tommy_new_portfolio

December 21, 2022

1 Calculating Tommy's Portfolio Using Non-Negative Least-Squares

- This allocation is the best one for a number of reasons:
  - 1. No leverage
  - 2. No high turnover
  - 3. All long positions
- Essentially, this is the most realistic, mathematically correct way to optimize your portfolio.

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import matplotlib as mpl
import seaborn as sns
import scipy as scs
import math
import yfinance as yf
from statsmodels.regression.rolling import RollingOLS
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from statsmodels.regression.quantile_regression import QuantReg

from sklearn.linear_model import QuantileRegressor
from sklearn.decomposition import PCA

from scipy.optimize import lsq_linear
import warnings
warnings.filterwarnings("ignore")
plt.style.use("seaborn")
mpl.rcParams['font.family'] = 'serif'
%matplotlib inline
```

[ ]:
```
def regression_based_performance(factor,fund_ret,rf,constant = True):
    """
        Returns the Regression based performance Stats for given set of returns␣
 ↪and factors
        Inputs:
```

```
            factor - Dataframe containing monthly returns of the regressors
            fund_ret - Dataframe containing monthly excess returns of the␣
↪regressand fund
            rf - Monthly risk free rate of return
        Output:
            summary_stats - (Beta of regression, treynor ratio, information␣
↪ratio, alpha).
    """
    if constant:
        X = sm.tools.add_constant(factor)
    else:
        X = factor
    y=fund_ret
    model = sm.OLS(y,X,missing='drop').fit()

    if constant:
        beta = model.params[1:]
        alpha = round(float(model.params['const']),6)

    else:
        beta = model.params
    treynor_ratio = ((fund_ret.values-rf.values).mean()*12)/beta[0]
    tracking_error = (model.resid.std()*np.sqrt(12))
    if constant:
        information_ratio = model.params[0]*12/tracking_error
    r_squared = model.rsquared
    unexplained_var = 1-r_squared
    if constant:
        return␣
↪(beta,treynor_ratio,information_ratio,alpha,r_squared,unexplained_var,tracking_error)
    else:
        return (beta,treynor_ratio,r_squared,unexplained_var,tracking_error)
```

2 Getting the Data

[ ]:
```
tickers = "AAPL MSFT GS VYM XOM JPM WTRG OXY ^GSPC BUD VNQ HD NFLX"
start = "2003-01-01"
end = "2022-12-01"
rf = pd.DataFrame(yf.download('^IRX', start = start, interval = '1mo', end =␣
↪end)['Adj Close'])
rf.dropna(inplace = True)
rf = rf *(10**-2)
rf.columns = ['3M Treasury']
adj_close = pd.DataFrame(yf.download(tickers, start,interval = '1mo', end =␣
↪end)["Adj Close"])
adj_close.dropna(inplace = True)
```

```
adj_close.columns = ["Apple", "Budweiser", "Goldman", "Home Depot" ,"JP␣
↪Morgan", "Microsoft", "Netflix", "Occidental Petroleum","Vanguard Real␣
↪Estate", "Vanguard High Dividend","Essential Utilities", "Exxon", "S&P 500"]
rets = adj_close.pct_change().dropna()
```

```
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  13 of 13 completed
```

[ ]:
```
rets = rets.join(rf, how = 'inner')
```

[ ]:
```
rets.head()
```

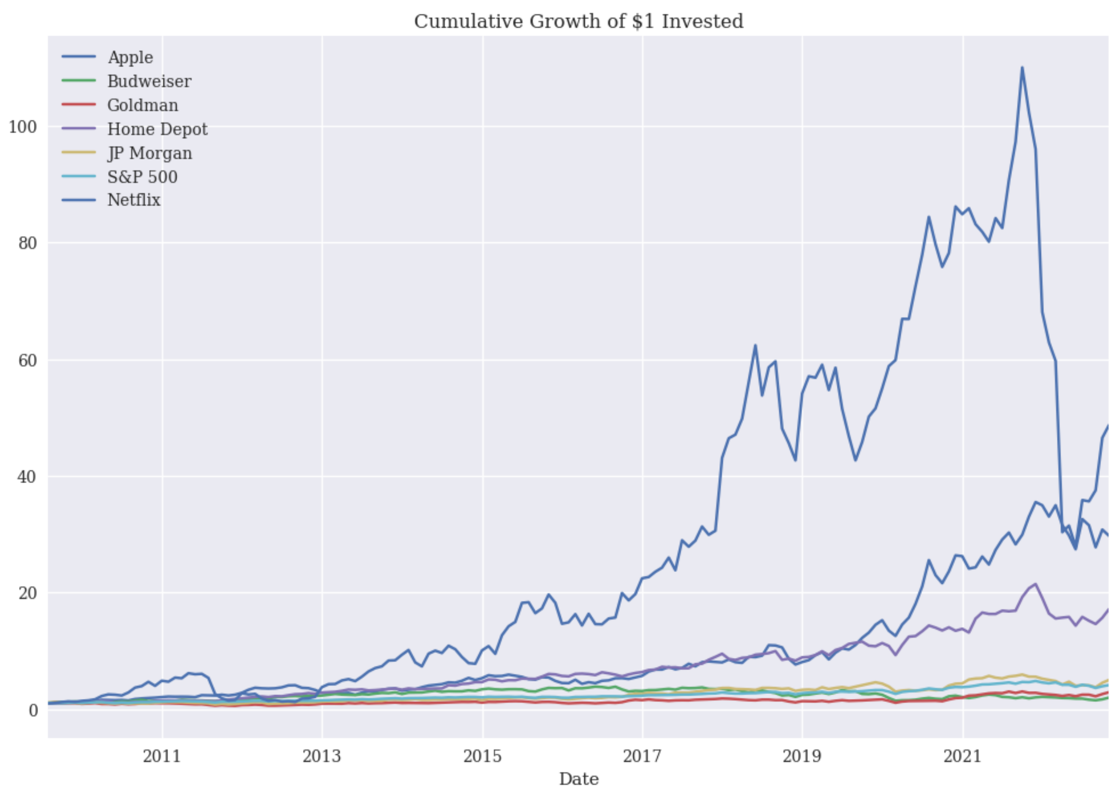
[ ]:
```
               Apple  Budweiser   Goldman  Home Depot  JP Morgan  Microsoft  \
Date
2009-08-01  0.029500   0.080946  0.013227    0.052044   0.126101   0.048044
2009-09-01  0.101897   0.058526  0.116574   -0.023819   0.008283   0.049274
2009-10-01  0.016995   0.026992 -0.076919   -0.050353  -0.046782   0.078150
2009-11-01  0.060530   0.065918 -0.002997    0.090474   0.018468   0.060584
2009-12-01  0.054124   0.034599 -0.002707    0.057383  -0.019299   0.040963

             Netflix  Occidental Petroleum  Vanguard Real Estate  \
Date
2009-08-01 -0.007055              0.024671              0.143231
2009-09-01  0.058217              0.072503              0.055513
2009-10-01  0.157678             -0.027731             -0.035206
2009-11-01  0.096913              0.064708              0.065656
2009-12-01 -0.060379              0.006931              0.060190

            Vanguard High Dividend  Essential Utilities     Exxon   S&P 500  \
Date
2009-08-01                0.049037            -0.066999 -0.017616  0.033560
2009-09-01                0.021703             0.055156 -0.001749  0.035723
2009-10-01               -0.020197            -0.124150  0.044599 -0.019762
2009-11-01                0.060733             0.056311  0.047440  0.057364
2009-12-01                0.002639             0.082649 -0.086361  0.017771

            3M Treasury
Date
2009-08-01      0.00130
2009-09-01      0.00115
2009-10-01      0.00045
2009-11-01      0.00050
2009-12-01      0.00050
```

[ ]:
```
pd.DataFrame([rets.columns, rets.mean()*12], index = ['Asset', 'Yearly␣
↪Return']).T
```

```
[ ]:                     Asset  Yearly Return
0                        Apple       0.292956
1                    Budweiser       0.085622
2                      Goldman       0.121511
3                   Home Depot       0.237138
4                    JP Morgan       0.154486
5                    Microsoft       0.223909
6                      Netflix       0.447962
7         Occidental Petroleum       0.142034
8          Vanguard Real Estate       0.126285
9     Vanguard High Dividend       0.130539
10         Essential Utilities       0.133595
11                       Exxon       0.102044
12                     S&P 500       0.117585
13                 3M Treasury       0.068968
```

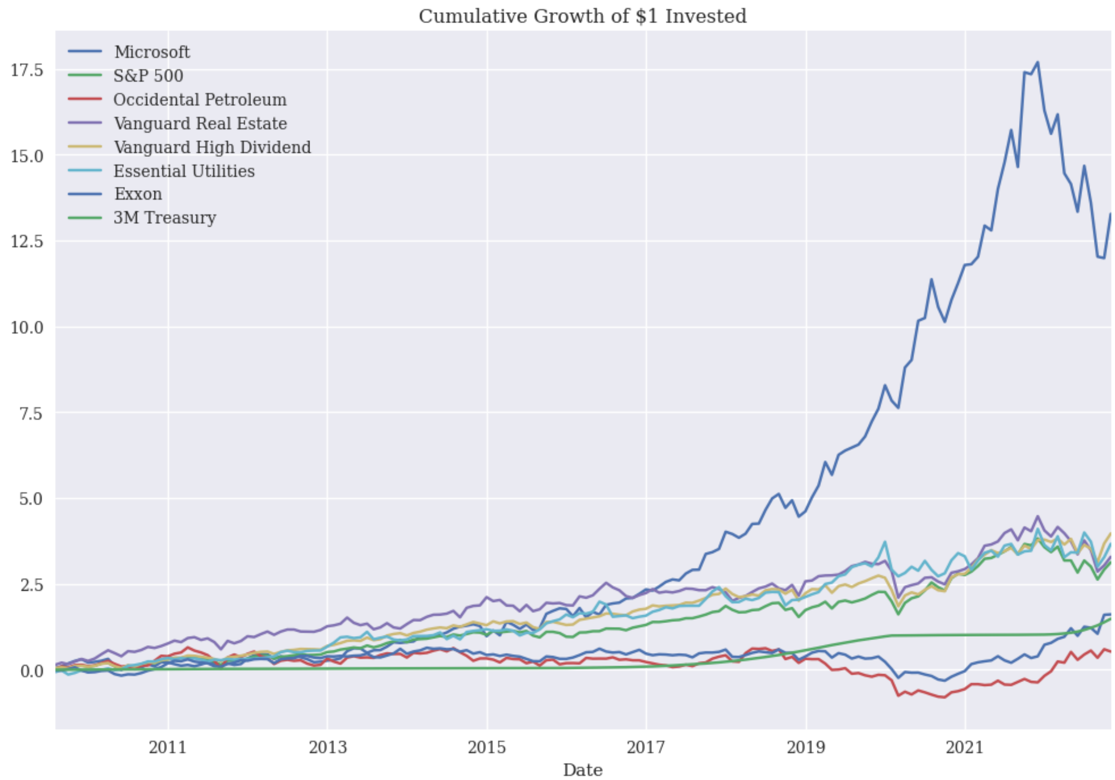
```
[ ]: ((1+rets[['Apple','Budweiser','Goldman','Home Depot','JP Morgan','S&P␣
     ↪500','Netflix']]).cumprod()).plot(title = "Cumulative Growth of $1␣
     ↪Invested",figsize=(12,8))
```

```
[ ]: <AxesSubplot:title={'center':'Cumulative Growth of $1 Invested'}, xlabel='Date'>
```



[ ]:
```python
((((1+rets[['Microsoft','S&P 500', 'Occidental Petroleum','Vanguard Real␣
↪Estate','Vanguard High Dividend','Essential Utilities','Exxon', '3M␣
↪Treasury']])).cumprod())-1).plot(title = "Cumulative Growth of $1␣
↪Invested",figsize=(12,8))
```

[ ]:
```
<AxesSubplot:title={'center':'Cumulative Growth of $1 Invested'}, xlabel='Date'>
```



[ ]:
```python
cum_returns = (((1+rets).cumprod())-1)
cum_returns = pd.DataFrame((cum_returns.tail().max()).
↪sort_values(ascending=False), columns = ['Cumaltive % Return'])
cum_returns
```

[ ]:

	Cumaltive % Return
Netflix	47.673417
Apple	31.571471
Home Depot	16.131567
Microsoft	14.678228
JP Morgan	4.002518
Essential Utilities	3.997796
Vanguard High Dividend	3.971195
Vanguard Real Estate	3.768826
S&P 500	3.182657

```
Goldman                            1.898954
Exxon                              1.614469
3M Treasury                        1.486758
Budweiser                          0.983229
Occidental Petroleum               0.598983
```

- The only two assets not returning back your money invested since 2003 is Budweiser and Occidental Petroleum. Netflix, if you got in early, would have returned you 47% gross return.At one point Netflix would have returned you an incredible 100%.
- Over the entire period you actually earn more than Budweiser and Occidental by holding 3-Month Treasuy Bills.

3 Univariate Risk Statistics

[ ]:
```python
def performance_summary(return_data, annualization = 12):
    """
        Returns the Performance Stats for given set of returns
        Inputs:
            return_data - DataFrame with Date index and Monthly Returns for␣
↪different assets/strategies.
        Output:
            summary_stats - DataFrame with annualized mean return, vol, sharpe␣
↪ratio. Skewness, Excess Kurtosis, Var (0.5) and
                            CVaR (0.5) and drawdown based on monthly returns.
    """
    summary_stats = return_data.mean().to_frame('Mean').apply(lambda x:␣
↪x*annualization)
    summary_stats['Volatility'] = return_data.std().apply(lambda x: x*np.
↪sqrt(annualization))
    summary_stats['Sharpe Ratio'] = summary_stats['Mean']/
↪summary_stats['Volatility']

    summary_stats['Skewness'] = return_data.skew()
    summary_stats['Excess Kurtosis'] = return_data.kurtosis()
    summary_stats['VaR (0.05)'] = return_data.quantile(.05, axis = 0)
    summary_stats['CVaR (0.05)'] = return_data[return_data <= return_data.
↪quantile(.05, axis = 0)].mean()

    wealth_index = 1000*(1+return_data).cumprod()
    previous_peaks = wealth_index.cummax()
    drawdowns = (wealth_index - previous_peaks)/previous_peaks

    summary_stats['Max Drawdown'] = drawdowns.min()
    summary_stats['Peak'] = [previous_peaks[col][:drawdowns[col].idxmin()].
↪idxmax() for col in previous_peaks.columns]
    summary_stats['Bottom'] = drawdowns.idxmin()
```

```python
    recovery_date = []
    for col in wealth_index.columns:
        prev_max = previous_peaks[col][:drawdowns[col].idxmin()].max()
        recovery_wealth = pd.DataFrame([wealth_index[col][drawdowns[col].
↪idxmin():]]).T
        recovery_date.append(recovery_wealth[recovery_wealth[col] >= prev_max].
↪index.min())
    summary_stats['Recovery'] = recovery_date

    return summary_stats
```

[ ]:
```python
performance_summary(rets).sort_values(by = 'Sharpe Ratio', ascending=False)
```

[ ]:

```
                           Mean  Volatility  Sharpe Ratio  Skewness  \
3M Treasury            0.068968    0.030312      2.275236  1.789315
Home Depot             0.237138    0.212117      1.117959 -0.128208
Apple                  0.292956    0.270585      1.082677 -0.078656
Microsoft              0.223909    0.215901      1.037095  0.019220
Vanguard High Dividend 0.130539    0.139285      0.937210 -0.367453
S&P 500                0.117585    0.146521      0.802511 -0.391257
Netflix                0.447962    0.563262      0.795300  0.749776
Essential Utilities    0.133595    0.186088      0.717913 -0.633429
Vanguard Real Estate   0.126285    0.181909      0.694220 -0.351645
JP Morgan              0.154486    0.256984      0.601153 -0.266176
Goldman                0.121511    0.289617      0.419558  0.097512
Exxon                  0.102044    0.246120      0.414609  0.327596
Budweiser              0.085622    0.261257      0.327729 -0.072879
Occidental Petroleum   0.142034    0.463839      0.306213  0.814077

                        Excess Kurtosis  VaR (0.05)  CVaR (0.05)  \
3M Treasury                    2.813032    0.000050     0.000043
Home Depot                     0.773732   -0.062385    -0.121388
Apple                         -0.323696   -0.107935    -0.132282
Microsoft                      0.302926   -0.080505    -0.109113
Vanguard High Dividend         1.684567   -0.050483    -0.088166
S&P 500                        0.669672   -0.069521    -0.090002
Netflix                        5.198140   -0.197629    -0.313974
Essential Utilities            1.123004   -0.092239    -0.126338
Vanguard Real Estate           1.351812   -0.069824    -0.106074
JP Morgan                      0.971896   -0.101988    -0.161171
Goldman                        0.230248   -0.120650    -0.162570
Exxon                          2.527887   -0.099233    -0.143597
Budweiser                      1.323511   -0.114314    -0.160251
Occidental Petroleum           8.749432   -0.155238    -0.245377

                        Max Drawdown       Peak     Bottom   Recovery
```

```
3M Treasury                  0.000000 2009-08-01 2009-08-01 2009-08-01
Home Depot                  -0.332386 2021-12-01 2022-06-01        NaT
Apple                       -0.395421 2012-09-01 2013-06-01 2014-06-01
Microsoft                   -0.305285 2021-12-01 2022-10-01        NaT
Vanguard High Dividend      -0.238734 2019-12-01 2020-03-01 2020-12-01
S&P 500                     -0.247695 2021-12-01 2022-09-01        NaT
Netflix                     -0.798966 2011-05-01 2012-09-01 2013-08-01
Essential Utilities         -0.215637 2021-12-01 2022-09-01        NaT
Vanguard Real Estate        -0.294162 2021-12-01 2022-09-01        NaT
JP Morgan                   -0.367319 2021-10-01 2022-09-01        NaT
Goldman                     -0.497975 2009-09-01 2011-12-01 2013-12-01
Exxon                       -0.583960 2014-04-01 2020-10-01 2022-01-01
Budweiser                   -0.625208 2016-06-01 2020-03-01        NaT
Occidental Petroleum        -0.879332 2011-04-01 2020-10-01        NaT
```

- Since 2003, Home Depot actually offers the highest sharpe ratio. Apple and Microsoft are greater than 1 as well. Sharpe is important for a variety of reasons because every investor wants to maximize return per risk invested. Even though Netflix averages an incredible 44% yearly return since 2003, it has an extreme amount of tail risk (**look at the excess Kurtosis; for a normally distributed random variable this measure should be 0**) because of the **streaming wars.** Some reasearch ought to go into who will win this battle in the long-run because the market punnished Netflix harshly as you can see that in the cumulative return chart.
- Disregard the **High Sharpe Ratio for the the 3- Month Treasury Bills.** They are risk-free for a reason and we expect them to have a high sharpe because their volatility (σ) is so low.
- Also you can see that **a ton of assets were just at their peak in 2021/2020**. Now that central banks are raising rates at the fastest rate since the 1980s, global fight against inflation, European Energy Crisis, Ukraine-Russia War, Increased Global Debt, etc there has been a major sell off causing a lot more hightend risk than normal periods - more risk today than 2008 since the global financial crisis was just one **big thing** where now we have several major things to uncover.

4 Performance Summary at Different Times

[ ]:
```
performance_summary(rets['2014']).sort_values(by = 'Sharpe Ratio', ascending =␣
↪False)
```

[ ]:
```
                            Mean  Volatility  Sharpe Ratio  Skewness  \
3M Treasury             0.002450    0.000424      5.773102  0.408234
Microsoft               0.249793    0.092978      2.686578  0.540553
Vanguard Real Estate    0.280149    0.147324      1.901580 -0.156600
Apple                   0.369116    0.224110      1.647028 -1.126673
Vanguard High Dividend  0.131267    0.084424      1.554861 -0.624126
Home Depot              0.284507    0.198782      1.431255  0.932763
S&P 500                 0.111364    0.081089      1.373352 -0.443025
```

```
Essential Utilities     0.165102    0.179835      0.918077 -0.310303
Goldman                 0.109745    0.121560      0.902810 -1.233202
JP Morgan               0.103957    0.141765      0.733303 -0.925389
Budweiser               0.099442    0.185482      0.536126 -0.549426
Netflix                 0.029728    0.484855      0.061312  0.606532
Occidental Petroleum   -0.071435    0.226750     -0.315040  0.050482
Exxon                  -0.051423    0.152319     -0.337602 -0.718171

                        Excess Kurtosis  VaR (0.05)  CVaR (0.05)  \
3M Treasury                   -0.459639    0.000041     0.000030
Microsoft                      1.054692   -0.017960    -0.022318
Vanguard Real Estate           2.287083   -0.031932    -0.069059
Apple                          0.656989   -0.085790    -0.107697
Vanguard High Dividend        -1.189929   -0.024888    -0.031597
Home Depot                     1.622710   -0.047384    -0.062089
S&P 500                       -0.415766   -0.024545    -0.035583
Essential Utilities            1.459197   -0.070846    -0.093059
Goldman                        1.633370   -0.046886    -0.074129
JP Morgan                      0.826278   -0.064403    -0.077911
Budweiser                      0.144350   -0.077983    -0.099286
Netflix                        0.345066   -0.165721    -0.210040
Occidental Petroleum          -1.268056   -0.086542    -0.103002
Exxon                         -0.462569   -0.075287    -0.089328

                        Max Drawdown       Peak     Bottom   Recovery
3M Treasury                 0.000000 2014-01-01 2014-01-01 2014-01-01
Microsoft                  -0.022318 2014-11-01 2014-12-01        NaT
Vanguard Real Estate       -0.069059 2014-08-01 2014-09-01 2014-10-01
Apple                      -0.067867 2014-11-01 2014-12-01        NaT
Vanguard High Dividend     -0.019398 2014-11-01 2014-12-01        NaT
Home Depot                 -0.035353 2014-02-01 2014-03-01 2014-08-01
S&P 500                    -0.015514 2014-08-01 2014-09-01 2014-10-01
Essential Utilities        -0.096389 2014-06-01 2014-09-01 2014-10-01
Goldman                    -0.036651 2014-02-01 2014-05-01 2014-06-01
JP Morgan                  -0.078895 2014-03-01 2014-05-01 2014-09-01
Budweiser                  -0.060553 2014-06-01 2014-07-01 2014-11-01
Netflix                    -0.284796 2014-08-01 2014-12-01        NaT
Occidental Petroleum       -0.225480 2014-08-01 2014-11-01        NaT
Exxon                      -0.103704 2014-04-01 2014-11-01        NaT
```

[ ]:
```
performance_summary(rets['2011']).sort_values(by = 'Sharpe Ratio', ascending =␣
↪False)
```

[ ]:
```
                            Mean  Volatility  Sharpe Ratio  Skewness  \
3M Treasury             0.006050    0.001778      3.403024  0.957466
Home Depot              0.225316    0.167197      1.347607  0.556803
Apple                   0.247796    0.203959      1.214935  1.238376
```

Exxon	0.188413	0.188266	1.000782	-0.014699
Vanguard High Dividend	0.107376	0.130652	0.821845	0.598206
Budweiser	0.099992	0.182112	0.549071	0.973345
Vanguard Real Estate	0.109519	0.242201	0.452183	0.251626
Essential Utilities	0.012711	0.093075	0.136564	0.110300
Occidental Petroleum	0.050440	0.425572	0.118522	1.221662
S&P 500	0.011461	0.159740	0.071750	0.756211
Microsoft	-0.035635	0.151773	-0.234794	0.449641
JP Morgan	-0.173035	0.315184	-0.548997	-0.151903
Netflix	-0.590542	0.723459	-0.816275	-0.963115
Goldman	-0.551479	0.300581	-1.834710	0.827970

	Excess Kurtosis	VaR (0.05)	CVaR (0.05)	\
3M Treasury	-0.579768	0.000050	0.000050	
Home Depot	-1.134208	-0.035570	-0.044374	
Apple	2.104481	-0.044330	-0.055783	
Exxon	-1.060801	-0.060736	-0.072315	
Vanguard High Dividend	0.533772	-0.037810	-0.049389	
Budweiser	0.193621	-0.041029	-0.041260	
Vanguard Real Estate	0.835167	-0.083553	-0.116994	
Essential Utilities	-1.449030	-0.032511	-0.037761	
Occidental Petroleum	2.757119	-0.143150	-0.175698	
S&P 500	1.932291	-0.063528	-0.071762	
Microsoft	-1.384059	-0.049080	-0.058379	
JP Morgan	0.816151	-0.144792	-0.198083	
Netflix	0.994853	-0.384557	-0.518020	
Goldman	2.214287	-0.159204	-0.183997	

	Max Drawdown	Peak	Bottom	Recovery
3M Treasury	0.000000	2011-01-01	2011-01-01	2011-01-01
Home Depot	-0.103544	2011-02-01	2011-09-01	2011-11-01
Apple	-0.055783	2011-10-01	2011-11-01	2011-12-01
Exxon	-0.164307	2011-04-01	2011-09-01	NaT
Vanguard High Dividend	-0.116717	2011-04-01	2011-09-01	NaT
Budweiser	-0.155979	2011-04-01	2011-09-01	NaT
Vanguard Real Estate	-0.181503	2011-05-01	2011-09-01	NaT
Essential Utilities	-0.072937	2011-01-01	2011-07-01	NaT
Occidental Petroleum	-0.371549	2011-04-01	2011-09-01	NaT
S&P 500	-0.170276	2011-04-01	2011-09-01	NaT
Microsoft	-0.092758	2011-01-01	2011-05-01	2011-07-01
JP Morgan	-0.347411	2011-02-01	2011-09-01	NaT
Netflix	-0.761706	2011-05-01	2011-11-01	NaT
Goldman	-0.441428	2011-02-01	2011-12-01	NaT

[ ]:
```
performance_summary(rets['2012'])
```

[ ]:

	Mean	Volatility	Sharpe Ratio	Skewness
Apple	0.325955	0.303115	1.075350	0.274173
Budweiser	0.409149	0.221021	1.851176	1.022577
Goldman	0.419368	0.343223	1.221855	-0.225908
Home Depot	0.427099	0.165073	2.587327	-0.893981
JP Morgan	0.372705	0.350009	1.064847	-1.434638
Microsoft	0.079005	0.223923	0.352824	0.504179
Netflix	0.674713	1.000125	0.674628	1.334591
Occidental Petroleum	-0.152303	0.227860	-0.668407	-0.265241
Vanguard Real Estate	0.168862	0.119325	1.415134	0.021416
Vanguard High Dividend	0.121729	0.081579	1.492162	-1.160913
Essential Utilities	0.175556	0.094190	1.863839	1.799221
Exxon	0.057332	0.157180	0.364756	0.000188
S&P 500	0.131634	0.105574	1.246845	-1.308430
3M Treasury	0.009150	0.000693	13.197517	-0.577250

	Excess Kurtosis	VaR (0.05)	CVaR (0.05)
Apple	-0.437936	-0.096044	-0.107600
Budweiser	0.792080	-0.034612	-0.046897
Goldman	1.600727	-0.116783	-0.168910
Home Depot	-0.619511	-0.046150	-0.047306
JP Morgan	3.190524	-0.136533	-0.223662
Microsoft	-0.020247	-0.076773	-0.088382
Netflix	1.918283	-0.251130	-0.303373
Occidental Petroleum	-0.774063	-0.107116	-0.131002
Vanguard Real Estate	-0.449867	-0.034783	-0.045059
Vanguard High Dividend	2.082824	-0.024857	-0.046637
Essential Utilities	5.283686	-0.012980	-0.024961
Exxon	1.757817	-0.058463	-0.089298
S&P 500	2.025151	-0.039077	-0.062651
3M Treasury	0.154054	0.000460	0.000350

	Max Drawdown	Peak	Bottom	Recovery
Apple	-0.198620	2012-09-01	2012-12-01	NaT
Budweiser	-0.048207	2012-03-01	2012-05-01	2012-06-01
Goldman	-0.230522	2012-03-01	2012-05-01	2012-12-01
Home Depot	-0.047306	2012-04-01	2012-05-01	2012-06-01
JP Morgan	-0.274315	2012-03-01	2012-05-01	NaT
Microsoft	-0.163913	2012-03-01	2012-11-01	NaT
Netflix	-0.547088	2012-01-01	2012-09-01	NaT
Occidental Petroleum	-0.265870	2012-02-01	2012-11-01	NaT
Vanguard Real Estate	-0.045059	2012-04-01	2012-05-01	2012-07-01
Vanguard High Dividend	-0.046637	2012-04-01	2012-05-01	2012-07-01
Essential Utilities	-0.028060	2012-07-01	2012-09-01	2012-11-01
Exxon	-0.093393	2012-03-01	2012-05-01	2012-07-01
S&P 500	-0.069678	2012-03-01	2012-05-01	2012-09-01
3M Treasury	0.000000	2012-01-01	2012-01-01	2012-01-01

5 Measure Assets For Sytematic and Idiosyncratic Risk

- Here I'll regress each asset in your portfolio to get a better understanding if your asset is generating excess return over the market - **on average**.
- The regression equation:

  $r_{i,t} = \alpha + \beta * r_{\text{SPY},t} + \epsilon_t$

  - Will calculate the the Treynor ratio, Information Ratio, Tracking Error, and Market Beta (i,e, *systematic risk*)

[ ]:
```
assets = rets.columns
market = rets['S&P 500']
risk_free = rets['3M Treasury']

df_list = []
for stock in assets:
    stock_ret = rets[stock]
    reg = regression_based_performance(market, stock_ret, risk_free)
    beta = reg[0][0]
    treynor_ratio = reg[1]
    information_ratio = reg[2]
    alpha = reg[3]
    r_squared = reg[4]
    idiosyncratic_risk = np.round(reg[5], 5)
    tracking_error = np.round(reg[6],4)
    df_list.append(pd.
 ↪DataFrame([[beta,treynor_ratio,information_ratio,alpha,r_squared,idiosyncratic_risk␣
 ↪,tracking_error]],columns=['Market Beta','Treynor Ratio','Information␣
 ↪Ratio','Alpha','r_squared','Idiosyncratic Risk' ,'tracking error'], index =␣
 ↪[stock]))
reg_performance = pd.concat(df_list)

reg_performance.sort_values(by  = 'Alpha', ascending = False)
```

[ ]:

	Market Beta	Treynor Ratio	Information Ratio	\
Netflix	1.129415	0.335567	0.585360	
Apple	1.134499	0.197434	0.747319	
Home Depot	1.003809	0.167532	0.779260	
Microsoft	0.991562	0.156260	0.671973	
3M Treasury	-0.007396	-0.000000	2.305402	
Essential Utilities	0.644609	0.100258	0.360481	
Vanguard High Dividend	0.887265	0.069394	0.524257	
Vanguard Real Estate	0.906988	0.063195	0.158084	
JP Morgan	1.274364	0.067107	0.026285	
S&P 500	1.000000	0.048617	0.000000	
Exxon	0.978411	0.033806	-0.064993	
Occidental Petroleum	1.579053	0.046272	-0.108549	

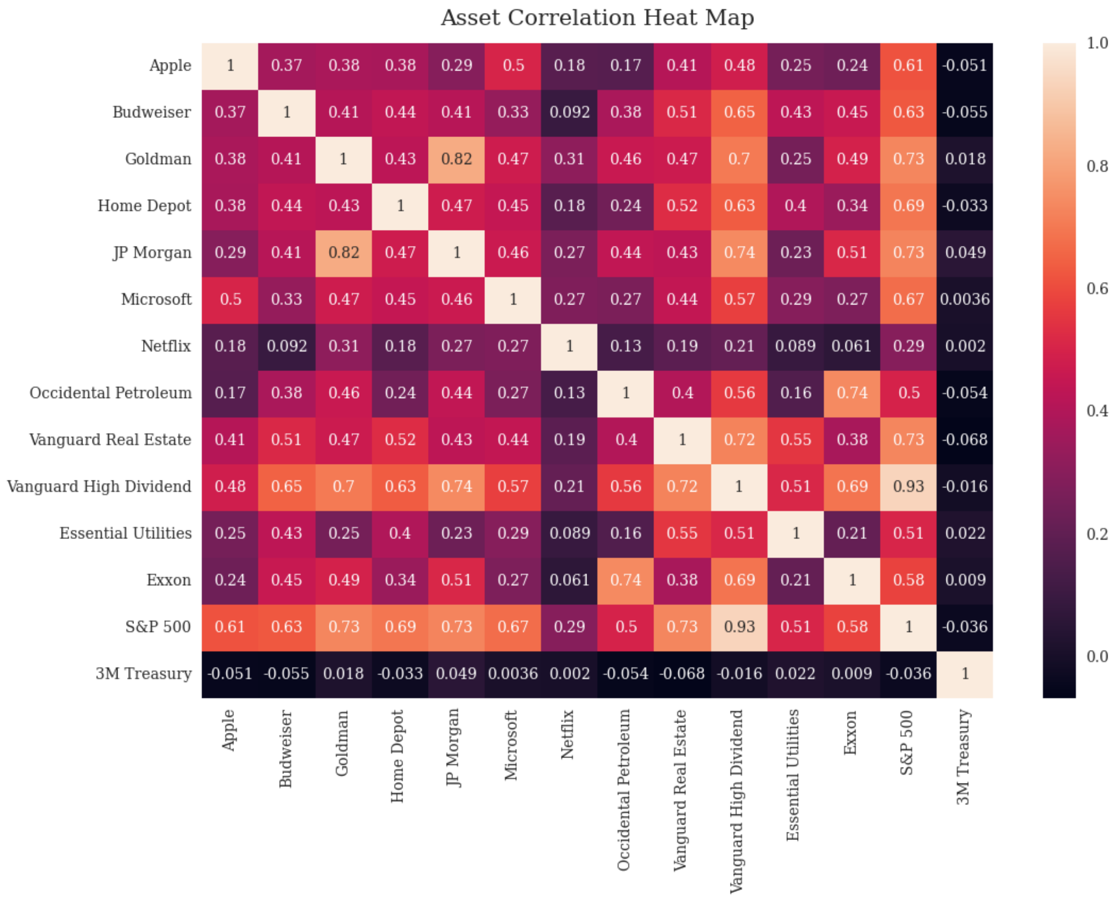
```
Budweiser                   1.120978       0.014856          -0.227337
Goldman                     1.445631       0.036346          -0.245411

                            Alpha  r_squared  Idiosyncratic Risk  \
Netflix                  0.026263   0.086315             0.91369
Apple                    0.013296   0.377397             0.62260
Home Depot               0.009925   0.480785             0.51922
Microsoft                0.008943   0.452824             0.54718
3M Treasury              0.005820   0.001278             0.99872
Essential Utilities      0.004817   0.257605             0.74239
Vanguard High Dividend   0.002184   0.871161             0.12884
Vanguard Real Estate     0.001636   0.533694             0.46631
JP Morgan                0.000387   0.527927             0.47207
S&P 500                  0.000000   1.000000             0.00000
Exxon                   -0.001084   0.339271             0.66073
Occidental Petroleum    -0.003637   0.248803             0.75120
Budweiser               -0.003849   0.395236             0.60476
Goldman                 -0.004039   0.534889             0.46511

                       tracking error
Netflix                        0.5384
Apple                          0.2135
Home Depot                     0.1528
Microsoft                      0.1597
3M Treasury                    0.0303
Essential Utilities            0.1603
Vanguard High Dividend         0.0500
Vanguard Real Estate           0.1242
JP Morgan                      0.1766
S&P 500                        0.0000
Exxon                          0.2001
Occidental Petroleum           0.4020
Budweiser                      0.2032
Goldman                        0.1975
```

- Most assets generate a **positive alpha** in this time period. *Goldman* has the highest systematic risk with a beta coeffcient of 1.129.
- ALl of these statistics are scaled to a yearly basis.

6 Correlation Heat Map

[ ]:
```
asset_corr = rets.corr()

plt.figure(figsize = (12,8))
heatmap = sns.heatmap(asset_corr,vmax = 1,annot = True)
heatmap.set_title("Asset Correlation Heat Map", fontdict={'fontsize':15}, pad =␣
 ↪12)
```

[ ]: Text(0.5, 1.0, 'Asset Correlation Heat Map')




```
[ ]: sorted_corr = asset_corr.unstack().sort_values().to_frame('Correlations')
     sorted_corr = sorted_corr[sorted_corr['Correlations'] != 1]
     sorted_corr.nlargest(len(rets.columns), 'Correlations')
```

[ ]:

		Correlations
S&P 500	Vanguard High Dividend	0.933360
Vanguard High Dividend	S&P 500	0.933360
JP Morgan	Goldman	0.823563
Goldman	JP Morgan	0.823563
Exxon	Occidental Petroleum	0.740123
Occidental Petroleum	Exxon	0.740123
Vanguard High Dividend	JP Morgan	0.738586
JP Morgan	Vanguard High Dividend	0.738586
Goldman	S&P 500	0.731361
S&P 500	Goldman	0.731361
	Vanguard Real Estate	0.730544
Vanguard Real Estate	S&P 500	0.730544

```
S&P 500              JP Morgan                 0.726586
JP Morgan            S&P 500                   0.726586
```

[ ]:
```
sorted_corr.nsmallest(len(rets.columns), 'Correlations')
```

[ ]:
```
                                               Correlations
Vanguard Real Estate   3M Treasury                -0.067830
3M Treasury            Vanguard Real Estate       -0.067830
                       Budweiser                  -0.055113
Budweiser              3M Treasury                -0.055113
Occidental Petroleum   3M Treasury                -0.053701
3M Treasury            Occidental Petroleum       -0.053701
                       Apple                      -0.050922
Apple                  3M Treasury                -0.050922
S&P 500                3M Treasury                -0.035752
3M Treasury            S&P 500                    -0.035752
                       Home Depot                 -0.032992
Home Depot             3M Treasury                -0.032992
Vanguard High Dividend 3M Treasury                -0.016380
3M Treasury            Vanguard High Dividend     -0.016380
```

7 Optimal Portfolio VIA Regression

- This is the method Harvard did to optimize their endowment fund.
- I am going to use excess returns to calculate this optimization because it is more realistic.

[ ]:
```
rets_excess = rets.subtract(rets['3M Treasury'], axis = 0).drop(columns = ['3M␣
↪Treasury'])
```

[ ]:
```
Ntime, Nassets = rets_excess.shape
# Description of Individual Asset Sharpe Ratios

(rets_excess.mean()/rets_excess.std()).to_frame().describe().rename({0:'Sharpe␣
↪Ratio Summary'},axis=1).drop(index = ['count']).style.format('{:.2%}'.format)
```

[ ]: <pandas.io.formats.style.Styler at 0x13a56cbbb20>

[ ]:
```
y = np.ones((Ntime,1))
x = rets_excess

beta = LinearRegression(fit_intercept=False, positive = True).fit(x,y).coef_.
↪transpose()

beta /= beta.sum()

beta
beta = pd.DataFrame(beta, columns = ['Weight'], index = [rets_excess.columns])
```

```
beta
beta_1 = np.matrix(beta)
```

8 Calculating Performance of this Hypothetical Portfolio

[ ]:
```
def mvo_performance_stats(asset_returns,cov_matrix,port_weights,␣
↪port_type,period):
    """
        Returns the Annualized Performance Stats for given asset returns,␣
↪portfolio weights and covariance matrix
        Inputs:
            asset_return - Excess return over the risk free rate for each asset␣
↪(n x 1) Vector
            cov_matrix = nxn covariance matrix for the assets
            port_weights = weights of the assets in the portfolio (1 x n) Vector
            port_type = Type of Portfolio | Eg - Tangency or Mean-Variance␣
↪Portfolio
            period = Monthly frequency
    """

    ret = np.dot(port_weights,asset_returns)*period
    vol = np.sqrt(port_weights @ cov_matrix @ port_weights.T)*np.sqrt(period)
    sharpe = ret/vol

    stats = pd.DataFrame([[ret,vol,sharpe]],columns= ["Annualized␣
↪Return","Annualized Volatility","Annualized Sharpe Ratio"], index =␣
↪[port_type])
    return stats
```

[ ]:
```
mvo_performance_stats(asset_returns=rets_excess.mean(), cov_matrix=rets_excess.
↪cov(), port_weights=np.array(beta).T ,port_type= "Non-Negative Least␣
↪Squares", period = 12)
```

[ ]:
```
                              Annualized Return  Annualized Volatility  \
Non-Negative Least Squares  [0.2190126752149181]           0
0  0.20404

                            Annualized Sharpe Ratio
Non-Negative Least Squares             0
0  1.073382
```

[ ]:
```
beta
```

[ ]:
```
                         Weight
Apple                  0.343494
Budweiser              0.000000
```

```
Goldman                   0.000000
Home Depot                0.392319
JP Morgan                 0.000000
Microsoft                 0.107241
Netflix                   0.156946
Occidental Petroleum      0.000000
Vanguard Real Estate      0.000000
Vanguard High Dividend    0.000000
Essential Utilities       0.000000
Exxon                     0.000000
S&P 500                   0.000000
```

- If you invest in just those assets, your annualized return would be $\mu = 21.9\%$ with annualized $\sigma = 20.404\%$
- The weights do make sense given the historical performances of all those assets. Notice how Home Depot receives the highest weight.

[ ]:
```
rets_excess.shape
```

[ ]:
```
(160, 13)
```

[ ]:
```
np.array(beta).shape
```

[ ]:
```
(13, 1)
```

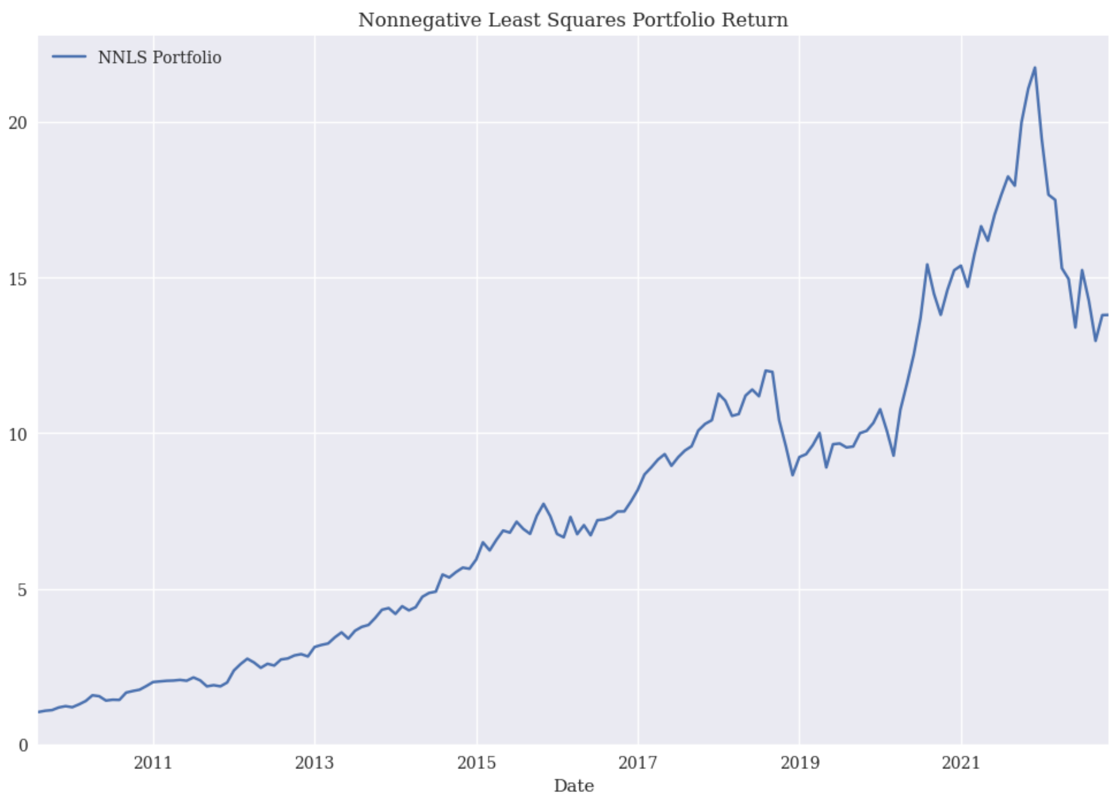
[ ]:
```
performance_NNLS = rets_excess@(np.array(beta))
performance_NNLS.columns = ['NNLS Portfolio']
```

[ ]:
```
performance_summary(performance_NNLS)
```

[ ]:
```
                    Mean  Volatility  Sharpe Ratio  Skewness  Excess Kurtosis  \
NNLS Portfolio  0.219013     0.20404      1.073382 -0.124825          0.26561

                VaR (0.05)  CVaR (0.05)  Max Drawdown       Peak     Bottom  \
NNLS Portfolio   -0.090742    -0.107336      -0.40393 2021-12-01 2022-09-01

               Recovery
NNLS Portfolio      NaT
```

[ ]:
```
performance = ((1+performance_NNLS).cumprod())
performance.plot(title = 'Nonnegative Least Squares Portfolio Return', figsize␣
↪= (12,8))
```

[ ]:
```
<AxesSubplot:title={'center':'Nonnegative Least Squares Portfolio Return'},
xlabel='Date'>
```



- You tail risk is not that bad considering you are heavily invested in Tech during this time. Your Max Draw down would have occrued recently, but a gross return of '1280% is not too bad over 13 years.
- If you put $\$10,000$ in this portfolio it would have grown to $\$138,000$

```
[ ]: 10000*13.8
```

```
[ ]: 138000.0
```

- The biggest limitation of your portfolio is that you do not have any diversification. There really isn't a way to find a better sharpe ratio unless you diversify across another asset classes/different sectors of the economy/different types of equities (especially value stocks instead of growth equity).
- Risk aversion is extremly contagious during market stress, and now that we are heading into a riskier world, risk aversion might be prolonged for a long time so getting back into the blue-chip companies that generates lots of cash and have lots of tangible assets unlike growth equity would definitely limit risk for the forseable future.