

# Stock\_Treynor\_Ratio\_Chart

September 29, 2021

## 1 Stock Treynor Ratio Chart

```
[1]: # Library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")

from pandas_datareader import data as pdr
import yfinance as yf
yf.pdr_override()

[2]: start = '2019-01-01' #input
end = '2020-07-01' #input
symbol1 = '^GSPC' #input
symbol2 = 'AMD' #input

[3]: market = yf.download(symbol1, start=start, end=end)['Adj Close']
stocks = yf.download(symbol2, start=start, end=end)['Adj Close']

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

[4]: market_returns = market.pct_change().dropna()
stocks_returns = stocks.pct_change().dropna()

[5]: # risk free
rf = yf.download('BIL', start=start, end=end)['Adj Close'].pct_change()[1:]

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

[6]: def treynor_ratio(stocks_returns, market_returns):
    m = np.matrix([stocks_returns, market_returns])
    beta = np.cov(m)[0][1] / np.std(market_returns)
    mrk_rate_ret = (market_returns[-1] - market_returns[0]) / market_returns[0]
```

```

er = rf + beta*(mrk_rate_ret-rf)
tr = (er - rf) / beta
return tr

```

```

[7]: # Compute the running Treynor Ratio
running = [treynor_ratio(stocks_returns[i-90:i], market_returns[i-90:i]) for i
↳ in range(90, len(stocks_returns))]

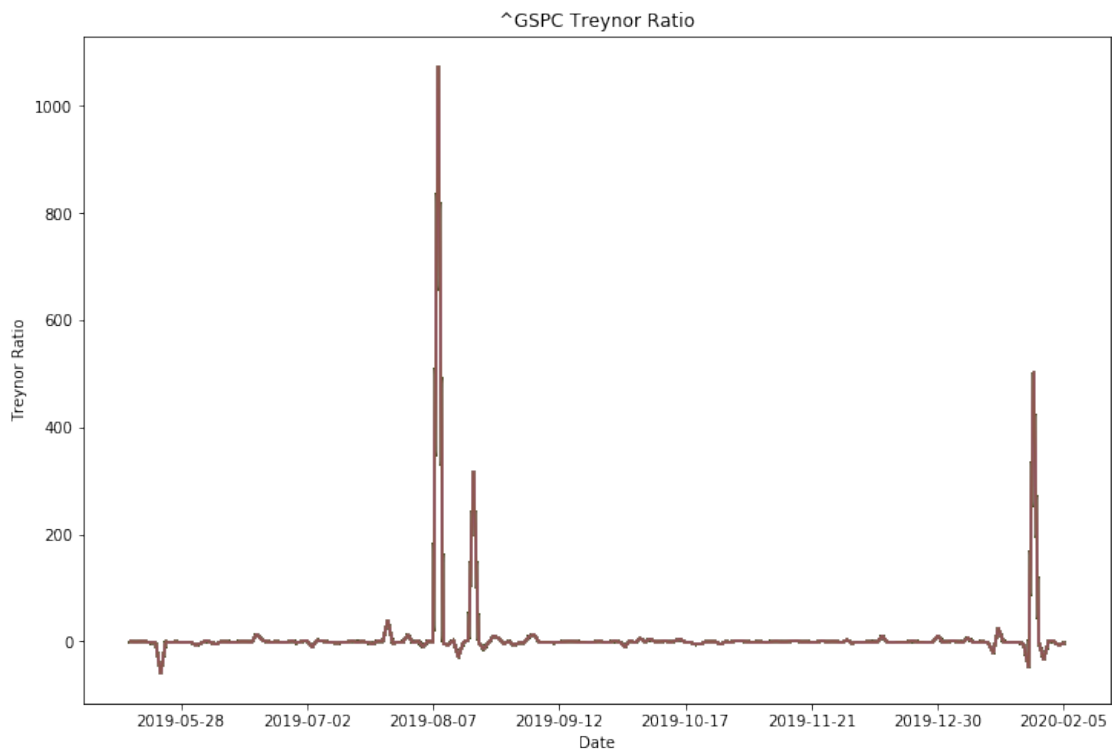
# Plot running Treynor Ratio up to 100 days before the end of the data set
_, ax1 = plt.subplots(figsize=(12,8))
ax1.plot(range(90, len(stocks_returns)-100), running[:100])
ticks = ax1.get_xticks()
ax1.set_xticklabels([stocks.index[int(i)].date() for i in ticks[:-1]]) # Label
↳ x-axis with dates
plt.title(symbol1 + ' Treynor Ratio')
plt.xlabel('Date')
plt.ylabel('Treynor Ratio')

```

```

[7]: Text(0, 0.5, 'Treynor Ratio')

```



```

[8]: TR = treynor_ratio(stocks_returns, market_returns)
TR

```

[8]: Date

2019-01-03	-1.622780
2019-01-04	-1.622561
2019-01-07	-1.622452
2019-01-08	-1.622452
2019-01-09	-1.622452
2019-01-10	-1.622561
2019-01-11	-1.622670
2019-01-14	-1.622343
2019-01-15	-1.622671
2019-01-16	-1.622561
2019-01-17	-1.622670
2019-01-18	-1.622452
2019-01-22	-1.622561
2019-01-23	-1.622452
2019-01-24	-1.622780
2019-01-25	-1.622343
2019-01-28	-1.622561
2019-01-29	-1.622452
2019-01-30	-1.622670
2019-01-31	-1.622452
2019-02-01	-1.622550
2019-02-04	-1.622561
2019-02-05	-1.622452
2019-02-06	-1.622671
2019-02-07	-1.622452
2019-02-08	-1.622671
2019-02-11	-1.622343
2019-02-12	-1.622671
2019-02-13	-1.622452
2019-02-14	-1.622671
...	
2020-05-19	-1.622452
2020-05-20	-1.622452
2020-05-21	-1.622561
2020-05-22	-1.622343
2020-05-26	-1.622561
2020-05-27	-1.622343
2020-05-28	-1.622452
2020-05-29	-1.622452
2020-06-01	-1.622561
2020-06-02	-1.622452
2020-06-03	-1.622452
2020-06-04	-1.622452
2020-06-05	-1.622233
2020-06-08	-1.622561
2020-06-09	-1.622452

```

2020-06-10    -1.622561
2020-06-11    -1.622233
2020-06-12    -1.622671
2020-06-15    -1.622452
2020-06-16    -1.622452
2020-06-17    -1.622452
2020-06-18    -1.622343
2020-06-19    -1.622452
2020-06-22    -1.622452
2020-06-23    -1.622452
2020-06-24    -1.622452
2020-06-25    -1.622561
2020-06-26    -1.622343
2020-06-29    -1.622452
2020-06-30    -1.622452
Name: Adj Close, Length: 376, dtype: float64

```

```

[9]: TR.plot(figsize=(12,8), title = symbol1 + ' Treynor Ratio')
plt.axhline(y=TR.mean(), color='r', linestyle='-')
plt.xlabel('Date')
plt.ylabel('Treynor Ratio')

```

```

[9]: Text(0, 0.5, 'Treynor Ratio')

```

