FINM 25000 Assignment 1

HW Group A 6

Lucas Morena, Tony Yu, Charlie Wu, Cerina Yao

<u>Question 1</u>

Table 1. Mean, Volatility and Sharpe Ratio of each asset

	Asset Class	Mean	Volatility	Sharpe Ratio
0	BWX	0.006390	0.076505	0.083525
1	DBC	0.042225	0.181134	0.233116
2	EEM	0.073746	0.193170	0.381770
3	EFA	0.085370	0.160429	0.532138
4	HYG	0.072104	0.085733	0.841033
5	IEF	0.023507	0.058628	0.400955
6	IYR	0.151888	0.182766	0.831048
7	PSP	0.138939	0.217208	0.639657
8	QAI	0.020549	0.048389	0.424667
9	SPY	0.150734	0.142042	1.061189
10	TIP	0.032002	0.045452	0.704078

	Asset Class	Optimal Weight
0	BWX	-0.286533
1	DBC	0.056437
2	EEM	0.037800
3	EFA	-0.002130
4	HYG	0.385248
5	IEF	0.635198
6	IYR	-0.108598
7	PSP	-0.117518
8	QAI	-1.276598
9	SPY	0.691554
10	TIP	0.279602

SPY has the best (highest) Sharpe Ratio, while BWX has the worst (lowest) Sharpe Ratio.

<u>Question 2</u>

Table 2. Correlation matrix

	BWX	DBC	EEM	EFA	HYG	IEF	IYR	PSP	QAI	SPY	TIP
BWX	1.000000	0.330982	0.609331	0.559939	0.507747	0.315212	0.388627	0.486088	0.624505	0.401463	0.551875
DBC	0.330982	1.000000	0.560808	0.580985	0.465019	-0.413355	0.285696	0.487789	0.535462	0.498145	0.075545
EEM	0.609331	0.560808	1.000000	0.851916	0.746547	-0.254467	0.604353	0.797594	0.801072	0.746713	0.233472
EFA	0.559939	0.580985	0.851916	1.000000	0.756339	-0.310853	0.671100	0.905540	0.834043	0.871234	0.160054
HYG	0.507747	0.465019	0.746547	0.756339	1.000000	-0.157887	0.738514	0.813852	0.750569	0.740805	0.235147
IEF	0.315212	-0.413355	-0.254467	-0.310853	-0.157887	1.000000	-0.059095	-0.301569	-0.085488	-0.328127	0.664207
IYR	0.388627	0.285696	0.604353	0.671100	0.738514	-0.059095	1.000000	0.737273	0.613673	0.730715	0.291943
PSP	0.486088	0.487789	0.797594	0.905540	0.813852	-0.301569	0.737273	1.000000	0.821420	0.898894	0.177978
QAI	0.624505	0.535462	0.801072	0.834043	0.750569	-0.085488	0.613673	0.821420	1.000000	0.828297	0.366972
SPY	0.401463	0.498145	0.746713	0.871234	0.740805	-0.328127	0.730715	0.898894	0.828297	1.000000	0.144071
TIP	0.551875	0.075545	0.233472	0.160054	0.235147	0.664207	0.291943	0.177978	0.366972	0.144071	1.000000

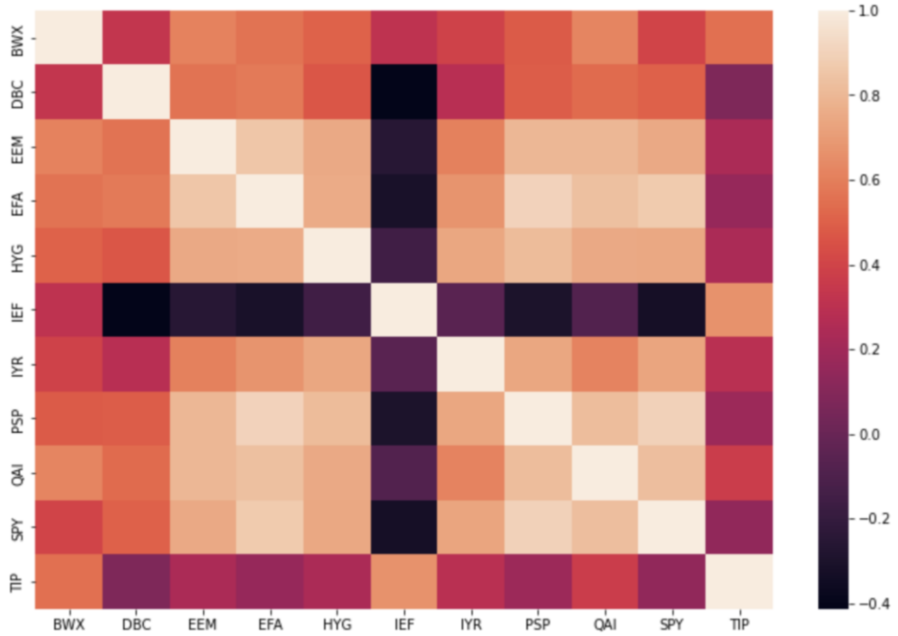


Figure 1. Heat map of correlation matrix

The assets with most negative correlation are DBC and IEF, with a correlation coefficient of -0.41. The assets with most positive correlation are PSP and EFA, with a correlation coefficient of 0.91. Using Sharpe Ratio as a measure of performance, TIP with SR = 0.70 outperformed both domestic bonds (IEF) with SR = 0.40 and foreign bonds (BWX) with SR = 0.08.

Looking at the heat map, TIP seems to have relatively low correlation with all assets except domestic and foreign bonds. Therefore, TIP does seem to expand the investment opportunity set, and Harvard should consider them a separate asset.

<u>Question 3</u>

Table 3. Weights of the tangency portfolio

	Asset Class	Tangency Weight
0	BWX	-0.973072
1	DBC	0.191662
2	EEM	0.128369
3	EFA	-0.007233
4	HYG	1.308310
5	IEF	2.157144
6	IYR	-0.368802
7	PSP	-0.399092
8	QAI	-4.335348
9	SPY	2.348530
10	TIP	0.949532

For the tangency portfolio:

Mean = 0.34

Volatility = 0.26

Sharpe Ratio = 1.28

Question 4

Table 4. Weights of the MV portfolio with target returns of 0.01

	Asset Class	Optimal Weight
0	BWX	-0.028653
1	DBC	0.005644
2	EEM	0.003780
3	EFA	-0.000213
4	HYG	0.038525
5	IEF	0.063520
6	IYR	-0.010860
7	PSP	-0.011752
8	QAI	-0.127660
9	SPY	0.069155
10	TIP	0.027960

For the MV portfolio:

Mean = 0.01

Volatility = 0.0049

Sharpe Ratio = 2.05

The asset that has the most long is SPY, and the asset that has the most short is QAI. This portfolio also has heavy long positions in IEF, HYG and TIP. Since this is an MV portfolio that aims to minimise volatility, the long positions in IEF, HYG and TIP make sense, since all three are bonds and therefore tend to be relatively low-risk and low-volatility. Although SPY is a domestic equity asset and has higher volatility than the three bonds, it has significantly higher mean returns than the bonds and lower volatility than assets with similar mean returns, such as IYR and PSP. This would explain SPY's heavy long position. On the other hand, QAI is heavily shorted since it has a relatively high correlation with all other assets (seen in the heat map) while having a low volatility, which means that its performance is relatively stable and the short position is a reliable counterweight against the other assets, which minimises volatility.

The long and short positions of the portfolios don't necessarily line up with the Sharpe Ratios of the assets; that is, assets with high Sharpe Ratio values don't necessarily have heavy long positions, and vice versa. For example, QAI, which has an extremely heavy short position, has a Sharpe Ratio higher than four other assets, while IYR, with the second highest Sharpe Ratio, also has a short position in the portfolio. On the other hand, IEF, which has the second highest long position in the portfolio, has a Sharpe Ratio even lower than QAI.

Question 5

For the equally weighted portfolio:

Weight of each asset = 0.0125

Mean = 0.01

Volatility = 0.014

Sharpe Ratio = 0.74

Table 5. Weights of the risk-parity portfolio with target returns of 0.01

	Asset Class	Optimal Weight
0	BWX	0.021312
1	DBC	0.009002
2	EEM	0.008441
3	EFA	0.010163
4	HYG	0.019018
5	IEF	0.027811
6	IYR	0.008921
7	PSP	0.007507
8	QAI	0.033690
9	SPY	0.011479
10	TIP	0.035873

For the risk-parity portfolio:

Mean = 0.01

Volatility = 0.0127

Sharpe Ratio = 0.79

Using Sharpe Ratio to measure performance, the MV portfolio with a Sharpe Ratio of 2.05 significantly outperformed both the equally weighted portfolio and the risk-parity portfolio, with Sharpe Ratios of 0.74 and 0.79 respectively.

Question 6

Table 6. Weights of the MV portfolio with target returns of 0.01, using data up to 2020

	Asset Class	Optimal Weights
0	BWX	-0.032512
1	DBC	-0.059029
2	EEM	0.022619
3	EFA	-0.063490
4	HYG	0.384518
5	IEF	0.703993
6	IYR	-0.174176
7	PSP	-0.020700
8	QAI	-0.998132
9	SPY	0.654522
10	TIP	0.047400

Sharpe Ratio using data through 2021: 2.25

Sharpe Ratio using data from 2022: -2.06

The raw data in 2022 had mostly negative excess returns, which means that 2022 was a bear market, leading to the negative Sharpe Ratio.

Appendix

Python code from Jupyter Notebook used to compute results:

HW 1- Group 6A

June 21, 2022

[ ]:

```
 
```

[167]:

```python
import pandas
import matplotlib.pyplot as plot
import seaborn as sbrn
import numpy as np
```

[168]:

```python
dataset = pandas.read_excel("hw1_data.xlsx")
```

[169]:

```python
import pandas as pda
```

[170]:

```python
dataset = pda.read_excel("hw1_data.xlsx")
```

[171]:

```python
dataset
```

[171]:

```
   Unnamed: 0                                     ETF Description
0         SPY                        Domestic Equity SPDR S&P500
1         EFA                        Foreign Equity iShares EAFE
2         EEM                           iShares Emerging Markets
3         PSP       Private Equity Invesco Global Private Equity
4         QAI                     Absolute Return IQ Multi-Strat
5         HYG       High Yield iShares High Yield Corporate Bond
6         DBC           Invesco DB Commodity Index Tracking Fund
7         IYR                 Real Estate iShares US Real Estate
8         IEF            Domestic Bonds iShares 7-10 Year Treasury
9         BWX  Foreign Bonds SPDR Bloomberg Barclay Internati…
10        TIP                Inflation-Indexed iShares TIPS Bond
11        SHV                         iShares Short Treasury Bond
```

[172]:

```python
dataset2 = pda.read_excel("hw1_data.xlsx", "total returns")
```

[173]:

```python
dataset2
```

[173]:

```
         Date       BWX       DBC       EEM       EFA       HYG       IEF  \
0  2009-04-30  0.008993 -0.001000  0.155582  0.115190  0.138461 -0.027452
1  2009-05-31  0.053672  0.162663  0.159400  0.131918  0.028554 -0.020773
2  2009-06-30  0.005149 -0.026259 -0.022495 -0.014049  0.033517 -0.005571
```

```
3   2009-07-31  0.031284  0.018568  0.110146  0.100415  0.069191  0.008317
4   2009-08-31  0.007628 -0.040365 -0.013136  0.045031 -0.016969  0.007633
..         …         …         …         …         …         …         …
153 2022-01-31 -0.026176  0.078922 -0.000205 -0.036350 -0.026549 -0.021130
154 2022-02-28 -0.011197  0.064674 -0.043202 -0.034292 -0.008591 -0.003041
155 2022-03-31 -0.042240  0.091747 -0.033811  0.005190 -0.012871 -0.040609
156 2022-04-30 -0.069696  0.056408 -0.061351 -0.067391 -0.041803 -0.042283
157 2022-05-31  0.011250  0.053396 -0.007787  0.026078  0.025120  0.016799

          IYR       PSP       QAI       SHV       SPY       TIP
0    0.296151  0.230202  0.022882  0.000553  0.099347 -0.017951
1    0.022727  0.053893  0.027865 -0.000471  0.058453  0.019966
2   -0.024863  0.045449 -0.003436  0.000599 -0.000655  0.001981
3    0.105799  0.143247  0.015326 -0.000027  0.074606  0.000879
4    0.131938  0.033413 -0.004151  0.000435  0.036940  0.008414
..        …         …         …         …         …         …
153 -0.082314 -0.086028 -0.020761 -0.000815 -0.052741 -0.020588
154 -0.045881 -0.073602 -0.006746 -0.000363 -0.029517  0.008557
155  0.068646 -0.007721 -0.002587 -0.000363  0.037590 -0.018755
156 -0.041305 -0.125679 -0.033398 -0.000290 -0.087769 -0.021830
157 -0.032675  0.013310 -0.002348  0.000935  0.007913 -0.002847

[158 rows x 13 columns]
```

[174]:
```python
dataset3 = pda.read_excel("hw1_data.xlsx", "excess returns")
```

[175]:
```python
dataset3
```

[175]:
```
          Date       BWX       DBC       EEM       EFA       HYG       IEF  \
0   2009-04-30  0.008440 -0.001553  0.155029  0.114637  0.137908 -0.028005
1   2009-05-31  0.054143  0.163134  0.159871  0.132389  0.029025 -0.020302
2   2009-06-30  0.004550 -0.026858 -0.023094 -0.014648  0.032918 -0.006170
3   2009-07-31  0.031311  0.018595  0.110173  0.100442  0.069218  0.008345
4   2009-08-31  0.007193 -0.040800 -0.013571  0.044596 -0.017404  0.007198
..         …         …         …         …         …         …         …
153 2022-01-31 -0.025361  0.079737  0.000610 -0.035535 -0.025734 -0.020315
154 2022-02-28 -0.010834  0.065037 -0.042840 -0.033929 -0.008228 -0.002679
155 2022-03-31 -0.041877  0.092110 -0.033449  0.005552 -0.012509 -0.040247
156 2022-04-30 -0.069406  0.056699 -0.061061 -0.067101 -0.041513 -0.041992
157 2022-05-31  0.010315  0.052461 -0.008722  0.025143  0.024185  0.015864

          IYR       PSP       QAI       SPY       TIP
0    0.295598  0.229649  0.022329  0.098794 -0.018504
1    0.023198  0.054364  0.028336  0.058924  0.020438
2   -0.025462  0.044850 -0.004035 -0.001254  0.001382
3    0.105826  0.143274  0.015353  0.074633  0.000906
4    0.131503  0.032977 -0.004586  0.036505  0.007979
```

```
..        ...       ...       ...       ...       ...
153 -0.081499 -0.085213 -0.019946 -0.051926 -0.019773
154 -0.045518 -0.073240 -0.006383 -0.029154  0.008919
155  0.069009 -0.007359 -0.002225  0.037953 -0.018393
156 -0.041014 -0.125388 -0.033108 -0.087479 -0.021540
157 -0.033610  0.012375 -0.003283  0.006978 -0.003782

[158 rows x 12 columns]
```

[176]:
```
import statistics
import math
```

[177]:
```
BWX= dataset3["BWX"]
DBC= dataset3["DBC"]
EEM= dataset3["EEM"]
EFA= dataset3["EFA"]
HYG= dataset3["HYG"]
IEF= dataset3["IEF"]
IYR= dataset3["IYR"]
PSP= dataset3["PSP"]
QAI= dataset3["QAI"]
SPY= dataset3["SPY"]
TIP= dataset3["TIP"]
```

[178]:
```
e1=statistics.mean(BWX)*12
e2=statistics.mean(DBC)*12
e3=statistics.mean(EEM)*12
e4=statistics.mean(EFA)*12
e5=statistics.mean(HYG)*12
e6=statistics.mean(IEF)*12
e7=statistics.mean(IYR)*12
e8=statistics.mean(PSP)*12
e9=statistics.mean(QAI)*12
e10=statistics.mean(SPY)*12
e11=statistics.mean(TIP)*12
```

[179]:
```
v1=statistics.stdev(BWX)*math.sqrt(12)
v2=statistics.stdev(DBC)*math.sqrt(12)
v3=statistics.stdev(EEM)*math.sqrt(12)
v4=statistics.stdev(EFA)*math.sqrt(12)
v5=statistics.stdev(HYG)*math.sqrt(12)
v6=statistics.stdev(IEF)*math.sqrt(12)
v7=statistics.stdev(IYR)*math.sqrt(12)
v8=statistics.stdev(PSP)*math.sqrt(12)
v9=statistics.stdev(QAI)*math.sqrt(12)
v10=statistics.stdev(SPY)*math.sqrt(12)
v11=statistics.stdev(TIP)*math.sqrt(12)
```

[180]:
```
sr1=e1/v1
sr2=e2/v2
sr3=e3/v3
sr4=e4/v4
sr5=e5/v5
sr6=e6/v6
sr7=e7/v7
sr8=e8/v8
sr9=e9/v9
sr10=e10/v10
sr11=e11/v11
```

[181]:
```
Q1 = {'Asset Class' : ['BWX', 'DBC', 'EEM', 'EFA', 'HYG','IEF', 'IYR', 'PSP',␣
 ↪'QAI', 'SPY','TIP'],
        'Mean' : [e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11],
        'Volatility' : [v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11],
      'Sharpe Ratio' : [sr1, sr2, sr3, sr4, sr5, sr6, sr7, sr8, sr9, sr10,␣
 ↪sr11]}
df1= pda.DataFrame(Q1)
display(df1)
```

	Asset Class	Mean	Volatility	Sharpe Ratio
0	BWX	0.006390	0.076505	0.083525
1	DBC	0.042225	0.181134	0.233116
2	EEM	0.073746	0.193170	0.381770
3	EFA	0.085370	0.160429	0.532138
4	HYG	0.072104	0.085733	0.841033
5	IEF	0.023507	0.058628	0.400955
6	IYR	0.151888	0.182766	0.831048
7	PSP	0.138939	0.217208	0.639657
8	QAI	0.020549	0.048389	0.424667
9	SPY	0.150734	0.142042	1.061189
10	TIP	0.032002	0.045452	0.704078

[182]:
```
M=max(sr1, sr2, sr3, sr4, sr5, sr6, sr7, sr8, sr9, sr10, sr11)
m=min(sr1, sr2, sr3, sr4, sr5, sr6, sr7, sr8, sr9, sr10, sr11)
print(M)
print(m)
```

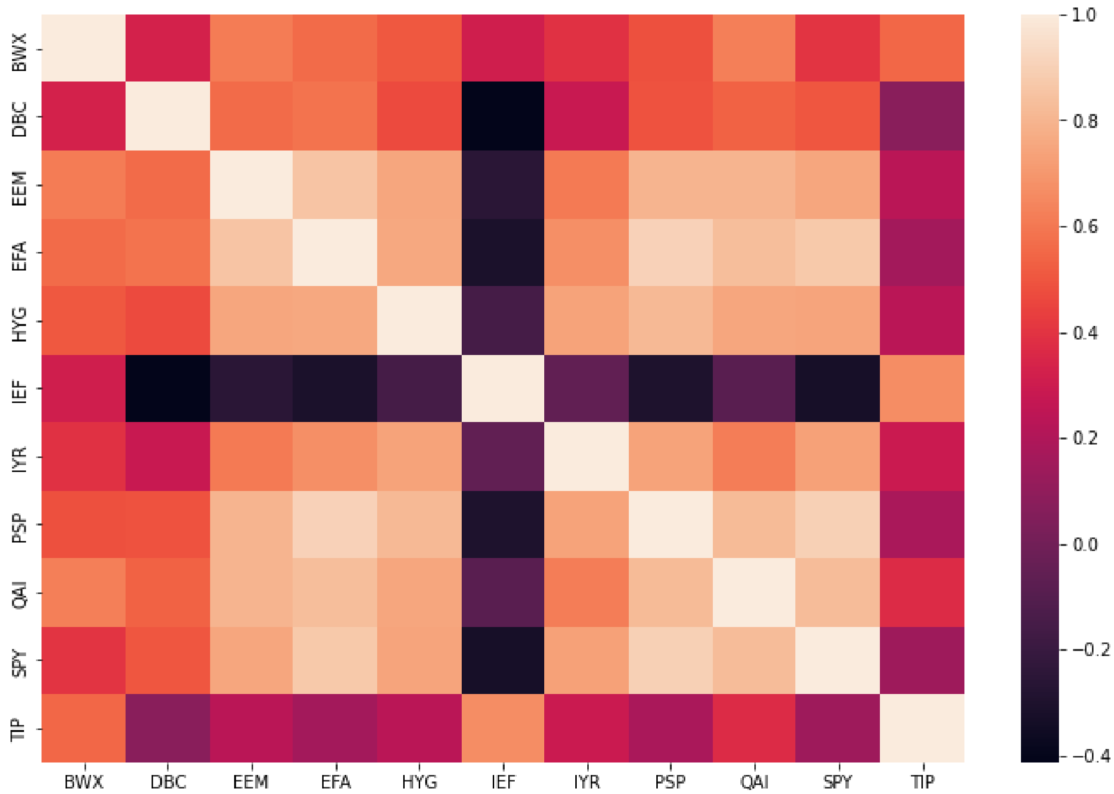
```
1.061188835567402
0.08352526973785378
```

[183]:
```
#SPY has the largest Sharpe ratio and BWX has the smallest.
```

[184]:
```
corr_matrix = dataset3.corr()

print("")
```

```python
print('CORRELATION MATRIX')
display(corr_matrix)
print("")

plot.figure(figsize = (12,8))
sbrn.heatmap(corr_matrix)
plot.show()
```

```
CORRELATION MATRIX
          BWX       DBC       EEM       EFA       HYG       IEF       IYR  \
BWX  1.000000  0.330982  0.609331  0.559939  0.507747  0.315212  0.388627
DBC  0.330982  1.000000  0.560808  0.580985  0.465019 -0.413355  0.285696
EEM  0.609331  0.560808  1.000000  0.851916  0.746547 -0.254467  0.604353
EFA  0.559939  0.580985  0.851916  1.000000  0.756339 -0.310853  0.671100
HYG  0.507747  0.465019  0.746547  0.756339  1.000000 -0.157887  0.738514
IEF  0.315212 -0.413355 -0.254467 -0.310853 -0.157887  1.000000 -0.059095
IYR  0.388627  0.285696  0.604353  0.671100  0.738514 -0.059095  1.000000
PSP  0.486088  0.487789  0.797594  0.905540  0.813852 -0.301569  0.737273
QAI  0.624505  0.535462  0.801072  0.834043  0.750569 -0.085488  0.613673
SPY  0.401463  0.498145  0.746713  0.871234  0.740805 -0.328127  0.730715
TIP  0.551875  0.075545  0.233472  0.160054  0.235147  0.664207  0.291943

          PSP       QAI       SPY       TIP
BWX  0.486088  0.624505  0.401463  0.551875
DBC  0.487789  0.535462  0.498145  0.075545
EEM  0.797594  0.801072  0.746713  0.233472
EFA  0.905540  0.834043  0.871234  0.160054
HYG  0.813852  0.750569  0.740805  0.235147
IEF -0.301569 -0.085488 -0.328127  0.664207
IYR  0.737273  0.613673  0.730715  0.291943
PSP  1.000000  0.821420  0.898894  0.177978
QAI  0.821420  1.000000  0.828297  0.366972
SPY  0.898894  0.828297  1.000000  0.144071
TIP  0.177978  0.366972  0.144071  1.000000
```



[185]:
```
#The strongest negative correlation is DBC and IEF at -0.41.
#The strongest positive correlation between distinct assets is PSP and EFA at 0.
↪91
#TIP has performed fairly well with an SR of 0.70 which is superior to IEF and␣
↪BWX
#which represent domestic and foreign bonds respectively.
#Given the correlation matrix, TIP has very weak correlation with most asset␣
↪classes and a low correlation of
#between 0.55-0.66 with the bonds. This is still lower than the correlation␣
↪between other distinct asssets so
#I agree that HMC is correct in considering TIP a disntinct asset.
```

[186]:
```
# Compute covariance matrix.
cov_mat = dataset3.cov() * 12

# To take the inverse, use the linalg sub-module within numpy
invcov_mat = np.linalg.inv(cov_mat)

# Calculate the return vector
mu = np.array([e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11])
```

```
# Calculate the unit vector, having the same shape as the return vector
unit= np.ones(mu.shape, dtype = 'int')

# Use np.dot to take the dot product of two vectors

scaling = np.dot(np.dot(unit, invcov_mat), mu)

numerator = np.dot(invcov_mat, mu)

tangency_weights = numerator/scaling


Q3 = {'Asset Class' : ['BWX', 'DBC', 'EEM', 'EFA', 'HYG','IEF', 'IYR', 'PSP',␣
 ↪'QAI', 'SPY','TIP'],
        'Tangency Weight' : [-0.97307183,  .19166205,  0.12836907, -0.00723276,␣
 ↪ 1.30831024,  2.15714398,
 -0.3688021,  -0.39909241, -4.33534801,  2.34853023,  0.94953156],}

df3= pda.DataFrame(Q3)
display(df3)
```

	Asset Class	Tangency Weight
0	BWX	-0.973072
1	DBC	0.191662
2	EEM	0.128369
3	EFA	-0.007233
4	HYG	1.308310
5	IEF	2.157144
6	IYR	-0.368802
7	PSP	-0.399092
8	QAI	-4.335348
9	SPY	2.348530
10	TIP	0.949532

[187]:
```
#I found the expected value using the tangency weights
tmu=np.dot(tangency_weights,mu)
```

[243]:
```
#This is a component of the variance calculation
mu2 = np.array([e1**2, e2**2, e3**2, e4**2, e5**2, e6**2, e7**2, e8**2, e9**2,␣
 ↪e10**2, e11**2])

tmu2=np.dot(tangency_weights,mu2)

#Here I used the formula VAR[X]=E[X^2]-E[X]^2
tvar=abs(tmu2-(tmu**2))
tvol=math.sqrt(tvar)
```

```
print(tmu)
print(tvol)
print(tmu/tvol)
```

```
0.33960168224022963
0.26476112954007863
1.2826719799471995
```

[244]:
```
#The tangency mean is 0.34
#The volatility is 0.26
#The Sharpe Ratio is 1.28
```

[245]:
```
#Q4, we find the delta star value
mup=0.01

denominator=np.dot(np.dot(np.transpose(mu),invcov_mat),mu)
delta=np.dot(scaling/denominator,mup)
print(delta)
```

```
0.029446261673480557
```

[246]:
```
opt_weights=np.dot(delta,tangency_weights)
print(opt_weights)
```

```
[-0.02865333  0.00564373  0.00377999 -0.00021298  0.03852485  0.06351983
 -0.01085984 -0.01175178 -0.12765979  0.06915544  0.02796015]
```

[241]:
```
Q4 = {'Asset Class' : ['BWX', 'DBC', 'EEM', 'EFA', 'HYG','IEF', 'IYR', 'PSP',␣
 ↪'QAI', 'SPY','TIP'],
        'Optimal Weight' :[-0.02865333,  0.00564373,  0.00377999, -0.00021298, ␣
 ↪0.03852485,  0.06351983,
 -0.01085984, -0.01175178, -0.12765979,  0.06915544,  0.02796015],}

df4= pda.DataFrame(Q4)
display(df4)
```

	Asset Class	Optimal Weight
0	BWX	-0.028653
1	DBC	0.005644
2	EEM	0.003780
3	EFA	-0.000213
4	HYG	0.038525
5	IEF	0.063520
6	IYR	-0.010860
7	PSP	-0.011752
8	QAI	-0.127660
9	SPY	0.069155
10	TIP	0.027960

[249]:
```
#Here we check if the expected return of our portfolio is as we stated
opt_mu=np.dot(opt_weights,mu)
print(opt_mu)
#Close enough
```

```
0.009999999999999998
```

[250]:
```
opt_var=mup**2/denominator
print(opt_var)
opt_vol=math.sqrt(opt_var)
print (opt_vol)
print(opt_mu/opt_vol)
```

```
2.3690636996292338e-05
0.0048673028461656605
2.0545259491871866
```

[251]:
```
#Thus we have found that the optimal volatility for a portfolio that returns 10␣
↪percent is 4.9 percent volatility.
#The sharpe ratio is an extremely good 2.05
#This portfolio has large long positions in SPY,TIP,IEF and HYG. Meaning a lot␣
↪of bonds both treasury and corporate.
#The protfolio shorts QAI very heavily and also short BWX and PSP.
#Meaning that it shorts hedge fund, foreign bonds and private equity.
#I don't see a connection with Sharpe ratios only that it plays SPY long with␣
↪the highest SR and shorts IYR and QAI
#which also had high SR's
```

[252]:
```
#Compute equal weights as 1/(number of securities)
eqw_vec = [1/11,1/11,1/11,1/11,1/11,1/11,1/11,1/11,1/11,1/11,1/11]
print(eqw_vec)

#To scale the returns of the portfolio,
mean_rescaler = 0.01/np.dot(np.transpose(mu),eqw_vec)
print(mean_rescaler)

#Rescaled weights are simply equal weights multiplied by the rescaling factor
eqw_vec_re = np.dot(eqw_vec,mean_rescaler)

print(eqw_vec_re)
#to check, rescaled weights produce the target portfolio return:
np.dot(np.transpose(mu),eqw_vec_re)
print(np.dot(np.transpose(mu),eqw_vec_re))
```

```
[0.09090909090909091, 0.09090909090909091, 0.09090909090909091,
0.09090909090909091, 0.09090909090909091, 0.09090909090909091,
0.09090909090909091, 0.09090909090909091, 0.09090909090909091,
```

```
0.09090909090909091, 0.09090909090909091]
0.13793901281192278
[0.01253991 0.01253991 0.01253991 0.01253991 0.01253991 0.01253991
 0.01253991 0.01253991 0.01253991 0.01253991 0.01253991]
0.009999999999999998
```

[253]:
```
#The volatility and sharpe ratio are as follows
eq_mu=0.01
print(eq_mu)

eq_var=np.dot(np.dot(np.transpose(eqw_vec_re),cov_mat),eqw_vec_re)
print(eq_var)
eq_vol=math.sqrt(eq_var)
print(eq_vol)

#Sharpe ratio
print(eq_mu/eq_vol)
```

```
0.01
0.00018370946337676133
0.013553946413379437
0.7377924993217291
```

[216]:
```
#risk parity

rp_weight_vec=[1/v1,1/v2,1/v3,1/v4,1/v5,1/v6,1/v7,1/v8,1/v9,1/v10,1/v11]
print(rp_weight_vec)

#rescale

mean_rescaler2 = 0.01/np.dot(np.transpose(mu),rp_weight_vec)
print(mean_rescaler)

#Rescaled weights are simply equal weights multiplied by the rescaling factor
rp_vec_re = np.dot(rp_weight_vec,mean_rescaler2)

print(rp_vec_re)
#to check, rescaled weights produce the target portfolio return:
np.dot(np.transpose(mu),rp_vec_re)
print(np.dot(np.transpose(mu),rp_vec_re))
```

```
[13.071086945619525, 5.520787762967549, 5.176795483206499, 6.233302422029516,
11.664115776896544, 17.056798190572465, 5.471470201848045, 4.603874476909501,
20.66573561578705, 7.040161761959947, 22.001341756356513]
1.3793901281192278
[0.0213121  0.00900152 0.00844064 0.01016325 0.01901807 0.02781071
 0.00892111 0.00750651 0.033695   0.01147882 0.03587268]
0.01
```

[218]:
```
#The volatility and sharpe ratio are as follows
rp_mu=0.01
print(rp_mu)

rp_var=np.dot(np.dot(np.transpose(rp_vec_re),cov_mat),rp_vec_re)
print(eq_var)
rp_vol=math.sqrt(rp_var)
print(rp_vol)

#Sharpe ratio
print(rp_mu/rp_vol)
```

```
0.01
0.018370946337676136
0.012711171359355884
0.7867095578598771
```

[200]:
```
#Both of these weightings underpermorm the optimal portfolio significantly
#Both have sharpe ratios under 1, rather similar at 0.74 and 0.79
#The risk parity has less volatility
```

[201]:
```
#QUESTION 6

dataset3 = dataset3.set_index("Date")
dataset3.index = pda.to_datetime(dataset3.index)
dataset3.head()
```

[201]:
```
                 BWX       DBC       EEM       EFA       HYG       IEF  \
Date
2009-04-30  0.008440 -0.001553  0.155029  0.114637  0.137908 -0.028005
2009-05-31  0.054143  0.163134  0.159871  0.132389  0.029025 -0.020302
2009-06-30  0.004550 -0.026858 -0.023094 -0.014648  0.032918 -0.006170
2009-07-31  0.031311  0.018595  0.110173  0.100442  0.069218  0.008345
2009-08-31  0.007193 -0.040800 -0.013571  0.044596 -0.017404  0.007198

                 IYR       PSP       QAI       SPY       TIP
Date
2009-04-30  0.295598  0.229649  0.022329  0.098794 -0.018504
2009-05-31  0.023198  0.054364  0.028336  0.058924  0.020438
2009-06-30 -0.025462  0.044850 -0.004035 -0.001254  0.001382
2009-07-31  0.105826  0.143274  0.015353  0.074633  0.000906
2009-08-31  0.131503  0.032977 -0.004586  0.036505  0.007979
```

[202]:
```
insample_data = dataset3.loc[:"2020"]
print("Part A Data")
display(insample_data)
```

```python
insample_data1 = dataset3.loc[:'2021']

print("Part B Data")
display(insample_data1)

outsample_data = dataset3.loc["2022"]

print("Part C Data")
display(outsample_data)
```

```
Part A Data

                 BWX       DBC       EEM       EFA       HYG       IEF  \
Date
2009-04-30  0.008440 -0.001553  0.155029  0.114637  0.137908 -0.028005
2009-05-31  0.054143  0.163134  0.159871  0.132389  0.029025 -0.020302
2009-06-30  0.004550 -0.026858 -0.023094 -0.014648  0.032918 -0.006170
2009-07-31  0.031311  0.018595  0.110173  0.100442  0.069218  0.008345
2009-08-31  0.007193 -0.040800 -0.013571  0.044596 -0.017404  0.007198
…                …         …         …         …         …         …
2020-08-31  0.004145  0.046404  0.028911  0.047256 -0.000187 -0.009662
2020-09-30 -0.012164 -0.035396 -0.010049 -0.020414 -0.009262  0.003332
2020-10-31  0.010069 -0.031511  0.013945 -0.035624  0.003916 -0.013925
2020-11-30  0.020691  0.102184  0.090120  0.142902  0.033720  0.003591
2020-12-31  0.023918  0.054320  0.071096  0.049955  0.019380 -0.002583

                 IYR       PSP       QAI       SPY       TIP
Date
2009-04-30  0.295598  0.229649  0.022329  0.098794 -0.018504
2009-05-31  0.023198  0.054364  0.028336  0.058924  0.020438
2009-06-30 -0.025462  0.044850 -0.004035 -0.001254  0.001382
2009-07-31  0.105826  0.143274  0.015353  0.074633  0.000906
2009-08-31  0.131503  0.032977 -0.004586  0.036505  0.007979
…                …         …         …         …         …
2020-08-31  0.002351  0.059905  0.012674  0.069833  0.008935
2020-09-30 -0.023360 -0.030325 -0.008266 -0.037389 -0.003726
2020-10-31 -0.029301 -0.007477 -0.004635 -0.025051 -0.006883
2020-11-30  0.086003  0.162395  0.033595  0.108985  0.012280
2020-12-31  0.024835  0.089630  0.021296  0.036849  0.010777

[141 rows x 11 columns]

Part B Data

                 BWX       DBC       EEM       EFA       HYG       IEF  \
Date
2009-04-30  0.008440 -0.001553  0.155029  0.114637  0.137908 -0.028005
2009-05-31  0.054143  0.163134  0.159871  0.132389  0.029025 -0.020302
```

```
2009-06-30  0.004550 -0.026858 -0.023094 -0.014648  0.032918 -0.006170
2009-07-31  0.031311  0.018595  0.110173  0.100442  0.069218  0.008345
2009-08-31  0.007193 -0.040800 -0.013571  0.044596 -0.017404  0.007198
…                …         …         …         …         …         …
2021-08-31 -0.008870 -0.016229  0.015879  0.014648  0.006268 -0.003763
2021-09-30 -0.023686  0.052138 -0.038733 -0.032614 -0.003701 -0.015971
2021-10-31 -0.004222  0.058250  0.010990  0.032062 -0.002836 -0.004158
2021-11-30 -0.005715 -0.087135 -0.040396 -0.044894 -0.011238  0.011374
2021-12-31 -0.001900  0.066463  0.014799  0.043638  0.022570 -0.005512

                 IYR       PSP       QAI       SPY       TIP
Date
2009-04-30  0.295598  0.229649  0.022329  0.098794 -0.018504
2009-05-31  0.023198  0.054364  0.028336  0.058924  0.020438
2009-06-30 -0.025462  0.044850 -0.004035 -0.001254  0.001382
2009-07-31  0.105826  0.143274  0.015353  0.074633  0.000906
2009-08-31  0.131503  0.032977 -0.004586  0.036505  0.007979
…                …         …         …         …         …
2021-08-31  0.019578  0.014458  0.001425  0.029941 -0.001460
2021-09-30 -0.056269 -0.083231 -0.013358 -0.046605 -0.007801
2021-10-31  0.073033  0.095065  0.010977  0.070435  0.011367
2021-11-30 -0.023673 -0.041620 -0.014500 -0.007582  0.009104
2021-12-31  0.094420  0.019151  0.007894  0.045976  0.003758

[153 rows x 11 columns]
```

```
Part C Data
```

```
                 BWX       DBC       EEM       EFA       HYG       IEF  \
Date
2022-01-31 -0.025361  0.079737  0.000610 -0.035535 -0.025734 -0.020315
2022-02-28 -0.010834  0.065037 -0.042840 -0.033929 -0.008228 -0.002679
2022-03-31 -0.041877  0.092110 -0.033449  0.005552 -0.012509 -0.040247
2022-04-30 -0.069406  0.056699 -0.061061 -0.067101 -0.041513 -0.041992
2022-05-31  0.010315  0.052461 -0.008722  0.025143  0.024185  0.015864

                 IYR       PSP       QAI       SPY       TIP
Date
2022-01-31 -0.081499 -0.085213 -0.019946 -0.051926 -0.019773
2022-02-28 -0.045518 -0.073240 -0.006383 -0.029154  0.008919
2022-03-31  0.069009 -0.007359 -0.002225  0.037953 -0.018393
2022-04-30 -0.041014 -0.125388 -0.033108 -0.087479 -0.021540
2022-05-31 -0.033610  0.012375 -0.003283  0.006978 -0.003782
```

[203]:
```
#Using the same approch from Q4 we do this with data A:
cols=['BWX', 'DBC', 'EEM', 'EFA', 'HYG','IEF', 'IYR', 'PSP', 'QAI', 'SPY','TIP']
mu6a=np.dot(12,np.array(insample_data[cols].mean()))
print(mu6a)
```

```
[ 0.02626422 -0.01329334  0.09766578  0.09495408  0.08295793  0.03664414
  0.15224937  0.15987845  0.02857646  0.15714301  0.03565006]
```

[204]:
```python
vol6a=np.dot(math.sqrt(12),np.array(insample_data1[cols].std()))
print(vol6a)
```

```
[0.07373315 0.1790072  0.19451558 0.16073692 0.08520802 0.05653255
 0.18129611 0.21413999 0.04769314 0.13968741 0.04476654]
```

[205]:
```python
###### Compute covariance matrix.
cov_mat6a = insample_data.cov() *12



# To take the inverse, use the linalg sub-module within numpy
invcov_mat6a = np.linalg.inv(cov_mat6a)

# Calculate the unit vector, having the same shape as the return vector
unit6a= np.ones(mu6a.shape, dtype = 'int')




# Use np.dot to take the dot product of two vectors

scaling6a = np.dot(np.dot(unit6a, invcov_mat6a), mu6a)

numerator6a = np.dot(invcov_mat6a, mu6a)

tangency_weights6a = numerator6a/scaling6a

#Q6a, we find the delta star value
mup6a=0.01

denominator6a=np.dot(np.dot(np.transpose(mu6a),invcov_mat6a),mu6a)
delta6a=np.dot(scaling6a/denominator6a,mup6a)
opt_weights6a=np.dot(delta6a,tangency_weights6a)
print(opt_weights6a)
Q6 = {'Asset Class' : ['BWX', 'DBC', 'EEM', 'EFA', 'HYG','IEF', 'IYR', 'PSP',␣
↪'QAI', 'SPY','TIP'],
        'Optimal Weights 2020' : [-0.03251228, -0.05902893,  0.02261935, -0.
↪06349035,  0.38451792,  0.70399312,
 -0.17417582, -0.02069998, -0.9981324,   0.65452248,  0.04740016],}

df6= pda.DataFrame(Q6)
display(df6)
```

```
[-0.00325123 -0.00590289  0.00226193 -0.00634904  0.03845179  0.07039931
 -0.01741758 -0.00207    -0.09981324  0.06545225  0.00474002]
```

```
   Asset Class  Optimal Weights 2020
0          BWX             -0.032512
1          DBC             -0.059029
2          EEM              0.022619
3          EFA             -0.063490
4          HYG              0.384518
5          IEF              0.703993
6          IYR             -0.174176
7          PSP             -0.020700
8          QAI             -0.998132
9          SPY              0.654522
10         TIP              0.047400
```

[206]:

```python
#Using the means from 2021, 6b)
mu6b=np.dot(opt_weights6a,np.dot(12,np.array(insample_data1[cols].mean())))
print(mu6b)
opt_var6b=mup6a**2/denominator6a
opt_vol6b=math.sqrt(opt_var6b)
opt_sr6b=mu6b/opt_vol6b
print(opt_sr6b)

#Now for part C


mu6c=np.dot(opt_weights6a,np.dot(5,np.array(outsample_data[cols].mean())))
print(mu6c)
opt_var6c=mup6a**2/denominator6a
opt_vol6c=math.sqrt(opt_var6c)
opt_sr6c=mu6c/opt_vol6c
print(opt_sr6c)
```

```
0.009807832380615444
2.24585199171848
-0.008978925984782265
-2.0560443963409862
```

[207]:

```python
#6b) The mean return is 0.0098 and the SR is 2.25
#6c) The mean return is -0.0090 and the SR is -2.06
#Applying the 2020 weights to the data up to 2021 yields a higher sharpe ratio.
#Applying the weights to data from 2022 yields negative values as we entered a
 ↪bear market this year
```