Analysis of my 5 favorite stocks

In [1]:

```
# import libraries

import pandas as pd
import yfinance as yf
from datetime import datetime
```

In [2]:

```
# create start and end date

start_date = datetime.now() - pd.DateOffset(months = 6)
end_date = datetime.now()
```

In [5]:

```
# MY FAVORITES :)

tickers = ['SPY', 'QQQ', 'AAPL', 'AMD', 'SHOP']
```

In [6]:

```
# download data for the last 3 months from yfinance

df_list = [] # start with an empty list

for ticker in tickers: # iterates through each ticker symbol
    data = yf.download(ticker, start=start_date, end=end_date) # downloads historical stock data
    data['DailyRange'] = data['High'] - data['Low']
    df_list.append(data) # append list and data

df = pd.concat(df_list, keys=tickers, names=['Ticker', 'Date']) # concat list into df

#print
print(df.head())
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
                         Open        High         Low       Close   Adj Close  \
Ticker Date
SPY    2023-06-27  432.350006  436.809998  431.880005  436.170013  432.881836
       2023-06-28  435.049988  437.440002  434.410004  436.390015  433.100189
       2023-06-29  435.959991  438.279999  435.540009  438.109985  434.807159
       2023-06-30  441.440002  444.299988  441.109985  443.279999  439.938202
       2023-07-03  442.920013  444.079987  442.630005  443.790009  440.444366

                      Volume  DailyRange
Ticker Date
SPY    2023-06-27   72813700    4.929993
       2023-06-28   75636000    3.029999
       2023-06-29   67882300    2.739990
       2023-06-30  104921500    3.190002
       2023-07-03   32793400    1.449982
```

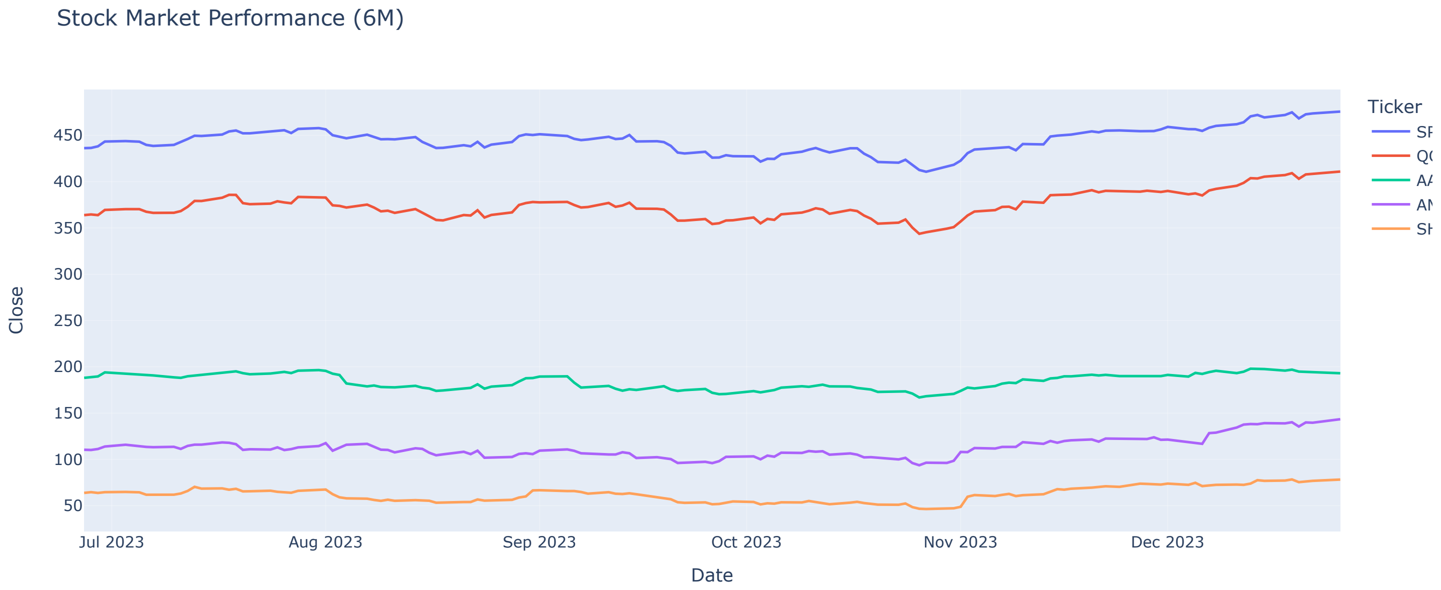
In [7]:

```
# reset index from Date col

df = df.reset_index()
print(df.head())
```

```
  Ticker       Date        Open        High         Low       Close  \
0    SPY 2023-06-27  432.350006  436.809998  431.880005  436.170013
1    SPY 2023-06-28  435.049988  437.440002  434.410004  436.390015
2    SPY 2023-06-29  435.959991  438.279999  435.540009  438.109985
3    SPY 2023-06-30  441.440002  444.299988  441.109985  443.279999
4    SPY 2023-07-03  442.920013  444.079987  442.630005  443.790009

    Adj Close     Volume  DailyRange
0  432.881836   72813700    4.929993
1  433.100189   75636000    3.029999
2  434.807159   67882300    2.739990
3  439.938202  104921500    3.190002
4  440.444366   32793400    1.449982
```

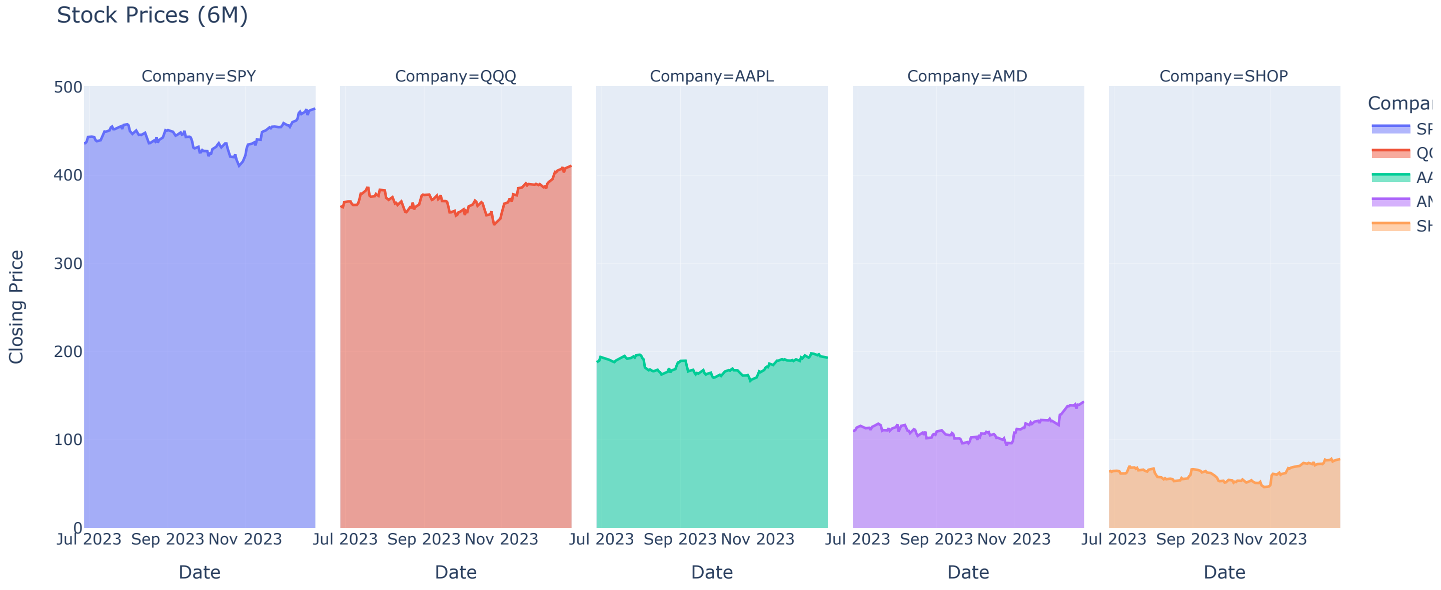
In [8]:

```
# visualize stock performance

import plotly.express as px
fig = px.line(df, x='Date', y='Close', color='Ticker', title='Stock Market Performance (6M)')

fig.show()
```



In [9]:

```python
# facet area chart

fig = px.area(df, x='Date',y='Close', color='Ticker',
             facet_col = 'Ticker',
             labels = {'Date':'Date', 'Close':'Closing Price', 'Ticker':'Company'},
             title = 'Stock Prices (6M)')

fig.show()
```



Moving Averages

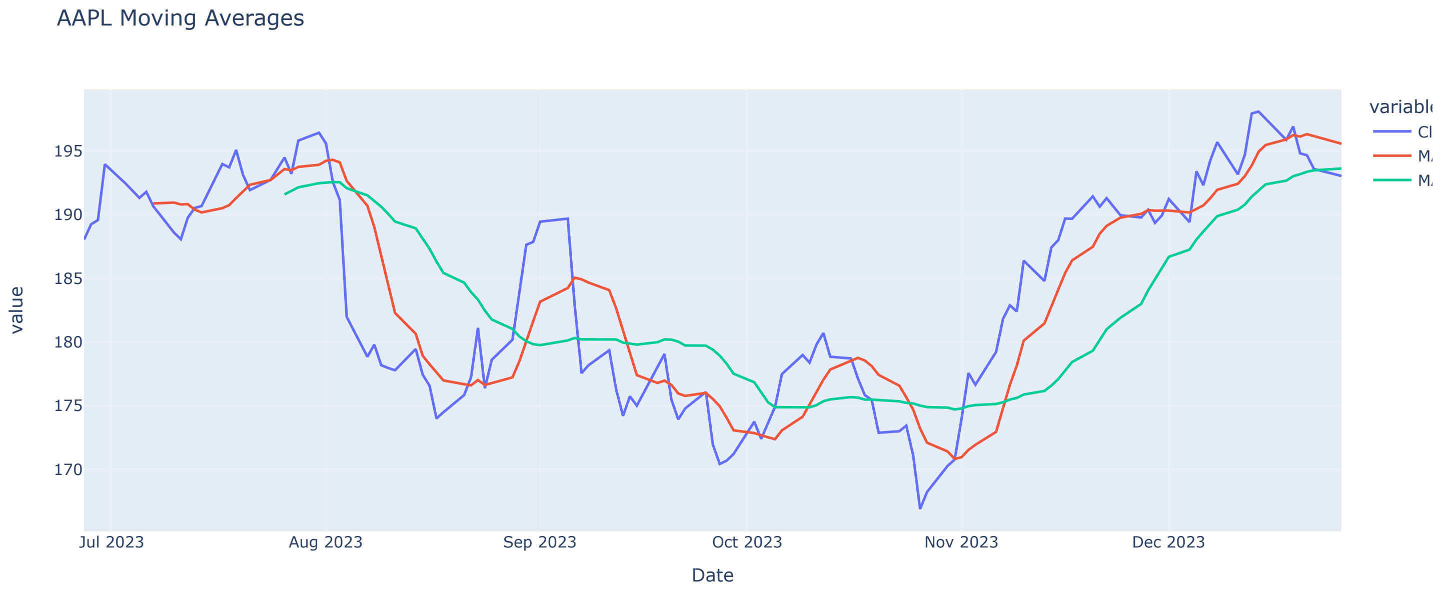
In [10]:

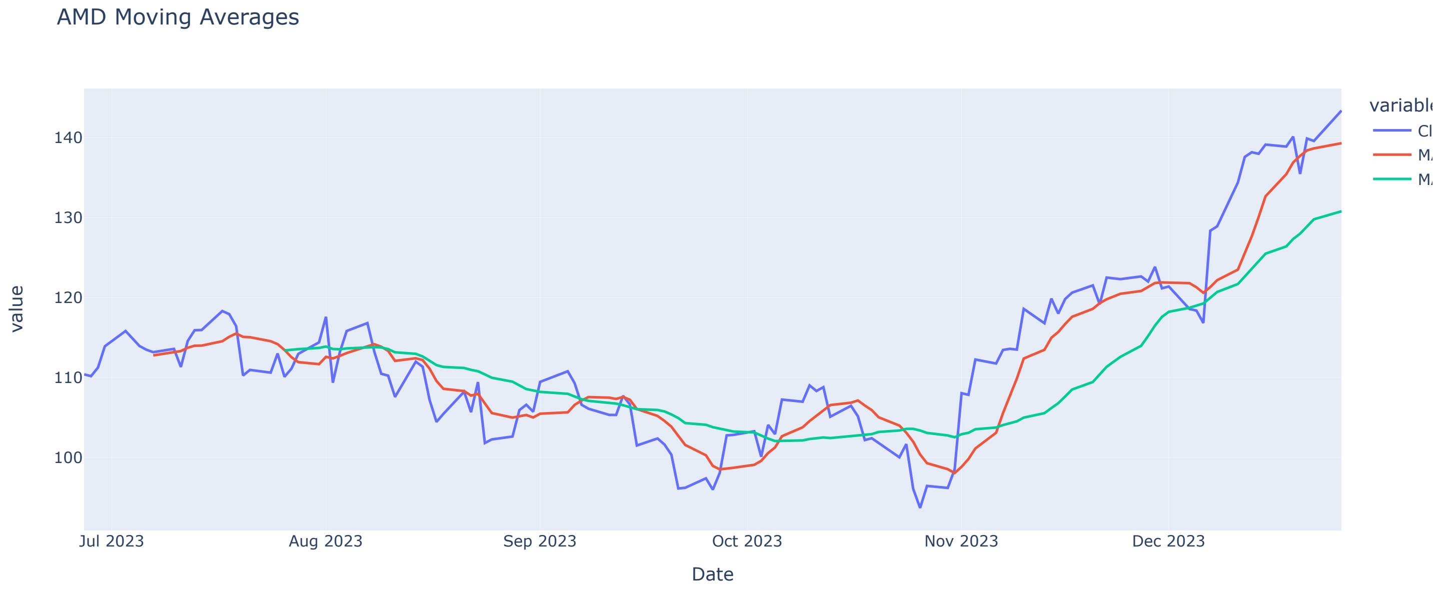
```python
# create moving averages (8 & 21)

df['MA8'] = df.groupby('Ticker')['Close'].rolling(window=8).mean().reset_index(0, drop=True)
df['MA21'] = df.groupby('Ticker')['Close'].rolling(window=21).mean().reset_index(0, drop=True)

# analyze moving averages

for ticker, group in df.groupby('Ticker'):
    print(f'Moving Averages for {ticker}')
    print(group[['MA8', 'MA21']])
```

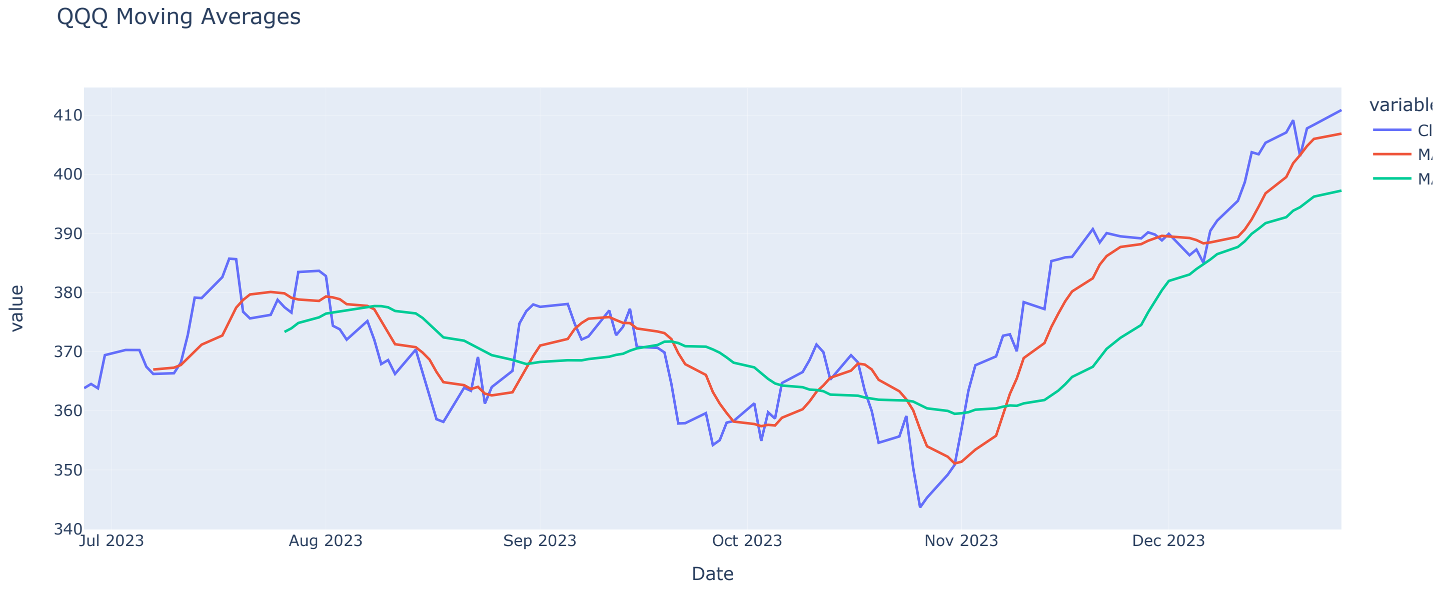
```
Moving Averages for AAPL
            MA8        MA21
254         NaN         NaN
255         NaN         NaN
256         NaN         NaN
257         NaN         NaN
258         NaN         NaN
..          ...         ...
376  196.258753  193.030000
377  196.148752  193.190953
378  196.336252  193.383333
379  196.197502  193.492382
380  195.583752  193.639048

[127 rows x 2 columns]
Moving Averages for AMD
            MA8        MA21
381         NaN         NaN
382         NaN         NaN
383         NaN         NaN
384         NaN         NaN
385         NaN         NaN
..          ...         ...
503  136.916248  127.335237
504  137.734999  127.999047
505  138.422499  128.987142
506  138.671249  129.800952
507  139.323750  130.805714

[127 rows x 2 columns]
Moving Averages for QQQ
            MA8        MA21
127         NaN         NaN
128         NaN         NaN
129         NaN         NaN
130         NaN         NaN
131         NaN         NaN
..          ...         ...
249  401.883751  393.851429
250  403.247498  394.439048
251  404.778748  395.358095
252  405.992496  396.230476
253  406.884998  397.248095

[127 rows x 2 columns]
Moving Averages for SHOP
           MA8       MA21
508        NaN        NaN
509        NaN        NaN
510        NaN        NaN
511        NaN        NaN
512        NaN        NaN
..         ...        ...
630  75.191249  73.315714
631  75.553749  73.597143
632  75.969999  73.902380
633  76.511250  74.181904
634  77.054999  74.556190

[127 rows x 2 columns]
Moving Averages for SPY
            MA8        MA21
0           NaN         NaN
1           NaN         NaN
2           NaN         NaN
3           NaN         NaN
4           NaN         NaN
..          ...         ...
122  468.117500  460.409047
123  469.125000  461.075714
124  470.463753  462.000953
125  471.657501  462.888096
126  472.301250  463.857144

[127 rows x 2 columns]
```

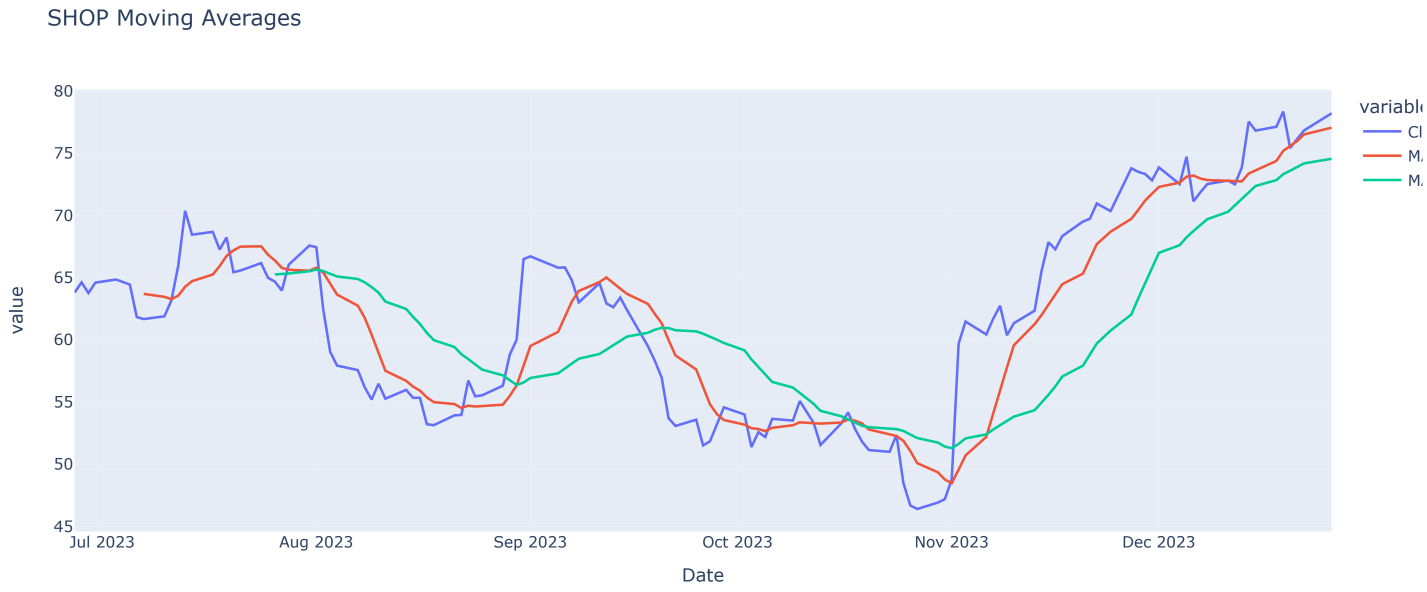
In [11]:

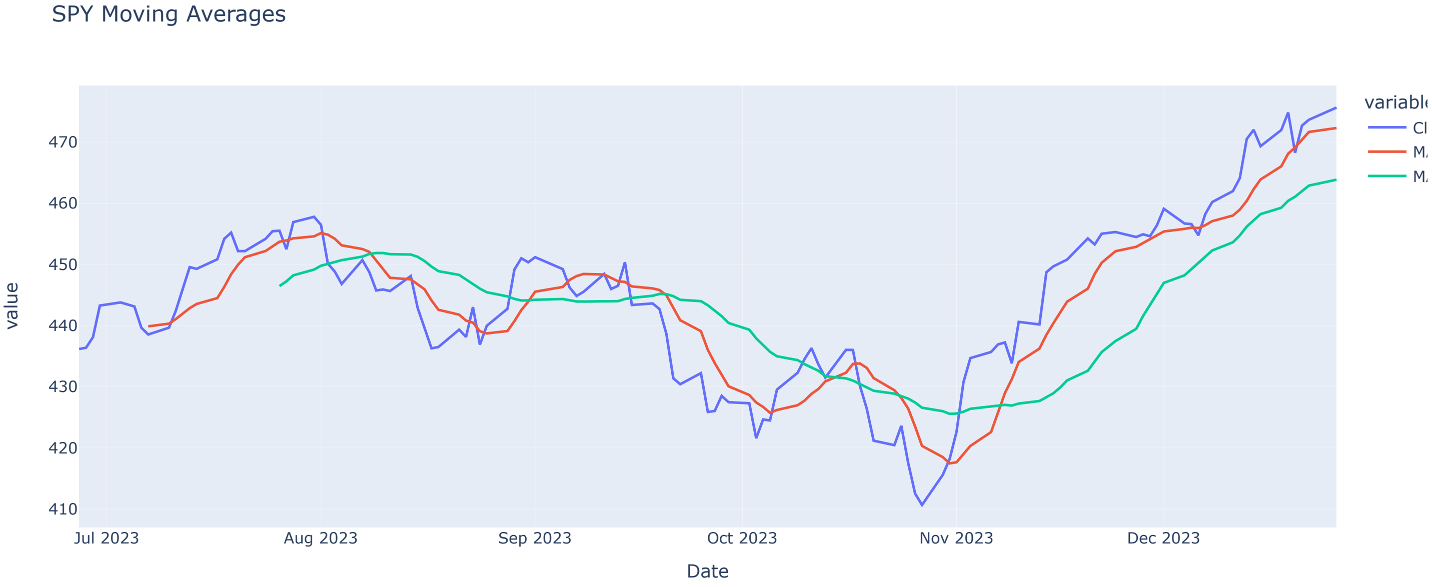
```
# visualize moving averages

for ticker, group in df.groupby('Ticker'):
    fig = px.line(group, x='Date', y=['Close', 'MA8', 'MA21'],
                  title= f'{ticker} Moving Averages')
    fig.show()
```


AAPL Moving Averages


AMD Moving Averages


QQQ Moving Averages



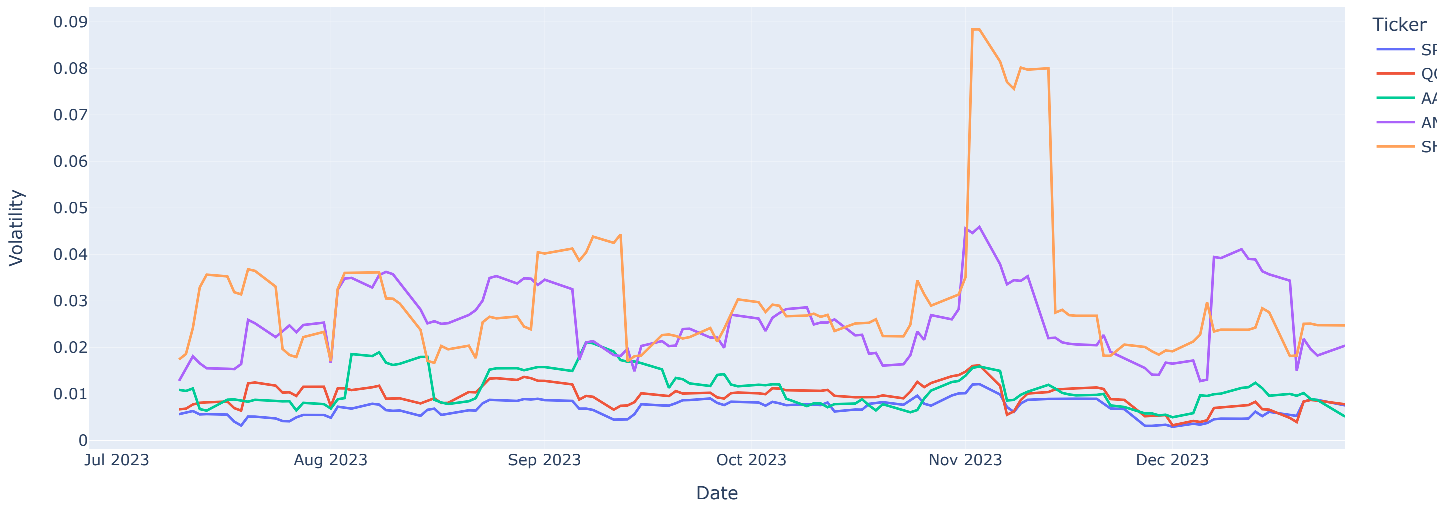


Volatility

In [13]:

```
# create volatility df

df['Volatility'] = df.groupby('Ticker')['Close'].pct_change().rolling(window=8).std().reset_index(0, drop=True)

#visualize
fig = px.line(df, x='Date', y='Volatility', color='Ticker', title='Volatility of My Top 5')
fig.show()
```

Volatility of My Top 5

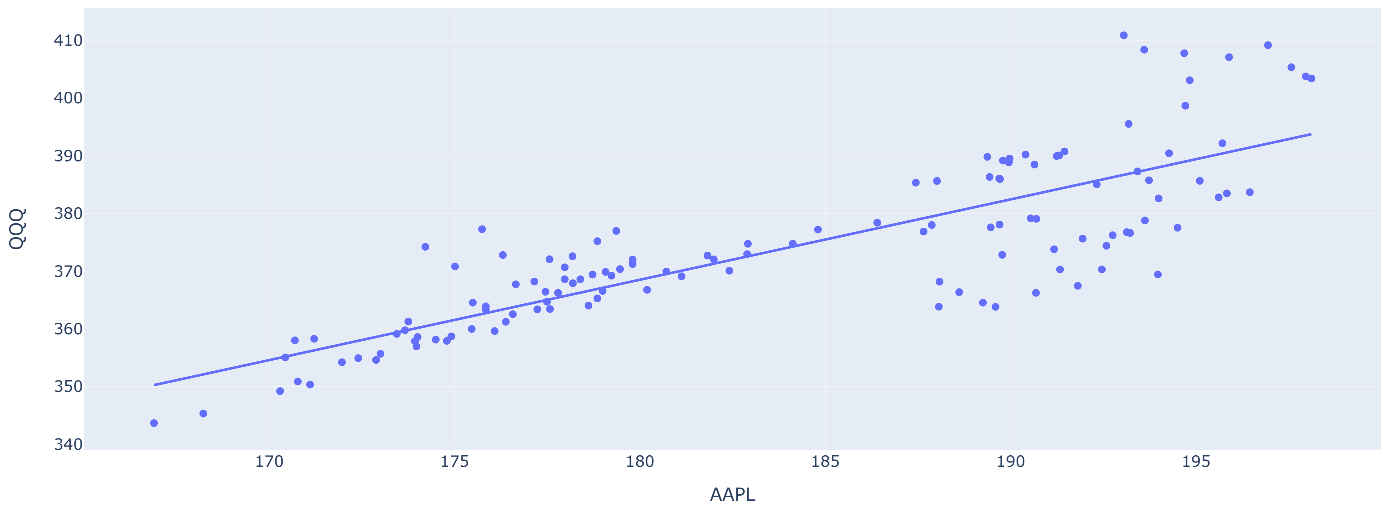


Correlation

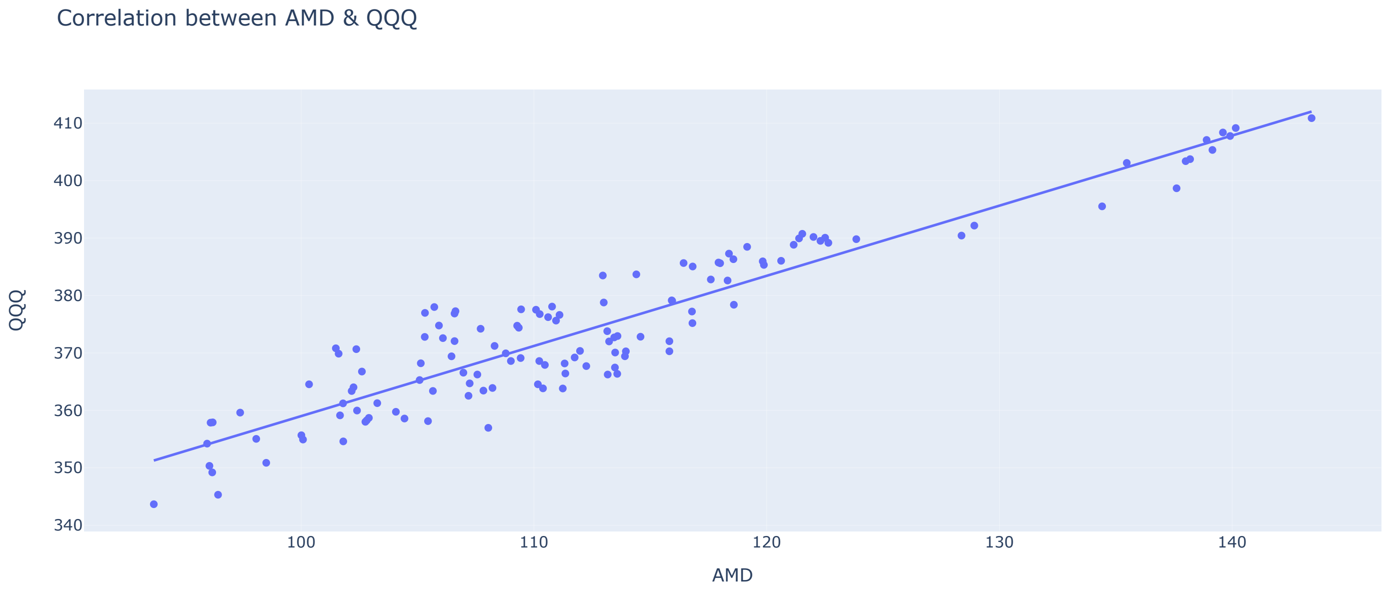
In [14]:
```python
import statsmodels.api as sm
```

In [15]:
```python
# create dataframe with AAPL & QQQ
apple = df.loc[df['Ticker'] == 'AAPL', ['Date', 'Close']].rename(columns={'Close':'AAPL'})
qqq = df.loc[df['Ticker'] == 'QQQ', ['Date', 'Close']].rename(columns={'Close':'QQQ'})
df_corr = pd.merge(apple, qqq, on= 'Date')

#create scatterplot to visualize

fig = px.scatter(df_corr, x='AAPL', y='QQQ', trendline='ols', title='Correlation between AAPL & QQQ')
fig.show()
```

Correlation between AAPL & QQQ

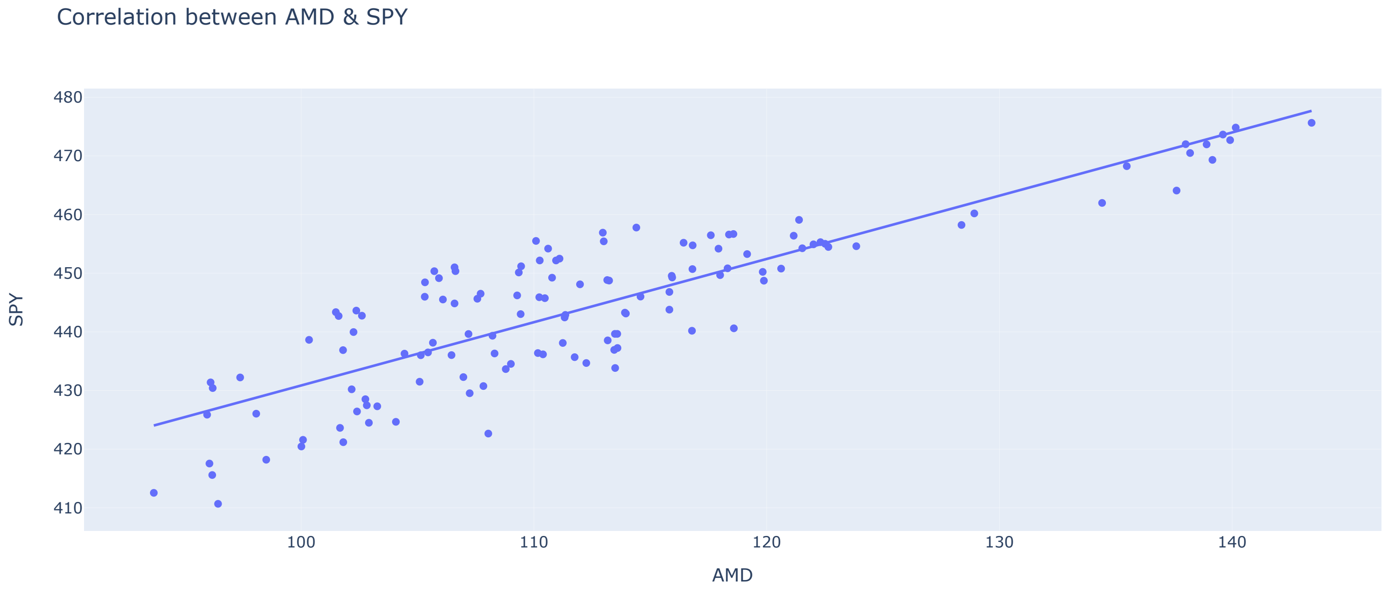


In [16]:
```python
# create dataframe with AMD & QQQ
amd = df.loc[df['Ticker'] == 'AMD', ['Date', 'Close']].rename(columns={'Close':'AMD'})
qqq = df.loc[df['Ticker'] == 'QQQ', ['Date', 'Close']].rename(columns={'Close':'QQQ'})
df_corr = pd.merge(amd, qqq, on= 'Date')

#create scatterplot to visualize

fig = px.scatter(df_corr, x='AMD', y='QQQ', trendline='ols', title='Correlation between AMD & QQQ')
fig.show()
```



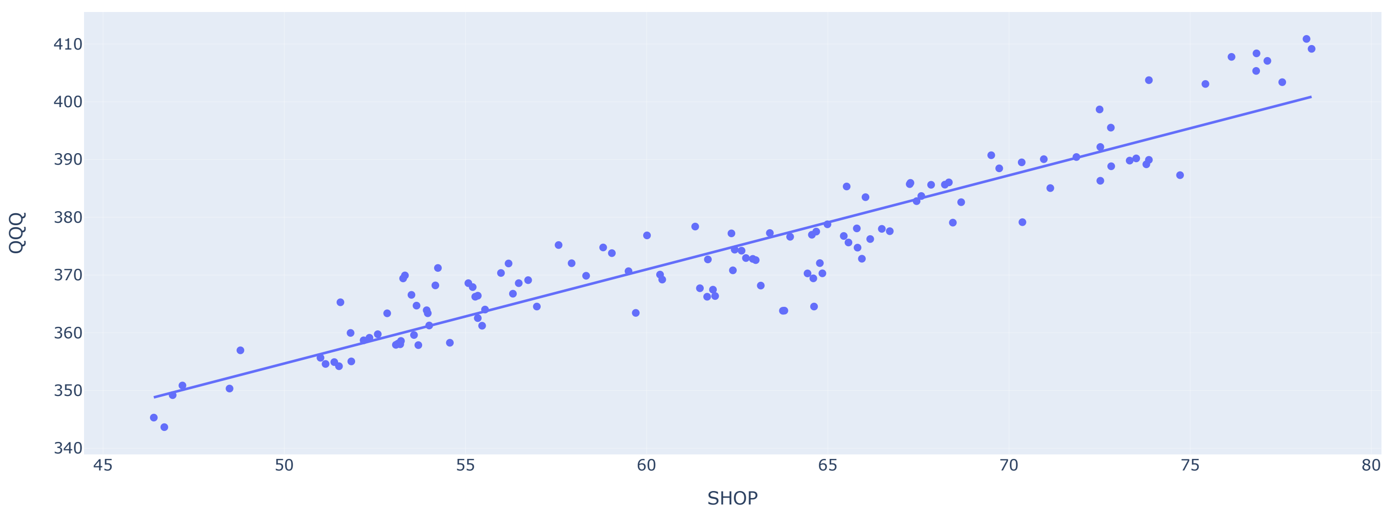
In [17]:

```
# create dataframe with AMD & SPY
amd = df.loc[df['Ticker'] == 'AMD', ['Date', 'Close']].rename(columns={'Close':'AMD'})
spy = df.loc[df['Ticker'] == 'SPY', ['Date', 'Close']].rename(columns={'Close':'SPY'})
df_corr = pd.merge(amd, spy, on= 'Date')

#create scatterplot to visualize

fig = px.scatter(df_corr, x='AMD', y='SPY', trendline='ols', title='Correlation between AMD & SPY')
fig.show()
```



In [18]:

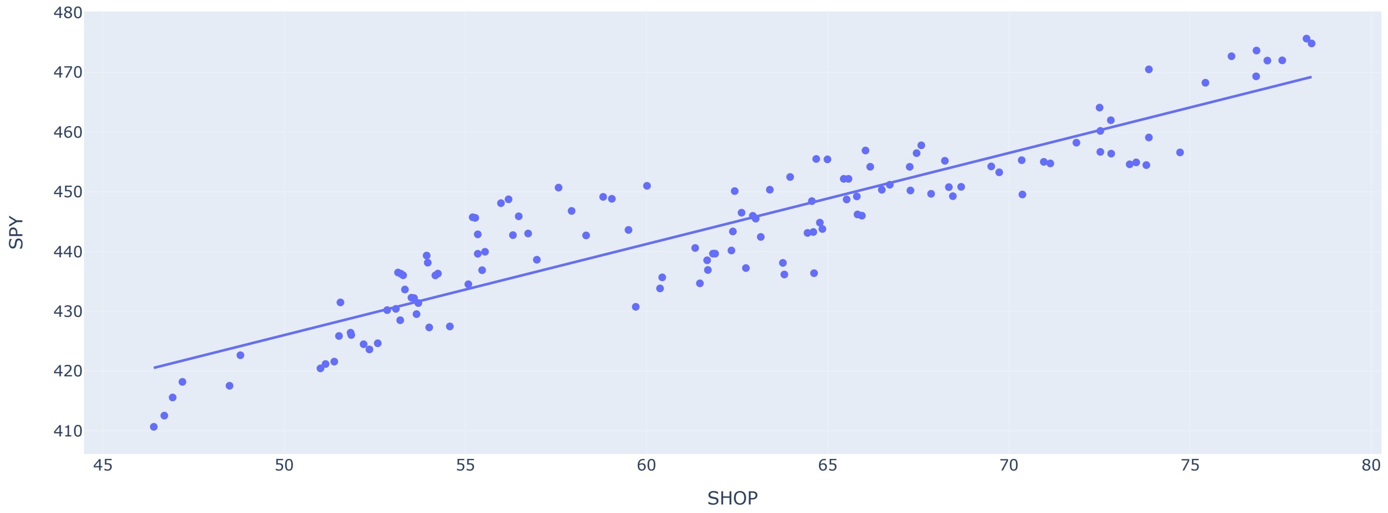
```
# create dataframe with SHOP & QQQ
shop = df.loc[df['Ticker'] == 'SHOP', ['Date', 'Close']].rename(columns={'Close':'SHOP'})
qqq = df.loc[df['Ticker'] == 'QQQ', ['Date', 'Close']].rename(columns={'Close':'QQQ'})
df_corr = pd.merge(shop, qqq, on= 'Date')

#create scatterplot to visualize

fig = px.scatter(df_corr, x='SHOP', y='QQQ', trendline='ols', title='Correlation between SHOP & QQQ')
fig.show()
```



Correlation between SHOP & QQQ

In [19]:

```
# create dataframe with SHOP & SPY
shop = df.loc[df['Ticker'] == 'SHOP', ['Date', 'Close']].rename(columns={'Close':'SHOP'})
spy = df.loc[df['Ticker'] == 'SPY', ['Date', 'Close']].rename(columns={'Close':'SPY'})
df_corr = pd.merge(shop, spy, on= 'Date')

#create scatterplot to visualize

fig = px.scatter(df_corr, x='SHOP', y='SPY', trendline='ols', title='Correlation between SHOP & SPY')
fig.show()
```

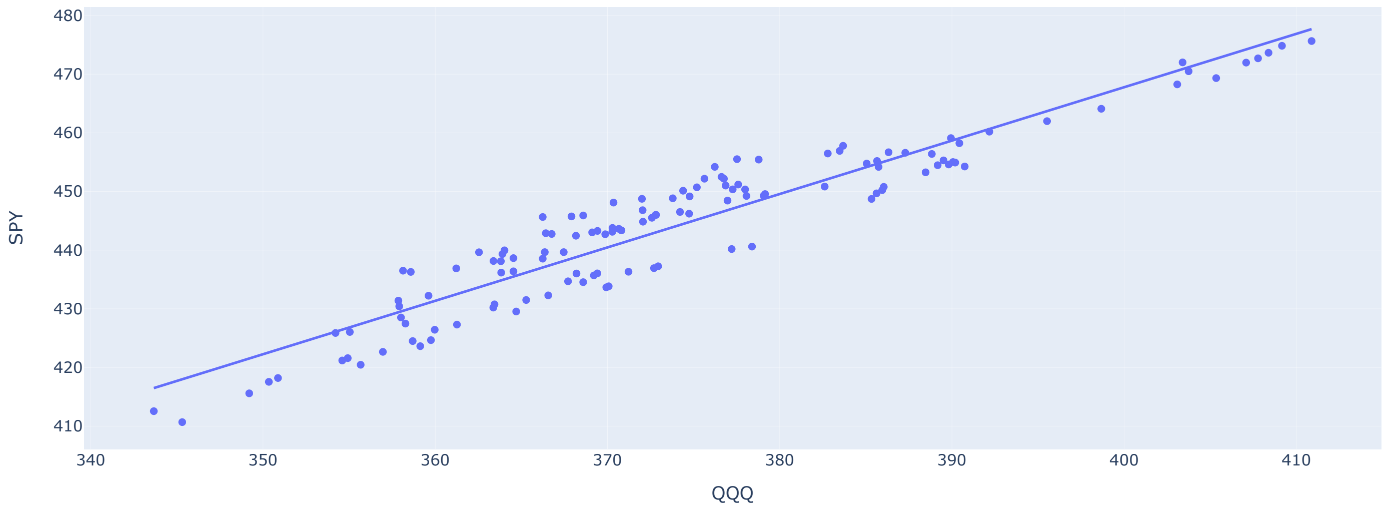


Correlation between SHOP & SPY

In [20]:

```
# create dataframe with QQQ & SPY
qqq = df.loc[df['Ticker'] == 'QQQ', ['Date', 'Close']].rename(columns={'Close':'QQQ'})
spy = df.loc[df['Ticker'] == 'SPY', ['Date', 'Close']].rename(columns={'Close':'SPY'})
df_corr = pd.merge(qqq, spy, on= 'Date')

#create scatterplot to visualize

fig = px.scatter(df_corr, x='QQQ', y='SPY', trendline='ols', title='Correlation between QQQ & SPY')
fig.show()
```

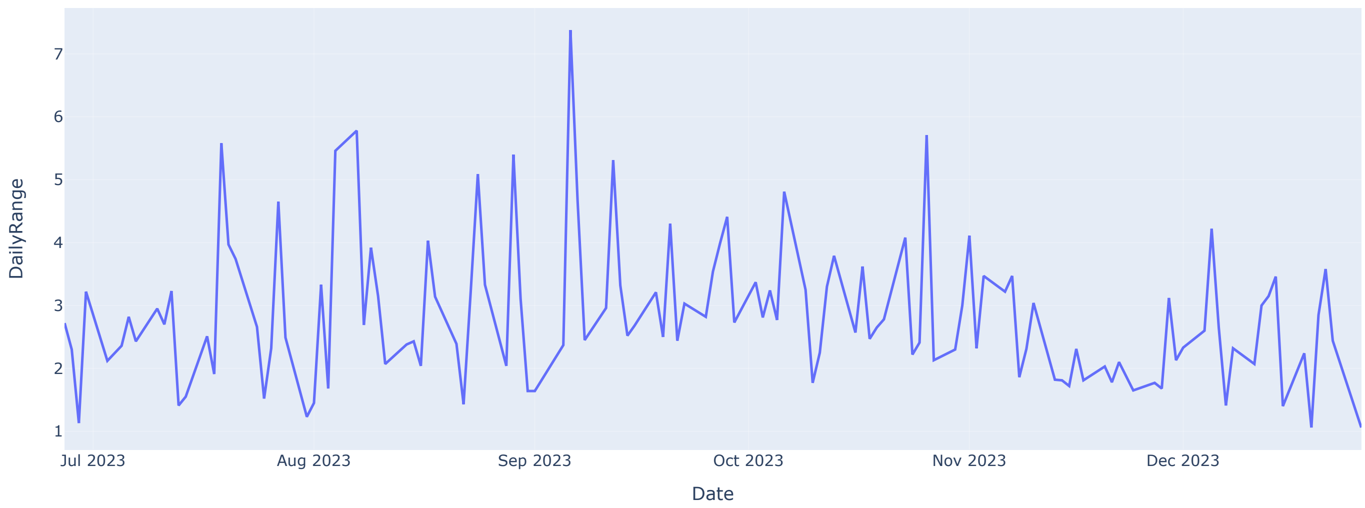
Correlation between QQQ & SPY



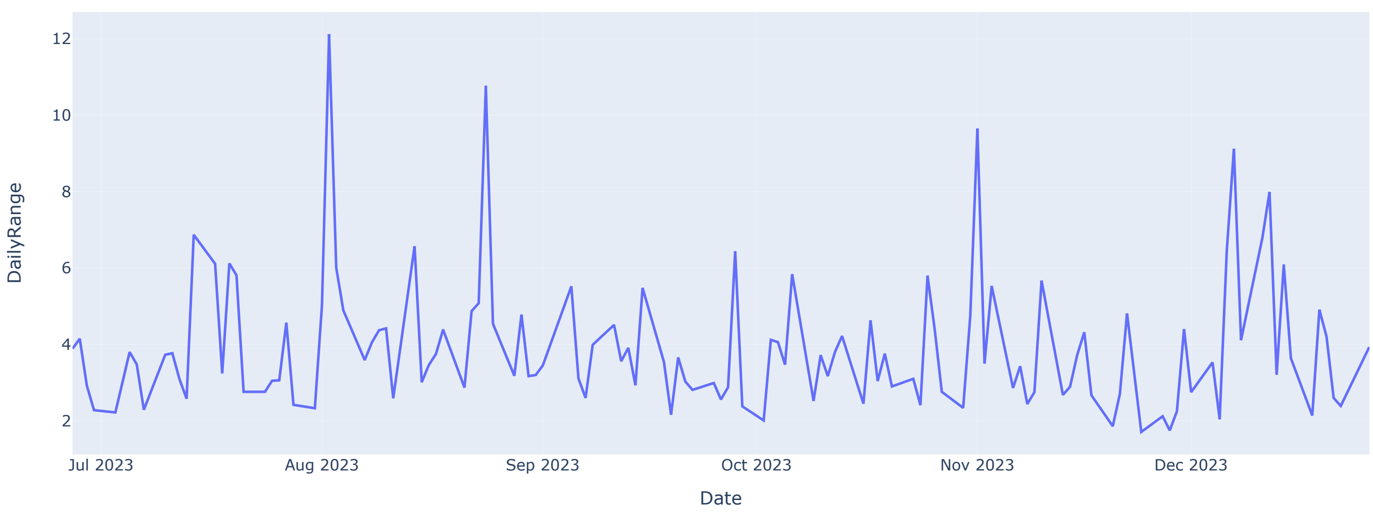
Daily Range

In [21]:
```
for ticker, group in df.groupby('Ticker'):
    # reset index to include data column
    group.reset_index(inplace=True)
    #plot daily range for each stock
    fig = px.line(group, x='Date', y='DailyRange',
                  title= f'{ticker} Daily Range')
    fig.show()
```

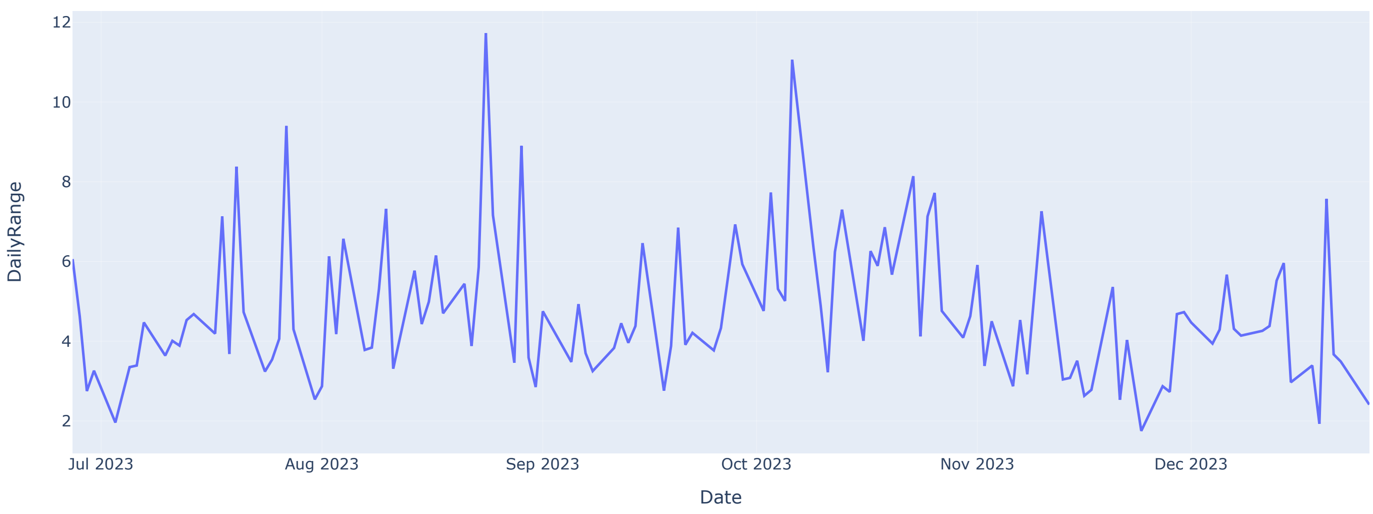
AAPL Daily Range



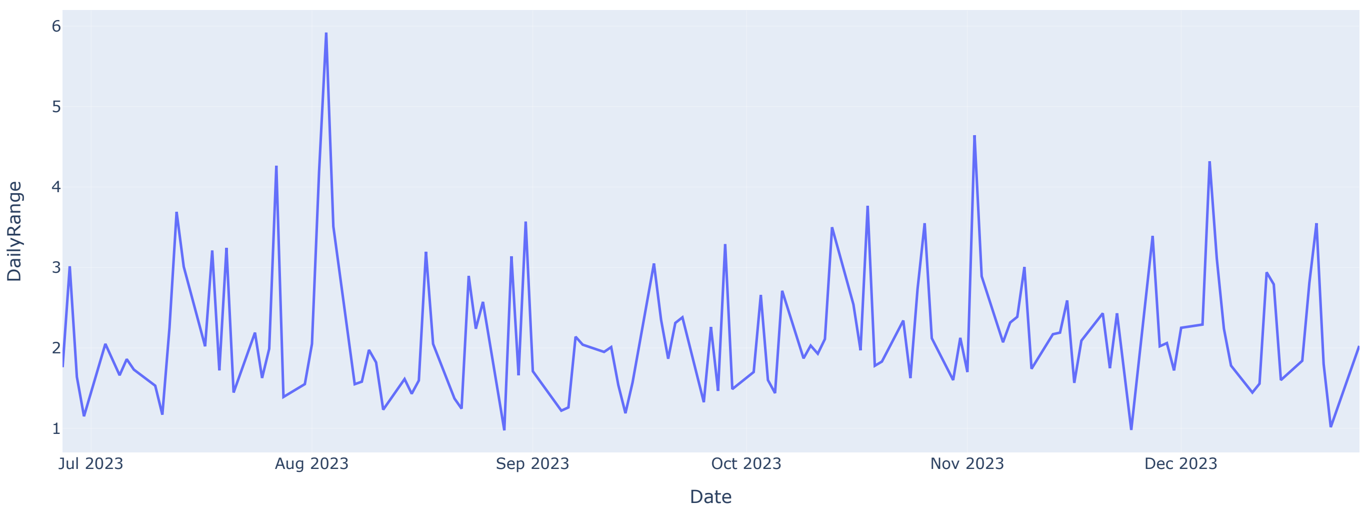
AMD Daily Range

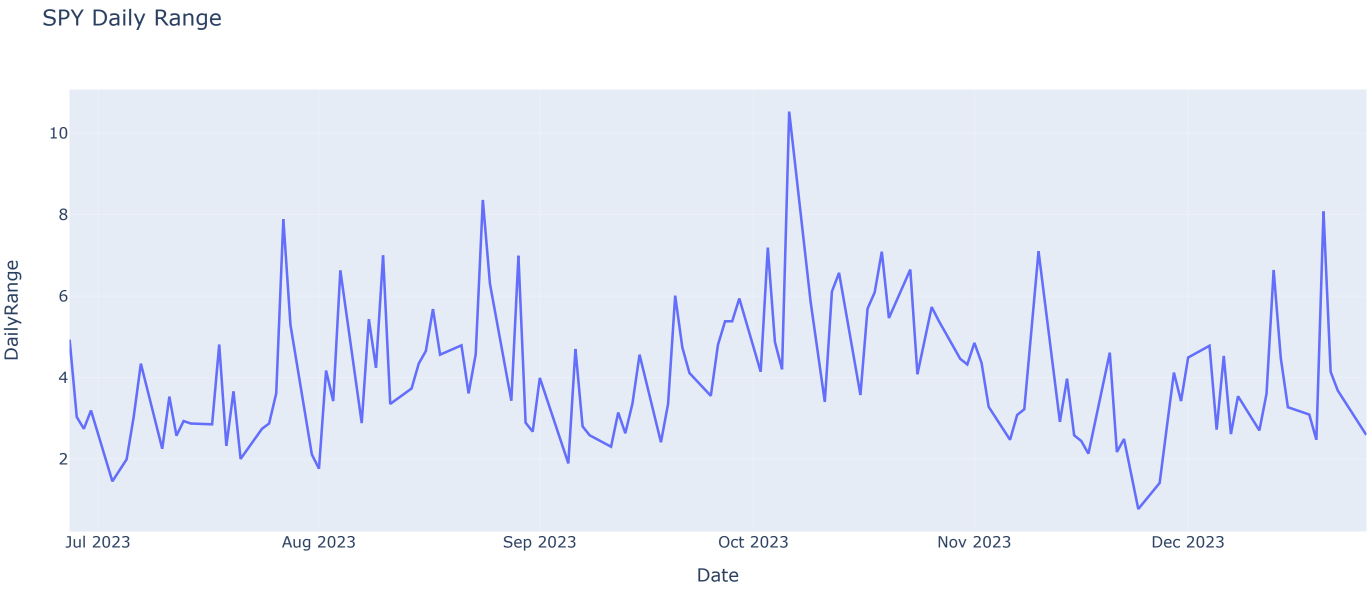


QQQ Daily Range



SHOP Daily Range




SPY Daily Range
DailyRange
10
8
6
4
2
Jul 2023
Aug 2023
Sep 2023
Oct 2023
Nov 2023
Dec 2023
Date