Risk and Return Between Amazon and Facebook

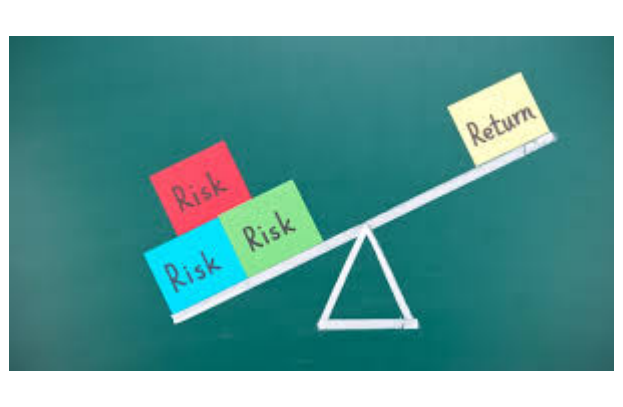


Table of Content

1. Conclusion & Summary of findings

Which investment should we make based on the two Sharpe ratios? Amazon had a Sharpe ratio twice as high as Facebook in 2016. This means that an investment in Amazon returned twice as much as an investment in the S&P 500 for each unit of risk assumed. In other words, the investment in Amazon would have been more appealing in risk-adjusted terms.

The difference between Amazon and Facebook was primarily driven by differences in return rather than risk. The risk of choosing Amazon over Facebook was only slightly higher (as measured by the standard deviation), so the higher Sharpe ratio for Amazon is primarily due to the higher average daily returns for Amazon.

When confronted with investment alternatives that offer both different returns and risks, the Sharpe Ratio aids decision-making by adjusting returns for differences in risk and allowing an investor to compare investment opportunities on equal terms, that is, on a 'apples-to-apples' basis.

2. Recommendation

Amazon had a Sharpe ratio twice as high as Facebook in 2016. This means that an investment in Amazon returned twice as much as an investment in the S&P 500 for each unit of risk assumed. In other words, the investment in Amazon would have been more appealing in risk-adjusted terms.

3. Introduction

An investment may make sense if we expect it to yield a higher return than it costs. However, returns are only part of the story because they are risky, with a variety of possible outcomes. How does one compare investments that may produce similar results on average but have varying levels of risk?</p>

Polarity Venture has seen and experienced the benefits of investing in stocks in order to generate high returns on investment. Polarity Ventures formed a team to force in stock investing; the team's leader tasked the analytical team with presenting companies in the S&P 500 to invest in. So far, we have chosen two companies that meet our criteria; now, we must dig deeper to find the company that will provide us with the best return, as we must choose a company.

3. Analysis Plan

Polarity Ventures wants to invest in one of the tech titans, Amazon or Facebook, by purchasing stock. The analytical team has been tasked with recommending the best option.

We will use William Sharpe a reward-to-variability ratio for our decision. We will be able to compare the expected returns for two investment opportunities and calculate the additional return per unit of risk Polarity Ventures could obtain by selecting one over the other (Sharpe Ratio). It specifically examines the difference in returns between two investments and compares the average difference to the standard deviation (as a risk measure) of this difference. A higher Sharpe ratio indicates that the reward for a given amount of risk will be greater. It is common to compare a specific opportunity to a benchmark that represents an entire investment category.

The Sharpe ratio is typically calculated for a portfolio, with the risk-free interest rate serving as the benchmark. Instead of a portfolio, we will use stocks. We will also use a stock index as a benchmark rather than the risk-free interest rate because both are available at daily frequencies and we will not have to convert interest rates from annual to daily frequency. Keep in mind that you would perform the same calculation with portfolio returns and your preferred risk-free rate, such as the 3-month Treasury Bill Rate.

5. Data Description

We will use the S&P 500 as a benchmark, which measures the performance of the 500 largest stocks in the United States. When we use a stock index instead of the risk-free rate, the result is known as the Information Ratio, and it is used to benchmark the return on active portfolio management because it tells you how much more return your portfolio manager earned for a given unit of risk compared to simply putting your money into a low-cost index fund.

Let's take a look at the data we'll be using to calculate the Sharpe Ratio, S&P 500 (stock data from Amazon and Facebook), and S&P 500 prices to see how many observations and variables we'll have at our disposal for risk calculation.

In [2]:
```
# Importing required modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Settings to produce nice plots in a Jupyter notebook
plt.style.use('fivethirtyeight')
%matplotlib inline

# Reading in the data
stock_data = pd.read_csv('stock_data.csv',
                         parse_dates = ['Date'], index_col = 'Date').dropna()
benchmark_data = pd.read_csv('benchmark_data.csv',
                         parse_dates = ['Date'], index_col = 'Date').dropna()


# Display summary for stock_data
print('Stocks\n')
stock_data.info()
print(stock_data.head())

# Display summary for benchmark_data
print('\nBenchmarks\n')
benchmark_data.info()
benchmark_data.head()
```

```
Stocks

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 252 entries, 2016-01-04 to 2016-12-30
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Amazon    252 non-null    float64
 1   Facebook  252 non-null    float64
dtypes: float64(2)
memory usage: 5.9 KB
                Amazon    Facebook
Date
2016-01-04  636.989990  102.220001
2016-01-05  633.789978  102.730003
2016-01-06  632.650024  102.970001
2016-01-07  607.940002   97.919998
2016-01-08  607.049988   97.330002

Benchmarks

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 252 entries, 2016-01-04 to 2016-12-30
Data columns (total 1 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   S&P 500  252 non-null    float64
dtypes: float64(1)
memory usage: 3.9 KB
```

Out[2]:

	S&P 500
Date	
2016-01-04	2012.66
2016-01-05	2016.71
2016-01-06	1990.26
2016-01-07	1943.09
2016-01-08	1922.03

a. Summary of daily prices for Amazon and Facebook

Before comparing an investment in Facebook or Amazon to the index of the 500 largest companies in the United States, let's visualize the data so we know what we're dealing with. To see how each of these companies has performed thus far.

In [17]:
```
# visualize the stock_data
stock_data.plot(subplots=True, title = 'Stock Data');

# summarize the stock_data
stock_data.describe()
```

Out[17]:

	Amazon	Facebook
count	252.000000	252.000000
mean	699.523135	117.035873
std	92.362312	8.899858
min	482.070007	94.160004
25%	606.929993	112.202499
50%	727.875000	117.765000
75%	767.882492	123.902502
max	844.359985	133.279999

Both companies have shown consistent growth, with the exception of a slight drop in stock prices in November 2016.

b. Summarize daily values for the S&P 500

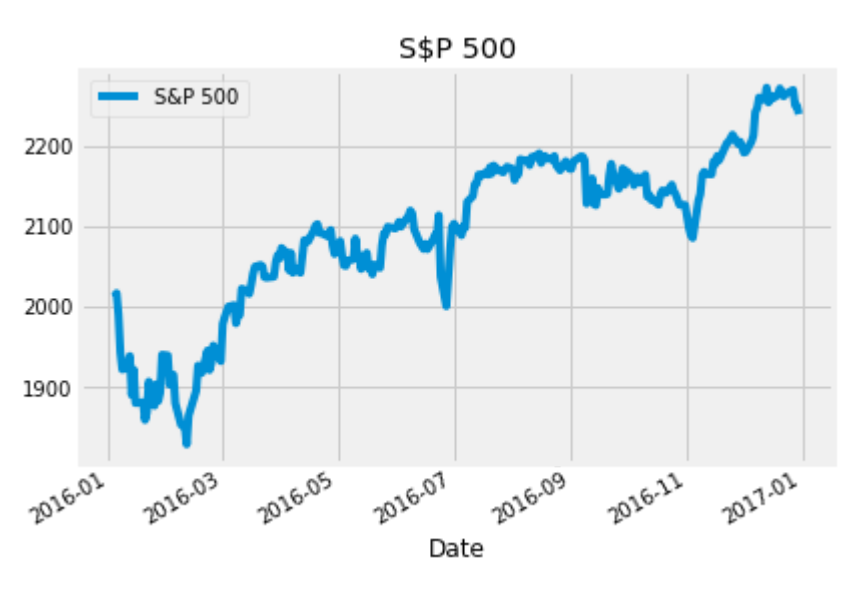
Let's also take a closer look at the S&P 500, our benchmark.

In [18]:
```
# plot the benchmark_data
benchmark_data.plot(title = 'S$P 500');

# summarize the benchmark_data
benchmark_data.describe()
```

Out[18]:

	S&P 500
count	252.000000
mean	2094.651310
std	101.427615
min	1829.080000
25%	2047.060000
50%	2104.105000
75%	2169.075000
max	2271.720000



The S&P 500 index has been steadily rising in 2016, with a 25% increase.

c. The inputs for the Sharpe Ratio: Starting with Daily Stock Returns

The Sharpe Ratio considers the difference in returns between two investment opportunities.

Our data, on the other hand, show the historical value of each investment rather than the return. To compute the return, we must first determine the percentage change in value from one day to the next. We'll also look at the summary statistics because they'll be used as inputs in the Sharpe Ratio calculation. Can you already guess the outcome?

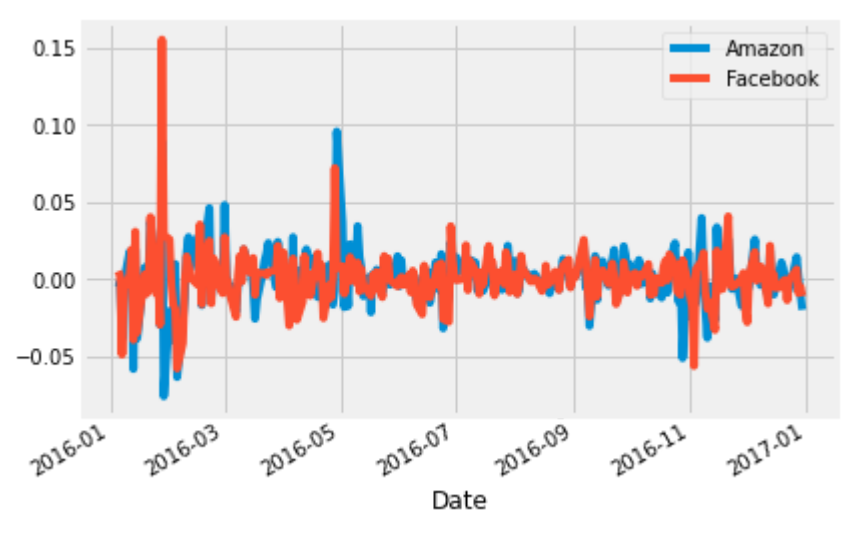
In [19]:
```
# calculate daily stock_data returns
stock_returns = stock_data.pct_change()

# plot the daily returns
stock_returns.plot()

# summarize the daily returns
stock_returns.describe()
```

Out[19]:

	Amazon	Facebook
count	251.000000	251.000000
mean	0.000818	0.000626
std	0.018383	0.017840
min	-0.076100	-0.058105
25%	-0.007211	-0.007220
50%	0.000857	0.000879
75%	0.009224	0.008108
max	0.095664	0.155214



According to our graph, Amazon and Facebook appear to be moving (changing) at nearly the same rate. to delve deeper in order to select the most suitable company for our portfolio.

In [20]:
```
stock_data.pct_change()
```

Out[20]:

	Amazon	Facebook
Date		
2016-01-04	NaN	NaN
2016-01-05	-0.005024	0.004989
2016-01-06	-0.001799	0.002336
2016-01-07	-0.039058	-0.049043
2016-01-08	-0.001464	-0.006025
...	...	...
2016-12-23	-0.007503	-0.001107
2016-12-27	0.014213	0.006310
2016-12-28	0.000946	-0.009237
2016-12-29	-0.009040	-0.004875
2016-12-30	-0.019970	-0.011173

252 rows × 2 columns

d. Daily S&P 500 returns

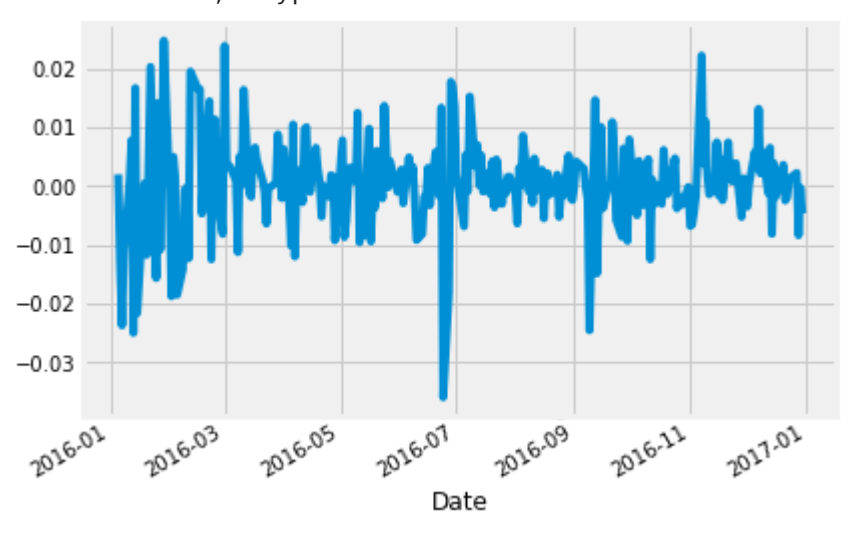
For the S&P 500, calculating daily returns works just the same way, we just need to make sure we select it as a `Series` using single brackets `[]` and not as a `DataFrame` to facilitate the calculations in the next step.

In [21]:
```
# calculate daily benchmark_data returns
sp_returns = benchmark_data['S&P 500'].pct_change()

# plot the daily returns
sp_returns.plot();

# summarize the daily returns
sp_returns.describe()
```

Out[21]:
```
count    251.000000
mean       0.000458
std        0.008205
min       -0.035920
25%       -0.002949
50%        0.000205
75%        0.004497
max        0.024760
Name: S&P 500, dtype: float64
```



e. Calculating Excess Returns for Amazon and Facebook vs. S&P 500

Next, we must compute the relative performance of stocks in comparison to the S&P 500 benchmark. For each day, this is calculated as the difference in returns between code>stock returns and sp returns.

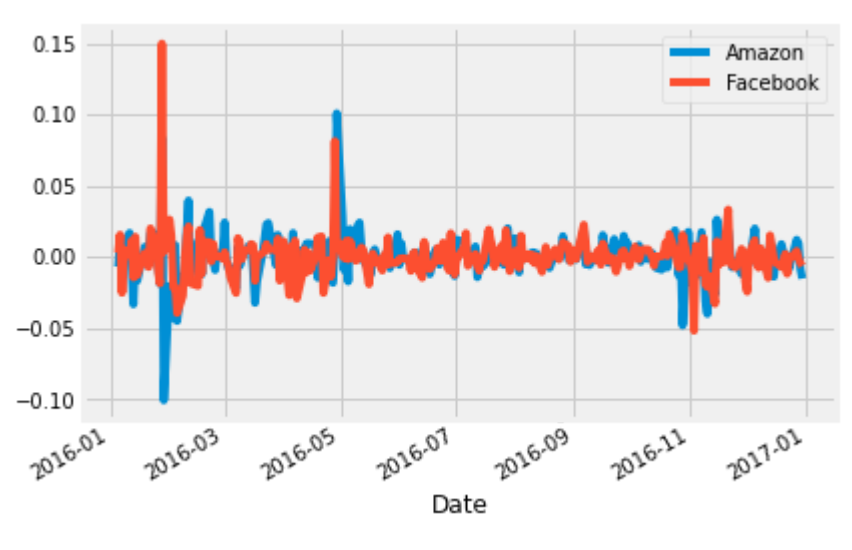
In [22]:
```
# calculate the difference in daily returns
excess_returns = stock_returns.sub(sp_returns,axis = 0)

# plot the excess_returns
excess_returns.plot()

# summarize the excess_returns
excess_returns.describe()
```

Out[22]:

	Amazon	Facebook
count	251.000000	251.000000
mean	0.000360	0.000168
std	0.016126	0.015439
min	-0.100860	-0.051958
25%	-0.006229	-0.005663
50%	0.000698	-0.000454
75%	0.007351	0.005814
max	0.100728	0.149686



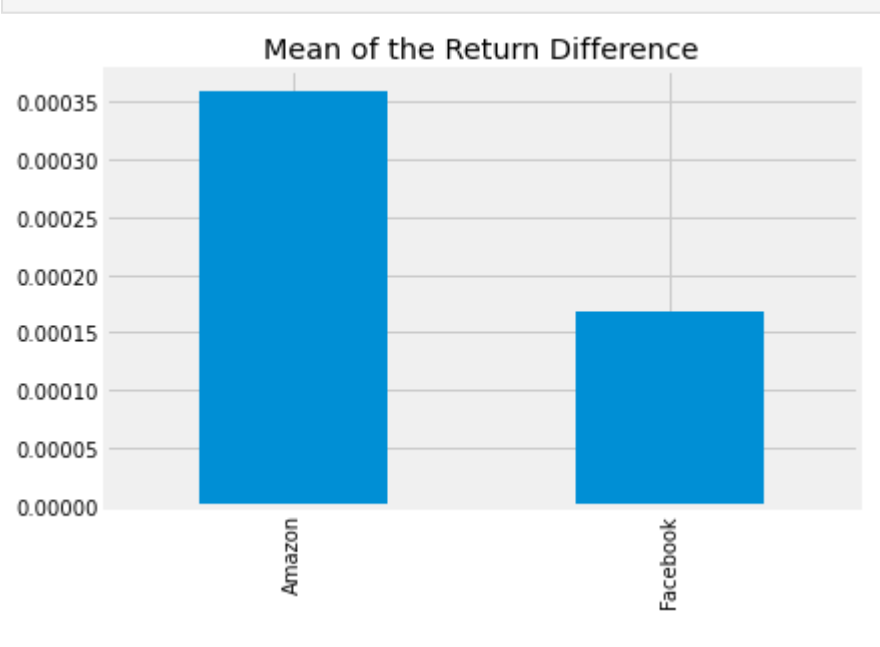
Looking at the graph, the daily returns of both companies appear to be on the same level, which does not provide us with much information to work with when selecting the stock with the highest return. We will need to calculate the risk of investing in the companies in order to make an informed decision.

f. The Sharpe Ratio, Step 1: The Average Difference in Daily Returns Stocks vs S&P 500

We can now begin computing the Sharpe Ratio. We must first compute the average of the excess returns. This indicates how much more or less the investment earns per day than the benchmark.

In [24]:
```
# calculate the mean of excess_returns
avg_excess_return = excess_returns.mean()

# plot avg_excess_returns
avg_excess_return.plot.bar(title = 'Mean of the Return Difference');
```



The results of calculating daily returns show that Amazon returns are nearly double those of Facebook. Simply put, Amazon returns are nearly double those of Facebook on a daily basis.

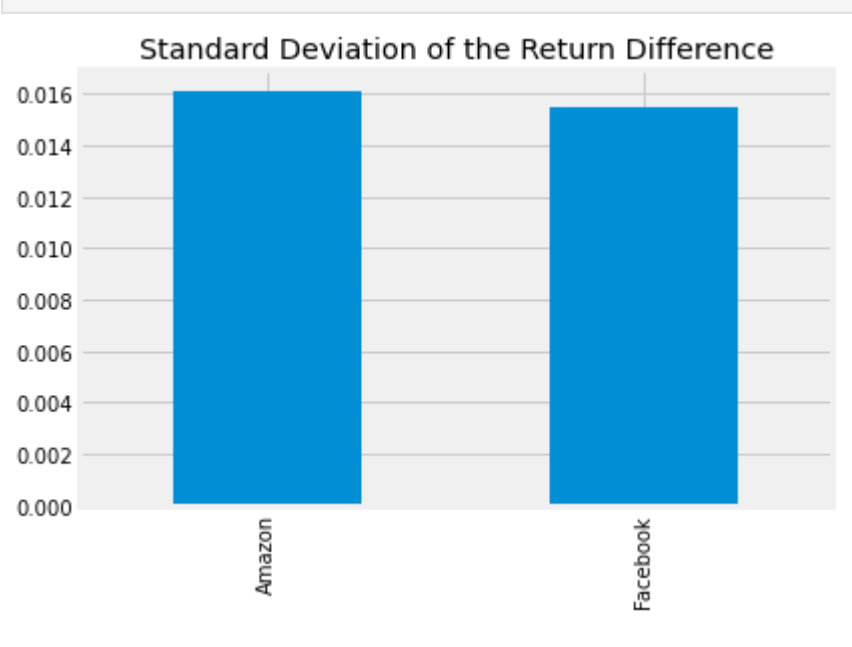
g. The Sharpe Ratio, Step 2: Standard Deviation of the Return Difference

There appears to be a significant difference in average daily returns between Amazon and Facebook.

The standard deviation of the excess returns is then computed. This shows how much risk an investment in stocks entails when compared to an investment in the S&P 500.

In [25]:
```
# calculate the standard deviations
sd_excess_return = excess_returns.std()

# plot the standard deviations
sd_excess_return.plot.bar(title = 'Standard Deviation of the Return Difference');
```



h. Putting it all together

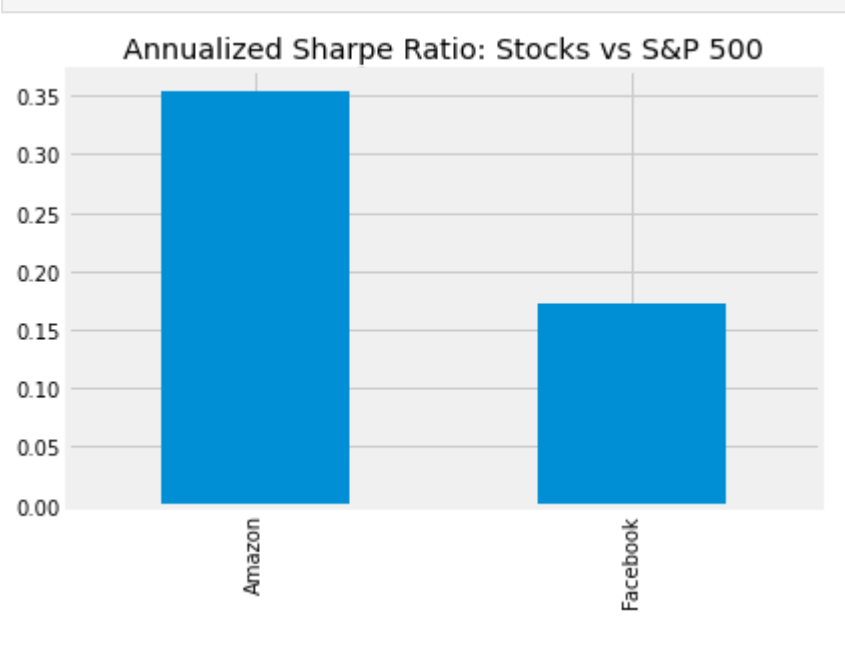
All that remains is to compute the ratio of avg excess returns to sd excess returns. Finally, the Sharpe ratio is calculated, which indicates how much more (or less) return the investment opportunity under consideration yields per unit of risk.

Annualizing the Sharpe Ratio involves multiplying it by the square root of the number of periods. We used daily data as input, so the square root of the number of trading days (5 days, 52 weeks, minus a few holidays) will be used: $\sqrt{252}$

In [26]:
```
# calculate the daily sharpe ratio
daily_sharpe_ratio = avg_excess_return.div(sd_excess_return)

# annualize the sharpe ratio
annual_factor = np.sqrt(252)
annual_sharpe_ratio = daily_sharpe_ratio.mul(annual_factor)

# plot the annualized sharpe ratio
annual_sharpe_ratio.plot.bar(title = 'Annualized Sharpe Ratio: Stocks vs S&P 500');
```



A higher Sharpe ratio indicates that the reward for a given amount of risk will be greater. Can you determine the best option based on this and the graph?

Obviously, you can.