financial-market-analysis

June 27, 2024

[1]:

```
!pip install yfinance
```

```
Requirement already satisfied: yfinance in c:\users\zeine\anaconda3\lib\site-
packages (0.2.40)
Requirement already satisfied: pandas>=1.3.0 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (2.1.4)
Requirement already satisfied: numpy>=1.16.5 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (1.26.4)
Requirement already satisfied: requests>=2.31 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (2.31.0)
Requirement already satisfied: multitasking>=0.0.7 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in c:\users\zeine\anaconda3\lib\site-
packages (from yfinance) (4.9.3)
Requirement already satisfied: platformdirs>=2.0.0 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (3.10.0)
Requirement already satisfied: pytz>=2022.5 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (2023.3.post1)
Requirement already satisfied: frozendict>=2.3.4 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (2.4.2)
Requirement already satisfied: peewee>=3.16.2 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (3.17.5)
Requirement already satisfied: beautifulsoup4>=4.11.1 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (4.12.2)
Requirement already satisfied: html5lib>=1.1 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in
c:\users\zeine\anaconda3\lib\site-packages (from
beautifulsoup4>=4.11.1->yfinance) (2.5)
Requirement already satisfied: six>=1.9 in c:\users\zeine\anaconda3\lib\site-
packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in
c:\users\zeine\anaconda3\lib\site-packages (from html5lib>=1.1->yfinance)
(0.5.1)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\zeine\anaconda3\lib\site-packages (from pandas>=1.3.0->yfinance)
(2.8.2)
Requirement already satisfied: tzdata>=2022.1 in
```

```
c:\users\zeine\anaconda3\lib\site-packages (from pandas>=1.3.0->yfinance)
(2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
c:\users\zeine\anaconda3\lib\site-packages (from requests>=2.31->yfinance)
(2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\zeine\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\zeine\anaconda3\lib\site-packages (from requests>=2.31->yfinance)
(2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\zeine\anaconda3\lib\site-packages (from requests>=2.31->yfinance)
(2024.6.2)
```

[3]:

```
!pip install quantstats
```

```
Requirement already satisfied: quantstats in c:\users\zeine\anaconda3\lib\site-
packages (0.0.62)
Requirement already satisfied: pandas>=0.24.0 in
c:\users\zeine\anaconda3\lib\site-packages (from quantstats) (2.1.4)
Requirement already satisfied: numpy>=1.16.5 in
c:\users\zeine\anaconda3\lib\site-packages (from quantstats) (1.26.4)
Requirement already satisfied: seaborn>=0.9.0 in
c:\users\zeine\anaconda3\lib\site-packages (from quantstats) (0.12.2)
Requirement already satisfied: matplotlib>=3.0.0 in
c:\users\zeine\anaconda3\lib\site-packages (from quantstats) (3.8.0)
Requirement already satisfied: scipy>=1.2.0 in
c:\users\zeine\anaconda3\lib\site-packages (from quantstats) (1.11.4)
Requirement already satisfied: tabulate>=0.8.0 in
c:\users\zeine\anaconda3\lib\site-packages (from quantstats) (0.9.0)
Requirement already satisfied: yfinance>=0.1.70 in
c:\users\zeine\anaconda3\lib\site-packages (from quantstats) (0.2.40)
Requirement already satisfied: python-dateutil>=2.0 in
c:\users\zeine\anaconda3\lib\site-packages (from quantstats) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\zeine\anaconda3\lib\site-packages (from matplotlib>=3.0.0->quantstats)
(1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\zeine\anaconda3\lib\site-packages (from matplotlib>=3.0.0->quantstats)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\zeine\anaconda3\lib\site-packages (from matplotlib>=3.0.0->quantstats)
(4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\zeine\anaconda3\lib\site-packages (from matplotlib>=3.0.0->quantstats)
(1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\zeine\anaconda3\lib\site-packages (from matplotlib>=3.0.0->quantstats)
```

```
(23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\zeine\anaconda3\lib\site-packages (from matplotlib>=3.0.0->quantstats)
(10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\zeine\anaconda3\lib\site-packages (from matplotlib>=3.0.0->quantstats)
(3.0.9)
Requirement already satisfied: pytz>=2020.1 in
c:\users\zeine\anaconda3\lib\site-packages (from pandas>=0.24.0->quantstats)
(2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\zeine\anaconda3\lib\site-packages (from pandas>=0.24.0->quantstats)
(2023.3)
Requirement already satisfied: six>=1.5 in c:\users\zeine\anaconda3\lib\site-
packages (from python-dateutil>=2.0->quantstats) (1.16.0)
Requirement already satisfied: requests>=2.31 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance>=0.1.70->quantstats)
(2.31.0)
Requirement already satisfied: multitasking>=0.0.7 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance>=0.1.70->quantstats)
(0.0.11)
Requirement already satisfied: lxml>=4.9.1 in c:\users\zeine\anaconda3\lib\site-
packages (from yfinance>=0.1.70->quantstats) (4.9.3)
Requirement already satisfied: platformdirs>=2.0.0 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance>=0.1.70->quantstats)
(3.10.0)
Requirement already satisfied: frozendict>=2.3.4 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance>=0.1.70->quantstats)
(2.4.2)
Requirement already satisfied: peewee>=3.16.2 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance>=0.1.70->quantstats)
(3.17.5)
Requirement already satisfied: beautifulsoup4>=4.11.1 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance>=0.1.70->quantstats)
(4.12.2)
Requirement already satisfied: html5lib>=1.1 in
c:\users\zeine\anaconda3\lib\site-packages (from yfinance>=0.1.70->quantstats)
(1.1)
Requirement already satisfied: soupsieve>1.2 in
c:\users\zeine\anaconda3\lib\site-packages (from
beautifulsoup4>=4.11.1->yfinance>=0.1.70->quantstats) (2.5)
Requirement already satisfied: webencodings in
c:\users\zeine\anaconda3\lib\site-packages (from
html5lib>=1.1->yfinance>=0.1.70->quantstats) (0.5.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
c:\users\zeine\anaconda3\lib\site-packages (from
requests>=2.31->yfinance>=0.1.70->quantstats) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
```

```
c:\users\zeine\anaconda3\lib\site-packages (from
requests>=2.31->yfinance>=0.1.70->quantstats) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\zeine\anaconda3\lib\site-packages (from
requests>=2.31->yfinance>=0.1.70->quantstats) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\zeine\anaconda3\lib\site-packages (from
requests>=2.31->yfinance>=0.1.70->quantstats) (2024.6.2)
```

[80]:

```
!pip install ta
```

```
Collecting ta
  Downloading ta-0.11.0.tar.gz (25 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: numpy in c:\users\zeine\anaconda3\lib\site-
packages (from ta) (1.26.4)
Requirement already satisfied: pandas in c:\users\zeine\anaconda3\lib\site-
packages (from ta) (2.1.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\zeine\anaconda3\lib\site-packages (from pandas->ta) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\users\zeine\anaconda3\lib\site-packages (from pandas->ta) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\zeine\anaconda3\lib\site-packages (from pandas->ta) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\zeine\anaconda3\lib\site-
packages (from python-dateutil>=2.8.2->pandas->ta) (1.16.0)
Building wheels for collected packages: ta
  Building wheel for ta (setup.py): started
  Building wheel for ta (setup.py): finished with status 'done'
  Created wheel for ta: filename=ta-0.11.0-py3-none-any.whl size=29420
sha256=fe1737b15e7c51553be508d12c2cb236a6c7fc3de91a6b587e8cc469699cd2c9
  Stored in directory: c:\users\zeine\appdata\local\pip\cache\wheels\a1\d7\29\77
81cc5eb9a3659d032d7d15bdd0f49d07d2b24fec29f44bc4
Successfully built ta
Installing collected packages: ta
Successfully installed ta-0.11.0
```

[86]:

```
# Importing Libraries

# Data handling and statistical analysis
import pandas as pd
from pandas_datareader import data
import numpy as np
from scipy import stats

# Data visualization
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# Financial data
import quantstats as qs
import ta
import yfinance as yf

# Linear Regression Model
from sklearn.linear_model import LinearRegression

# Enabling Plotly offline
from plotly.offline import init_notebook_mode
init_notebook_mode(connected=True)

# Datetime and hiding warnings
import datetime as dt
import warnings
warnings.filterwarnings("ignore")
```

1 Daily Returns for Stocks

Definition: Daily returns measure the percentage change in the value of an investment from one day to the next. For example, if a stock's price increases from 100 to 102 in one day, the daily return is 2%.

[98]:
```
azn = qs.utils.download_returns('AZN')
azn = azn.loc['2010-07-01':'2024-06-20']

bp = qs.utils.download_returns('BP')
bp = bp.loc['2010-07-01':'2024-06-20']

aapl = qs.utils.download_returns('AAPL')
aapl = aapl.loc['2010-07-01':'2024-06-20']

msft = qs.utils.download_returns('MSFT')
msft = msft.loc['2010-07-01':'2024-06-20']
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
```

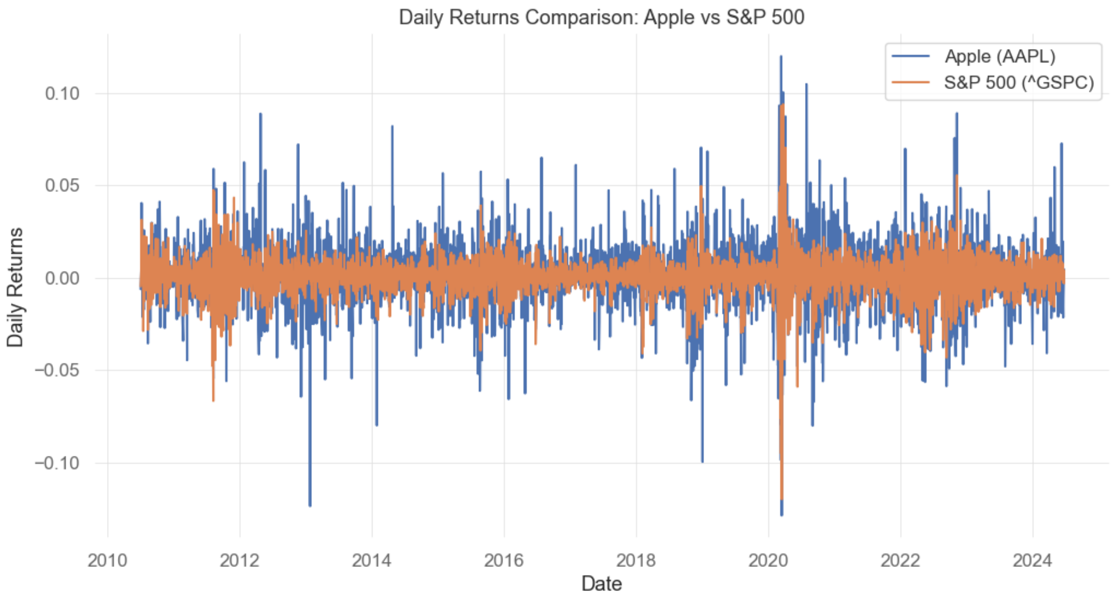
[100]:

```python
try:
    azn.index = azn.index.tz_localize('UTC').tz_convert(None)
    bp.index = bp.index.tz_localize('UTC').tz_convert(None)
    aapl.index = aapl.index.tz_localize('UTC').tz_convert(None)
    msft.index = msft.index.tz_localize('UTC').tz_convert(None)
except Exception as e:
    print(f"An error occurred during timezone localization and conversion: {e}")
```

[96]:

```python
# Function to download daily returns
def download_daily_returns(ticker):
    data = yf.download(ticker, start='2010-07-01', end='2024-06-26')
    returns = data['Adj Close'].pct_change().dropna()
    returns.index = returns.index.tz_localize('UTC')
    return returns

# Downloading Apple (AAPL) and S&P 500 (^GSPC) daily returns
aapl = download_daily_returns('AAPL')
sp500 = download_daily_returns('^GSPC')

# Plotting daily returns For Apple
print('\n')
print('\nApple Daily Returns Plot:\n')
plt.figure(figsize=(12, 6))
plt.plot(aapl.index, aapl, label='Apple (AAPL)')
plt.plot(sp500.index, sp500, label='S&P 500 (^GSPC)')
plt.title('Daily Returns Comparison: Apple vs S&P 500')
plt.xlabel('Date')
plt.ylabel('Daily Returns')
plt.legend()
plt.grid(True)
plt.show()
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed



Apple Daily Returns Plot:
```

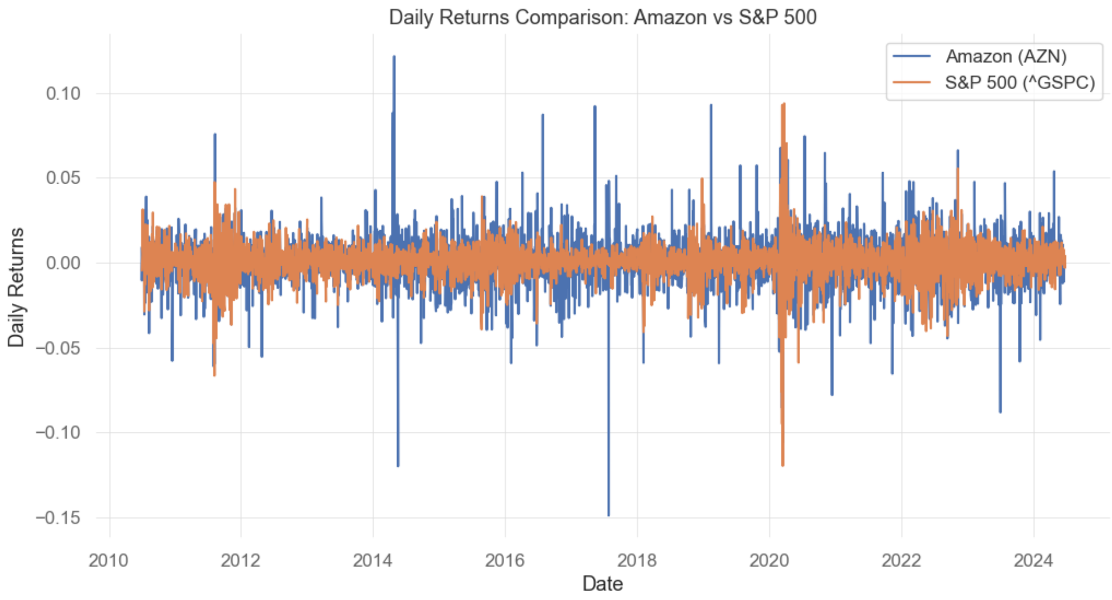


1.1 Apple analysis:

- Huge volatility in 2013 and 2020 (due to COVID)
- In general stocks seem stable and a predictable investment option

[110]:
```
# Plotting daily returns for Amazon
print('\n')
print('\nAmazon Daily Returns Plot:\n')
plt.figure(figsize=(12,6))
plt.plot(azn.index, azn, label='Amazon (AZN)')
plt.plot(sp500.index, sp500, label='S&P 500 (^GSPC)')
plt.title('Daily Returns Comparison: Amazon vs S&P 500')
plt.xlabel('Date')
plt.ylabel('Daily Returns')
plt.legend()
plt.grid(True)
plt.show()
```

```
Amazon Daily Returns Plot:
```



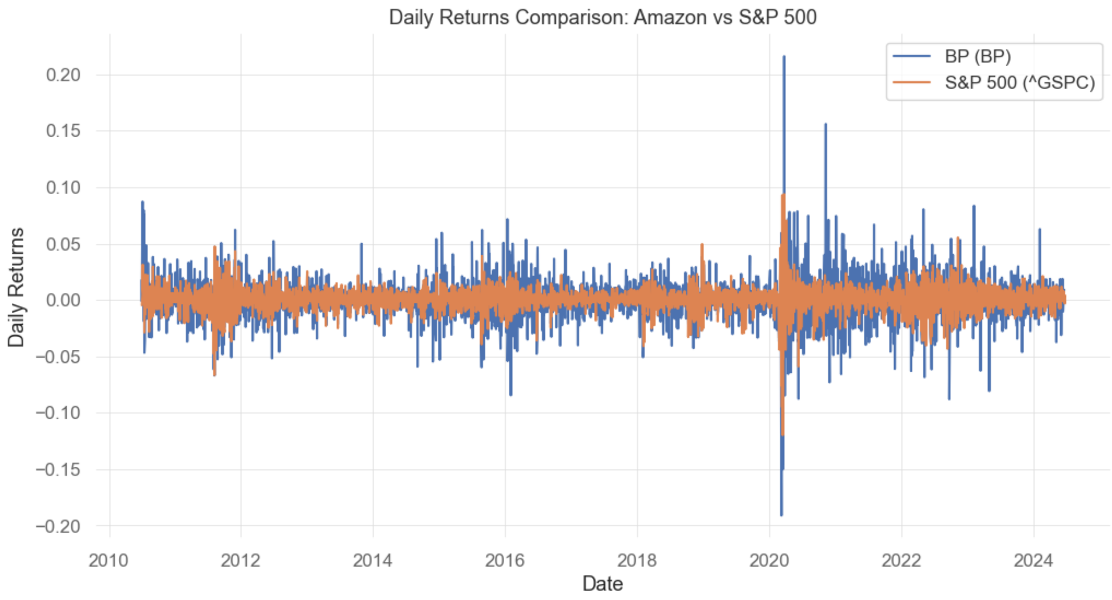
1.2 Amazon analysis:

- Unusual variation seen in late 2014/early 2015 and late 2016/early2017
- This indivates higher volatility suggesting it could be a riskier investment

[120]:

```
# Plotting daily returns for Amazon
print('\n')
print('\nBP Daily Returns Plot:\n')
plt.figure(figsize=(12,6))
plt.plot(bp.index, bp, label= 'BP (BP)')
plt.plot(sp500.index, sp500, label='S&P 500 (^GSPC)')
plt.title('Daily Returns Comparison: Amazon vs S&P 500')
plt.xlabel('Date')
plt.ylabel('Daily Returns')
plt.legend()
plt.grid(True)
plt.show()
```

```
BP Daily Returns Plot:
```

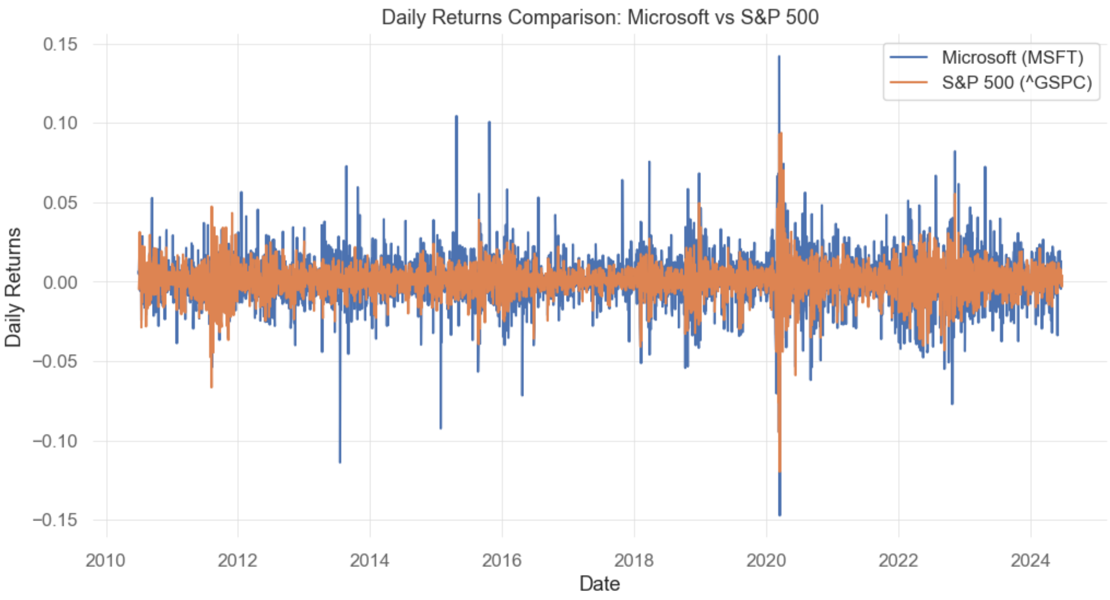


1.3 BP analysis:

- As expected, volatility seen in 2020
- In geenral the stocks seem stable and a predictable investment option

[126]:
```
# Plotting daily returns for Microsoft
print('\n')
print('\nMicrosoft Daily Returns Plot:\n')
plt.figure(figsize=(12,6))
plt.plot(msft.index, msft, label = 'Microsoft (MSFT)')
plt.plot(sp500.index, sp500, label = 'S&P 500 (^GSPC)')
plt.title('Daily Returns Comparison: Microsoft vs S&P 500')
plt.xlabel('Date')
plt.ylabel('Daily Returns')
plt.grid(True)
plt.legend()
plt.show()
```

```
Microsoft Daily Returns Plot:
```



1.4 Microsoft analysis:

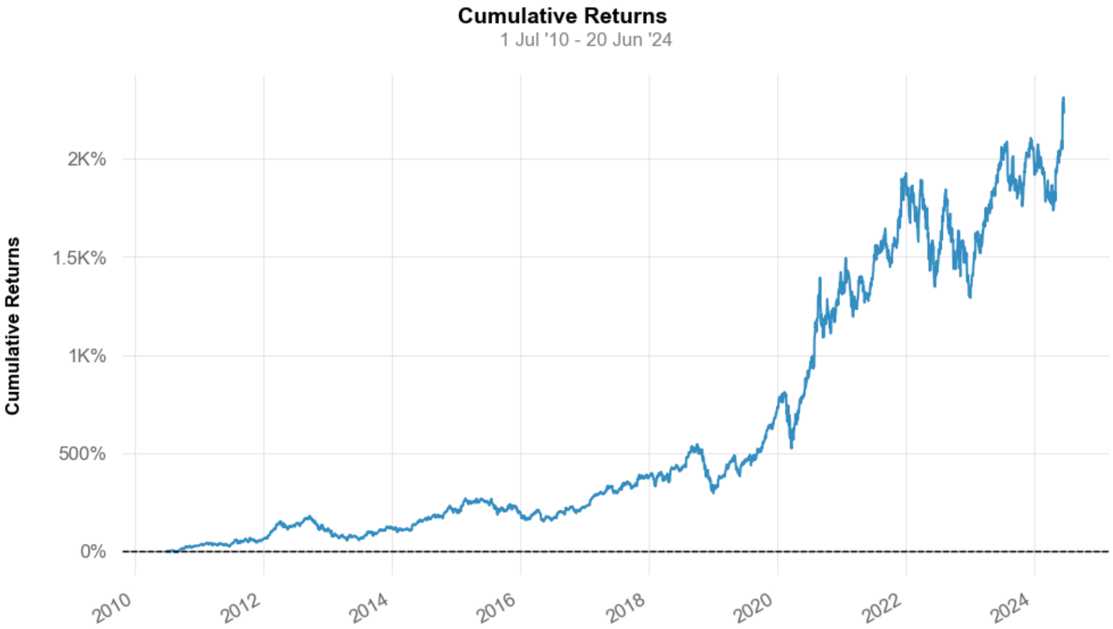
- As expected, volatility seen in 2020- but bigger than usual
- IUnusal volatility seen around 2013, and 2015-2016 year.
- This might be a riskier investment

2 Cumulative Returns

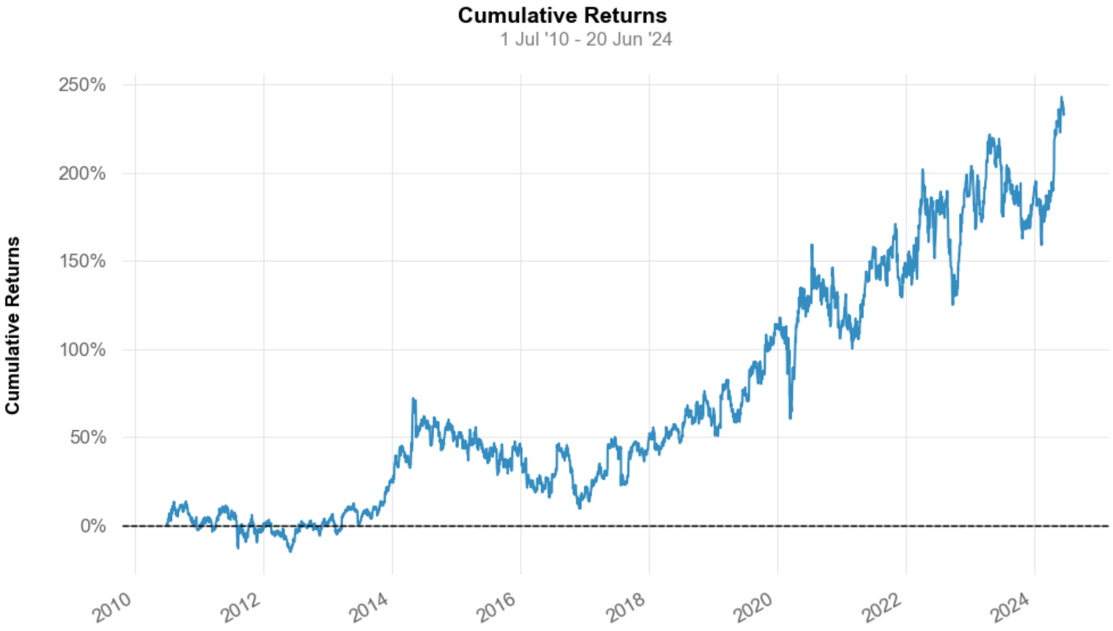
Definition: Cumulative return is the total change in the value of an investment over a specified period, expressed as a percentage of the initial value. For example, if you invest 1,000 in a stock and its value grows to 1,500 over three years, the cumulative return is 50%.

[160]:
```
# print('\n')
print('\nApple Cumulative Returns Plot\n')
qs.plots.returns(aapl)
print('\n')
print('\n')
print('\nAmazon Cumulative Returns Plot\n')
qs.plots.returns(azn)
print('\n')
print('\n')
print('\nBP Cumulative Returns Plot\n')
qs.plots.returns(bp)
print('\n')
print('\n')
print('\nMicrosoft Cumulative Returns Plot\n')
qs.plots.returns(msft)
```

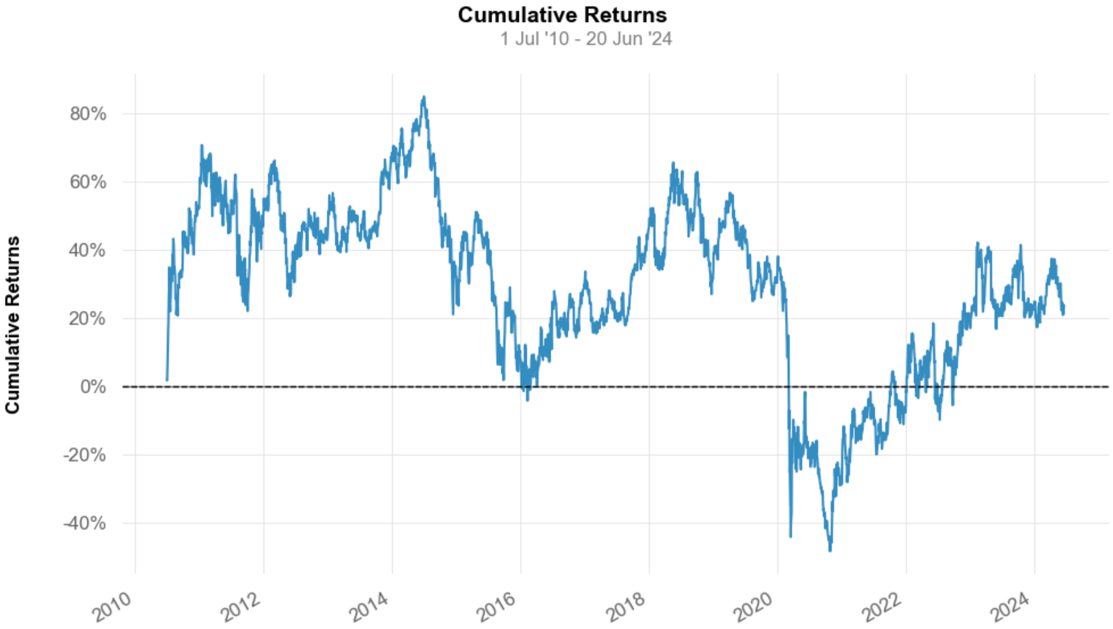
```
Apple Cumulative Returns Plot
```



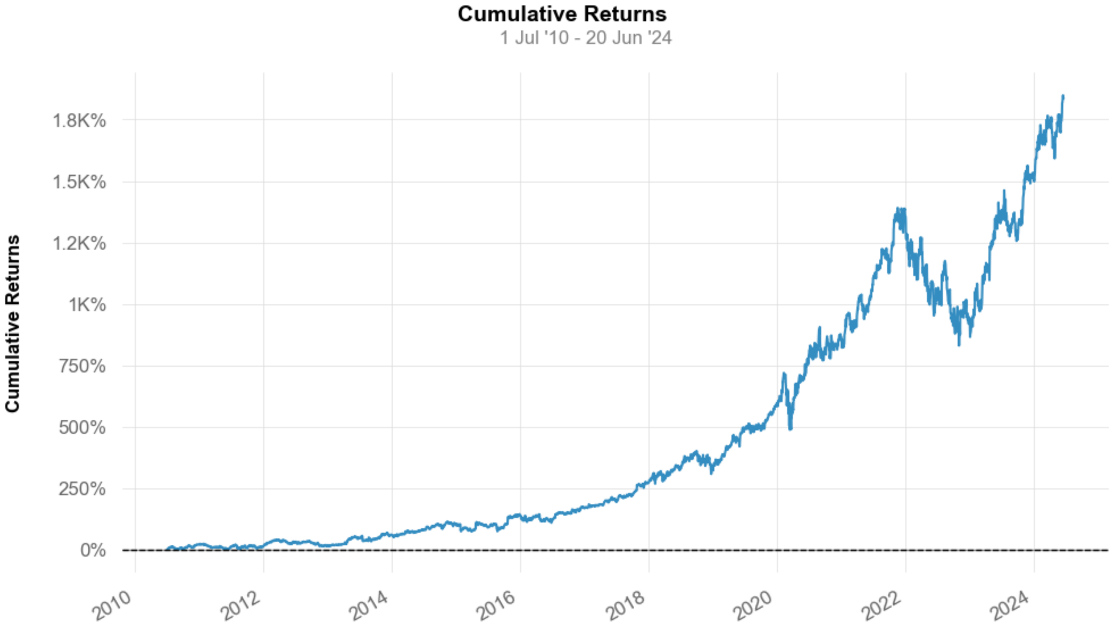
```
Amazon Cumulative Returns Plot
```



BP Cumulative Returns Plot



```
Microsoft Cumulative Returns Plot
```



2.1 Cumulative Returns Analysis

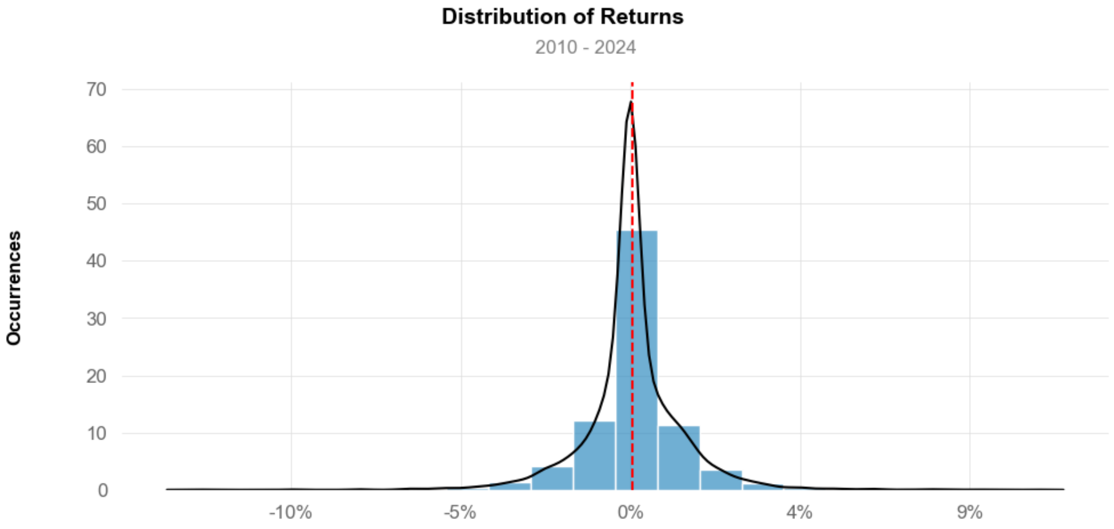
- Apple has the highest percentage of Cumulative return peak- surpassing 2k in 2024 as opposed to the lowest peak of above 80% in 2015
- Stark difference in negative cumulative return between BP and all the other stocks- BP cumulative returns was at an alaraming low percentage, exceeding -40% in 2020-2021
- However, in 2024 it has now increased back to just above 40%
- BP seems to be yield extreme volatile cumulative returns
- Microsoft also yielded good cumulative returns, exceeding 1.8k in 2024

3 Histogram analysis

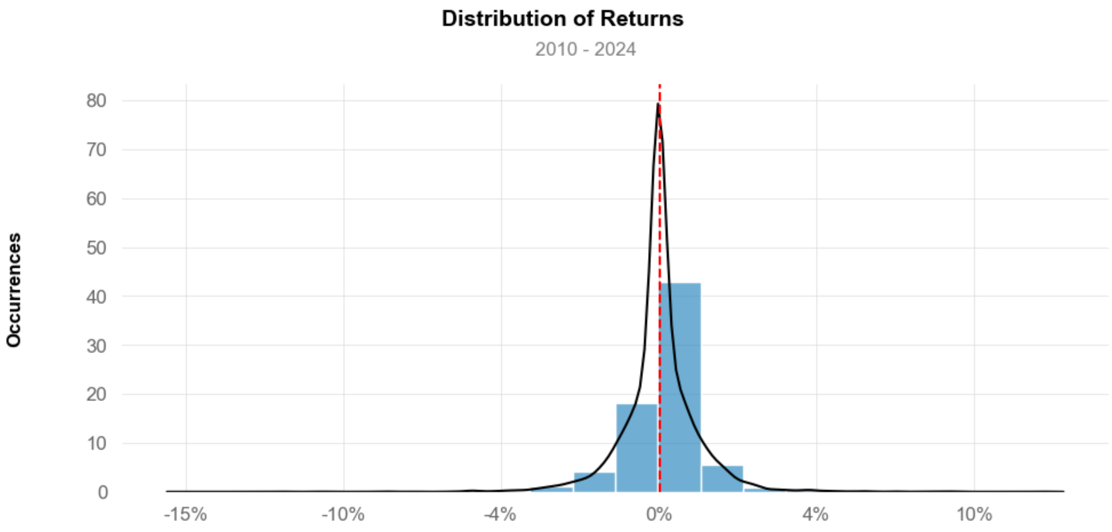
[164]:
```
# Plotting histograms for daily returns
print('\n')
print('\nApple Daily Returns Histogram')
```

```
qs.plots.histogram(aapl, resample = 'D')
print('\n')
print('\nAmazon Daily Returns Histogram')
qs.plots.histogram(azn, resample = 'D')
print('\n')
print('\nBP Daily Returns Histogram')
qs.plots.histogram(bp, resample = 'D')
print('\n')
print('\nMicrosoft Daily Returns Histogram')
qs.plots.histogram(msft, resample = 'D')
```

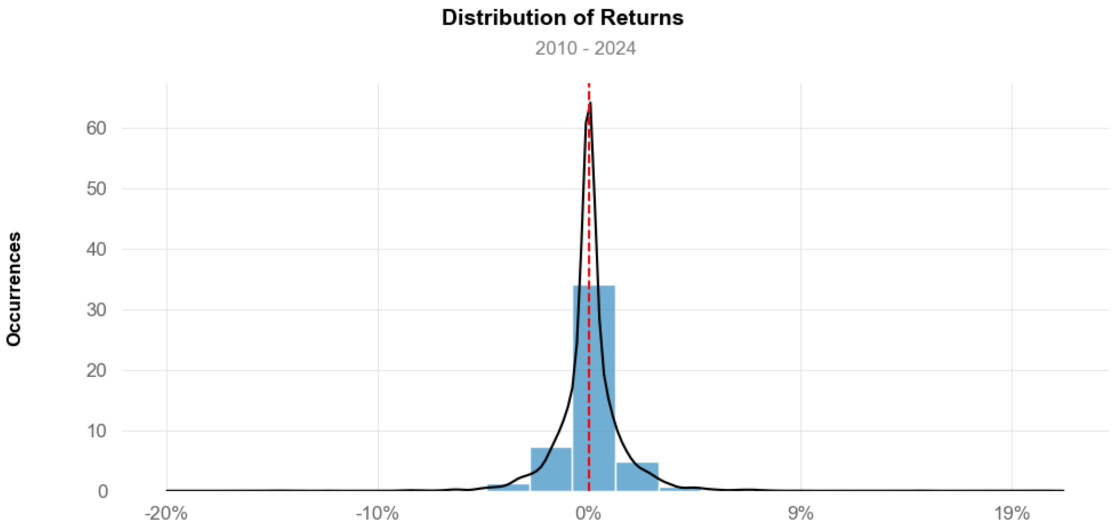
```
Apple Daily Returns Histogram
```



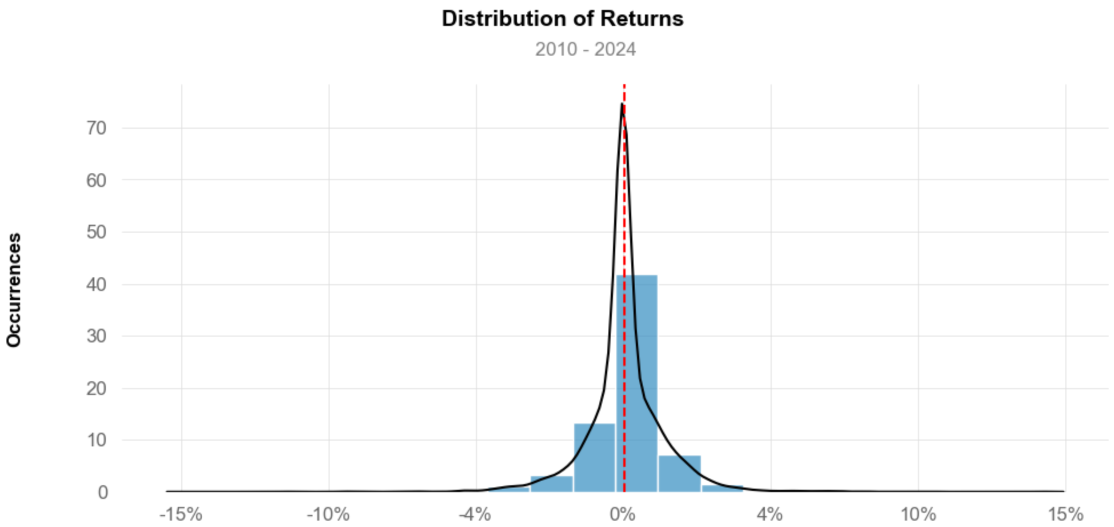
```
Amazon Daily Returns Histogram
```



BP Daily Returns Histogram



Microsoft Daily Returns Histogram



Distribution of Returns

2010 - 2024

4 Histogram Analysis

- Most Daily returns are close to 0
- Extreme values seem to be consistent across stocks- never exceedeing -20% and 20%
- All stocks seem to have balaced retrns- values on extreme sides being closer to eachother

4.1 Kurtosis

[172]:

```
print('\n')
print("Apple's kurtosis: ", qs.stats.kurtosis(aapl).round(2))
print('\n')
print("Amazon's kurtosis: ", qs.stats.kurtosis(azn).round(2))
print('\n')
print("BP's kurtosis: ", qs.stats.kurtosis(bp).round(3))
print('\n')
print("Microsoft's kurtosis: ", qs.stats.kurtosis(msft).round(3))
```

```


Apple's kurtosis:  5.35



Amazon's kurtosis:  9.71



BP's kurtosis:  14.151



Microsoft's kurtosis:  7.807
```

4.2 Analysis:

- All four stocks have positive and high kurtosis- indicating that there is a high concentration of daily return observations in the tails of their distribution
- This suggests all four stocks can be subject to high levels of volatility and therefore risk- there can be a good amount of price fluctuations which deviate significantly from their average returns
- BP has thew highest kurtosis level- meaning it is subjected to high levels of volatility and risk (as seen from the other above ghraphs)

4.3 Skewness

[176]:

```
print('\n')
print("Apple's skewness: ", qs.stats.skew(aapl).round(2))
print('\n')
print("Amazon's skewness: ", qs.stats.skew(azn).round(2))
print('\n')
print("BP's skewness: ", qs.stats.skew(bp).round(3))
print('\n')
print("Microsoft's skewness: ", qs.stats.skew(msft).round(3))
```

```


Apple's skewness:  -0.04


Amazon's skewness:  -0.23


BP's skewness:  0.053


Microsoft's skewness:  0.054
```

- Generally slight level of skewness across all four stocks - confirming insights from Histogram

4.4 Correlation Matrix and Pairplots

[193]:

```
df = pd.concat([aapl, azn, bp, msft], join = 'outer', axis = 1)
df.columns = ['Apple', 'Amazon', 'BP', 'Microsoft']
df
```

[193]:

```
                              Apple    Amazon        BP  Microsoft
Date
2010-07-02 00:00:00+00:00 -0.006198 -0.010515 -0.001361   0.004750
2010-07-06 00:00:00+00:00  0.006844  0.015303  0.087223   0.023635
2010-07-07 00:00:00+00:00  0.040381  0.011723  0.040113   0.020151
2010-07-08 00:00:00+00:00 -0.002242  0.009104  0.016571   0.004527
2010-07-09 00:00:00+00:00  0.005928  0.004716  0.009188  -0.005735
```

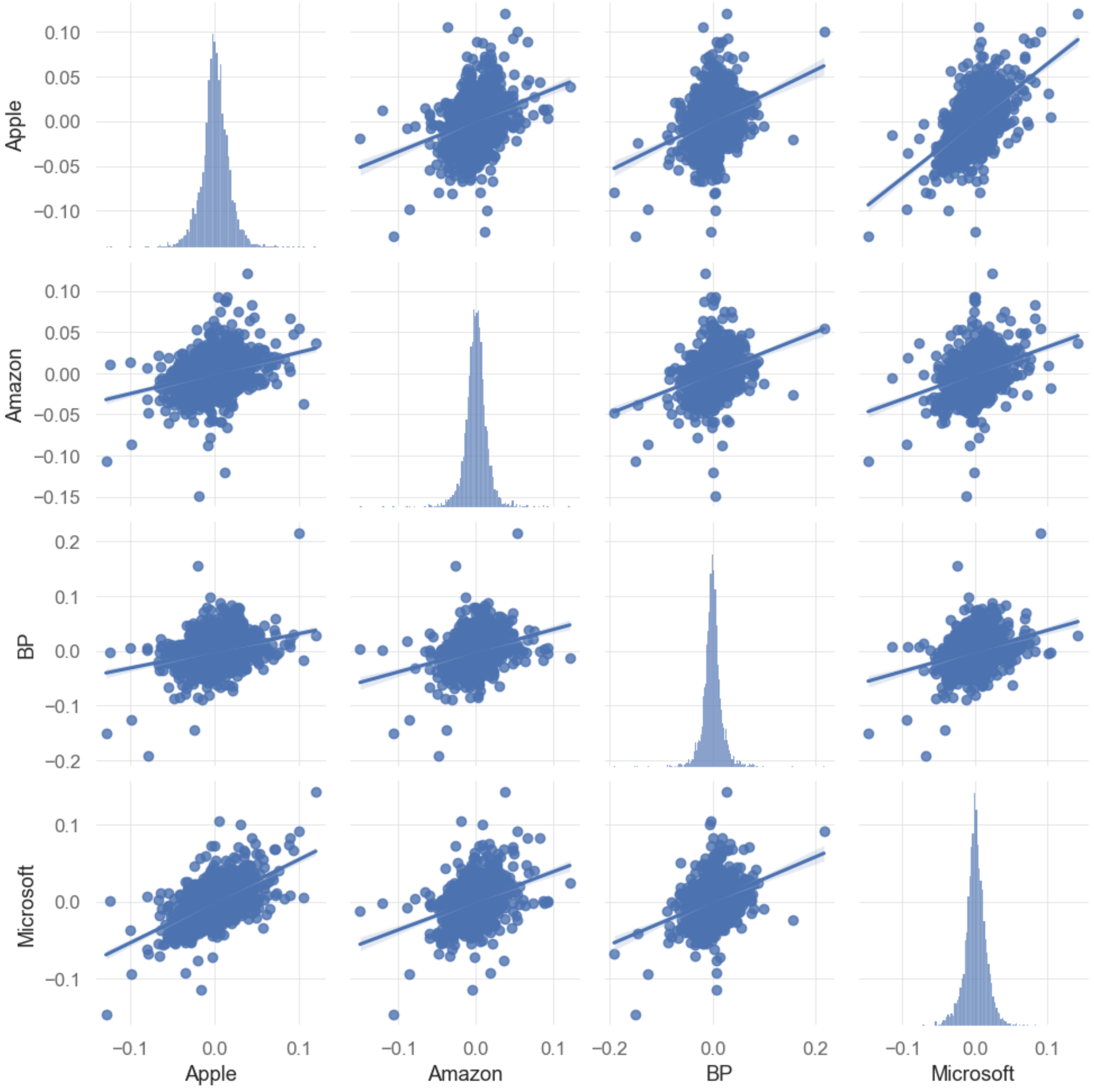
```
…                                 …         …         …         …
2024-06-18 00:00:00+00:00 -0.010984 -0.011222  0.007975  -0.004528
2024-06-20 00:00:00+00:00 -0.021513  0.001785  0.009042  -0.001434
2024-06-21 00:00:00+00:00 -0.010444  0.004073 -0.005601   0.009154
2024-06-24 00:00:00+00:00  0.003133  0.006212  0.020839  -0.004691
2024-06-25 00:00:00+00:00  0.004468  0.004788 -0.004690   0.007327

[3518 rows x 4 columns]
```

[195]:

```
sns.pairplot(df, kind='reg')
plt.show
```

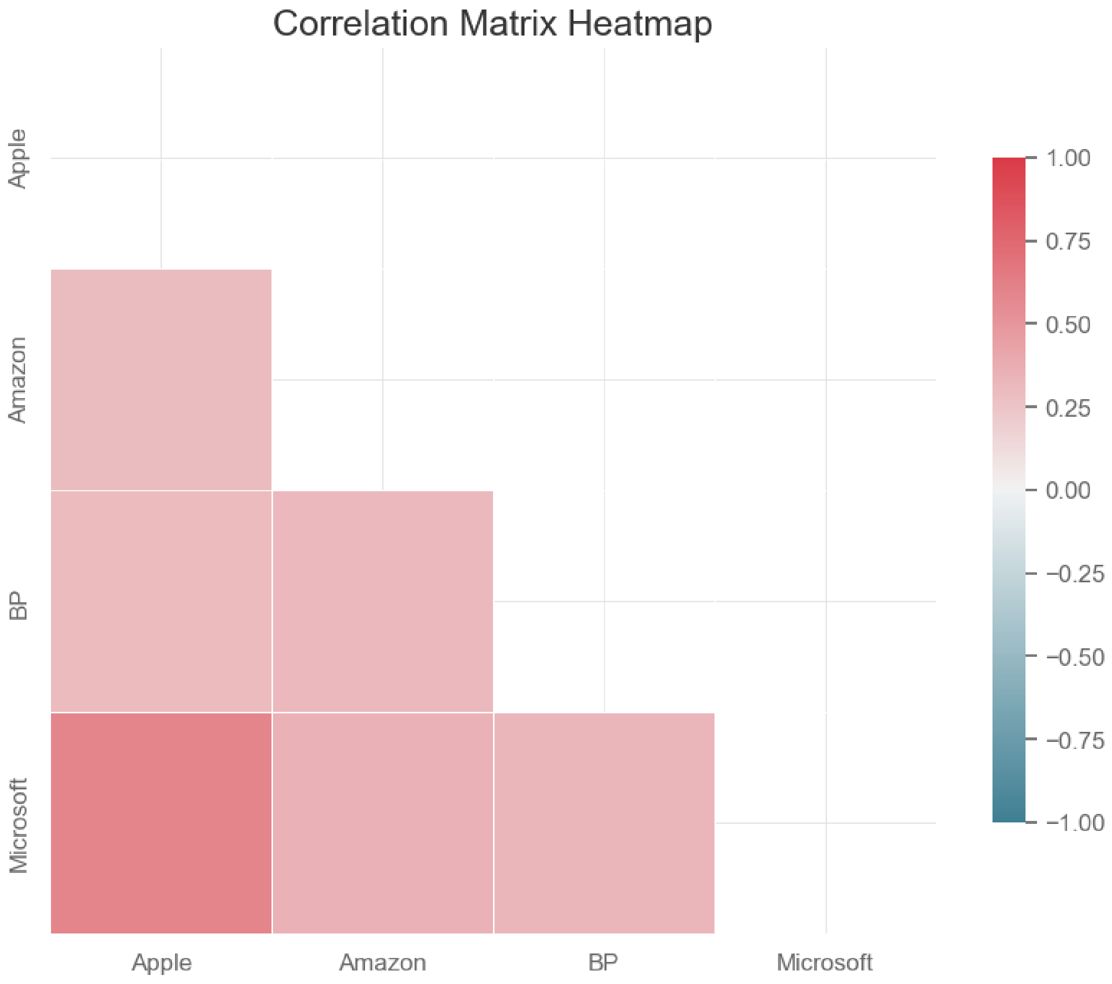
[195]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



[211]:

```
mask = np.triu(np.ones_like(corr, dtype=bool))

plt.figure(figsize=(10, 8))

cmap = sns.diverging_palette(220, 10, as_cmap=True)

sns.heatmap(corr, mask=mask, cmap=cmap, vmax=1, vmin=-1, center=0,
            square=True, linewidths=.5, annot=True, fmt=".2f", cbar_kws={"shrink": .75})

# Set the title
plt.title('Correlation Matrix Heatmap', fontsize=18)

# Show the plot
plt.show()
```



4.5 Analysis

- Strong Positive correlation between Apple and Microsoft
- Relative positive correlation between all other stocks
- Should not use Long-Short strategies

5 Sharpe Ratio

[226]:

```
print('\n')
print("Sharpe Ratio for AAPL: ", qs.stats.sharpe(aapl).round(2))
print('\n')
print("Sharpe Ratio for AZN: ", qs.stats.sharpe(azn).round(2))
print('\n')
print("Sharpe Ratio for BP: ", qs.stats.sharpe(bp).round(2))
print('\n')
print("Sharpe Ratio for MSFT: ", qs.stats.sharpe(msft).round(2))
```

```


Sharpe Ratio for AAPL:  0.99



Sharpe Ratio for AZN:  0.65



Sharpe Ratio for BP:  0.38



Sharpe Ratio for MSFT:  1.03
```

5.1 Analysis

- Apple and Microsoft offer a better risk-return relationship

[ ]:

```

```