Portfolio Optimization

Introduction

When an investor chooses to invest money in different assets or stocks of companies, they create a portfolio of the stocks. These stocks earn returns over time making the investment profitable. The returns are based on the market movement, economic factors, company performance and various other factors which influence the value of stocks and investors take risk when they are investing in a company's stock.

There are many models that try to exploit the risk-return relationship such that investors make most of their money. This project focuses on Modern Portfolio Theory and Capital Asset Pricing Model, which are two popular empirical models that establish a relationship between risky assets and expected returns enabling an investor to create an optimal portfolio.

This project contains python code that creates a sample optimal portfolio based on 20 diversified stocks and explores different concepts like Sharpe ratio, tangency portfolio and efficient portfolio derived from the two models stated above. The code also contains statistical elements like correlation, covariance and their role in determining the optimal portfolio.

Source of Data

The prices of the 20 stocks have been taken from [Yahoo Finance] for the time-period of 1st January 2019 to 31st December 2020. The risk-free rate has been considered as 0.02 based on 13 week treasury bill sourced from the [government website].

[4]:
```
#2 Import Data
df = data.DataReader(['GOOGL', 'WMT','LUV','CCF','TSLA','BABA','GLD','PFE','FB','NIO','WTM', 'BIO','FCX','VIRT','JNJ','DIS','AAPL','NRC','ALL','ETSY'], 'yahoo', start='2019/01/01', end='2020/12/31')
df.head()
```

[4]:

Attributes									Adj Close		...										Volume
Symbols	GOOGL	WMT	LUV	CCF	TSLA	BABA	GLD	PFE	FB	NIO	...	WTM	BIO	FCX	VIRT	JNJ	DIS	AAPL	NRC	ALL	ETSY
Date																					
2019-01-02	1054.680054	89.610855	46.326736	97.973686	62.023998	136.699997	121.330002	37.603287	135.679993	6.20	...	11900.0	165800.0	17940900.0	801200.0	7631700.0	9723500.0	148158800.0	46000.0	2208000.0	1968600.0
2019-01-03	1025.469971	89.150032	44.823132	92.828590	60.071999	130.600006	122.430000	36.551258	131.740005	6.05	...	10500.0	180200.0	22417600.0	1118800.0	8654500.0	10594700.0	365248800.0	10200.0	3034900.0	1803100.0
2019-01-04	1078.069946	89.706871	47.034313	96.919044	63.537998	139.750000	121.440002	37.385921	137.949997	6.36	...	12100.0	446800.0	25597600.0	1390700.0	8831700.0	10122800.0	234428400.0	43000.0	3061600.0	4075000.0
2019-01-07	1075.920044	90.762917	47.044140	95.608131	66.991997	143.100006	121.860001	37.585892	138.050003	6.50	...	16300.0	354300.0	16813100.0	1423200.0	8404700.0	6714700.0	219111200.0	40500.0	4135800.0	3153900.0
2019-01-08	1085.369995	91.396545	46.719833	94.494347	67.070000	146.789993	121.529999	37.759781	142.529999	6.40	...	22400.0	235800.0	22809600.0	1140600.0	9351600.0	8730700.0	164101200.0	9200.0	2627500.0	2824800.0

5 rows × 120 columns

[5]:
```
#3 Closing price of company's stock on the given day
df = df['Adj Close']
df.head()
```

[5]:

Symbols	GOOGL	WMT	LUV	CCF	TSLA	BABA	GLD	PFE	FB	NIO	WTM	BIO	FCX	VIRT	JNJ	DIS	AAPL	NRC	ALL	ETSY
Date																				
2019-01-02	1054.680054	89.610855	46.326736	97.973686	62.023998	136.699997	121.330002	37.603287	135.679993	6.20	852.304443	228.080002	10.135902	23.510199	120.209579	107.654343	38.505024	37.549950	77.756592	47.000000
2019-01-03	1025.469971	89.150032	44.823132	92.828590	60.071999	130.600006	122.430000	36.551258	131.740005	6.05	848.067810	224.300003	9.852178	24.161501	118.299408	105.046211	34.669640	36.471489	76.480324	46.029999
2019-01-04	1078.069946	89.706871	47.034313	96.919044	63.537998	139.750000	121.440002	37.385921	137.949997	6.36	851.736206	226.279999	10.585953	24.749481	120.284859	108.286606	36.149662	37.334255	78.737595	49.700001
2019-01-07	1075.920044	90.762917	47.044140	95.608131	66.991997	143.100006	121.860001	37.585892	138.050003	6.50	854.208374	233.669998	10.762061	24.279097	119.513252	109.225136	36.069202	37.598969	78.613770	51.570000
2019-01-08	1085.369995	91.396545	46.719833	94.494347	67.070000	146.789993	121.529999	37.759781	142.529999	6.40	874.793335	235.529999	10.957734	24.441921	122.289139	110.074753	36.756794	37.824463	78.604263	53.880001

Components of an Optimal Portfolio

1. Mean of the individual stock returns.
   The reason for calculating the mean for individual stocks is the data is from past 2 years and mean provides the aggregated value of all the prices of each stock.

```python
#7 monthly returns for individual companies
ind_er = df.resample('M').last().pct_change().mean()
ind_er
```

```
Symbols
GOOGL    0.021881
WMT      0.020528
LUV     -0.002873
CCF      0.003673
TSLA     0.139801
BABA     0.018007
GLD      0.016537
PFE      0.002254
FB       0.025350
NIO      0.152727
WTM      0.006940
BIO      0.040787
FCX      0.050233
VIRT     0.006219
JNJ      0.011314
DIS      0.027666
AAPL     0.057004
NRC      0.010107
ALL      0.013785
ETSY     0.072267
dtype: float64
```

2. Standard Deviation of each stock
   The standard deviation in this scenario is representation of volatility in the stocks.

```python
[16]: #12 Volatility is given by the annual standard deviation. We multiply by 250 because there are 250 trading days/year.
      ann_sd = df.pct_change().apply(lambda x: np.log(1+x)).std().apply(lambda x: x*np.sqrt(250))
      ann_sd
```

```
[16]: Symbols
      GOOGL    0.318924
      WMT      0.241640
      LUV      0.469920
      CCF      0.527314
      TSLA     0.721281
      BABA     0.361893
      GLD      0.160236
      PFE      0.285506
      FB       0.378130
      NIO      0.988628
      WTM      0.307639
      BIO      0.357210
      FCX      0.585664
      VIRT     0.390418
      JNJ      0.242504
      DIS      0.376916
      AAPL     0.377517
      NRC      0.484141
      ALL      0.326238
      ETSY     0.611939
      dtype: float64
```

3. Correlation matrix

   MPT is based on zero correlation between the stocks. Zero correlation implies that there is no influence or interference from one stock to another. A well-diversified portfolio is comprised of stocks which are not correlated to each other.

```
#5 Correlation matrix
corr_matrix = df.pct_change().apply(lambda x: np.log(1+x)).corr()
corr_matrix
```

Symbols	GOOGL	WMT	LUV	CCF	TSLA	BABA	GLD	PFE	FB	NIO	WTM	BIO	FCX	VIRT	JNJ	DIS	AAPL	NRC	ALL	ETSY
Symbols																				
GOOGL	1.000000	0.431242	0.415517	0.450793	0.408503	0.506703	0.054785	0.490127	0.743314	0.230212	0.353908	0.503737	0.555861	0.001473	0.525393	0.557220	0.707680	0.528347	0.526153	0.402995
WMT	0.431242	1.000000	0.186581	0.217555	0.156920	0.241041	0.132525	0.398998	0.354366	0.038558	0.179489	0.407964	0.248231	0.023612	0.495475	0.300078	0.464867	0.400684	0.370724	0.293044
LUV	0.415517	0.186581	1.000000	0.419045	0.211959	0.301615	0.014463	0.387413	0.343160	0.182980	0.422604	0.188630	0.486466	-0.076458	0.339526	0.558799	0.387193	0.392653	0.521924	0.185905
CCF	0.450793	0.217555	0.419045	1.000000	0.313295	0.280856	0.039717	0.308718	0.354843	0.169136	0.487770	0.259497	0.580944	0.083258	0.285014	0.522887	0.396869	0.642192	0.556578	0.214333
TSLA	0.408503	0.156920	0.211959	0.313295	1.000000	0.336423	0.124655	0.170789	0.373681	0.287638	0.219803	0.214798	0.433152	0.169294	0.173877	0.282632	0.452450	0.327921	0.340357	0.278024
BABA	0.506703	0.241041	0.301615	0.280856	0.336423	1.000000	-0.013665	0.290717	0.485922	0.313140	0.228287	0.349593	0.448624	0.008589	0.343060	0.325795	0.537764	0.270933	0.294950	0.329383
GLD	0.054785	0.132525	0.014463	0.039717	0.124655	-0.013665	1.000000	0.048645	0.092056	0.007598	0.166650	0.125834	0.082079	0.153194	0.076574	0.016221	0.083563	0.108528	0.038036	0.193292
PFE	0.490127	0.398998	0.387413	0.308718	0.170789	0.290717	0.048645	1.000000	0.374533	0.070735	0.327328	0.401161	0.373635	0.012519	0.634063	0.385840	0.442124	0.371603	0.518858	0.298378
FB	0.743314	0.354366	0.343160	0.354843	0.373681	0.485922	0.092056	0.374533	1.000000	0.242950	0.301460	0.433782	0.458689	-0.000855	0.417125	0.456255	0.698051	0.443211	0.438368	0.389278
NIO	0.230212	0.038558	0.182980	0.169136	0.287638	0.313140	0.007598	0.070735	0.242950	1.000000	0.162245	0.126421	0.308099	0.039757	0.101680	0.144214	0.234275	0.171213	0.170606	0.201583
WTM	0.353908	0.179489	0.422604	0.487770	0.219803	0.228287	0.166650	0.327328	0.301460	0.162245	1.000000	0.210093	0.431367	0.095404	0.355964	0.482638	0.370271	0.458080	0.559534	0.157328
BIO	0.503737	0.407964	0.188630	0.259497	0.214798	0.349593	0.125834	0.401161	0.433782	0.126421	0.210093	1.000000	0.351492	0.035102	0.495816	0.299400	0.476779	0.299977	0.370560	0.409890
FCX	0.555861	0.248231	0.486466	0.580944	0.433152	0.448624	0.082079	0.373635	0.458689	0.308099	0.431367	0.351492	1.000000	0.073321	0.428564	0.581487	0.541364	0.509915	0.550062	0.348628
VIRT	0.001473	0.023612	-0.076458	0.083258	0.169294	0.008589	0.153194	0.012519	-0.000855	0.039757	0.095404	0.035102	0.073321	1.000000	-0.018289	0.022107	0.054959	0.098463	0.074539	0.051559
JNJ	0.525393	0.495475	0.339526	0.285014	0.173877	0.343060	0.076574	0.634063	0.417125	0.101680	0.355964	0.495816	0.428564	-0.018289	1.000000	0.443463	0.525322	0.395411	0.545234	0.329581
DIS	0.557220	0.300078	0.558799	0.522887	0.282632	0.325795	0.016221	0.385840	0.456255	0.144214	0.482638	0.299400	0.581487	0.022107	0.443463	1.000000	0.483393	0.523059	0.543378	0.209011
AAPL	0.707680	0.464867	0.387193	0.396869	0.452450	0.537764	0.083563	0.442124	0.698051	0.234275	0.370271	0.476779	0.541364	0.054959	0.525322	0.483393	1.000000	0.433876	0.548974	0.450672
NRC	0.528347	0.400684	0.392653	0.642192	0.327921	0.270933	0.108528	0.371603	0.443211	0.171213	0.458080	0.299977	0.509915	0.098463	0.395411	0.523059	0.433876	1.000000	0.569442	0.223877
ALL	0.526153	0.370724	0.521924	0.556578	0.340357	0.294950	0.038036	0.518858	0.438368	0.170606	0.559534	0.370560	0.550062	0.074539	0.545234	0.543378	0.548974	0.569442	1.000000	0.331373
ETSY	0.402995	0.293044	0.185905	0.214333	0.278024	0.329383	0.193292	0.298378	0.389278	0.201583	0.157328	0.409890	0.348628	0.051559	0.329581	0.209011	0.450672	0.223877	0.331373	1.000000

4. Covariance matrix

   Covariance depicts a directional relationship between assets. MPT uses covariance to assets and reduce the overall portfolio risk. A negative covariance ensures that change in price of one stock does not impact the price of other stock.

```
#4 Covariance
cov_matrix = df.pct_change().apply(lambda x: np.log(1+x)).cov()
cov_matrix
```

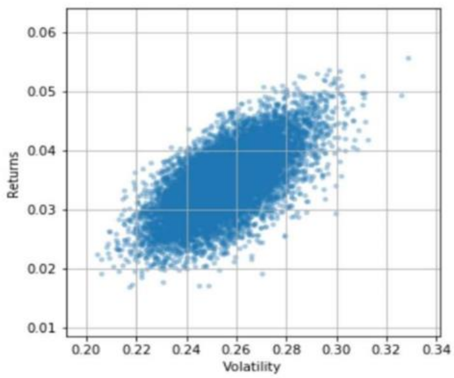
Symbols	GOOGL	WMT	LUV	CCF	TSLA	BABA	GLD	PFE	FB	NIO	WTM	BIO	FCX	VIRT	JNJ	DIS	AAPL	NRC	ALL	ETSY
Symbols																				
GOOGL	4.068508e-04	0.000133	0.000249	0.000303	0.000376	0.000234	0.000011	0.000179	3.585592e-04	0.000290	0.000139	0.000230	0.000415	7.334031e-07	0.000163	0.000268	0.000341	0.000326	0.000219	0.000315
WMT	1.329343e-04	0.000234	0.000085	0.000111	0.000109	0.000084	0.000021	0.000110	1.295152e-04	0.000037	0.000053	0.000141	0.000141	8.910347e-06	0.000116	0.000109	0.000170	0.000188	0.000117	0.000173
LUV	2.490925e-04	0.000085	0.000883	0.000415	0.000287	0.000205	0.000004	0.000208	2.439055e-04	0.000340	0.000244	0.000127	0.000536	-5.610933e-05	0.000155	0.000396	0.000275	0.000357	0.000320	0.000214
CCF	3.032453e-04	0.000111	0.000415	0.001112	0.000477	0.000214	0.000013	0.000186	2.830133e-04	0.000353	0.000317	0.000196	0.000718	6.856216e-05	0.000146	0.000416	0.000316	0.000656	0.000383	0.000277
TSLA	3.758783e-04	0.000109	0.000287	0.000477	0.002081	0.000351	0.000058	0.000141	4.076682e-04	0.000820	0.000195	0.000221	0.000732	1.906930e-04	0.000122	0.000307	0.000493	0.000458	0.000320	0.000491
BABA	2.339274e-04	0.000084	0.000205	0.000214	0.000351	0.000524	-0.000003	0.000120	2.659790e-04	0.000448	0.000102	0.000181	0.000380	4.854330e-06	0.000120	0.000178	0.000294	0.000190	0.000139	0.000292
GLD	1.119884e-05	0.000021	0.000004	0.000013	0.000058	-0.000003	0.000103	0.000009	2.231061e-05	0.000005	0.000033	0.000029	0.000031	3.833470e-05	0.000012	0.000004	0.000020	0.000034	0.000008	0.000076
PFE	1.785140e-04	0.000110	0.000208	0.000186	0.000141	0.000120	0.000009	0.000326	1.617361e-04	0.000080	0.000115	0.000164	0.000250	5.581879e-06	0.000176	0.000166	0.000191	0.000205	0.000193	0.000209
FB	3.585592e-04	0.000130	0.000244	0.000283	0.000408	0.000266	0.000022	0.000162	5.719283e-04	0.000363	0.000140	0.000234	0.000406	-5.047325e-07	0.000153	0.000260	0.000399	0.000325	0.000216	0.000360
NIO	2.903408e-04	0.000037	0.000340	0.000353	0.000820	0.000448	0.000005	0.000080	3.632871e-04	0.003910	0.000197	0.000179	0.000714	6.138130e-05	0.000098	0.000215	0.000350	0.000328	0.000220	0.000488
WTM	1.388926e-04	0.000053	0.000244	0.000317	0.000195	0.000102	0.000033	0.000115	1.402723e-04	0.000197	0.000379	0.000092	0.000311	4.583515e-05	0.000106	0.000224	0.000172	0.000273	0.000225	0.000118
BIO	2.295489e-04	0.000141	0.000127	0.000196	0.000221	0.000181	0.000029	0.000164	2.343668e-04	0.000179	0.000092	0.000510	0.000294	1.958132e-05	0.000172	0.000161	0.000257	0.000208	0.000173	0.000358
FCX	4.153004e-04	0.000141	0.000536	0.000718	0.000732	0.000380	0.000031	0.000250	4.063196e-04	0.000714	0.000311	0.000294	0.001372	6.706075e-05	0.000243	0.000513	0.000479	0.000578	0.000420	0.000500
VIRT	7.334031e-07	0.000009	-0.000056	0.000069	0.000191	0.000005	0.000038	0.000006	-5.047325e-07	0.000061	0.000046	0.000020	0.000067	6.097043e-04	-0.000007	0.000013	0.000032	0.000074	0.000038	0.000049
JNJ	1.625367e-04	0.000116	0.000155	0.000146	0.000122	0.000120	0.000012	0.000176	1.529981e-04	0.000098	0.000106	0.000172	0.000243	-6.926400e-06	0.000235	0.000162	0.000192	0.000186	0.000173	0.000196
DIS	2.679287e-04	0.000109	0.000396	0.000416	0.000307	0.000178	0.000004	0.000166	2.601076e-04	0.000215	0.000224	0.000161	0.000513	1.301234e-05	0.000162	0.000568	0.000275	0.000382	0.000267	0.000193
AAPL	3.408163e-04	0.000170	0.000275	0.000316	0.000493	0.000294	0.000020	0.000191	3.985881e-04	0.000350	0.000172	0.000257	0.000479	3.240121e-05	0.000192	0.000275	0.000570	0.000317	0.000270	0.000416
NRC	3.263166e-04	0.000188	0.000357	0.000656	0.000458	0.000190	0.000034	0.000205	3.245511e-04	0.000328	0.000273	0.000208	0.000578	7.444503e-05	0.000186	0.000382	0.000317	0.000938	0.000360	0.000265
ALL	2.189746e-04	0.000117	0.000320	0.000383	0.000320	0.000139	0.000008	0.000193	2.163085e-04	0.000220	0.000225	0.000173	0.000420	3.797571e-05	0.000173	0.000267	0.000270	0.000360	0.000426	0.000265
ETSY	3.145980e-04	0.000173	0.000214	0.000277	0.000491	0.000292	0.000076	0.000209	3.603035e-04	0.000488	0.000118	0.000358	0.000500	4.927245e-05	0.000196	0.000193	0.000416	0.000265	0.000265	0.001498

5. Efficient Frontier

This frontier is all possible combinations of highest expected returns at every level of risk. It helps the investors understand the returns associated with each risk and caters to all kind of risk profiles.

```
#17 Plot efficient frontier
portfolios.plot.scatter(x='Volatility', y='Returns', marker='o', s=10, alpha=0.3, grid=True, figsize=[5,5])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f71bf664e90>
```



6. Minimum Variance Portfolio

This portfolio is the combination highest returns at lowest level of risk. It is usually located at the bottom of the efficient frontier. The below code gives the weights of 20 stocks which will make a minimum variance portfolio. The Sharpe ratio of this portfolio is 0.12.

```
#18 Minimum Variance Portfolio
min_vol_port = portfolios.iloc[portfolios['Volatility'].idxmin()]
# idxmin() gives us the minimum value in the column specified.
min_vol_port
```

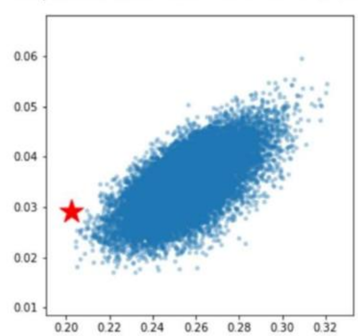
```
Returns         0.029095
Volatility      0.202541
GOOGL weight    0.025566
WMT weight      0.119650
LUV weight      0.053298
CCF weight      0.030737
TSLA weight     0.047964
BABA weight     0.052673
GLD weight      0.124873
PFE weight      0.094193
FB weight       0.038626
NIO weight      0.042846
WTM weight      0.062311
BIO weight      0.003021
FCX weight      0.018016
VIRT weight     0.117905
JNJ weight      0.033720
DIS weight      0.042687
AAPL weight     0.027983
NRC weight      0.018694
ALL weight      0.002467
ETSY weight     0.042770
Name: 17030, dtype: float64
```

```
#19 Sharpe ratio of Minimum Variance Portfolio
min_var_port_sharpe_ratio=(0.022511-0.02)/0.204274
min_var_port_sharpe_ratio*10
```

```
0.12292313265515921
```

```
#20 plotting the minimum volatility portfolio
plt.subplots(figsize=[5,5])
plt.scatter(portfolios['Volatility'], portfolios['Returns'],marker='o', s=10, alpha=0.3)
plt.scatter(min_vol_port[1], min_vol_port[0], color='r', marker='*', s=500)
```

```
<matplotlib.collections.PathCollection at 0x7f152b21c8d0>
```



7. Optimal Portfolio

   This portfolio has the maximum returns at any given level of risk. It is also referred to as tangency portfolio. This portfolio will have the highest Sharpe ratio. This ratio is the calculated by subtracting risk free return from excess expected returns of the portfolio and divide the result by standard deviation of the portfolio with excess returns. Sharpe ratio of optimal portfolio is 1.13. Below code shows the weights of the 20 stocks at which this tangency/optimal portfolio is formed.

```
#21 Optimal Portfolio
rf = 0.02 # risk free ratio
optimal_risky_port = portfolios.iloc[((portfolios['Returns']-rf)/portfolios['Volatility']).idxmax()]
optimal_risky_port
```

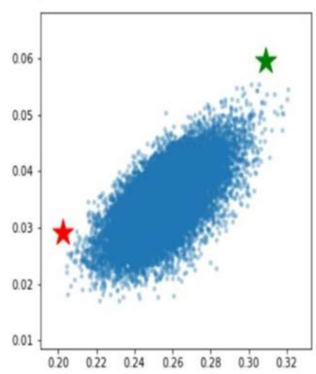
```
Returns         0.059579
Volatility      0.308588
GOOGL weight    0.039746
WMT weight      0.006135
LUV weight      0.008442
CCF weight      0.008565
TSLA weight     0.150797
BABA weight     0.094864
GLD weight      0.013188
PFE weight      0.035244
FB weight       0.047532
NIO weight      0.089721
WTM weight      0.007745
BIO weight      0.038022
FCX weight      0.010845
VIRT weight     0.065020
JNJ weight      0.009014
DIS weight      0.070840
AAPL weight     0.117561
NRC weight      0.019179
ALL weight      0.051935
ETSY weight     0.115604
Name: 11657, dtype: float64
```

```
#22 Tangency portfolio of the optimal risky asset
tangency_portfolio_sharpe_ratio=(0.050658-0.02)/0.269230
tangency_portfolio_sharpe_ratio*10
```

```
1.138728967797051
```

```
#23 Plotting optimal portfolio
plt.subplots(figsize=(5, 5))
plt.scatter(portfolios['Volatility'], portfolios['Returns'],marker='o', s=10, alpha=0.3)
plt.scatter(min_vol_port[1], min_vol_port[0], color='r', marker='*', s=500)
plt.scatter(optimal_risky_port[1], optimal_risky_port[0], color='g', marker='*', s=500)
```

```
<matplotlib.collections.PathCollection at 0x7f152b189d50>
```



8. Capital Allocation Line

   This line represents all possible portfolio combinations with risky and risk-free assets. The line shows the returns investor may earn at the assumed level of risk. This line intersects with the efficient frontier at Tangency portfolio.

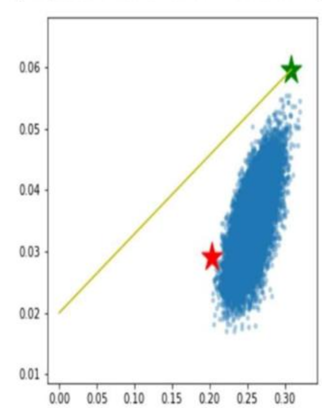
   In this project we have chosen a risk averse investor at A = 3. The below code is the capital allocation line for the given 20 stocks. The blue dot on CAL is the investor's optimal portfolio for risk aversion level A=3

```
#26
data2 = {'utility':utility, 'cal_y':cal_y, 'cal_x':cal_x}
cal = pd.DataFrame(data2)
cal.head()
```

	utility	cal_y	cal_x
0	0.020000	0.020000	0.000000
1	0.021687	0.022083	0.016241
2	0.022584	0.024166	0.032483
3	0.022688	0.026249	0.048724
4	0.022002	0.028332	0.064966

```
#27
plt.subplots(figsize=(5, 5))
plt.scatter(portfolios['Volatility'], portfolios['Returns'],marker='o', s=10, alpha=0.3)
plt.scatter(min_vol_port[1], min_vol_port[0], color='r', marker='*', s=500)
plt.scatter(optimal_risky_port[1], optimal_risky_port[0], color='g', marker='*', s=500)
plt.plot(cal_x, cal_y, color='y')
```

```
[<matplotlib.lines.Line2D at 0x7f152b071d90>]
```




```
#28 plotting optimal portfolio for A=3
cal['utility'].idxmax()
```

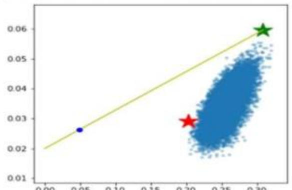
```
3
```

```
investors_port = cal.iloc [cal['utility'].idxmax()]
investors_port
```

```
utility    0.022688
cal_y      0.026249
cal_x      0.048724
Name: 3, dtype: float64
```

```
plt.subplots(figsize=(5, 5))
plt.scatter(portfolios['Volatility'], portfolios['Returns'],marker='o', s=10, alpha=0.3)
plt.scatter(min_vol_port[1], min_vol_port[0], color='r', marker='*', s=500)
plt.scatter(optimal_risky_port[1], optimal_risky_port[0], color='g', marker='*', s=500)
plt.plot(cal_x, cal_y, color='y')
plt.plot(investors_port[2], investors_port[1], 'o', color='b')
```

```
[<matplotlib.lines.Line2D at 0x7f152afe5390>]
```



Conclusion

A portfolio's Sharpe ratio with value more than 1 is a good investment. Our selection of stocks has resulted in a portfolio with 1.13 Sharpe ratio.

A good portfolio is a sign of diversified portfolio. There is a relatively higher correlation between tech companies such as Google, Apple and Facebook. On the other hand, there is a lower correlation between different industries. For example, between Google and SPDR Gold Trust or between Google and Virtual Financial. This shows that benefits of diversification came from portfolios composed of stocks from different industries. In another words, diversification is spreading risk across different types of investments to reduce the consequences of a wrong forecast. This observation matches with the general expectations of investors.

This portfolio is ideal for risk averse, risk neutral and risk lover. The below plots from the code indicate investors at risk aversion level of A=3, A= 0 and A= -3 have favorable portfolios on the capital allocation line. According to our code, we have observed that both investors at A = 0 and A = -3 will choose same portfolio with higher returns. For A = - 3 the higher returns come by taking higher risks.

```
#28 plotting optimal portfolio for A=3
cal['utility'].idxmax()
```

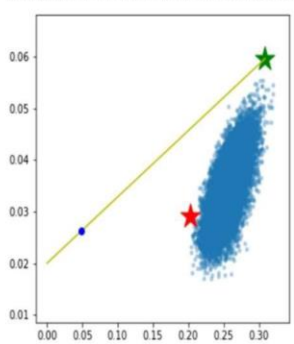
```
3
```

```
investors_port = cal.iloc [cal['utility'].idxmax()]
investors_port
```

```
utility    0.022688
cal_y      0.026249
cal_x      0.048724
Name: 3, dtype: float64
```

```
plt.subplots(figsize=(5, 5))
plt.scatter(portfolios['Volatility'], portfolios['Returns'],marker='o', s=10, alpha=0.3)
plt.scatter(min_vol_port[1], min_vol_port[0], color='r', marker='*', s=500)
plt.scatter(optimal_risky_port[1], optimal_risky_port[0], color='g', marker='*', s=500)
plt.plot(cal_x, cal_y, color='y')
plt.plot(investors_port[2], investors_port[1], 'o', color='b')
```

```
[<matplotlib.lines.Line2D at 0x7f152afe5390>]
```




```
43]: #28 Defining utility function
     cal['utility'].idxmax()
```

```
43]: 19
```

```
44]: #29 Identifying the utility for Investor with A=3 risk aversion
     investors_port = cal.iloc [cal['utility'].idxmax()]
     investors_port
```

```
44]: utility    0.055444
     cal_y      0.055444
     cal_x      0.304998
     Name: 19, dtype: float64
```

```
45]: #30 Plotting the optimal portfolio for investor with A= 3 risk aversion
     plt.subplots(figsize=(5, 5))
     plt.scatter(portfolios['Volatility'], portfolios['Returns'],marker='o', s=10, alpha=0.3)
     plt.scatter(min_vol_port[1], min_vol_port[0], color='r', marker='*', s=500)
     plt.scatter(optimal_risky_port[1], optimal_risky_port[0], color='g', marker='*', s=500)
     plt.plot(cal_x, cal_y, color='y')
     plt.plot(investors_port[2], investors_port[1], 'o', color='b')
```

```
45]: [<matplotlib.lines.Line2D at 0x7fabafbc4c90>]
```

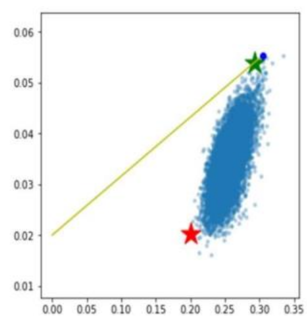


Figure 1 Left Plot A= 3 & Right Plot A = 0 , A= -3