

```

In [1]: # Importing the necessary Python modules required in this notebook
import pandas as pd
import math
import numpy as np
import matplotlib.pyplot as plt
import yfinance as yf
import Plib as pl
from pandas_datareader import data as pdr
from tabulate import tabulate
import datetime
import statsmodels.api as sm
from statistics import mean
from scipy.stats import norm
from scipy.stats import rankdata

dt_start="2010-08-09"
dt_end="2019-08-01"
weights=(0.24,0.18,0.20,0.19,0.19)

api = pl.EODH()

In [2]: df11 = api.get_eod_data_with_date('USAG','LSE', dt_start, dt_end)['Adjusted_close'].rename('USAG').interpolate()
df12 = api.get_eod_data_with_date('SXLPL','LSE', dt_start, dt_end)['Adjusted_close'].rename('SXLPL').interpolate()
df13 = api.get_eod_data_with_date('SXLUL','LSE', dt_start, dt_end)['Adjusted_close'].rename('SXLUL').interpolate()
df14 = api.get_eod_data_with_date('IDVY','LSE', dt_start, dt_end)['Adjusted_close'].rename('IDVY').interpolate()
df15 = api.get_eod_data_with_date('IDUS','LSE', dt_start, dt_end)['Adjusted_close'].rename('IUSA').interpolate()

# Use S&P500 as benchmark
market = api.get_eod_data_with_date('GSPC','INDX', dt_start, dt_end)['Adjusted_close'].rename('market').interpolate()
# Use CBOE Interest Rate 10-Year T-Note Index
riskfree = api.get_eod_data_with_date('TNX','INDX', dt_start, dt_end)['Adjusted_close'].rename('cash').interpolate().mean()

/mniconda3/lib/python3.7/site-packages/eod_historical_data/data.py:42: ParserWarning: Falling back to the 'python' engine because the 'c' engine does not support skipfooter; you can avoid this warning by specifying engine='python'.
  parse_dates=[0], index_col=0)

```

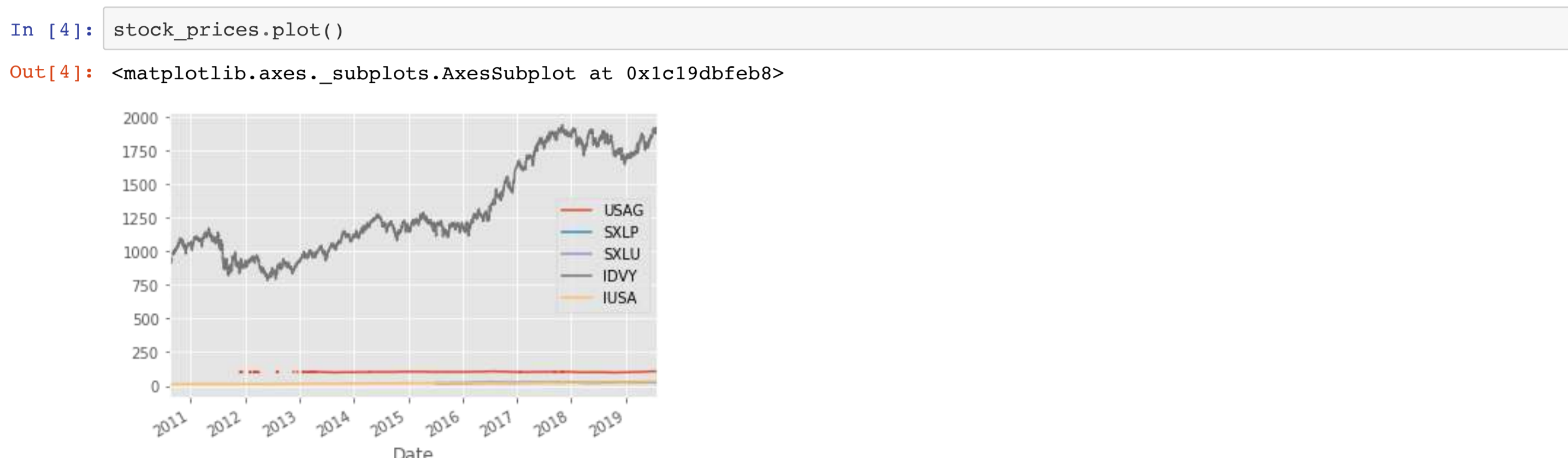
```

In [3]: stock_prices=pd.concat([df11, df12,df13,df14,df15], axis=1)
file_name="mycsv_etfprices.csv"
stock_prices.to_csv(file_name, encoding='utf-8')
stock_prices.tail()

```

```
Out[3]:
```

	USAG	SXLPL	SXLUL	IDVY	IUSA
Date					
2019-07-26	107.4341	27.3987	31.6925	1889.4	30.1025
2019-07-29	107.4243	27.5375	31.6663	1919.6	30.1062
2019-07-30	107.5772	27.7025	31.7875	1894.2	30.0188
2019-07-31	107.7251	27.3663	31.6088	1879.3	30.0325
2019-08-01	107.9750	27.4250	31.6837	1891.0	30.0200



```

In [5]: portfolio_value=stock_prices.pct_change()[1:].dot(weights).rename("portfolio")
portfolio_value=portfolio_value.replace([np.inf, -np.inf], np.nan).interpolate()

```

```

In [6]: market=market.pct_change()
market=market.replace([np.inf, -np.inf], np.nan).interpolate()
returns=pd.concat([portfolio_value, market], axis=1).dropna()
returns.head()

```

```
Out[6]:
```

	portfolio	market
Date		
2015-07-10	0.008051	0.012338
2015-07-13	0.002198	0.011066
2015-07-14	0.000952	0.004453
2015-07-15	0.000181	-0.000735
2015-07-16	0.007688	0.008015

```

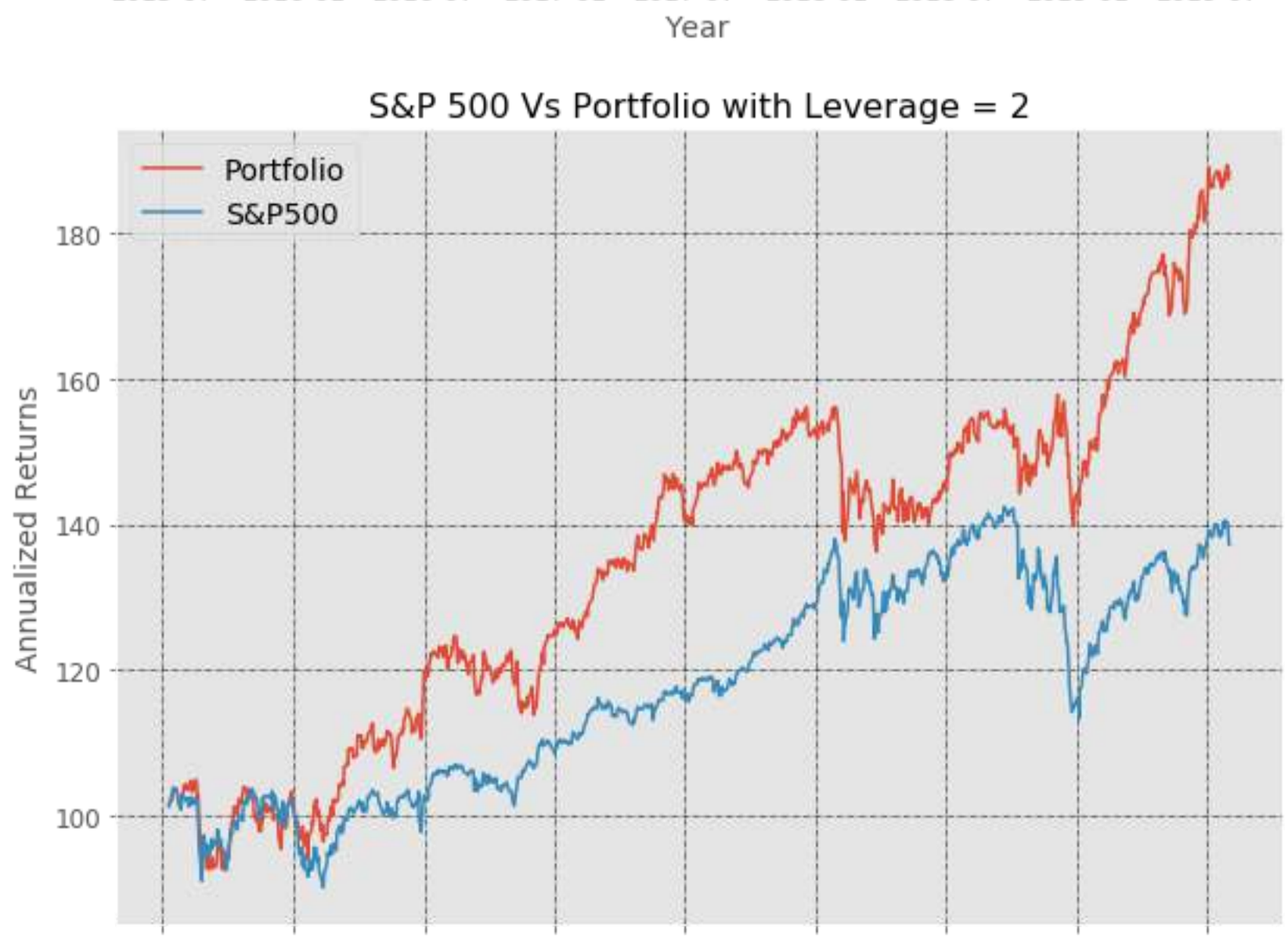
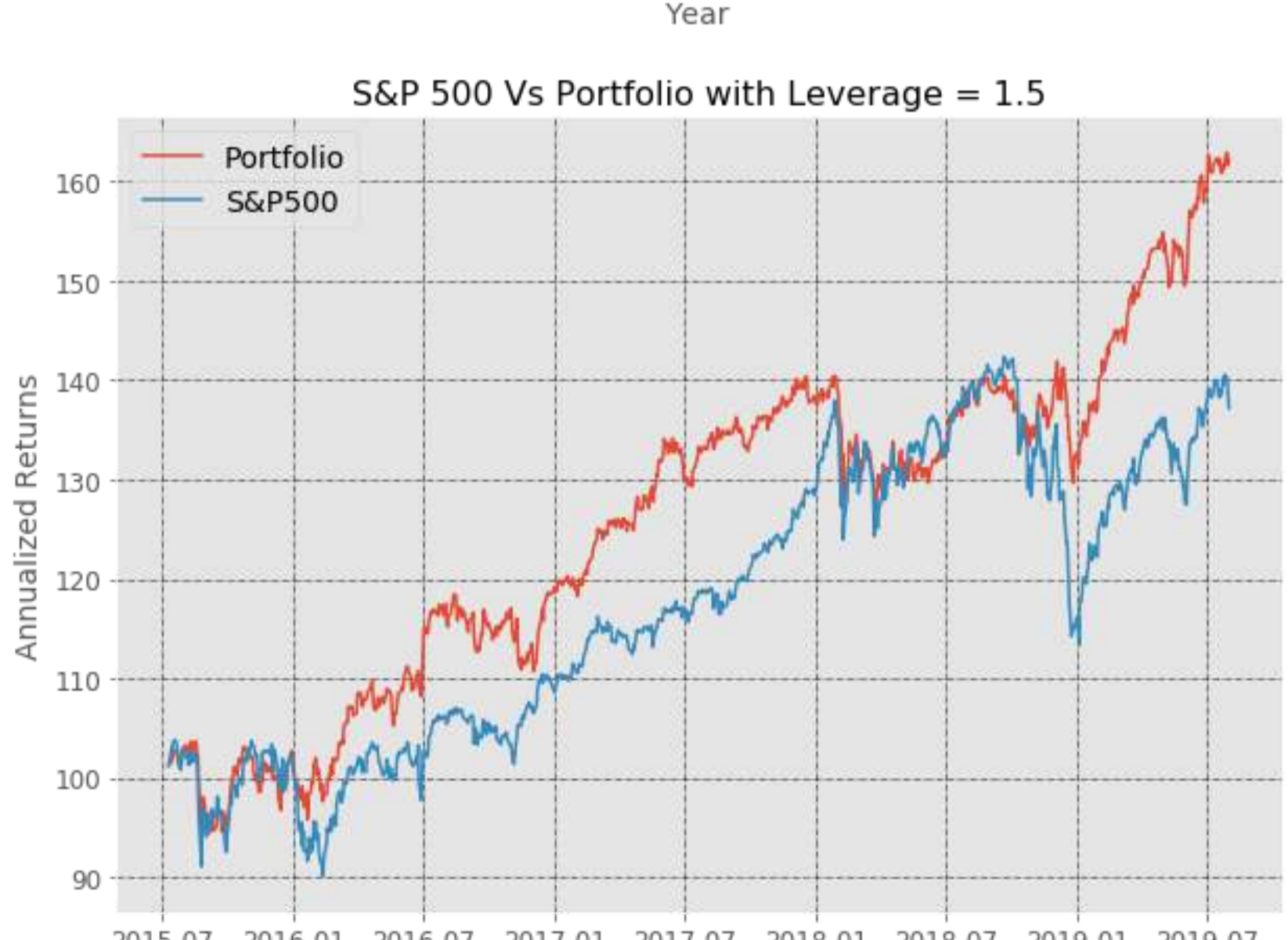
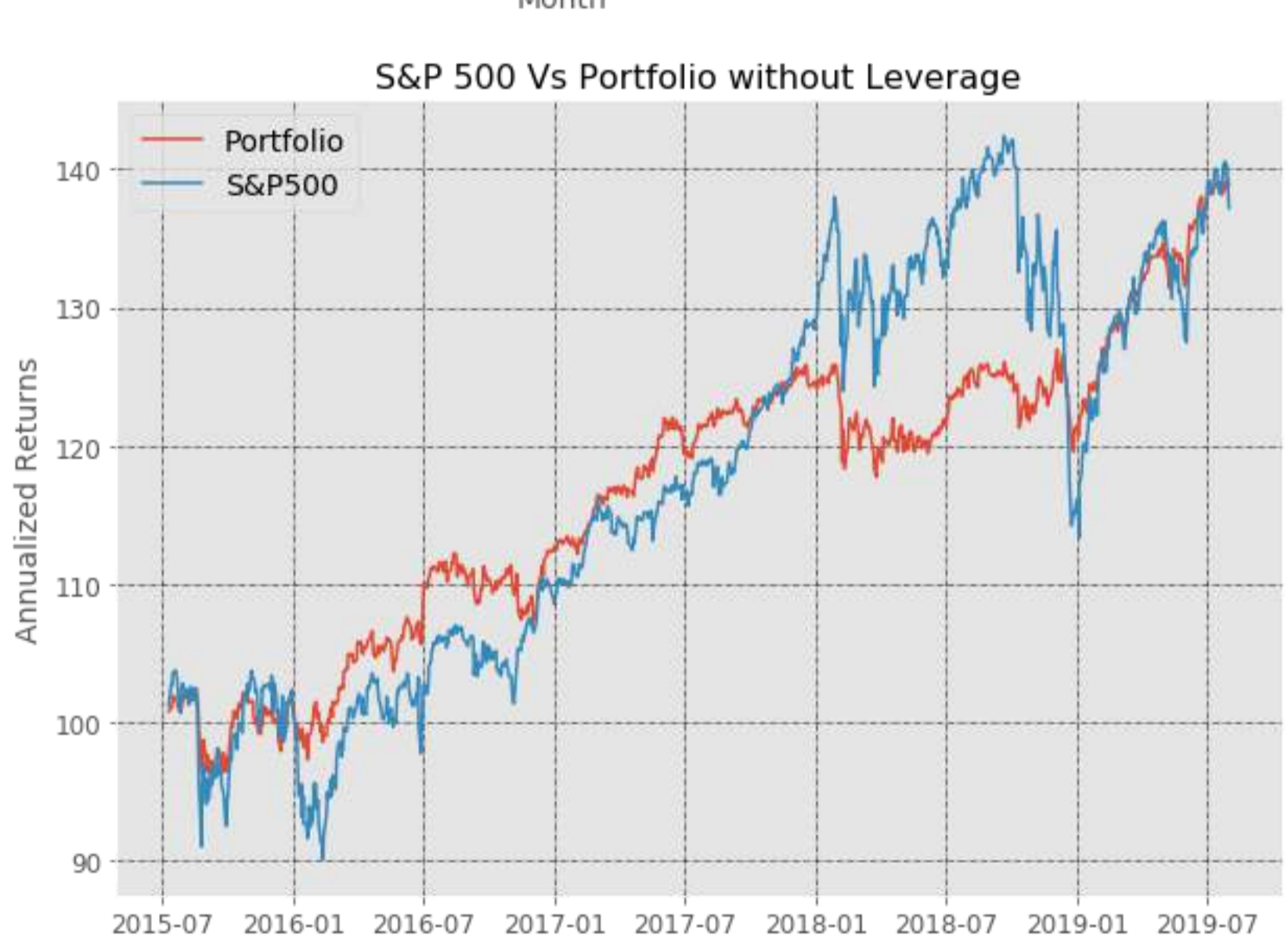
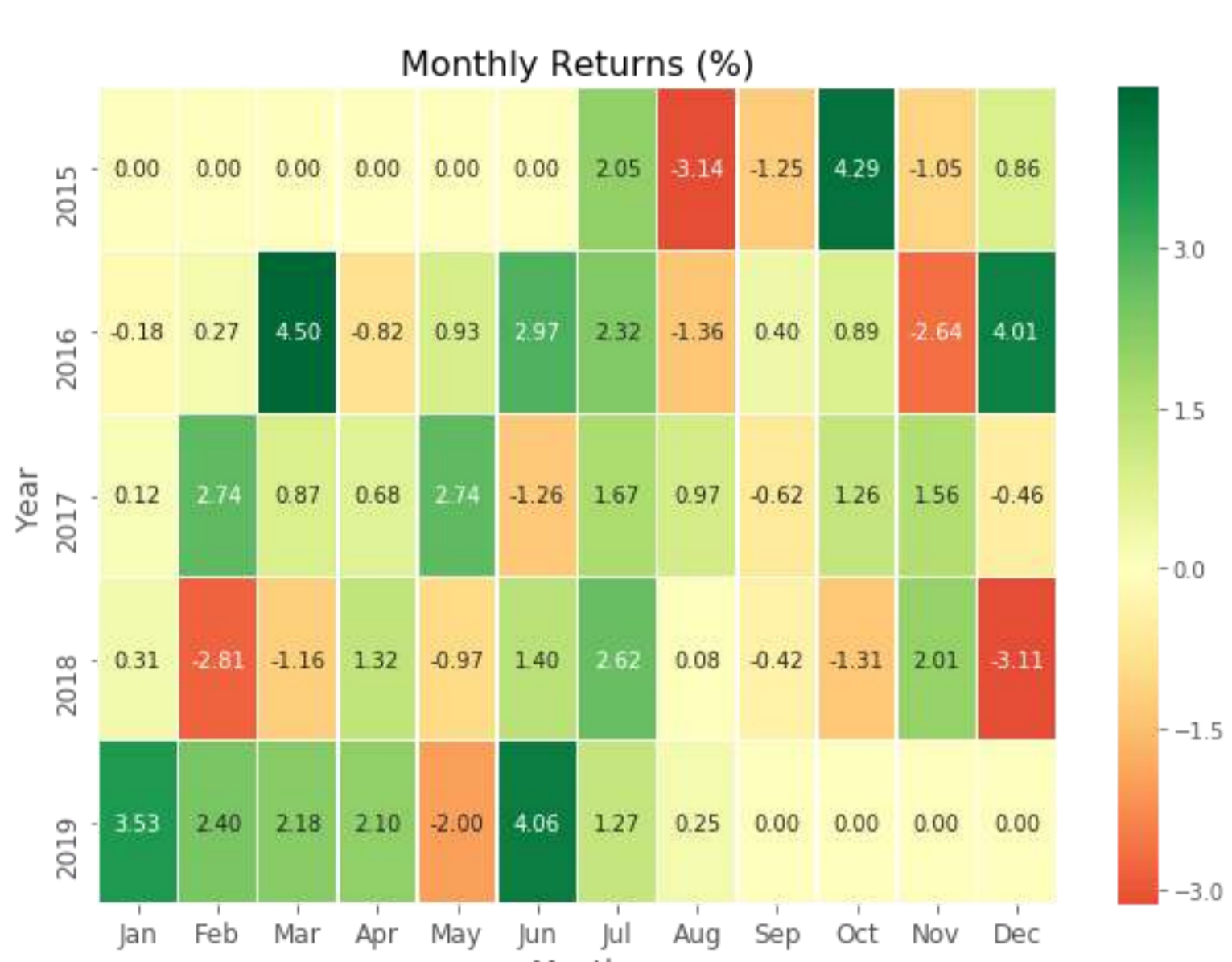
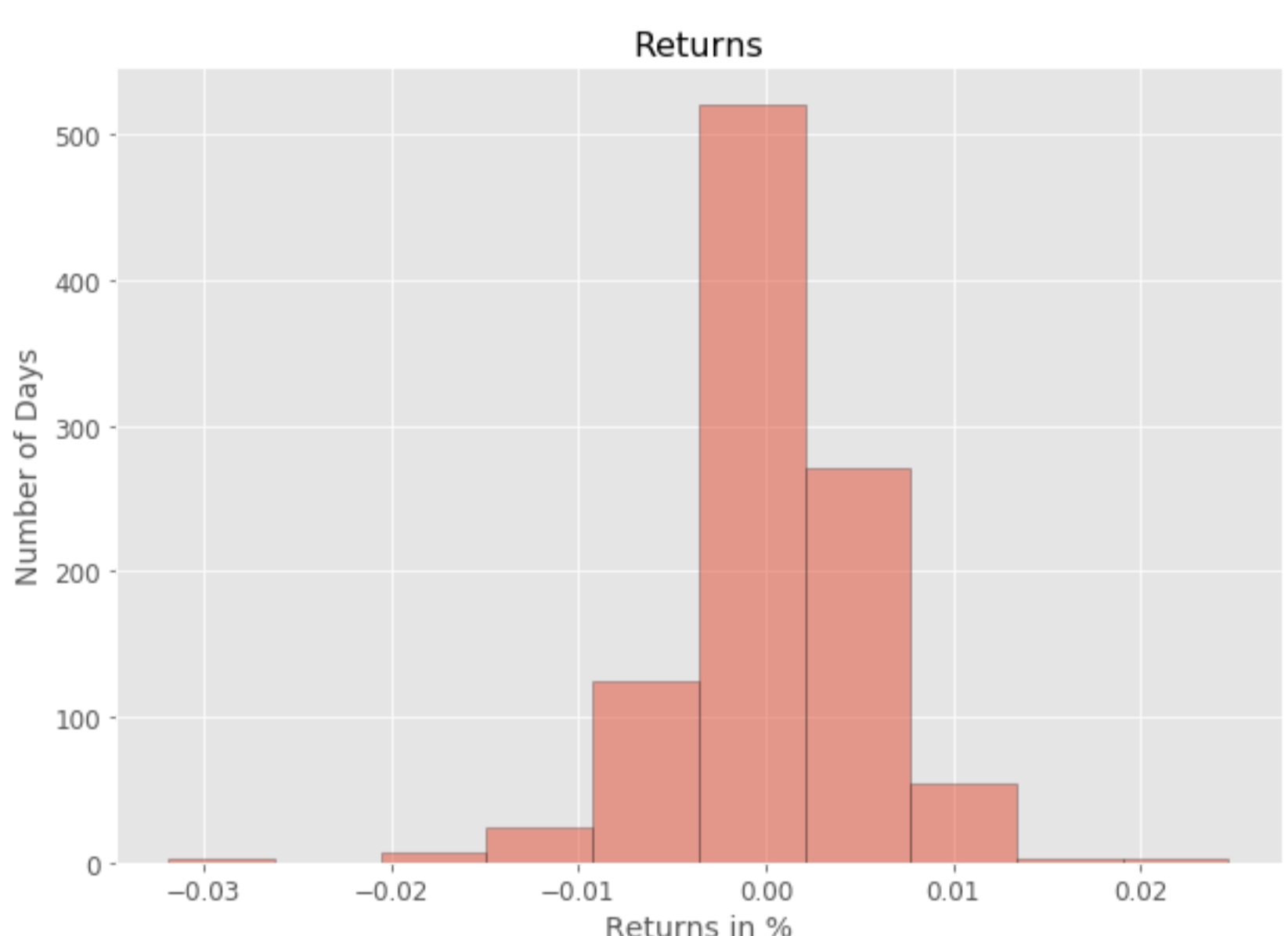
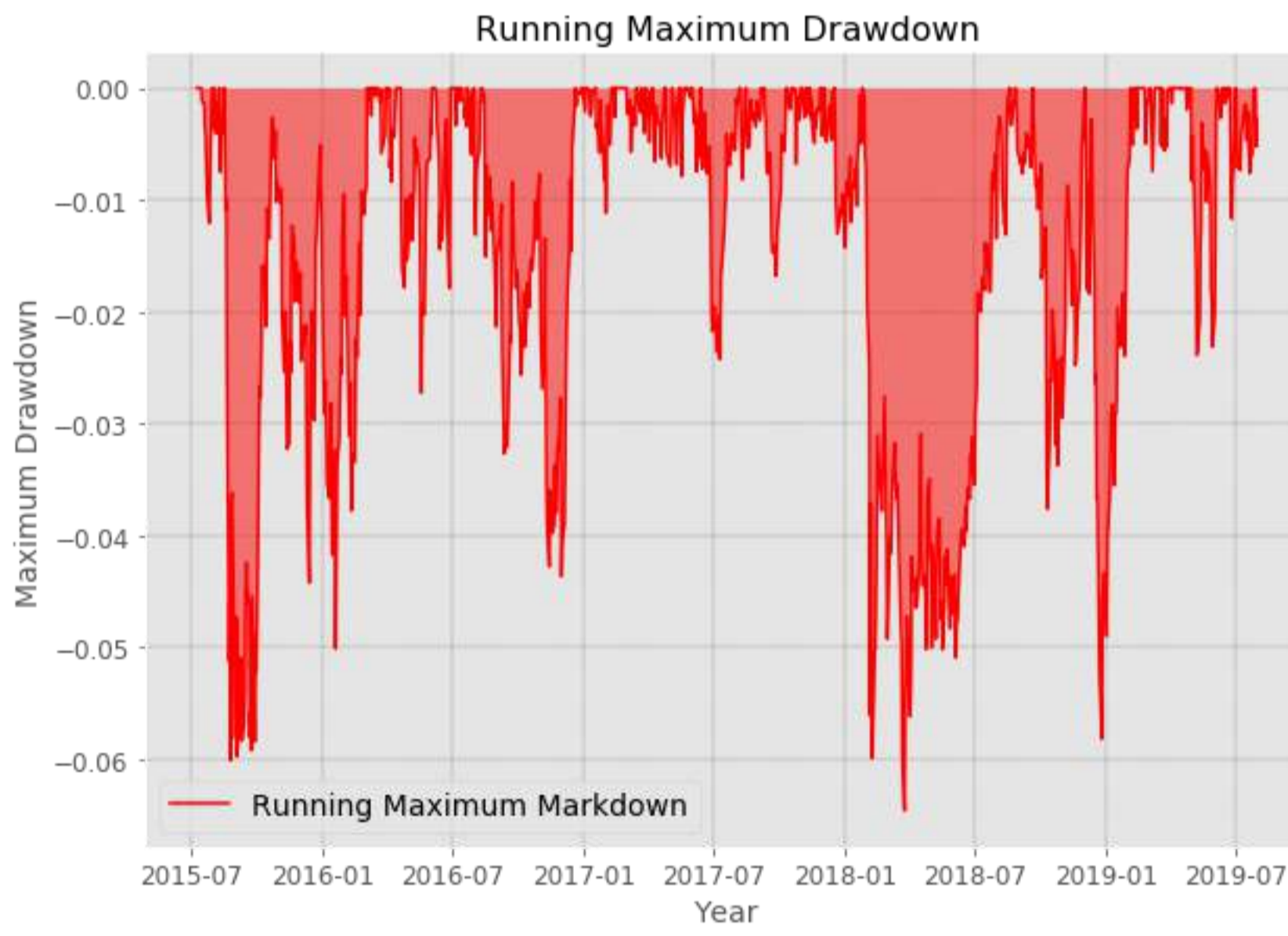
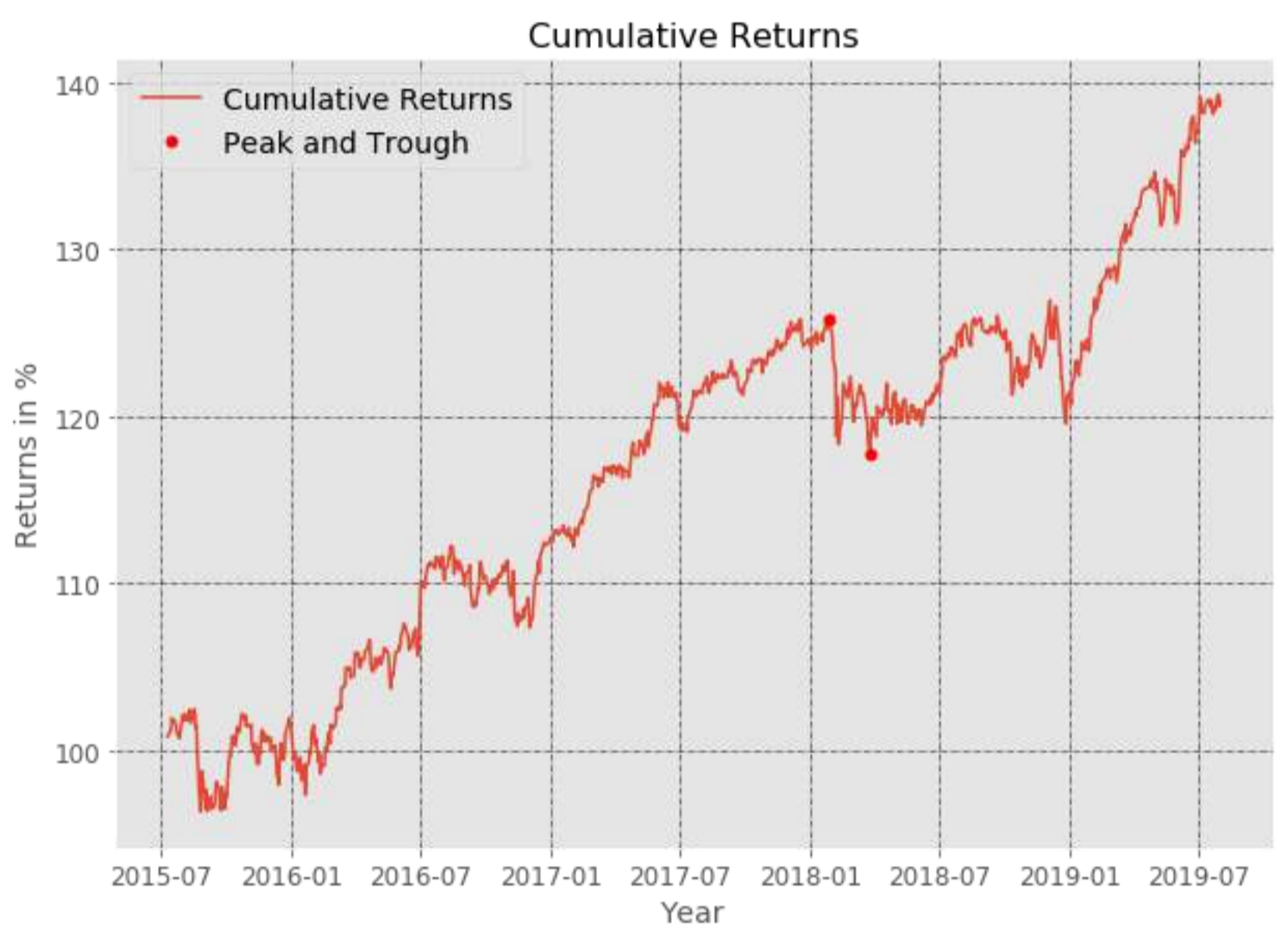
In [7]: file_name="mycsv_etf.csv"
returns.to_csv(file_name, encoding='utf-8')

```

```

In [8]: # Return Analysis with Leverage=1
pl.portfolioAnalysis(riskfree/100, returns,1)

```



```
Out[8]:
```

	Parameters	Value
0	Annual Returns	8.91097
1	Annual Volatility	7.8485
2	Sharpe Ratio	0.788114
3	Sortino Ratio	1.06472
4	Beta	0.258804
5	Treynor Ratio	0.239004
6	Information Ratio	-0.022942
7	Skewness	-0.540133
8	Kurtosis	4.41454
9	Maximum Drawdown	-6.46345