Stocks Visualization - Python / Plotly / D3 / Babel / .JS / Node / Heroku

Stocks Visualization and Analysis - Scraping / Fetching data Yahoo Finance & Quandly & Alpha_vantage using APIs and libs.

Fetching data using Yahoo Finance

In [1]:

```
#!pip install pandas-datareader
#!pip install fix-yahoo-finance
```

In [2]:

```
from pandas_datareader import data
import matplotlib.pyplot as plt
import pandas as pd
pd.core.common.is_list_like = pd.api.types.is_list_like
import pandas_datareader as pdr
```

In [3]:

In [4]:

```
start_date = datetime(200,1,1)
```

In [5]:

In [6]:

In [7]:

Out[7]:

```
{'zip': '98052',
 'sector': 'Technology',
 'fullTimeEmployees': 144000,
 'longBusinessSummary': "Microsoft Corporation develops, licenses, and supports software, se
rvices, devices, and solutions worldwide. The company's Productivity and Business Processes
segment offers Office, Exchange, SharePoint, Microsoft Teams, Office 365 Security and Compli
ance, and Skype for Business, as well as related Client Access Licenses (CAL); and Skype, Ou
tlook.com, and OneDrive. It also provides LinkedIn that includes Talent and marketing soluti
ons, and subscriptions; and Dynamics 365, a set of cloud-based and on-premises business solu
tions for small and medium businesses, large organizations, and divisions of enterprises. Th
e company's Intelligent Cloud segment licenses SQL and Windows Servers, Visual Studio, Syste
m Center, and related CALs; GitHub that provides a collaboration platform and code hosting s
ervice for developers; and Azure, a cloud platform. It also provides support services and Mi
crosoft consulting services to assist customers in developing, deploying, and managing Micro
soft server and desktop solutions; and training and certification to developers and IT profe
ssionals on various Microsoft products. The company's More Personal Computing segment offers
Windows OEM licensing and other non-volume licensing of the Windows operating system; Window
s Commercial comprising volume licensing of the Windows operating system, Windows cloud serv
ices, and other Windows commercial offerings; patent licensing; Windows Internet of Things;
```

In [8]:
```
hist = msft.history(period="max")
```

```
                 Open    High     Low   Close      Volume  Dividends  \
Date
1986-03-13    0.06    0.07    0.06    0.06  1031788800        0.0
1986-03-14    0.06    0.07    0.06    0.06   308160000        0.0
1986-03-17    0.06    0.07    0.06    0.07   133171200        0.0
1986-03-18    0.07    0.07    0.06    0.06    67766400        0.0
1986-03-19    0.06    0.06    0.06    0.06    47894400        0.0
...            ...     ...     ...     ...         ...        ...
2019-11-22  150.07  150.30  148.82  149.59    15901800        0.0
2019-11-25  150.00  151.35  149.92  151.23    22420900        0.0
2019-11-26  151.36  152.42  151.32  152.03    24620100        0.0
2019-11-27  152.33  152.50  151.52  152.32    15184400        0.0
2019-11-29  152.10  152.30  151.28  151.38    11977300        0.0

            Stock Splits
Date
1986-03-13           0.0
1986-03-14           0.0
1986-03-17           0.0
1986-03-18           0.0
1986-03-19           0.0
...                  ...
2019-11-22           0.0
2019-11-25           0.0
2019-11-26           0.0
2019-11-27           0.0
2019-11-29           0.0

[8500 rows x 7 columns]
```

In [9]:

```
[*********************100%***********************]  2 of 2 completed
```

In [10]:

Out[10]:

	Adj Close		Close		High		Low		Open	
	AAPL	SPY	AAPL	SPY	AAPL	SPY	AAPL	SPY	AAPL	SPY
Date										
2000-01-03	3.478462	100.470520	3.997768	145.437500	4.017857	148.250000	3.631696	143.875000	3.745536	148.
2000-01-04	3.185191	96.541473	3.660714	139.750000	3.950893	144.062500	3.613839	139.640625	3.866071	143.
2000-01-05	3.231803	96.714218	3.714286	140.000000	3.948661	141.531250	3.678571	137.250000	3.705357	139.
2000-01-06	2.952128	95.159882	3.392857	137.750000	3.821429	141.500000	3.392857	137.750000	3.790179	139.
2000-01-07	3.091966	100.686371	3.553571	145.750000	3.607143	145.750000	3.410714	140.062500	3.446429	140.
...	...	...	...	...	...	...	...	...	...	
2019-11-22	261.779999	310.959991	261.779999	310.959991	263.179993	311.239990	260.839996	309.850006	262.589996	311.
2019-11-25	266.369995	313.369995	266.369995	313.369995	266.440002	313.369995	262.519989	311.980011	262.709991	311.
2019-11-26	264.290009	314.079987	264.290009	314.079987	267.160004	314.279999	262.500000	313.059998	266.940002	313.
2019-11-27	267.839996	315.480011	267.839996	315.480011	267.980011	315.480011	265.309998	314.369995	265.579987	314.
2019-11-29	267.250000	314.309998	267.250000	314.309998	268.000000	315.130005	265.899994	314.059998	266.600006	314.

5010 rows × 12 columns

In [11]:
```
from pandas_datareader import data as pdr
import yfinance as yf
yf.pdr_override()
```

In [12]:

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

In [13]:

```
data
```

Out[13]:

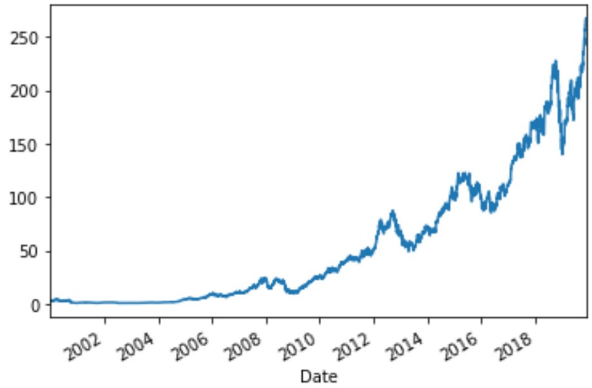
	Open	High	Low	Close	Adj Close	Volume
Date						
2000-01-03	3.745536	4.017857	3.631696	3.997768	3.478462	133949200
2000-01-04	3.866071	3.950893	3.613839	3.660714	3.185191	128094400
2000-01-05	3.705357	3.948661	3.678571	3.714286	3.231803	194580400
2000-01-06	3.790179	3.821429	3.392857	3.392857	2.952128	191993200
2000-01-07	3.446429	3.607143	3.410714	3.553571	3.091966	115183600
...	...	...	...	...	...	...
2019-11-22	262.589996	263.179993	260.839996	261.779999	261.779999	16331300
2019-11-25	262.709991	266.440002	262.519989	266.369995	266.369995	21005100
2019-11-26	266.940002	267.160004	262.500000	264.290009	264.290009	26301900
2019-11-27	265.579987	267.980011	265.309998	267.839996	267.839996	16308900
2019-11-29	266.600006	268.000000	265.899994	267.250000	267.250000	11654400

5010 rows × 6 columns

In [14]:

In [15]:

```
data['Adj Close'].plot()
plt.show()
```



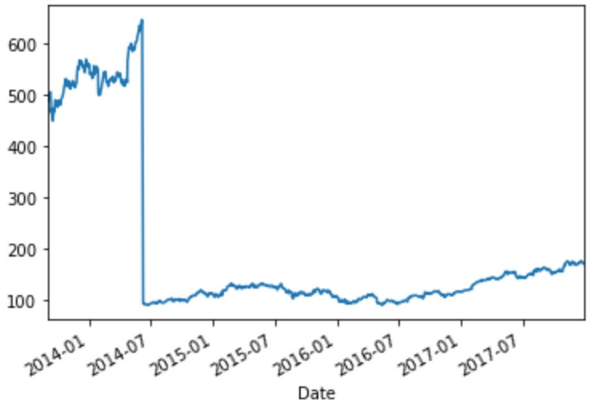
Fetching data using Quandl

In [16]:

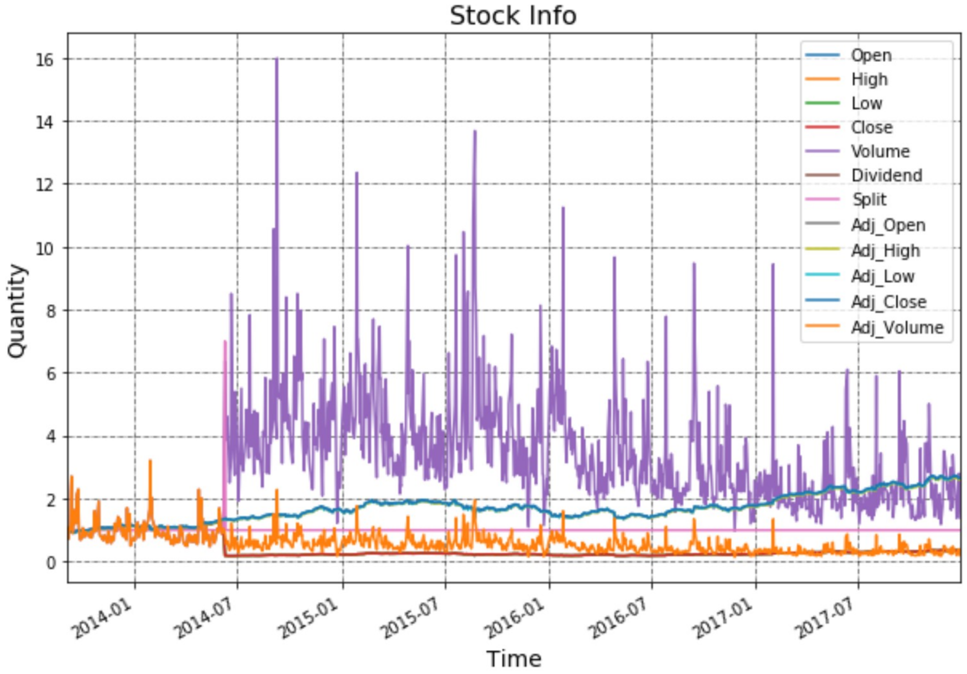
In [17]:

In [18]:

In [19]:
```
data.Close.plot()
```



In [20]:
```
((data.pct_change()+1).cumprod()).plot(figsize=(10, 7))
plt.legend()
plt.title('Stock Info', fontsize=16)
plt.ylabel('Quantity', fontsize=14)
plt.xlabel('Time', fontsize=14)
plt.grid(which="major", color='k', linestyle='-.', linewidth=0.5)
```



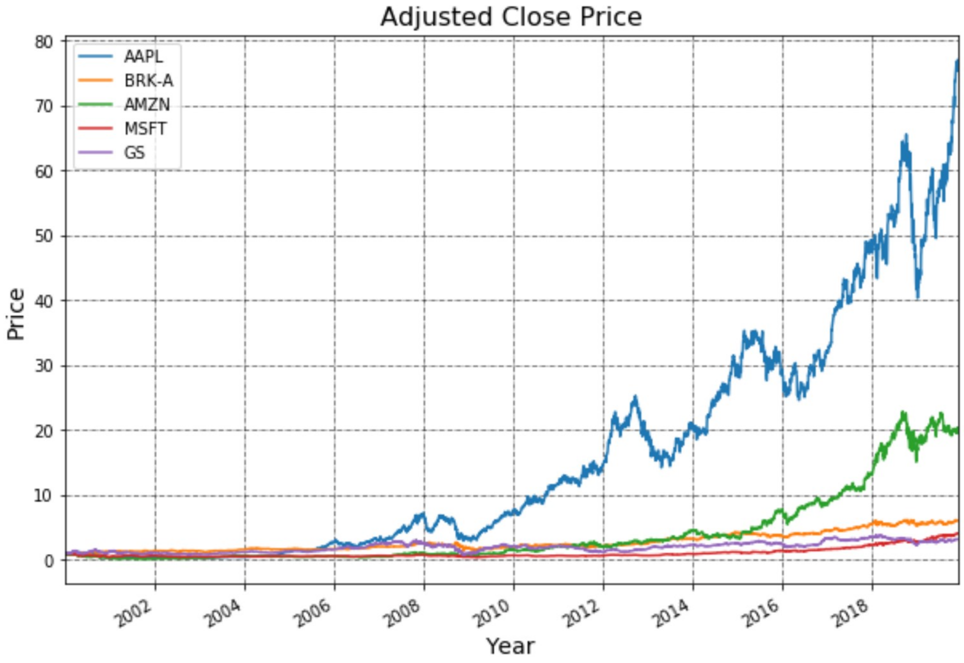
In [21]:
```
#Data for multiple stocks
import pandas as pd
tickers_list = ["AAPL","BRK-A","AMZN","MSFT","GS"]
data = pd.DataFrame(columns=tickers_list)
import yfinance as yf
for ticker in tickers_list:
 data[ticker] = yf.download(ticker,"2000-01-01","2020-01-01")['Adj Close']
```

```
[*********************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
```

Out[21]:

	AAPL	BRK-A	AMZN	MSFT	GS
Date					
2000-01-03	3.478462	54800.0	89.3750	37.495686	70.762970
2000-01-04	3.185191	52000.0	81.9375	36.229057	66.305847
2000-01-05	3.231803	53200.0	69.7500	36.611080	63.200893
2000-01-06	2.952128	55000.0	65.5625	35.384666	65.905197
2000-01-07	3.091966	56500.0	69.5625	35.847076	66.155640

In [22]:
```
((data.pct_change()+1).cumprod()).plot(figsize=(10, 7))
plt.legend()
plt.title('Adjusted Close Price', fontsize=16)
plt.ylabel('Price', fontsize=14)
plt.xlabel('Year', fontsize=14)
plt.grid(which="major", color='k', linestyle='-.', linewidth=0.5)
```



Fetching data using Alpha_Vantage

In [23]:
```
#Minute level data using the Alpha vantage - Intraday Trading - High-Frequency Trading
```

In [24]:

In [25]:
```
ts = TimeSeries(key='Your-API-Key', output_format='pandas')
data, meta_data = ts.get_intraday(symbol='MSFT',interval='1min', outputsize='compact')
```

```
                     1. open  2. high  3. low  4. close  5. volume
date
2019-11-29 13:01:00  151.3800   151.38  151.38  151.3800  3873242.0
2019-11-29 13:00:00  151.5200   151.75  151.38  151.3800   810279.0
2019-11-29 12:59:00  151.4400   151.53  151.42  151.5200   158602.0
2019-11-29 12:58:00  151.3700   151.47  151.37  151.4500   108235.0
2019-11-29 12:57:00  151.4747   151.48  151.36  151.3656    79403.0
```

Stock Analysis using Plotly

In [26]:

In [27]:
```
import pandas as pd
import numpy as np
import datetime
import matplotlib.pyplot as plt
%matplotlib inline
from plotly import __version__
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot

import plotly.graph_objs as go

print(__version__) # requires version >= 1.9.0
```

```
4.3.0
```

In [28]:
```
import pandas as pd
from pandas import ExcelWriter
from pandas import ExcelFile

portfolio_df = pd.read_excel('stocks acquisition costs dates.xlsx')
```

Out[28]:

	Acquisition Date	Ticker	Quantity	Unit Cost	Cost Basis	Start of Year
0	2013-02-07	AAPL	125	65.40	8175.00	2018-12-28
1	2014-02-27	JNJ	100	81.90	8190.00	2018-12-28
2	2013-06-04	MCD	100	84.99	8499.00	2018-12-28
3	2015-12-14	MTCH	600	13.63	8178.00	2018-12-28
4	2016-01-14	NFLX	75	108.71	8153.25	2018-12-28
5	2013-08-14	WMT	125	68.30	8537.50	2018-12-28
6	2013-12-13	FB	150	53.32	7998.00	2018-12-28
7	2015-01-05	TWTR	225	36.38	8185.50	2018-12-28

In [29]:
```
start_sp = datetime.datetime(2010, 1, 1)
end_sp = datetime.datetime(2019, 11, 29)

# This variable is used for YTD performance.
end_of_last_year = datetime.datetime(2018, 12, 28)
stocks_start = datetime.datetime(2010, 1, 1)
```

In [30]:
```
from pandas_datareader import data as pdr
import yfinance as yf
```

In [31]:
```
sp500 = pdr.get_data_yahoo("^GSPC", start="2000-01-01", end="2020-01-01")
```

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

Out[31]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2000-01-03	1469.250000	1478.000000	1438.359985	1455.219971	1455.219971	931800000
2000-01-04	1455.219971	1455.219971	1397.430054	1399.420044	1399.420044	1009000000
2000-01-05	1399.420044	1413.270020	1377.680054	1402.109985	1402.109985	1085500000
2000-01-06	1402.109985	1411.900024	1392.099976	1403.449951	1403.449951	1092300000
2000-01-07	1403.449951	1441.469971	1400.729980	1441.469971	1441.469971	1225200000

In [32]:

Out[32]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2019-11-22	3111.409912	3112.870117	3099.260010	3110.290039	3110.290039	3226780000
2019-11-25	3117.439941	3133.830078	3117.439941	3133.639893	3133.639893	3511530000
2019-11-26	3134.850098	3142.689941	3131.000000	3140.520020	3140.520020	4595590000
2019-11-27	3145.489990	3154.260010	3143.409912	3153.629883	3153.629883	3033090000
2019-11-29	3147.179932	3150.300049	3139.340088	3140.979980	3140.979980	174302000000

In [33]:

In [34]:

Out[34]:

	Date	Adj Close
5005	2019-11-22	3110.290039
5006	2019-11-25	3133.639893
5007	2019-11-26	3140.520020
5008	2019-11-27	3153.629883
5009	2019-11-29	3140.979980

In [35]:
```
sp_500_adj_close_start = sp_500_adj_close[sp_500_adj_close['Date']==end_of_last_year]
```

Out[35]:

	Date	Adj Close
4777	2018-12-28	2485.73999

In [36]:
```
tickers = portfolio_df['Ticker'].unique()
```

Out[36]:
```
array(['AAPL', 'JNJ', 'MCD', 'MTCH', 'NFLX', 'WMT', 'FB', 'TWTR'],
      dtype=object)
```

In [37]:

```
def get(tickers, startdate, enddate):
    def data(ticker):
        return (pdr.get_data_yahoo(ticker, start=startdate, end=enddate))
    datas = map(data, tickers)
    return(pd.concat(datas, keys=tickers, names=['Ticker', 'Date']))
```

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

In [38]:

Out[38]:

		Open	High	Low	Close	Adj Close	Volume
Ticker	**Date**						
AAPL	**2010-01-04**	30.490000	30.642857	30.340000	30.572857	26.601469	123432400
	2010-01-05	30.657143	30.798571	30.464285	30.625713	26.647457	150476200
	2010-01-06	30.625713	30.747143	30.107143	30.138571	26.223597	138040000
	2010-01-07	30.250000	30.285715	29.864286	30.082857	26.175119	119282800
	2010-01-08	30.042856	30.285715	29.865715	30.282858	26.349140	111902700

In [39]:

```
adj_close = all_data[['Adj Close']].reset_index()
```

Out[39]:

	Ticker	Date	Adj Close
0	AAPL	2010-01-04	26.601469
1	AAPL	2010-01-05	26.647457
2	AAPL	2010-01-06	26.223597
3	AAPL	2010-01-07	26.175119
4	AAPL	2010-01-08	26.349140

In [40]:

```
adj_close_start = adj_close[adj_close['Date']==end_of_last_year]
```

Out[40]:

	Ticker	Date	Adj Close
2262	AAPL	2018-12-28	153.917389
4757	JNJ	2018-12-28	123.780724
7252	MCD	2018-12-28	171.422867
8266	MTCH	2018-12-28	42.070000
10761	NFLX	2018-12-28	256.079987

In [41]:
```
adj_close_latest = adj_close[adj_close['Date']==stocks_end]
```

Out[41]:

	Ticker	Date	Adj Close
2494	AAPL	2019-11-29	267.250000
4989	JNJ	2019-11-29	137.490005
7484	MCD	2019-11-29	194.479996
8498	MTCH	2019-11-29	70.480003
10993	NFLX	2019-11-29	314.660004
13488	WMT	2019-11-29	119.089996
15384	FB	2019-11-29	201.639999
16910	TWTR	2019-11-29	30.910000

In [42]:
```
adj_close_latest.set_index('Ticker', inplace=True)
```

Out[42]:

	Date	Adj Close
Ticker		
AAPL	2019-11-29	267.250000
JNJ	2019-11-29	137.490005
MCD	2019-11-29	194.479996
MTCH	2019-11-29	70.480003
NFLX	2019-11-29	314.660004

In [43]:
```
portfolio_df.set_index(['Ticker'], inplace=True)
```

Out[43]:

	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year
Ticker					
AAPL	2013-02-07	125	65.40	8175.00	2018-12-28
JNJ	2014-02-27	100	81.90	8190.00	2018-12-28
MCD	2013-06-04	100	84.99	8499.00	2018-12-28
MTCH	2015-12-14	600	13.63	8178.00	2018-12-28
NFLX	2016-01-14	75	108.71	8153.25	2018-12-28

In [44]:
```
merged_portfolio = pd.merge(portfolio_df, adj_close_latest, left_index=True, right_index=True)
```

Out[44]:

	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Date	Adj Close
Ticker							
AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000
JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005
MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996
MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003
NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004

In [45]:
```
merged_portfolio['ticker return'] = merged_portfolio['Adj Close'] / merged_portfolio['Unit Cost
```

Out[45]:

	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Date	Adj Close	ticker return
Ticker								
AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391
JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755
MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269
MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947
NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490
WMT	2013-08-14	125	68.30	8537.50	2018-12-28	2019-11-29	119.089996	0.743631
FB	2013-12-13	150	53.32	7998.00	2018-12-28	2019-11-29	201.639999	2.781695
TWTR	2015-01-05	225	36.38	8185.50	2018-12-28	2019-11-29	30.910000	-0.150357

In [46]:
```

```

In [47]:
```
merged_portfolio_sp = pd.merge(merged_portfolio, sp_500_adj_close, left_on='Acquisition Date',
```

Out[47]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Date_x	Adj Close_x	ticker return	Date_y	Adj Close_y
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	2013-02-07	1509.390015
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	2014-02-27	1854.290039
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	2013-06-04	1631.380005
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2015-12-14	2021.939941
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	2016-01-14	1921.839966

In [48]:
```
del merged_portfolio_sp['Date_y']

merged_portfolio_sp.rename(columns={'Date_x': 'Latest Date', 'Adj Close_x': 'Ticker Adj Close'
                                    , 'Adj Close_y': 'SP 500 Initial Close'}, inplace=True)
```

Out[48]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966

In [49]:
```
merged_portfolio_sp['Equiv SP Shares'] = merged_portfolio_sp['Cost Basis'] / merged_portfolio_s
merged_portfolio_sp.head()
```

Out[49]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	Equiv SP Shares
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	5.416095
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	4.416785
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	5.209700
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	4.044631
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	4.242419

In [50]:

```
merged_portfolio_sp_latest = pd.merge(merged_portfolio_sp, sp_500_adj_close, left_on='Latest Da
```

Out[50]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	Equiv SP Shares	D
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	5.416095	2019-11
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	4.416785	2019-11
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	5.209700	2019-11
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	4.044631	2019-11
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	4.242419	2019-11

In [51]:

```
del merged_portfolio_sp_latest['Date']

merged_portfolio_sp_latest.rename(columns={'Adj Close': 'SP 500 Latest Close'}, inplace=True)
```

Out[51]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	Equiv SP Shares	SP La Cl
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	5.416095	3140.97
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	4.416785	3140.97
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	5.209700	3140.97
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	4.044631	3140.97
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	4.242419	3140.97

In [52]:

```
merged_portfolio_sp_latest['SP Return'] = merged_portfolio_sp_latest['SP 500 Latest Close'] / m

merged_portfolio_sp_latest['Abs. Return Compare'] = merged_portfolio_sp_latest['ticker return']

merged_portfolio_sp_latest['Ticker Share Value'] = merged_portfolio_sp_latest['Quantity'] * mer

merged_portfolio_sp_latest['SP 500 Value'] = merged_portfolio_sp_latest['Equiv SP Shares'] * me

merged_portfolio_sp_latest['Abs Value Compare'] = merged_portfolio_sp_latest['Ticker Share Valu

merged_portfolio_sp_latest['Stock Gain / (Loss)'] = merged_portfolio_sp_latest['Ticker Share Va

merged_portfolio_sp_latest['SP 500 Gain / (Loss)'] = merged_portfolio_sp_latest['SP 500 Value']

merged_portfolio_sp_latest.head()
```

Out[52]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	Equiv SP Shares	SP La Cl
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	5.416095	3140.97
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	4.416785	3140.97
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	5.209700	3140.97
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	4.044631	3140.97
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	4.242419	3140.97

In [53]:

```
merged_portfolio_sp_latest_YTD = pd.merge(merged_portfolio_sp_latest, adj_close_start, on='Tick
```

Out[53]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	...	SP 500 Latest Close	
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	...	3140.97998	1
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	...	3140.97998	0
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	...	3140.97998	0
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	...	3140.97998	0
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	...	3140.97998	0

5 rows × 21 columns

In [54]:

```
del merged_portfolio_sp_latest_YTD['Date']

merged_portfolio_sp_latest_YTD.rename(columns={'Adj Close': 'Ticker Start Year Close'}, inplace
```

Out[54]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	Equiv SP Shares	SP La Cl
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	5.416095	3140.97
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	4.416785	3140.97
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	5.209700	3140.97
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	4.044631	3140.97
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	4.242419	3140.97

In [55]:

```
merged_portfolio_sp_latest_YTD_sp = pd.merge(merged_portfolio_sp_latest_YTD, sp_500_adj_close_s
                                             , left_on='Start of Year', right_on='Date')
```

Out[55]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	...	SP Return	I Co
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	...	1.080960	2.0
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	...	0.693899	-0.0
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	...	0.925352	0.3
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	...	0.553449	3.6
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	...	0.634361	1.2

5 rows × 22 columns

In [56]:
```
del merged_portfolio_sp_latest_YTD_sp['Date']

merged_portfolio_sp_latest_YTD_sp.rename(columns={'Adj Close': 'SP Start Year Close'}, inplace=

merged_portfolio_sp_latest_YTD_sp['Share YTD'] = merged_portfolio_sp_latest_YTD_sp['Ticker Adj

merged_portfolio_sp_latest_YTD_sp['SP 500 YTD'] = merged_portfolio_sp_latest_YTD_sp['SP 500 Lat

merged_portfolio_sp_latest_YTD_sp.head()
```

Out[56]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	...	Abs. Return Compare	Ti
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	...	2.005432	33
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	...	-0.015144	13
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	...	0.362918	19
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	...	3.617498	42
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	...	1.260129	23

5 rows × 23 columns

In [57]:
```
merged_portfolio_sp_latest_YTD_sp = merged_portfolio_sp_latest_YTD_sp.sort_values(by='Ticker',
```

Out[57]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	...	Abs. Return Compare	T
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	...	2.005432	33
6	FB	2013-12-13	150	53.32	7998.00	2018-12-28	2019-11-29	201.639999	2.781695	1775.319946	...	2.012448	30
1	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	...	-0.015144	13
2	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	...	0.362918	19
3	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	...	3.617498	42
4	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	...	1.260129	23
7	TWTR	2015-01-05	225	36.38	8185.50	2018-12-28	2019-11-29	30.910000	-0.150357	2020.579956	...	-0.704852	6
5	WMT	2013-08-14	125	68.30	8537.50	2018-12-28	2019-11-29	119.089996	0.743631	1685.390015	...	-0.120021	14

8 rows × 23 columns

In [58]:
```
merged_portfolio_sp_latest_YTD_sp['Cum Invst'] = merged_portfolio_sp_latest_YTD_sp['Cost Basis'

merged_portfolio_sp_latest_YTD_sp['Cum Ticker Returns'] = merged_portfolio_sp_latest_YTD_sp['Ti

merged_portfolio_sp_latest_YTD_sp['Cum SP Returns'] = merged_portfolio_sp_latest_YTD_sp['SP 500

merged_portfolio_sp_latest_YTD_sp['Cum Ticker ROI Mult'] = merged_portfolio_sp_latest_YTD_sp['C

merged_portfolio_sp_latest_YTD_sp.head()
```

Out[58]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	...	Stock Gain / (Loss)	S
0	AAPL	2013-02-07	125	65.40	8175.0	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	...	25231.250000	8
6	FB	2013-12-13	150	53.32	7998.0	2018-12-28	2019-11-29	201.639999	2.781695	1775.319946	...	22247.999908	6
1	JNJ	2014-02-27	100	81.90	8190.0	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	...	5559.000549	5
2	MCD	2013-06-04	100	84.99	8499.0	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	...	10948.999573	7
3	MTCH	2015-12-14	600	13.63	8178.0	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	...	34110.002014	4

5 rows × 27 columns

In [59]:

Out[59]:

	Ticker	Date	Adj Close
0	AAPL	2010-01-04	26.601469
1	AAPL	2010-01-05	26.647457
2	AAPL	2010-01-06	26.223597
3	AAPL	2010-01-07	26.175119
4	AAPL	2010-01-08	26.349140

In [60]:

Out[60]:

	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year
Ticker					
AAPL	2013-02-07	125	65.40	8175.00	2018-12-28
JNJ	2014-02-27	100	81.90	8190.00	2018-12-28
MCD	2013-06-04	100	84.99	8499.00	2018-12-28
MTCH	2015-12-14	600	13.63	8178.00	2018-12-28
NFLX	2016-01-14	75	108.71	8153.25	2018-12-28

In [61]:
```
portfolio_df.reset_index(inplace=True)

adj_close_acq_date = pd.merge(adj_close, portfolio_df, on='Ticker')
```

Out[61]:

	Ticker	Date	Adj Close	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year
0	AAPL	2010-01-04	26.601469	2013-02-07	125	65.4	8175.0	2018-12-28
1	AAPL	2010-01-05	26.647457	2013-02-07	125	65.4	8175.0	2018-12-28
2	AAPL	2010-01-06	26.223597	2013-02-07	125	65.4	8175.0	2018-12-28
3	AAPL	2010-01-07	26.175119	2013-02-07	125	65.4	8175.0	2018-12-28
4	AAPL	2010-01-08	26.349140	2013-02-07	125	65.4	8175.0	2018-12-28

In [62]:
```
del adj_close_acq_date['Quantity']
del adj_close_acq_date['Unit Cost']
del adj_close_acq_date['Cost Basis']
del adj_close_acq_date['Start of Year']
```

In [63]:
```
adj_close_acq_date['Date Delta'] = adj_close_acq_date['Date'] - adj_close_acq_date['Acquisition

adj_close_acq_date['Date Delta'] = adj_close_acq_date[['Date Delta']].apply(pd.to_numeric)
```

Out[63]:

	Ticker	Date	Adj Close	Acquisition Date	Date Delta
0	AAPL	2010-01-04	26.601469	2013-02-07	-97632000000000000
1	AAPL	2010-01-05	26.647457	2013-02-07	-97545600000000000
2	AAPL	2010-01-06	26.223597	2013-02-07	-97459200000000000
3	AAPL	2010-01-07	26.175119	2013-02-07	-97372800000000000
4	AAPL	2010-01-08	26.349140	2013-02-07	-97286400000000000

In [64]:
```
adj_close_acq_date_modified = adj_close_acq_date[adj_close_acq_date['Date Delta']>=0]
```

Out[64]:

	Ticker	Date	Adj Close	Acquisition Date	Date Delta
779	AAPL	2013-02-07	59.058853	2013-02-07	0
780	AAPL	2013-02-08	59.911518	2013-02-07	86400000000000
781	AAPL	2013-02-11	60.535877	2013-02-07	345600000000000
782	AAPL	2013-02-12	59.018475	2013-02-07	432000000000000
783	AAPL	2013-02-13	58.906216	2013-02-07	518400000000000

In [65]:
```
adj_close_pivot = adj_close_acq_date_modified.pivot_table(index=['Ticker', 'Acquisition Date'],

adj_close_pivot.reset_index(inplace=True)

adj_close_pivot
```

Out[65]:

	Ticker	Acquisition Date	Adj Close
0	AAPL	2013-02-07	267.839996
1	FB	2013-12-13	217.500000
2	JNJ	2014-02-27	143.786758
3	MCD	2013-06-04	218.628891
4	MTCH	2015-12-14	91.769997
5	NFLX	2016-01-14	418.970001
6	TWTR	2015-01-05	52.869999
7	WMT	2013-08-14	120.980003

In [66]:
```
adj_close_pivot_merged = pd.merge(adj_close_pivot, adj_close
                                             , on=['Ticker', 'Adj Close'])
```

Out[66]:

	Ticker	Acquisition Date	Adj Close	Date
0	AAPL	2013-02-07	267.839996	2019-11-27
1	FB	2013-12-13	217.500000	2018-07-25
2	JNJ	2014-02-27	143.786758	2018-12-13
3	MCD	2013-06-04	218.628891	2019-09-06
4	MTCH	2015-12-14	91.769997	2019-08-07

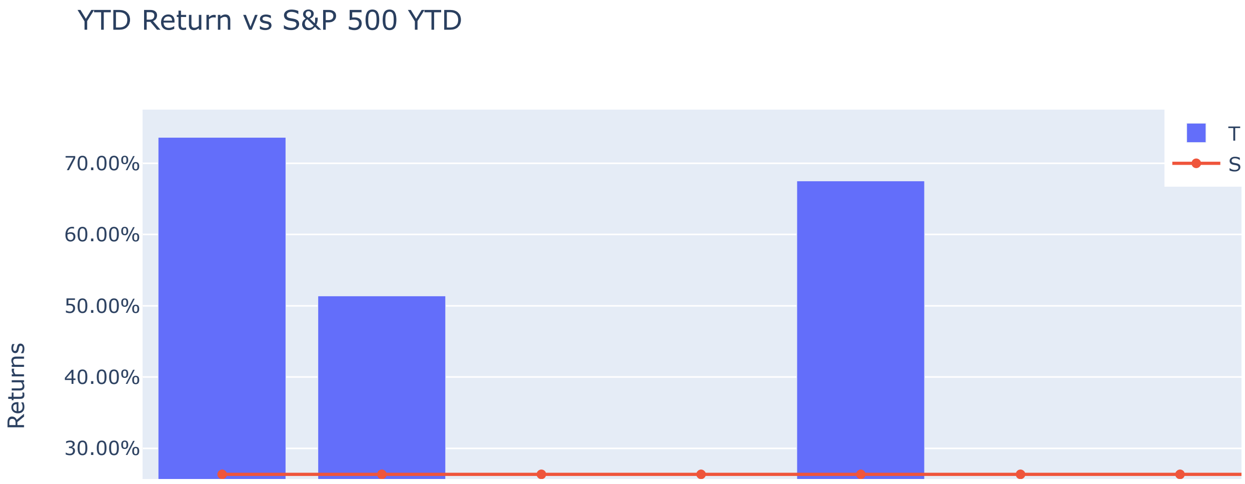
In [67]:
```
merged_portfolio_sp_latest_YTD_sp_closing_high = pd.merge(merged_portfolio_sp_latest_YTD_sp, ac
                                             , on=['Ticker', 'Acquisition Date'])

merged_portfolio_sp_latest_YTD_sp_closing_high.rename(columns={'Adj Close': 'Closing High Adj C

merged_portfolio_sp_latest_YTD_sp_closing_high['Pct off High'] = merged_portfolio_sp_latest_YTD

merged_portfolio_sp_latest_YTD_sp_closing_high
```

Out[67]:

	Ticker	Acquisition Date	Quantity	Unit Cost	Cost Basis	Start of Year	Latest Date	Ticker Adj Close	ticker return	SP 500 Initial Close	...	SP Start Year Close
0	AAPL	2013-02-07	125	65.40	8175.00	2018-12-28	2019-11-29	267.250000	3.086391	1509.390015	...	2485.73999
1	FB	2013-12-13	150	53.32	7998.00	2018-12-28	2019-11-29	201.639999	2.781695	1775.319946	...	2485.73999
2	JNJ	2014-02-27	100	81.90	8190.00	2018-12-28	2019-11-29	137.490005	0.678755	1854.290039	...	2485.73999
3	MCD	2013-06-04	100	84.99	8499.00	2018-12-28	2019-11-29	194.479996	1.288269	1631.380005	...	2485.73999
4	MTCH	2015-12-14	600	13.63	8178.00	2018-12-28	2019-11-29	70.480003	4.170947	2021.939941	...	2485.73999
5	NFLX	2016-01-14	75	108.71	8153.25	2018-12-28	2019-11-29	314.660004	1.894490	1921.839966	...	2485.73999
6	TWTR	2015-01-05	225	36.38	8185.50	2018-12-28	2019-11-29	30.910000	-0.150357	2020.579956	...	2485.73999
7	WMT	2013-08-14	125	68.30	8537.50	2018-12-28	2019-11-29	119.089996	0.743631	1685.390015	...	2485.73999

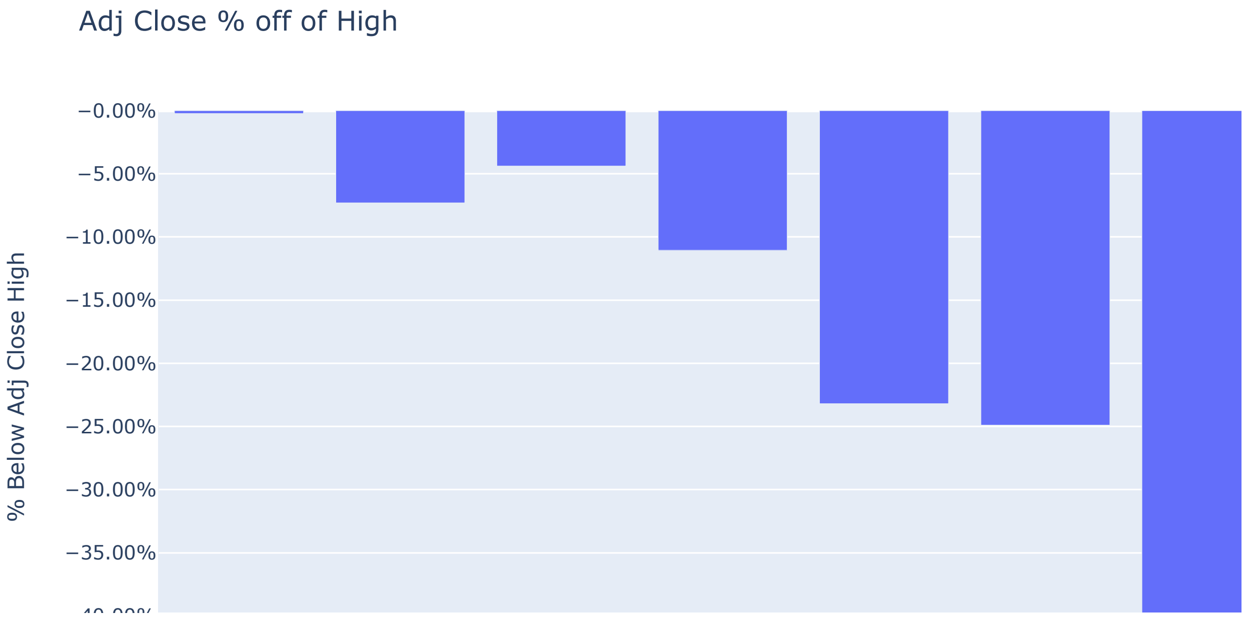
8 rows × 30 columns

In [68]:
```
trace1 = go.Bar(
    x = merged_portfolio_sp_latest_YTD_sp['Ticker'][0:10],
    y = merged_portfolio_sp_latest_YTD_sp['Share YTD'][0:10],
    name = 'Ticker YTD')

trace2 = go.Scatter(
    x = merged_portfolio_sp_latest_YTD_sp['Ticker'][0:10],
    y = merged_portfolio_sp_latest_YTD_sp['SP 500 YTD'][0:10],
    name = 'SP500 YTD')

data = [trace1, trace2]

layout = go.Layout(title = 'YTD Return vs S&P 500 YTD'
    , barmode = 'group'
    , yaxis=dict(title='Returns', tickformat=".2%")
    , xaxis=dict(title='Ticker')
    , legend=dict(x=.8,y=1)
    )

fig = go.Figure(data=data, layout=layout)
```

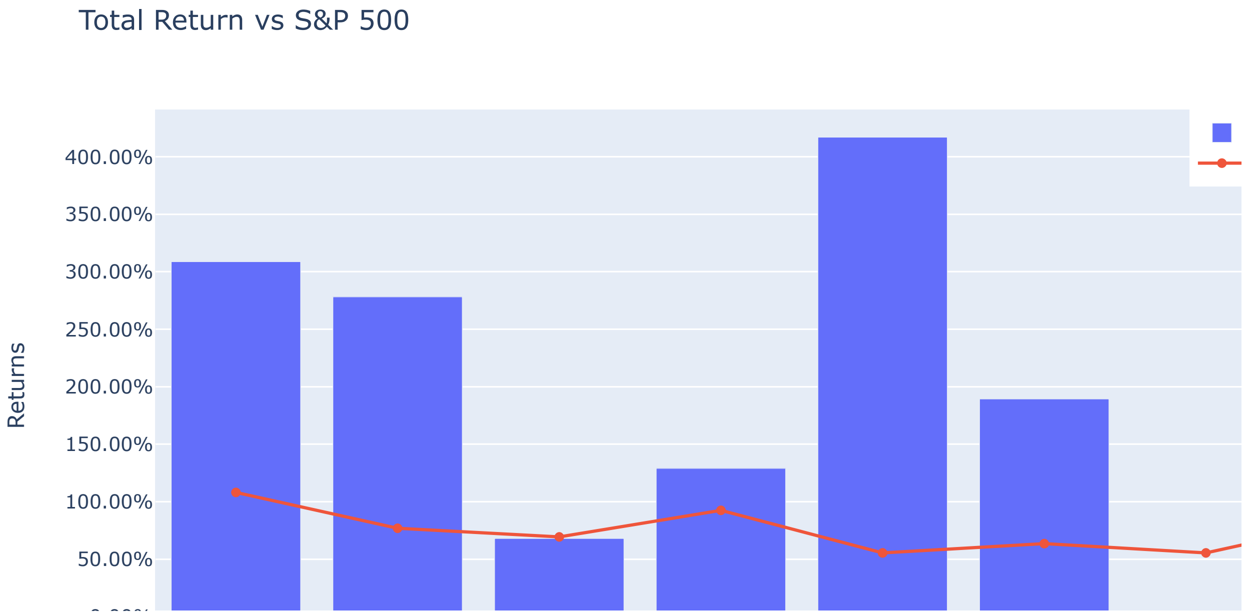


In [69]:

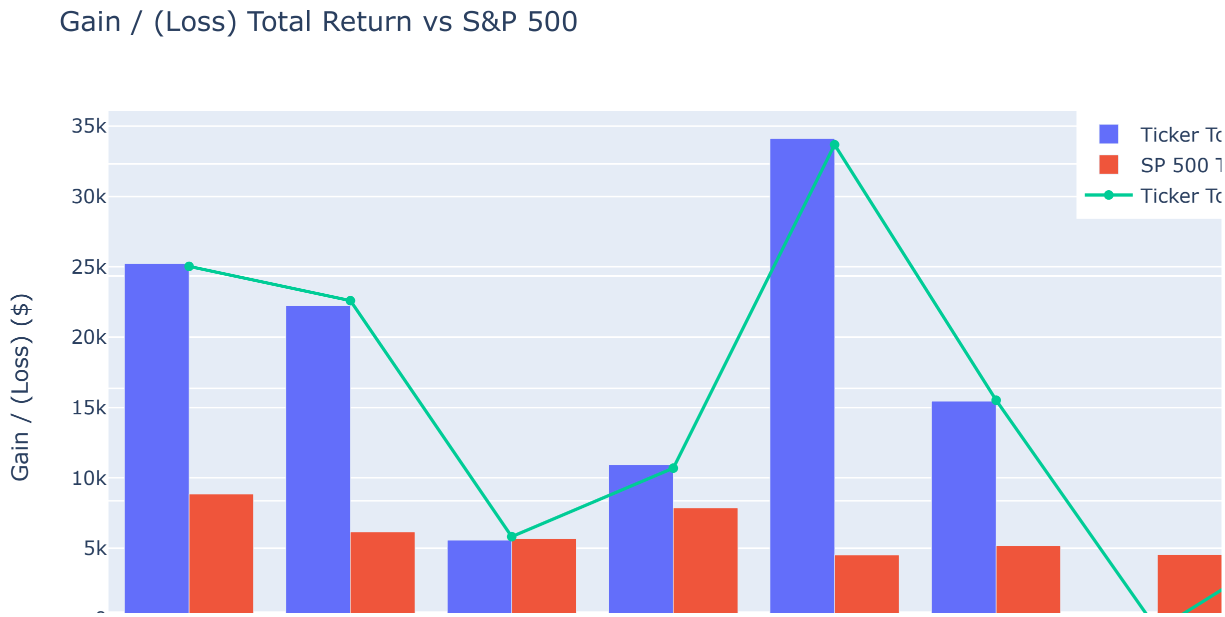
```
trace1 = go.Bar(
    x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'][0:10],
    y = merged_portfolio_sp_latest_YTD_sp_closing_high['Pct off High'][0:10],
    name = 'Pct off High')

data = [trace1]

layout = go.Layout(title = 'Adj Close % off of High'
    , barmode = 'group'
    , yaxis=dict(title='% Below Adj Close High', tickformat=".2%")
    , xaxis=dict(title='Ticker')
    , legend=dict(x=.8,y=1)
    )

fig = go.Figure(data=data, layout=layout)
```



```
In [70]: trace1 = go.Bar(
             x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'][0:10],
             y = merged_portfolio_sp_latest_YTD_sp_closing_high['ticker return'][0:10],
             name = 'Ticker Total Return')

         trace2 = go.Scatter(
             x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'][0:10],
             y = merged_portfolio_sp_latest_YTD_sp_closing_high['SP Return'][0:10],
             name = 'SP500 Total Return')

         data = [trace1, trace2]

         layout = go.Layout(title = 'Total Return vs S&P 500'
             , barmode = 'group'
             , yaxis=dict(title='Returns', tickformat=".2%")
             , xaxis=dict(title='Ticker', tickformat=".2%")
             , legend=dict(x=.8,y=1)
             )

         fig = go.Figure(data=data, layout=layout)
```

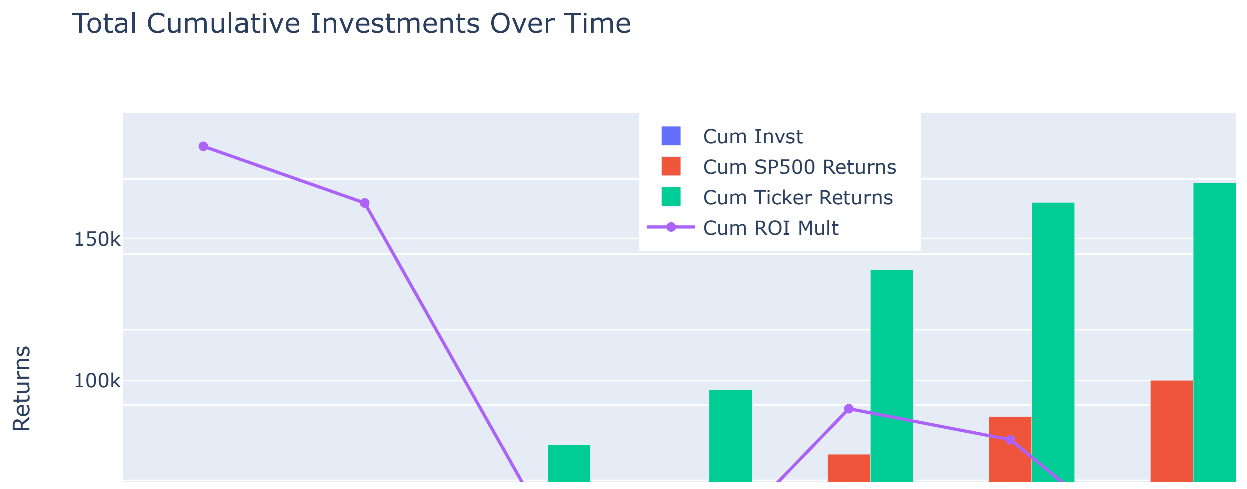


In [71]:
```
trace1 = go.Bar(
    x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'][0:10],
    y = merged_portfolio_sp_latest_YTD_sp_closing_high['Stock Gain / (Loss)'][0:10],
    name = 'Ticker Total Return ($)')

trace2 = go.Bar(
    x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'][0:10],
    y = merged_portfolio_sp_latest_YTD_sp_closing_high['SP 500 Gain / (Loss)'][0:10],
    name = 'SP 500 Total Return ($)')

trace3 = go.Scatter(
    x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'][0:10],
    y = merged_portfolio_sp_latest_YTD_sp_closing_high['ticker return'][0:10],
    name = 'Ticker Total Return %',
    yaxis='y2')

data = [trace1, trace2, trace3]

layout = go.Layout(title = 'Gain / (Loss) Total Return vs S&P 500'
    , barmode = 'group'
    , yaxis=dict(title='Gain / (Loss) ($)')
    , yaxis2=dict(title='Ticker Return', overlaying='y', side='right', tickformat=".2%")
    , xaxis=dict(title='Ticker')
    , legend=dict(x=.75,y=1)
    )

fig = go.Figure(data=data, layout=layout)
```



In [72]:

```
trace1 = go.Bar(
    x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'],
    y = merged_portfolio_sp_latest_YTD_sp_closing_high['Cum Invst'],
    # mode = 'lines+markers',
    name = 'Cum Invst')

trace2 = go.Bar(
    x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'],
    y = merged_portfolio_sp_latest_YTD_sp_closing_high['Cum SP Returns'],
    # mode = 'lines+markers',
    name = 'Cum SP500 Returns')

trace3 = go.Bar(
    x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'],
    y = merged_portfolio_sp_latest_YTD_sp_closing_high['Cum Ticker Returns'],
    # mode = 'lines+markers',
    name = 'Cum Ticker Returns')

trace4 = go.Scatter(
    x = merged_portfolio_sp_latest_YTD_sp_closing_high['Ticker'],
    y = merged_portfolio_sp_latest_YTD_sp_closing_high['Cum Ticker ROI Mult'],
    # mode = 'lines+markers',
    name = 'Cum ROI Mult'
    , yaxis='y2')


data = [trace1, trace2, trace3, trace4]

layout = go.Layout(title = 'Total Cumulative Investments Over Time'
    , barmode = 'group'
    , yaxis=dict(title='Returns')
    , xaxis=dict(title='Ticker')
    , legend=dict(x=.4,y=1)
    , yaxis2=dict(title='Cum ROI Mult', overlaying='y', side='right')
    )

fig = go.Figure(data=data, layout=layout)
```



In [ ]: