)

red = (vix > lower) & (vix.shift() <= lower)

mapping = {1: 'blue', 2: 'yellow', 3: 'green', 4: 'red'}

indicator = pd.Series(np.where(blue, 1., np.where(yellow, 2.,

np.where(green, 3., np.where(red, 4., np.nan)))),

index=vix.index).ffill().map(mapping).dropna()

vix = vix.reindex(indicator.index)

plt.scatter(vix.index, vix, c=indicator, marker='\*')

plt.title('VIX regime')

plt.ylabel('VIX')

## efficient set of portfolios — Efficient Frontier”

## check with worst performance stocks ('NWL', 'TRUE', 'LOGM')

import quandl

# get adjusted closing prices of 3 selected companies with Quandl

quandl.ApiConfig.api\_key = 'MvzrXUa5F8WQX-6oSUH4'

selected = ['NWL', 'TRUE', 'LOGM']

data = quandl.get\_table('WIKI/PRICES', ticker = selected,

qopts = { 'columns': ['date', 'ticker', 'adj\_close'] },

date = { 'gte': '2018-1-1', 'lte': '2019-02-01' }, paginate=True)

# reorganise data pulled by setting date as index with

# columns of tickers and their corresponding adjusted prices

clean = data.set\_index('date')

table = clean.pivot(columns='ticker')

# calculate daily and annual returns of the stocks

returns\_daily = table.pct\_change()

returns\_annual = returns\_daily.mean() \* 250

# get daily and covariance of returns of the stock

cov\_daily = returns\_daily.cov()

cov\_annual = cov\_daily \* 250

# empty lists to store returns, volatility and weights of imiginary portfolios

port\_returns = []

port\_volatility = []

stock\_weights = []

# set the number of combinations for imaginary portfolios

num\_assets = len(selected)

num\_portfolios = 50000

# populate the empty lists with each portfolios returns,risk and weights

for single\_portfolio in range(num\_portfolios):

weights = np.random.random(num\_assets)

weights /= np.sum(weights)

returns = np.dot(weights, returns\_annual)

volatility = np.sqrt(np.dot(weights.T, np.dot(cov\_annual, weights)))

port\_returns.append(returns)

port\_volatility.append(volatility)

stock\_weights.append(weights)

# a dictionary for Returns and Risk values of each portfolio

portfolio = {'Returns': port\_returns,

'Volatility': port\_volatility}

# extend original dictionary to accomodate each ticker and weight in the portfolio

for counter,symbol in enumerate(selected):

portfolio[symbol+' Weight'] = [Weight[counter] for Weight in stock\_weights]

# make a nice dataframe of the extended dictionary

df = pd.DataFrame(portfolio)

# get better labels for desired arrangement of columns

column\_order = ['Returns', 'Volatility'] + [stock+' Weight' for stock in selected]

# reorder dataframe columns

df = df[column\_order]

# plot the efficient frontier with a scatter plot

plt.style.use('seaborn')

df.plot.scatter(x='Volatility', y='Returns', figsize=(10, 8), grid=True)

plt.xlabel('Volatility (Std. Deviation)')

plt.ylabel('Expected Returns')

plt.title('Efficient Frontier')

plt.show()

## check performance with stocks related to computer ('NVDA', 'GOOG', 'AMZN', 'AAPL', 'IBM', 'MSFT', 'INTC')

# get adjusted closing prices of 7 selected companies with Quandl

quandl.ApiConfig.api\_key = 'MvzrXUa5F8WQX-6oSUH4'

selected = ['NVDA', 'GOOG', 'AMZN', 'AAPL', 'IBM', 'MSFT', 'INTC']

data = quandl.get\_table('WIKI/PRICES', ticker = selected,

qopts = { 'columns': ['date', 'ticker', 'adj\_close'] },

date = { 'gte': '2018-3-1', 'lte': '2019-03-01' }, paginate=True)

# reorganise data pulled by setting date as index with

# columns of tickers and their corresponding adjusted prices

clean = data.set\_index('date')

table = clean.pivot(columns='ticker')

# calculate daily and annual returns of the stocks

returns\_daily = table.pct\_change()

returns\_annual = returns\_daily.mean() \* 250

# get daily and covariance of returns of the stock

cov\_daily = returns\_daily.cov()

cov\_annual = cov\_daily \* 250

# empty lists to store returns, volatility and weights of imiginary portfolios

port\_returns = []

port\_volatility = []

stock\_weights = []

# set the number of combinations for imaginary portfolios

num\_assets = len(selected)

num\_portfolios = 50000

# populate the empty lists with each portfolios returns,risk and weights

for single\_portfolio in range(num\_portfolios):

weights = np.random.random(num\_assets)

weights /= np.sum(weights)

returns = np.dot(weights, returns\_annual)

volatility = np.sqrt(np.dot(weights.T, np.dot(cov\_annual, weights)))

port\_returns.append(returns)

port\_volatility.append(volatility)

stock\_weights.append(weights)

# a dictionary for Returns and Risk values of each portfolio

portfolio = {'Returns': port\_returns,

'Volatility': port\_volatility}

# extend original dictionary to accomodate each ticker and weight in the portfolio

for counter,symbol in enumerate(selected):

portfolio[symbol+' Weight'] = [Weight[counter] for Weight in stock\_weights]

# make a nice dataframe of the extended dictionary

df = pd.DataFrame(portfolio)

# get better labels for desired arrangement of columns

column\_order = ['Returns', 'Volatility'] + [stock+' Weight' for stock in selected]

# reorder dataframe columns

df = df[column\_order]

# plot the efficient frontier with a scatter plot

plt.style.use('seaborn')

df.plot.scatter(x='Volatility', y='Returns', grid=True)

plt.xlabel('Volatility (Std. Deviation)')

plt.ylabel('Expected Returns')

# plt.title('Efficient Frontier')

#plt.show()

#plt.tick\_params(labelsize=12)

plt.rc('figure', figsize=(8, 6))

plt.rc('xtick', labelsize=14)

plt.rc('ytick', labelsize=14)

plt.rc('axes', labelsize=14)

plt.savefig('scatter.png', dpi=300)

vol\_min=df["Volatility"].min()

ret\_max=df["Returns"].max()

extremes=df.loc[(df['Volatility'] == df["Volatility"].min()) | (df["Returns"]==df["Returns"].max())]

extremes["Portfolio"]=["Min Volatility","Max Return"]

extremes

x\_labels=list(extremes.columns.values)[2:9]

x=np.arange(len(x\_labels))

width=0.35

min\_v\_y=list(extremes.iloc[0,2:9])

max\_r\_y=list(extremes.iloc[1,2:9])

ax=plt.subplot()

ax.bar(x,min\_v\_y,width,label="Minimum Risk")

ax.bar(x+width,max\_r\_y,width,label="Maxiumum Return")

plt.xticks(x+width/2, x\_labels, rotation=45)

plt.ylim(0,0.8)

plt.legend(loc="best")

# plt.title("Stock Weights By Portfolio")

# plt.xlabel("Stocks")

plt.ylabel("Weight (min Violtility/max Return)")

#plt.show()

plt.rc('figure', figsize=(12, 8))

plt.rc('xtick', labelsize=12)

plt.rc('ytick', labelsize=12)

plt.rc('axes', labelsize=12)

plt.legend(fontsize=12)

plt.savefig('weight.png', dpi=300)