Table of Contents

Part 1

1.1. Preprocessing

1. Calculate the excess return of the 48 industry portfolios

Below are the first 5 rows of the excess returns. Please refer to the code for the full dataframe.

	Month	Mkt-RF	RF	Agric	Food	Soda	Beer	Smoke	Toys	Fun	...	Boxes	Trans	Whlsl	Rtail	Meals	Banks	Insur	RlEst	Fin	Other
0	198601	0.65	0.56	7.36	1.82	-1.76	-1.42	4.99	1.58	3.35	...	1.35	5.04	1.92	1.11	-0.65	2.26	6.07	1.52	1.91	3.35
1	198602	7.13	0.53	13.45	7.36	11.15	7.18	11.67	10.19	7.75	...	10.08	6.50	8.32	6.68	8.74	10.51	6.88	6.46	7.65	10.75
2	198603	4.88	0.60	2.14	7.24	10.29	8.57	10.92	5.05	6.26	...	5.01	2.62	6.81	8.51	5.82	6.62	4.71	8.72	4.89	5.92
3	198604	-1.31	0.52	4.72	-1.09	3.92	-3.44	3.12	3.39	7.84	...	-1.45	-3.16	1.18	-1.22	3.34	-1.10	-6.62	-3.74	-3.81	-5.21
4	198605	4.62	0.49	0.59	8.06	7.46	5.88	9.67	2.67	5.42	...	4.58	0.21	5.03	11.46	4.40	3.71	5.23	-1.70	4.70	5.83

5 rows × 51 columns

1.2 Basic Portfolio Construction and In-sample analysis (Deliverables)

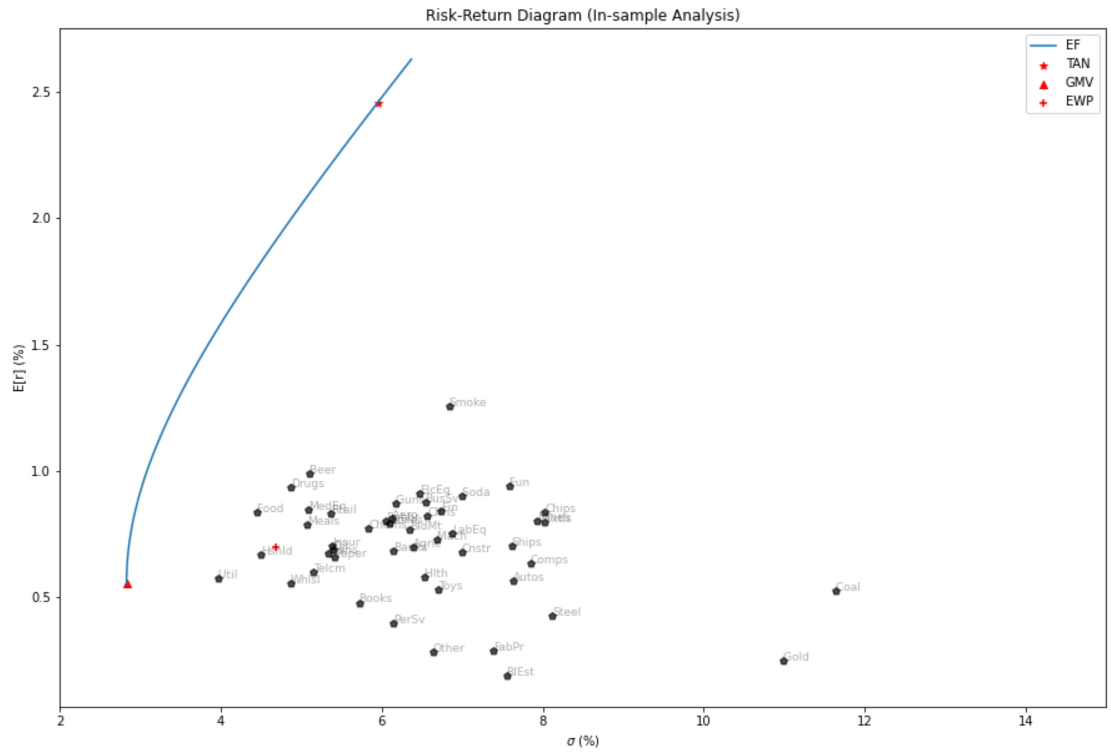
1. In a 3 by 4 table, summarize the three in-sample performance metrics (i.e., expected return, standard deviation, and Sharpe ratio) of the four portfolios (MKT, EWP, TAN, and GMV).

Below is our 3 by 4 summary table of in-sample performance metrics:

	Expected Return	Standard Deviation	Sharpe Ratio
EWP	0.699881	4.689313	0.149250
TAN	2.456390	5.950674	0.412792
GMV	0.555440	2.829671	0.196291
MKT	0.636361	4.484527	0.141902

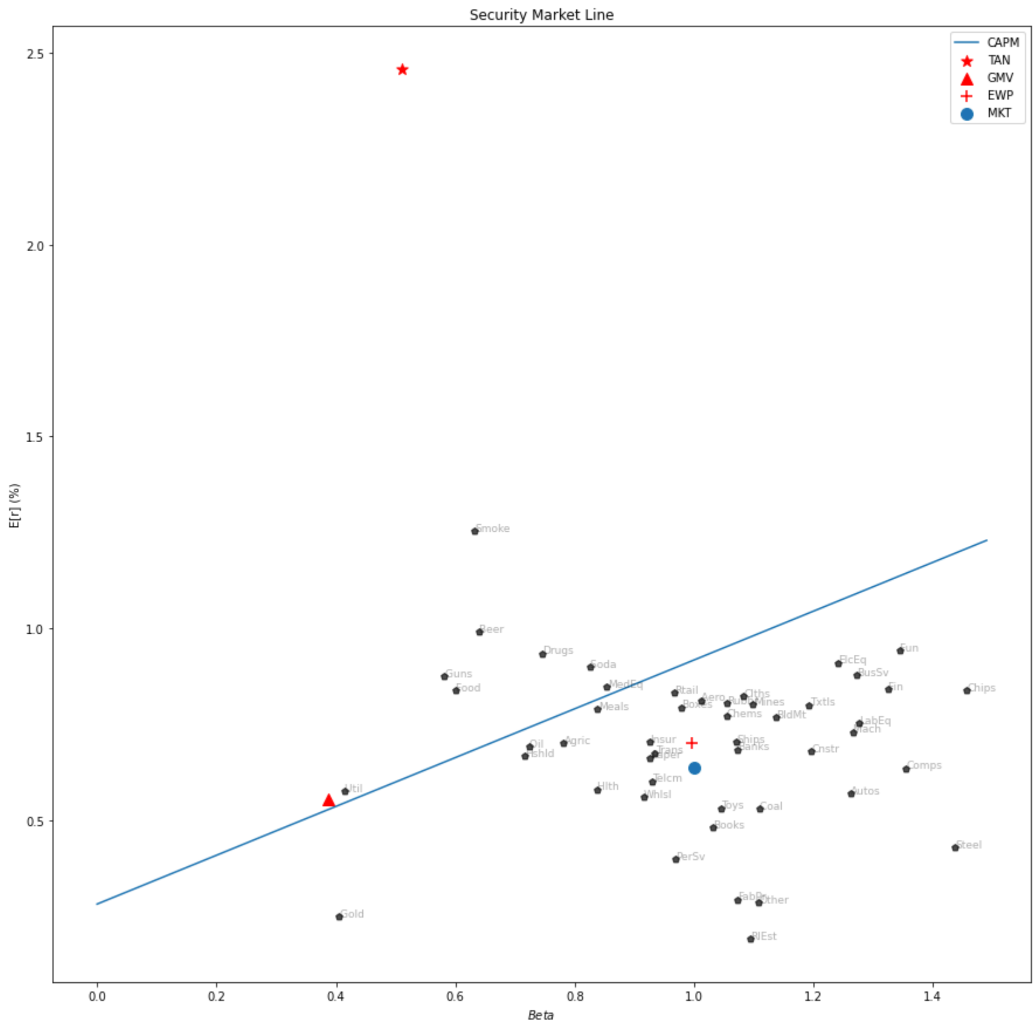
2. Plot the following two graphs related to in-sample analysis:

(a) The σ vs. E[r] diagram. This diagram includes all of the 48 industry portfolios, the special portfolios (i.e., EWP, TAN, and GMV) constructed from the 30-year data, the market portfolio (MKT), and the in-sample efficient frontier.



The efficient frontier sits on the left of all the assets, with its GMV at the bottom of the curve, having expected returns of 0.55% with a 2.83 standard deviation. The steepest line is at the top of the efficient frontier, meaning that for any given level of **σ**, the portfolio on the line will have greater returns than the efficient frontier. Therefore, the tangency is at the top of the efficient frontier, with the expected returns of 2.45% and 5.95 standard deviation. The tangency portfolio dominates the returns for the equally weighted portfolio and the expected returns for each asset. The equally weighted portfolio has returns of 0.69%. This indicates that an optimum combination of a risk free asset and assets located along the efficient frontier would give better returns.

(b) The β vs. E[r] diagram. This diagram includes the same set of portfolios (the 48 industry portfolios treated as risky assets, MKT, EWP, TAN, and GMV) and a straight line that represents the relationship between β and E[r] implied by CAPM (also known as the security market line).



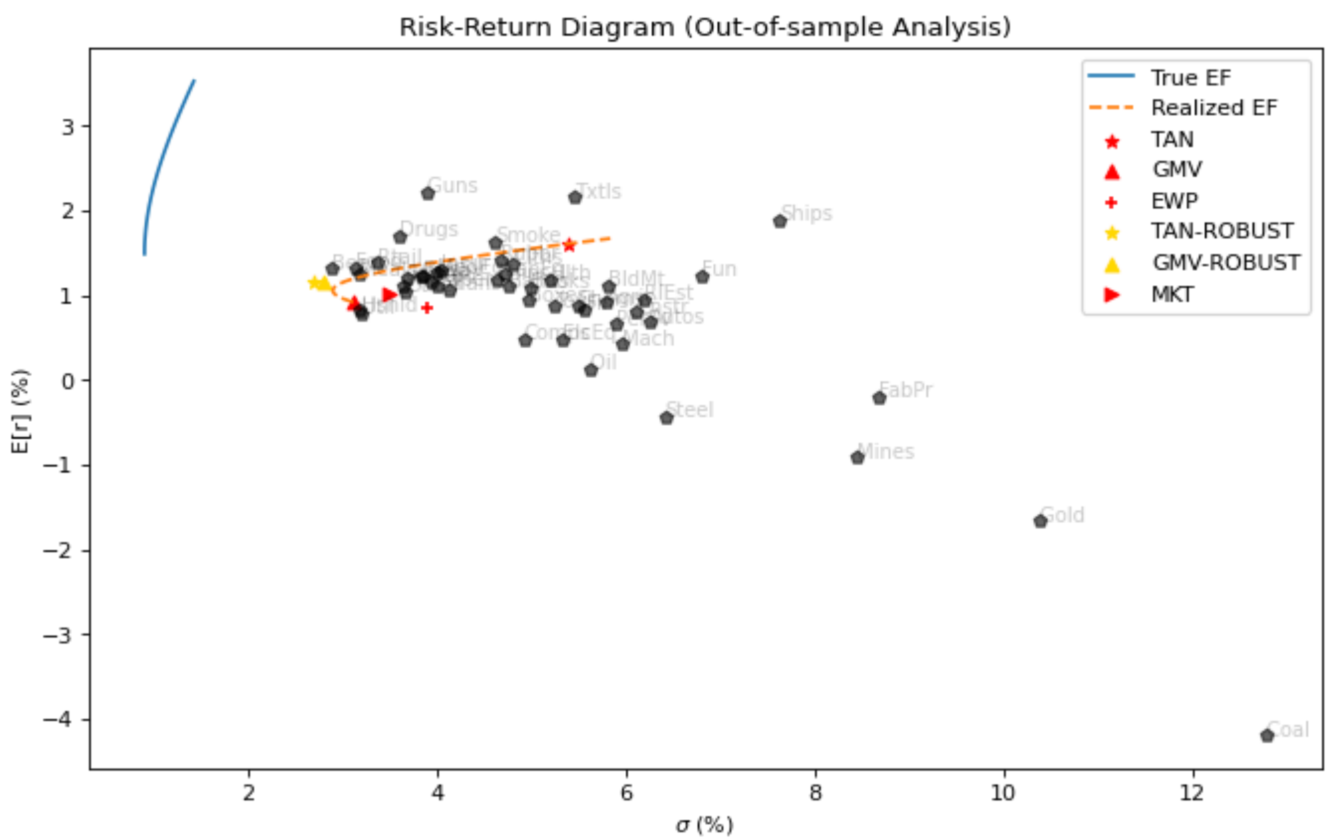
From the graph, there are 2 points of view that can be adopted. Firstly, if we were to assume that CAPM holds true, the SML can be used to explain the value of assets above or below the line. For instance, for assets below the SML, it indicates that they are overvalued because their expected returns are not as high as the required returns as indicated by the SML. Likewise, for assets above the SML, it indicates that the assets are undervalued because their expected returns are more than the required returns.

GMV, with the lowest total risk, has a beta lower than all the other assets while the tangency portfolio is located high above the SML due to higher expected returns can be received with an optimum combination of risk free and portfolio assets. The EWP and MKT portfolios are located within the cluster of other assets.

However, another point of view would be that the graph shows the limitation of CAPM in explaining the relationship between expected returns and beta because the points (assets) are scattered across the graph and are not located along a straight line. This could also indicate that CAPM does not hold true in determining the value of an asset based on their beta-expected returns. As mentioned in prof's lectures, other stock characteristics like size, book-to-market and profitability can explain expected returns better than beta. This would pave the way to more advanced models such as the Fama-French 3 factor model.

1.3 Robust Portfolio Construction and out of sample analysis

1. The out-of-sample σ vs. E[r] diagram. This diagram includes all of the 48 industry portfolios (treated as risky assets), MKT, and the special portfolios constructed from training data (EWP, TAN, GMV, TAN-robust, and GMV-robust). All of them are evaluated on the test data. Also, please include the “true" and “realized" EF curves.



2. Compare the out-of-sample performances of MKT, EWP, TAN, TAN-robust, GMV, and GMV- robust using the definitions and methods introduced above. Report your results in a 3 by 6 table. (Recall that we have 3 performance metrics and 6 portfolios to compare).

```
Out-of-Sample Performances
```

	Portfolio	Excess Returns	Standard Deviation	Sharpe Ratio
0	MKT	1.010833	3.502378	0.288613
1	TAN	1.606994	5.395611	0.297834
2	GMV	0.917728	3.111275	0.294968
3	EWP	0.845597	3.891562	0.217290
4	TAN-ROBUST	1.159521	2.702104	0.429118
5	GMW-ROBUST	1.156145	2.798023	0.413201

	Expected Return	Standard Deviation	Sharpe Ratio
EWP	0.699881	4.689313	0.149250
TAN	2.456390	5.950674	0.412792
GMV	0.555440	2.829671	0.196291
MKT	0.636361	4.484527	0.141902

3. Contrast the table for out-of-sample performance with that for in-sample performance. What are your findings and insights? Also, are there any limitations to the current evaluation methodology?

In general, the out-of-sample performance seems to be better than the in-sample performance across the various portfolios. Namely, the excess returns and sharpe ratio across all portfolios with the exception of the out-of-sample tangency portfolios are higher in the out-of-sample data. Furthermore, with the use of Robust GMV and Robust Tangency portfolios, it has a lower standard deviation than the GMV, with better excess returns. Therefore, this puts it closer to the True Efficient Frontier and to the left of the Realized Efficient Frontier.

Being closer to the true Efficient Frontier is good because it shows the effect of shrinkage estimators reducing the sample risk from the training data. As such, it highlights the importance of having robust estimation for out-of-sample analysis.

Our second finding is that the tangency portfolio (based on the sharpe ratio) is better in the sample analysis as compared to the out of sample analysis. The tangency portfolio in the out-of-sample is almost the same as the GMV, with sharpe ratio of 0.2978, just above 0.2949. This means that the model trained in the in-sample does not perform as well as the out-of-sample. However, when looking at the Tan-robust, its sharpe ratio of 0.429 is much higher than the other portfolios in the out-of-sample analysis. This shows that there is sample risk on training the model, but with the use of the shrinkage estimator, the Tan-robust reduces sample risk in training data and increases performance for out-of-sample analysis.

There are also 2 limitations observed by our team. Firstly, the training data assumes no seasonality. Instead, it considers all data points with uniform importance. Namely, the group observed that the out-of-sample analysis is to perform better than the in-sample analysis. This could be due to the difference in data used for both analyses. The dataset contains returns for assets across 29 years from 1986 to 2015. However, the training data corresponds to the first 25 years of data from 1986 to 2010 while the test data corresponds to the last 5 years between

2011 and 2016. With the difference in time duration, it is important to consider the seasonality of the data sets.

Similarly, there seems to be a difference in the extent of tail risks between the training and test datasets. This is because not only is there a difference in time duration, but there is also a different set of events happening over the last 25 years in the training data. For instance, over the period between 1986 to 2010, there was the 1980s debt crisis in Latin America, 1997 Asian Financial Crisis, 2001 Dot Com bubble (and subsequent crash) and the 2008 Global Financial Crisis. However, in the test data which is between 2011 to 2015, the global economy is on its recovery phase from the 2008 Financial Crisis, which is different from the extent of crises in the training data. While one might refer to the 2013 European Sovereign Debt Crisis, it is largely localized within Europe and is a smaller tail risk as compared to the 2008 crisis. As such, with our training data potentially having a higher tail risk than that of the test data, the subpar performance of the base tangency portfolio on our test data might reflect the inherent sample risk in both our training and test datasets. This once again showcases the importance of robust parameter estimation in reducing sample risks in training our model.

Part 2

2.1. Introduction and Objective

The clear goal of this Data Challenge is to aim to produce the optimal weightage of investment mix to maximize the capital return over a five year time horizon. Given the amount of data available, it is essential to build an efficient portfolio by leveraging existing financial models which have proven to predict lower loss rates than others.

From 1.2(b), we can see that CAPM is not very accurate in estimating the expected returns for the 48 industry portfolios and the market portfolio as very few portfolios fall on the security market line despite the 30 year timeframe.

Therefore, we have decided to use the 3 Factor model to improve the estimation of the expected returns. It is because the 3 factor model considers other important factors besides market excess returns, namely the effect of size and value factors on the returns of individual sectors.

2.2. Three Factor Model

Risk factors have been a key ingredient to quantitative models since the CAPM explained the expected returns of all N assets, using their respective exposure B to a single factor, excess return of the overall market over the risk free rate. Subsequently, Kenneth French and Eugene Fama (who won the 2013 Nobel Prize) identified additional risk factors that depend on a firm’s characteristics which are widely used today [1]. In particular, this is known as the Three-Factor model which adds the relative size and value of firms to the single CAPM source of risk.

The model assumes that idiosyncratic risk is completely uncorrelated between different assets. In other words, a company's firm specific risk does not affect the market environment. The model aims to take into consideration a greater variety of macro factors which could potentially impact the variability and comovement of different assets it is considering investing in.

$$r_i - r_f = \alpha_i + \beta_{i,1}\ F_1 + \cdots + \beta_{i,K}\ F_K + \varepsilon_i$$

Formula to calculate the expected return according to the 3 factor model

The model finds the expected return by multiplying the beta of an asset to a given factor´s covariance to another factor. The three factor model is instead of only considering the effect of one factor such as the CAPM with the market excess return on the variability and covariance between assets. As a result, the model adds realistic market mechanisms into the evaluation as in reality factors such as Small versus Big (Size of companies) and High Minus Low (Value vs Growth).

Nevertheless, given the theoretical approach in this project, we decided to use the Fama-French 3-factor model. This model uses:

- Excess return on the market
- Zero-cost portfolio long in small-cap and short in large-cap stocks (SMB)
- Book-to-market portfolio.[2]

Given that there is no arbitrage, we consider alpha to be zero. Also, considering the amount of assets and diversification opportunities present, we assume epsilon to be zero as well[3].

2.3. Parameters according to the 3 Factor Model

The 3-factor Model requires us to evaluate the proposed parameters by the model. One of them is the expected return based on the betas between the assets and respective factors as well as the covariance matrix, which follows a similar logic. The following will explain step by step how we were able to build optimal parameters for our final weights recommendation.

2.3.1. Preprocessing

To be able to start with constructing the optimal portfolio using the 3-factor model, we split the data set into factors and assets, where we further divide into train (1986 - 2010) and test (2011-2015) data sets. We do this to be able to fit our model to the training data and evaluate the accuracy of the realized portfolios on the test data.

---

[1] https://www.chicagobooth.edu/faculty/directory/f/eugene-f-fama#:~:text=Biography-,Eugene%20F.,particularly%20the%20efficient%20markets%20hypothesis.
[2] https://www.investopedia.com/terms/f/famaandfrenchthreefactormodel.asp
[3] https://finance.yahoo.com/news/does-fama-french-3-factor-182045163.html

2.3.2. Betas

```
def get_betas(assets, F):

  dic = {}

  for i in assets.columns:
    curr_asset = assets[i].values
    factors = F.values

    dic[i] = {}
    lr = LinearRegression(fit_intercept=False) # set to false as the 3 factor assumes zero alpha and zero epsilon
    results = lr.fit(np.array(factors), np.array(curr_asset).reshape(-1,1))
    beta = results.coef_
    dic[i] = beta[0]

  return dic
```

Function to calculate betas given asset against factors using Linear Regression

Then we calculated the betas of each asset in the training data with each factor by using a linear regression. Here we iterate through each asset and find the coefficient of the asset to each factor which hence acts as the respective betas.

$$\beta_p = \frac{Cov(r_p, r_b)}{Var(r_b)}$$

Alternative Method to calculate beta

We could also use the method based on the formula above. However, by using the linear regression we move away from model risk towards more sample risk, which we believe gives us a more realistic evaluation, based on the randomness present in the market which could invalidate the static models present.

2.3.3. Expected Return according to 3 Factor Model

```
def estimate_mu(data, betas):

    ## The data entry is to find the mean Rf, mean Mkt-Rf, mean SMB and mean HML
    ## The betas are for the individual weights for each factor, for each individual industry groups

    mean_rf = data.loc[:,'RF'].mean()
    mean_mkt_er = data.loc[:, 'Mkt-RF'].mean()

    mu_factor = mean_rf + betas["Mkt-RF"] * mean_mkt_er  + betas["SMB"] * data["SMB"].mean() + betas["HML"] * data["HML"

    return mu_factor
```

Function to estimate expected return according to the 3 Factor Model

Using the received betas we are able to calculate the expected return according to the formula proposed by the 3 factor model. As it can be seen, different to a One-Factor model, this model allows three factors to determine the expected return, and adds reliability to the evaluation by the model.

2.3.4. Covariance Matrix according to K Factor Model

Finally we need one more parameter to calculate the realized tangency and global minimum variance portfolio - the covariance matrix. Here we again follow the formula proposed by the 3 Factor model.

$$cov(r_i, r_j) = \sum_{k=1}^{K} \sum_{l=1}^{K} \beta_{k,i}\, \beta_{l,j}\, cov(F_k, F_l)$$

Formula to calculate covariance matrix according to 3-factor Model

```
# Find the different i and j combinations
n = len(betas)

combs = list(itertools.permutations(np.arange(0, n), 2))
for i in range(n):
    combs.append((i, i))  # to add the combinations where i and j are the same
```

Itertools used to receive all combinations of i and j

To be able to get all combinations of i and j of assets, we used itertools, as it gives us the ability to extract all possible permutations of size n with a size of 2.

```
for com in combs:

    i = com[0]
    j = com[1]

    assets_data = betas.iloc[[i,j]]
    count = 0

    total_sum = 0
    for k in range(3):
        for l in range(3):

            overall += 1
            count += 1
            addition = assets_data.iloc[0, k] * assets_data.iloc[1, l] * F_cov.iloc[k,l]

            total_sum += addition

            FF_cov.iloc[i,j] = total_sum
```

Programmatic solution for finding the covariance matrix according to 3 factor model

Then we use the received combinations to iterate through the sample covariance matrix. Using the covariance formula given by the 3 factor model we multiply the respective betas with the factor covariance to receive the value for the complementary indices in the final covariance matrix.

2.4. Results

2.4.1 Enforcing Numerical Stability

After creating the covariance matrix, we checked its numerical stability for it to be implemented using methods involving linear algebra (where the inverse of the covariance matrix was used to create the tangency portfolio). As such, we introduced the following checks on the matrix to ensure that the numbers are positive.

```
[ ] def is_pos_def(x):
        return np.all(np.linalg.eigvals(x) > 0)

    is_pos_def(FF_cov)

    False
```

```
[ ] # See the property of the covariance matrix
    min( np.linalg.eigvals(FF_cov) )

    (-4.5692562556729753e-14+0j)
```

Function to check for numerical stability of the covariance matrix

Based on the above, we note that not all values are > 0 and the minimum appears to be near 0 which could be an indication of machine error in computing zero.

To resolve these issues, we adopt 2 methods to adjust our covariance matrix.

- Direct intervention through the introduction of epsilon to nudge the minimum eigenvalue in the positive range
- Introduction of Shrinkage constant with sample covariance where the final covariance matrix = lambda * FF_cov + (1-lambda) * sample_cov

2.4.2 Direct Intervention Method

```
# For numerical stability

epsilon = 0.00001
FF_cov2 = FF_cov + epsilon * np.identity(48)
FF_cov2
```

Direct Intervention Method

We introduced an identity matrix with the same size as the covariance matrix of 48 rows and columns. This creates a square matrix with 1 on the main diagonal and 0s elsewhere.

Multiplying this with the epsilon and adding it back to the covariance matrix will marginally increase the main diagonal.
After which, we checked the covariance matrix once again for numerical stability and all numbers are now > 0.

```
[ ]  def is_pos_def(x):
         return np.all(np.linalg.eigvals(x) > 0)

     is_pos_def(FF_cov2)

     True
```

```
[ ]  # See the property of the covariance matrix
     min( np.linalg.eigvals(FF_cov2) )

     (9.999999956640345e-06+0j)
```

Function to check for numerical stability of the covariance matrix

Therefore, we can now create the tangency and GMV portfolios using this new covariance matrix and obtain the respective weights of the various assets.

GMV Portfolio

```
[ ] # Find the weights for the GMV portfolio
    V = FF_cov2
    w_g = gmv(V)

    # Prepare the parameters for portfolio evaluation
    w = w_g
    mu = mu

    print("Weights for the GMV:", w_g)
    evaluate_portfolio_performance_on_data(w, mu, V, rf = 0)
```

```
Weights for the GMV: [ 0.18164204  0.12897399  0.02149461  0.10736233  0.10498022  0.03337833
 -0.11804785 -0.01230371  0.10907897 -0.05332379  0.10172551  0.12719163
  0.09959637 -0.04783883  0.03705802 -0.07674051 -0.07020138 -0.05488458
 -0.10835052  0.09933265 -0.0480077  -0.12273716 -0.15077002 -0.06330543
 -0.02123706  0.16322744  0.3746005  -0.00185342  0.02007328  0.11840476
  0.21548899  0.02170537  0.08211301 -0.00276967 -0.00087651 -0.05031307
  0.02648834  0.00293552  0.00937739  0.01511283  0.09716083  0.02316356
  0.05388644 -0.15309052 -0.06954553  0.04888515 -0.13529636 -0.06294448]
{'Er': 0.33686683674663437,
 'Sharpe': 146.7269909592765,
 'sigma': 0.002295875043468522,
 'var': 5.2710422152215876e-06}
```

Tangency Portfolio

```
[ ] # Find the weights for the Tangency portfolio
    V = FF_cov2
    w_t = tangency(mu, V)

    # Prepare the parameters for portfolio evaluation
    w = w_t
    mu = mu

    print("Weights for the Tangency Portfolio:", w_t)
    evaluate_portfolio_performance_on_data(w, mu, V, rf = 0)
```

```
Weights for the Tangency Portfolio: [ 0.18164195  0.12897393  0.02149463  0.10736225  0.10498018  0.03337834
 -0.11804777 -0.01230369  0.10907891 -0.05332374  0.1017255   0.12719155
  0.09959627 -0.04783879  0.03705803 -0.07674038 -0.07020131 -0.05488451
 -0.10835047  0.09933263 -0.04800767 -0.12273712 -0.1507699  -0.06330538
 -0.02123699  0.16322741  0.37460033 -0.00185338  0.02007329  0.11840471
  0.21548891  0.02170534  0.08211297 -0.00276975 -0.00087658 -0.05031311
  0.02648829  0.00293554  0.00937739  0.01511284  0.09716079  0.02316354
  0.05388642 -0.1530904  -0.06954547  0.04888522 -0.13529629 -0.06294445]
{'Er': 0.3368673492958216,
 'Sharpe': 146.72710255708805,
 'sigma': 0.0022958767904842563,
 'var': 5.27105023708429e-06}
```

In this case, we note that both portfolios actually share very similar portfolio performance metrics and weights allocated to the assets. This observation could be caused by the extremely low variance from the GMV and the tangency portfolio which is nearly 'risk-free'.

Therefore, with the expected returns almost on the y-axis of the risk-return diagram, it makes sense for the sharpe ratio to be extremely high of exceedingly 100 (potentially infinity). Given that the tangency portfolio seeks for the portfolio with the highest sharpe ratio, the GMV with near infinity sharpe ratio would also be the tangency portfolio.

However, while we could recommend this portfolio for our client, we should also seek caution to this method as a risk-free monthly excess returns of 0.28% (which translates into annual effective excess risk free returns of 3.4%) sounds unrealistic. This suggests that there might lie with rather high model risks through the usage of the 3-factor model to estimate the parameters.

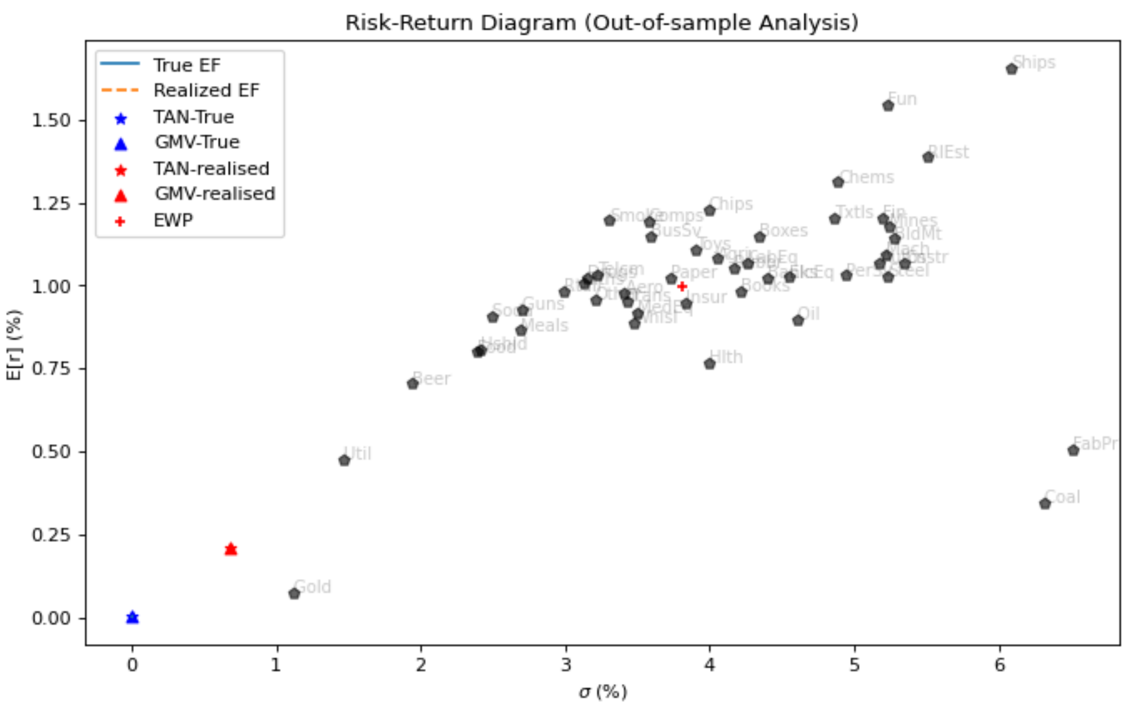
Therefore, we performed an out-of-sample analysis for this model.

2.4.2.1 Out of Sample Analysis on the model (Direct Intervention Method)

Similar to the previous section, we separated the dataset into:

- Training data : 1986 - 2010
- Test data : 2011 - 2015

We then generated the covariance matrix and mean of both datasets using the aforementioned method, creating numerically stable covariance matrices. We then plotted the relevant portfolios and efficient frontiers, and went to find the difference in sharpe ratios between the in-sample and out-sample tangency portfolios.
Thereby, obtaining the following:



```
******
The tangency portfolio constructed from the testing data is:  [ 0.006  0.11   0.104  0.147  0.002 -0.012 -0.126  0.014  0.098  0.051
  0.085  0.07   0.061 -0.085  0.018 -0.034 -0.042 -0.022 -0.037  0.115
 -0.037  0.001 -0.03   0.039 -0.183  0.072  0.313 -0.074  0.113 -0.003
  0.203  0.023 -0.017 -0.002 -0.02  -0.042  0.016  0.019 -0.029  0.04
  0.058  0.061  0.095 -0.015  0.03  -0.103 -0.078  0.03 ]
The global minimum variance portfolio constructed from the testing data  is:  [ 0.006  0.11   0.104  0.147  0.002 -0.012 -0.126  0.014  0.098  0.051
  0.085  0.07   0.061 -0.085  0.018 -0.034 -0.042 -0.022 -0.037  0.115
 -0.037  0.001 -0.03   0.039 -0.183  0.072  0.313 -0.074  0.113 -0.003
  0.203  0.023 -0.017 -0.002 -0.02  -0.042  0.016  0.019 -0.029  0.04
  0.058  0.061  0.095 -0.015  0.03  -0.103 -0.078  0.03 ]
******
The tangency portfolio constructed from the training data is:  [ 0.182  0.129  0.021  0.107  0.105  0.033 -0.118 -0.012  0.109 -0.053
  0.102  0.127  0.1   -0.048  0.037 -0.077 -0.07  -0.055 -0.108  0.099
 -0.048 -0.123 -0.151 -0.063 -0.021  0.163  0.375 -0.002  0.02   0.118
  0.215  0.022  0.082 -0.003 -0.001 -0.05   0.026  0.003  0.009  0.015
  0.097  0.023  0.054 -0.153 -0.07   0.049 -0.135 -0.063]
The global minimum variance portfolio constructed from the training data  is:  [ 0.182  0.129  0.021  0.107  0.105  0.033 -0.118 -0.012  0.109 -0.053
  0.102  0.127  0.1   -0.048  0.037 -0.077 -0.07  -0.055 -0.108  0.099
 -0.048 -0.123 -0.151 -0.063 -0.021  0.163  0.375 -0.002  0.02   0.118
  0.215  0.022  0.082 -0.003 -0.001 -0.05   0.026  0.003  0.009  0.015
  0.097  0.023  0.054 -0.153 -0.07   0.049 -0.135 -0.063]
******
Realised portfolio sigma:  0.6869126294137696
Realised portfolio Er:  0.20867161798193823
Realised portfolio Sharpe:  0.3037818916796222
True portfolio sigma:  0.002015641596315066
True portfolio Er:  0.0020935986239524196
True portfolio Sharpe:  1.03867603634489
Difference in Sharpe:  0.7348941446652677
```

From the out-of-sample analysis, we can see that both the "true" and the "realised" tangency and GMV portfolios are at the same point respectively. Furthermore, we can also observe that the difference in sharpe ratio between both realised and true portfolios is high as well. It is 0.735 which is nearly 70% of the sharpe ratio of that of the true portfolio.

We used sharpe ratio as the benchmark of portfolio performance because:

- Sharpe ratio incorporates both ER and sigma, hence is more comprehensive as a benchmark.
- The objective of this challenge is to seek the portfolio that gives the highest sharpe ratio, therefore the model that is most robust in maximizing sharpe ratio in out-of-sample datasets would best achieve our objective.

This suggests that the model’s prediction power for out-of-sample datasets is low. Given that we would be investing money into portfolios that are subjected to future price movements, we are essentially subjecting our portfolio allocations to out-of-sample datasets. Henceforth, we require a more robust model that would have stronger out-of-sample performance.

2.4.3 Introduction of Shrinkage Constant

Since the 1st model of an epsilon adjustment has poor out-of-sample performance, we adopt another method to estimate the covariance matrix which is the usage of "shrinkage" estimator.

The "shrinkage" estimator concept was first proposed by Ledoit and Wolf[4] in their seminal paper published in 2003. We modified it to suit our parameter estimation to incorporate the sample covariance (from assets) to provide our final covariance matrix with more numerical stability. It has the following expression:

---

[4] http://www.ledoit.net/honey.pdf

Covariance = lambda * Covariance_implied_by_FF + (1-lambda) * Sample Covariance

Similar to the previous method for obtaining the covariance matrix, we inserted an additional line to inculcate the formula

```
FF_cov2 = lamb * FF_cov + (1 - lamb) * assets.cov()

return FF_cov2
```

Snippet of the modified covariance matrix function

To test the method, we ran the model on a sample lambda of 0.5. Therefore, we can now create the tangency and GMV portfolios using this new covariance matrix and obtain the respective weights of the various assets.

GMV Portfolio

```
[ ] # Find the weights for the GMV portfolio using training data ("df_FF")
    V = estimate_cov_shrinkage(df_FF, 0.5)
    mu = estimate_mu_general(df_FF).values
    w_g = gmv(V)

    # Prepare the parameters for portfolio evaluation
    w = w_g

    print("Weights for the GMV:", w_g)
    evaluate_portfolio_performance_on_data(w, mu, V, rf = 0)

    OVERALL NUMBER OF ITERATIONS 20736
    Weights for the GMV: [ 0.10218907  0.0448029   0.00677071 -0.00255064 -0.03902967 -0.00519058
     -0.05655058  0.12266085  0.25866408 -0.08506313 -0.05392484  0.09998972
      0.01203524 -0.24368887 -0.09517246 -0.0131883  -0.1203047  -0.18853963
     -0.08460881  0.09032646 -0.03785442 -0.18603927 -0.10404071 -0.17109589
     -0.01714403  0.12281099  0.07124089 -0.00240461 -0.01927682  0.166024
      0.49174235  0.09867213 -0.03314477 -0.14242445  0.11705834  0.07654161
     -0.03556775  0.19786986  0.13748309  0.30371456  0.04659028  0.21575046
      0.08490748 -0.14908383 -0.09005902  0.12698887  0.00597665 -0.02486284]
    {'Er': 0.48082651689833183,
     'Sharpe': 0.21515270040879972,
     'sigma': 2.2348151614399447,
     'var': 4.994398805801847}
```

Tangency Portfolio

```
[ ] # Find the weights for the Tangency portfolio
    V = estimate_cov_shrinkage(df_FF, 0.5)
    mu = estimate_mu_general(df_FF).values
    w_t = tangency(mu, V)

    # Prepare the parameters for portfolio evaluation
    w = w_t

    print("Weights for the Tangency Portfolio:", w_t)
    evaluate_portfolio_performance_on_data(w, mu, V, rf = 0)
```

```
OVERALL NUMBER OF ITERATIONS 20736
Weights for the Tangency Portfolio: [ 0.07432985  0.04342623  0.01131064 -0.05302578 -0.00673857  0.00372218
 -0.01561805  0.05324612  0.19411116 -0.10526293 -0.03272917  0.07323855
 -0.04345103 -0.18739706 -0.05126859  0.0456586  -0.07165072 -0.09650709
 -0.05084977  0.0520292  -0.04710684 -0.14310519 -0.05837716 -0.11348858
 -0.00241621  0.10047091  0.05481265  0.01588022 -0.01585877  0.13391274
  0.40979377  0.07902571 -0.02294022 -0.1884834   0.07656881  0.04607158
 -0.01165872  0.15327377  0.08313402  0.26114731  0.03641613  0.16646984
  0.03639716  0.02186241 -0.05524657  0.11564967  0.06372829 -0.03250712]
{'Er': 0.7381336089679815,
 'Sharpe': 0.2665754102631052,
 'sigma': 2.7689485997206447,
 'var': 7.667076347894919}
```

The training set gives us a sharpe ratio of 0.267 with a more reasonable variance of 7.667 as compared to the near-zero variance from the direct intervention method. Moving forward, we need to conduct out-of-sample analysis to find the right model hyperparameters, namely the optimal lambda.

2.4.3.1 Out of Sample Analysis on the model (Shrinkage Constant Method)

To determine the optimal lambda, we ran multiple out-of-sample analyses with both the 'realized' and 'true' portfolios over a range of lambdas. More specifically, we decided to use the difference of the sharpe ratio between the 'realized' portfolio and the 'true' portfolio as the key benchmark. As such, we would choose the lambda that gives us the minimum sharpe ratio difference.

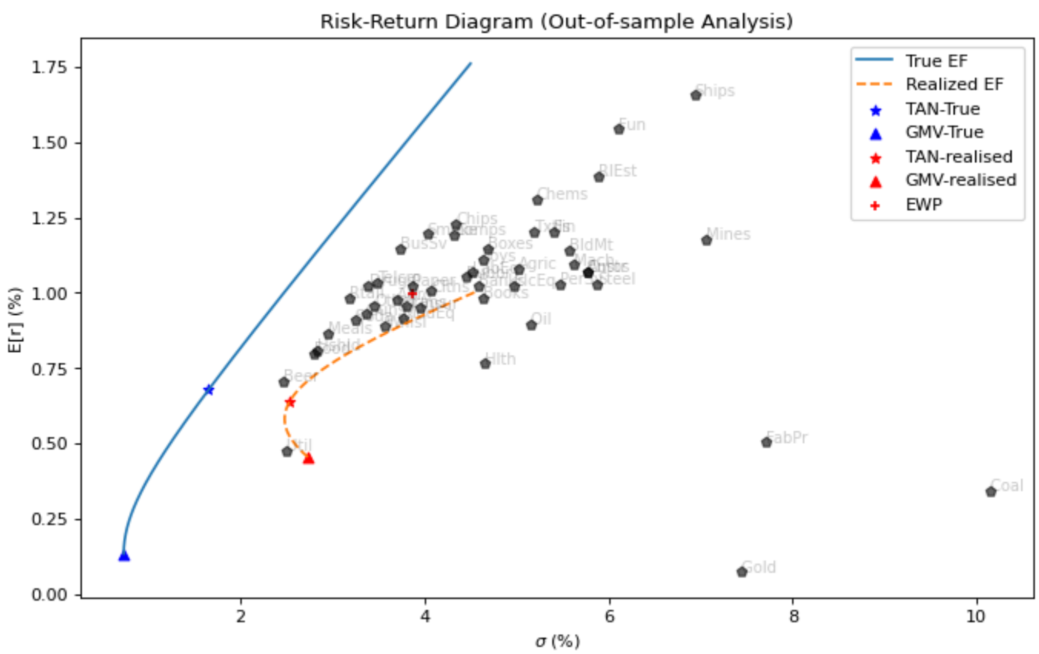
The reason is that we wish to maximize our model's prediction power (on maximum sharpe ratios) in out-of-sample datasets. Hence, the model with the minimal sharpe ratio difference would reflect the more generalized nature of the model in accounting new data.
In particular, we picked Sharpe ratio as the benchmark instead of ER or Sigma due to the following reasons:

- Sharpe ratio incorporates both ER and sigma, hence is more comprehensive as a benchmark.
- The objective of this challenge is to seek the portfolio that gives the highest sharpe ratio, therefore the model that is most robust in maximizing sharpe ratio in out-of-sample datasets would best achieve our objective.

As such, we ran experiments where lambda = 0.5, 0.3, 0.1, 0.75, 0.8, 0.85 and 0.9.

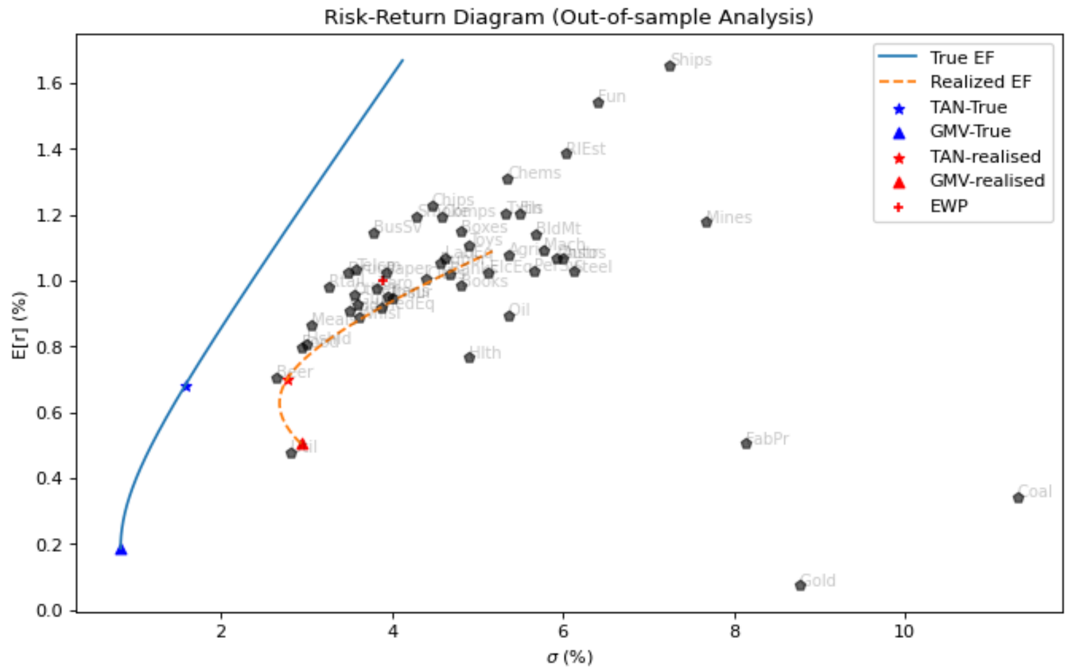
While this method is not comprehensive, iterating each lambda to derive multiple covariance matrices and the weights from the optimal portfolios are overly intensive computationally. Hence, we adopted a rather discrete approach.

Lambda = 0.5




```
******
The tangency portfolio constructed from the testing data is:  [ 0.105  0.512 -0.553  0.266  0.146 -0.154 -0.083 -0.027 -0.227  0.09
  0.052 -0.379  0.03  -0.21   0.448  0.19  -0.18  -0.257 -0.475 -0.114
  1.01  -0.349  0.012 -0.436 -0.303  0.638  0.05   0.003  0.079  0.066
 -0.095  0.403 -0.116  0.347 -0.456  0.482  0.317 -0.24  -0.383  0.138
  0.548  0.313  0.113  0.806 -0.893  0.12  -0.414  0.061]
The global minimum variance portfolio constructed from the testing data  is:  [ 0.107  0.117 -0.257  0.395  0.113 -0.311 -0.089 -0.064  0.051  0.091
  0.097 -0.252 -0.457 -0.097 -0.217  0.113 -0.255 -0.184 -0.292  0.021
  0.446 -0.24  -0.094 -0.178 -0.264  0.366  0.133  0.114  0.067 -0.056
  0.197  0.171  0.015  0.395 -0.3   -0.246  0.649  0.252 -0.301 -0.08
  0.978  0.234  0.069  0.399 -0.159  0.039 -0.238  0.001]
******
The tangency portfolio constructed from the training data is:  [ 0.074  0.043  0.011 -0.053 -0.007  0.004 -0.016  0.053  0.194 -0.105
 -0.033  0.073 -0.043 -0.187 -0.051  0.046 -0.072 -0.097 -0.051  0.052
 -0.047 -0.143 -0.058 -0.113 -0.002  0.1    0.055  0.016 -0.016  0.134
  0.41   0.079 -0.023 -0.188  0.077  0.046 -0.012  0.153  0.083  0.261
  0.036  0.166  0.036  0.022 -0.055  0.116  0.064 -0.033]
The global minimum variance portfolio constructed from the training data  is:  [ 0.102  0.045  0.007 -0.003 -0.039 -0.005 -0.057  0.123  0.259 -0.085
 -0.054  0.1    0.012 -0.244 -0.095 -0.013 -0.12  -0.189 -0.085  0.09
 -0.038 -0.186 -0.104 -0.171 -0.017  0.123  0.071 -0.002 -0.019  0.166
  0.492  0.099 -0.033 -0.142  0.117  0.077 -0.036  0.198  0.137  0.304
  0.047  0.216  0.085 -0.149 -0.09   0.127  0.006 -0.025]
******
Realised portfolio sigma:  2.535641530219763
Realised portfolio Er:  0.6374923654396207
Realised portfolio Sharpe:  0.25141265350089514
True portfolio sigma:  1.6584033072664826
True portfolio Er:  0.6780009776437605
True portfolio Sharpe:  0.40882756002295834
Difference in Sharpe:  0.1574149065220632
```

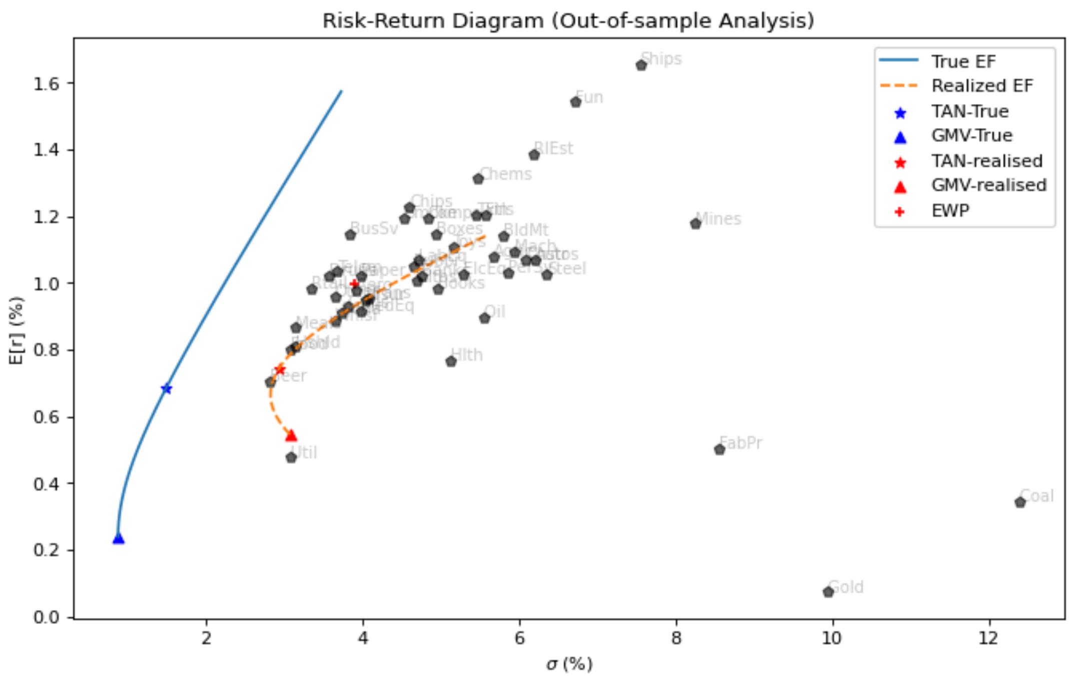
Lambda = 0.3



σ (%)

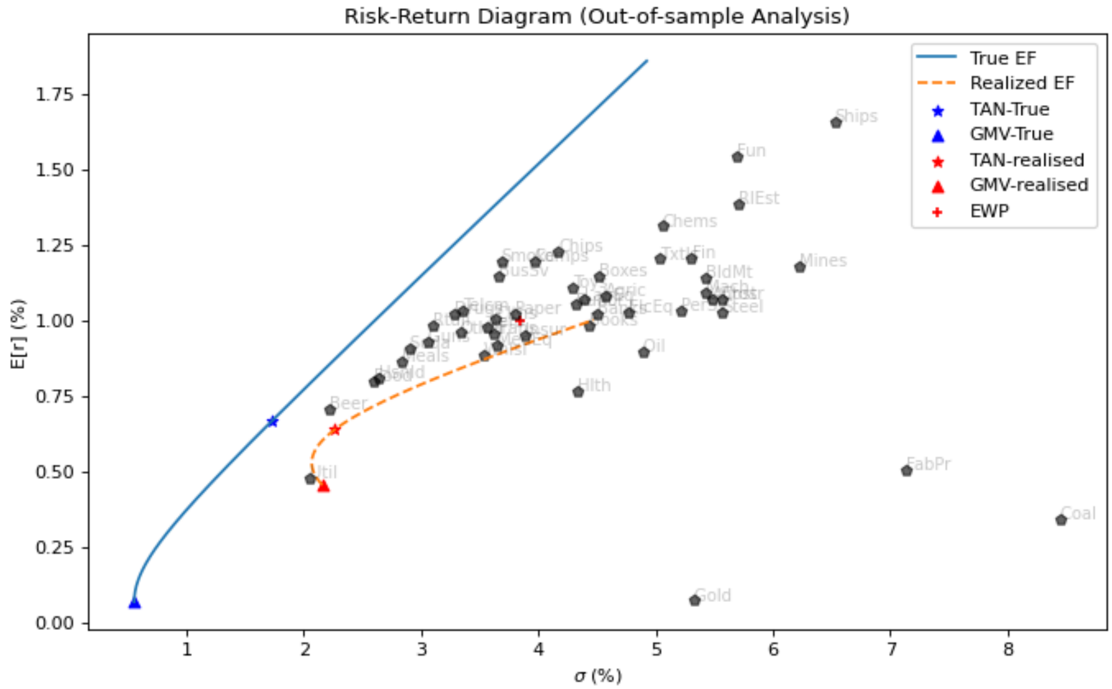
```
******
The tangency portfolio constructed from the testing data is:  [ 0.106  0.514 -0.556  0.261  0.147 -0.153 -0.086 -0.029 -0.231  0.089
  0.05  -0.375  0.037 -0.211  0.455  0.193 -0.179 -0.257 -0.476 -0.113
  1.017 -0.353  0.012 -0.438 -0.303  0.64   0.049  0.003  0.078  0.064
 -0.095  0.404 -0.117  0.353 -0.458  0.488  0.316 -0.239 -0.386  0.139
  0.542  0.315  0.116  0.805 -0.895  0.12  -0.417  0.056]
The global minimum variance portfolio constructed from the testing data  is:  [ 0.106  0.155 -0.286  0.383  0.116 -0.295 -0.089 -0.06   0.025  0.09
  0.093 -0.264 -0.411 -0.107 -0.153  0.12  -0.247 -0.191 -0.309  0.008
  0.498 -0.249 -0.084 -0.202 -0.267  0.392  0.124  0.103  0.068 -0.043
  0.169  0.193  0.003  0.389 -0.315 -0.176  0.617  0.203 -0.309 -0.059
  0.935  0.241  0.073  0.441 -0.231  0.046 -0.254  0.008]
******
The tangency portfolio constructed from the training data is:  [ 0.06   0.044  0.012 -0.057  0.003  0.    -0.006  0.038  0.17  -0.103
 -0.028  0.064 -0.041 -0.152 -0.043  0.049 -0.055 -0.074 -0.042  0.039
 -0.046 -0.114 -0.042 -0.094 -0.001  0.09   0.047  0.019 -0.015  0.128
  0.369  0.078 -0.022 -0.176  0.066  0.041 -0.011  0.135  0.069  0.233
  0.025  0.152  0.025  0.062 -0.04   0.1    0.075 -0.031]
The global minimum variance portfolio constructed from the training data  is:  [ 0.089  0.045  0.007 -0.002 -0.032 -0.01  -0.05   0.112  0.239 -0.081
 -0.051  0.092  0.019 -0.212 -0.09  -0.014 -0.107 -0.172 -0.078  0.08
 -0.036 -0.16  -0.091 -0.156 -0.017  0.114  0.065 -0.001 -0.018  0.162
  0.456  0.099 -0.033 -0.126  0.109  0.074 -0.037  0.182  0.128  0.278
  0.036  0.205  0.077 -0.122 -0.077  0.112  0.013 -0.022]
******
Realised portfolio sigma:  2.774968957672382
Realised portfolio Er:  0.699835946884856
Realised portfolio Sharpe:  0.252195955183539
True portfolio sigma:  1.58543931044009
True portfolio Er:  0.6823915357539274
True portfolio Sharpe:  0.43041164127847165
Difference in Sharpe:  0.17821568609493266
```

Lambda = 0.1



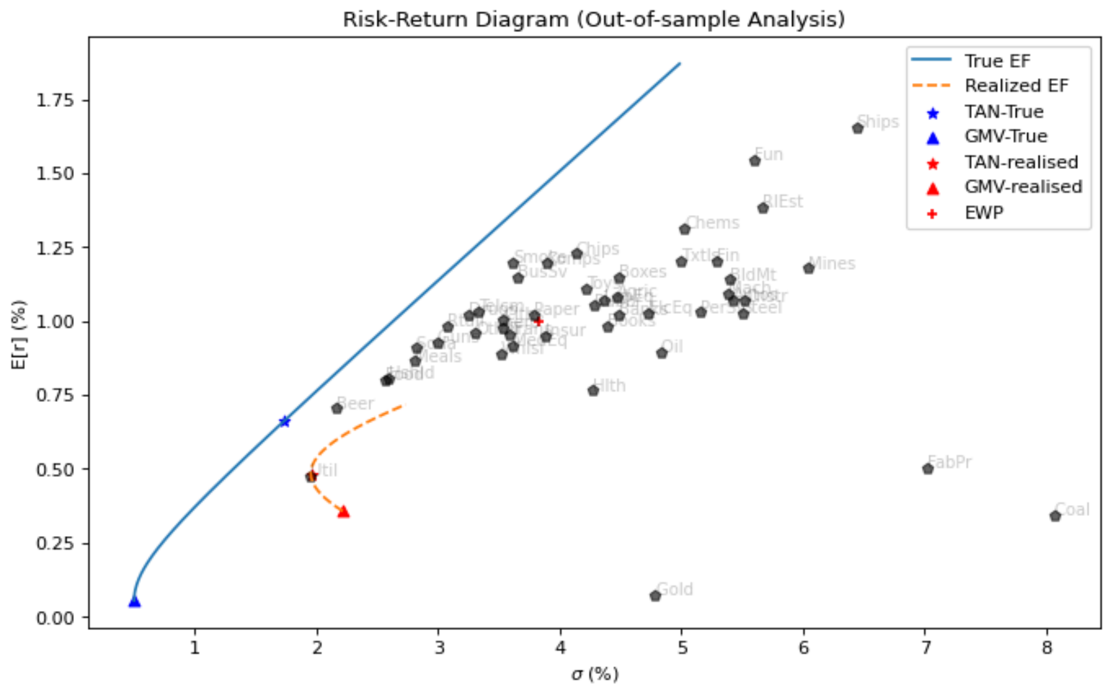

```
******
The tangency portfolio constructed from the testing data is:  [ 0.107  0.515 -0.558  0.256  0.147 -0.152 -0.089 -0.031 -0.235  0.087
  0.049 -0.372  0.043 -0.212  0.46   0.195 -0.179 -0.257 -0.477 -0.112
  1.023 -0.357  0.012 -0.44  -0.303  0.642  0.049  0.003  0.078  0.062
 -0.095  0.403 -0.118  0.359 -0.46   0.491  0.317 -0.238 -0.389  0.14
  0.538  0.317  0.118  0.804 -0.896  0.12  -0.419  0.051]
The global minimum variance portfolio constructed from the testing data  is:  [ 0.106  0.194 -0.315  0.369  0.119 -0.28  -0.089 -0.056 -0.002  0.09
  0.088 -0.276 -0.364 -0.117 -0.089  0.129 -0.239 -0.197 -0.327 -0.005
  0.552 -0.26  -0.074 -0.227 -0.271  0.417  0.116  0.092  0.069 -0.031
  0.142  0.216 -0.009  0.384 -0.331 -0.105  0.585  0.155 -0.317 -0.038
  0.892  0.249  0.077  0.482 -0.302  0.054 -0.272  0.013]
******
The tangency portfolio constructed from the training data is:  [ 0.049  0.044  0.013 -0.057  0.009 -0.003 -0.     0.027  0.153 -0.1
 -0.025  0.057 -0.036 -0.126 -0.038  0.05  -0.043 -0.058 -0.036  0.03
 -0.044 -0.093 -0.03  -0.081  0.     0.083  0.041  0.02  -0.014  0.124
  0.338  0.078 -0.022 -0.164  0.058  0.038 -0.011  0.121  0.06   0.212
  0.016  0.141  0.018  0.088 -0.028  0.088  0.082 -0.029]
The global minimum variance portfolio constructed from the training data  is:  [ 0.079  0.045  0.008 -0.002 -0.026 -0.013 -0.045  0.103  0.223 -0.077
 -0.048  0.086  0.025 -0.186 -0.086 -0.015 -0.096 -0.159 -0.073  0.071
 -0.034 -0.139 -0.08  -0.143 -0.016  0.107  0.059  0.    -0.018  0.159
  0.427  0.1   -0.034 -0.113  0.103  0.072 -0.037  0.169  0.12   0.257
  0.027  0.195  0.071 -0.099 -0.066  0.099  0.018 -0.02 ]
******
Realised portfolio sigma:  2.943551257113907
Realised portfolio Er:  0.7443289155866024
Realised portfolio Sharpe:  0.2528676590182438
True portfolio sigma:  1.5033749747457679
True portfolio Er:  0.6855338571043059
True portfolio Sharpe:  0.45599658675988997
Difference in Sharpe:  0.20312892774164615
```

Lambda = 0.75



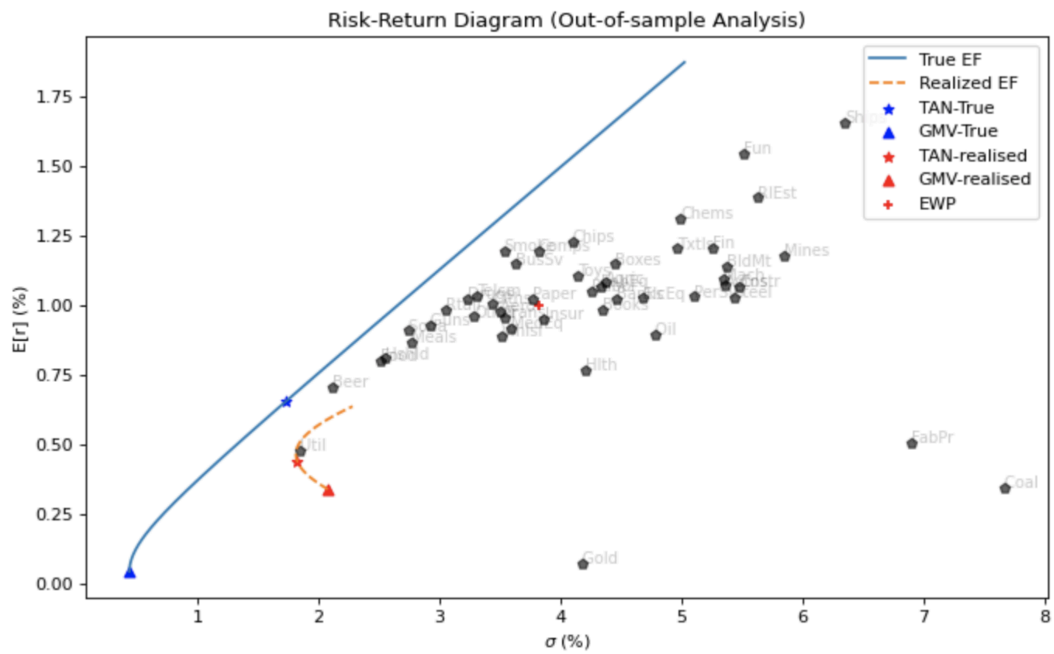

```
******
The tangency portfolio constructed from the testing data is:  [ 0.104  0.505 -0.547  0.275  0.145 -0.158 -0.08  -0.024 -0.219  0.092
  0.054 -0.382  0.016 -0.208  0.432  0.186 -0.182 -0.256 -0.472 -0.114
  0.995 -0.342  0.011 -0.431 -0.302  0.633  0.052  0.004  0.079  0.068
 -0.092  0.401 -0.113  0.339 -0.452  0.467  0.32  -0.236 -0.377  0.135
  0.559  0.31   0.109  0.803 -0.882  0.12  -0.409  0.067]
The global minimum variance portfolio constructed from the testing data  is:  [ 0.107  0.068 -0.222  0.41   0.11  -0.331 -0.091 -0.07   0.082  0.091
  0.103 -0.235 -0.512 -0.085 -0.295  0.103 -0.265 -0.175 -0.27   0.039
  0.382 -0.229 -0.107 -0.147 -0.26   0.334  0.143  0.128  0.066 -0.074
  0.232  0.142  0.029  0.406 -0.281 -0.335  0.69   0.315 -0.292 -0.107
  1.031  0.225  0.066  0.345 -0.068  0.03  -0.217 -0.009]
******
The tangency portfolio constructed from the training data is:  [ 0.074  0.043  0.011 -0.053 -0.007  0.004 -0.016  0.053  0.194 -0.105
 -0.033  0.073 -0.043 -0.187 -0.051  0.046 -0.072 -0.097 -0.051  0.052
 -0.047 -0.143 -0.058 -0.113 -0.002  0.1    0.055  0.016 -0.016  0.134
  0.41   0.079 -0.023 -0.188  0.077  0.046 -0.012  0.153  0.083  0.261
  0.036  0.166  0.036  0.022 -0.055  0.116  0.064 -0.033]
The global minimum variance portfolio constructed from the training data  is:  [ 0.102  0.045  0.007 -0.003 -0.039 -0.005 -0.057  0.123  0.259 -0.085
 -0.054  0.1    0.012 -0.244 -0.095 -0.013 -0.12  -0.189 -0.085  0.09
 -0.038 -0.186 -0.104 -0.171 -0.017  0.123  0.071 -0.002 -0.019  0.166
  0.492  0.099 -0.033 -0.142  0.117  0.077 -0.036  0.198  0.137  0.304
  0.047  0.216  0.085 -0.149 -0.09   0.127  0.006 -0.025]
******
Realised portfolio sigma:  2.259900689745566
Realised portfolio Er:  0.6374923654396207
Realised portfolio Sharpe:  0.2820886636002548
True portfolio sigma:  1.7266782678882822
True portfolio Er:  0.6666599616814222
True portfolio Sharpe:  0.3860939087956111
Difference in Sharpe:  0.10400524519535631
```

Lambda = 0.8



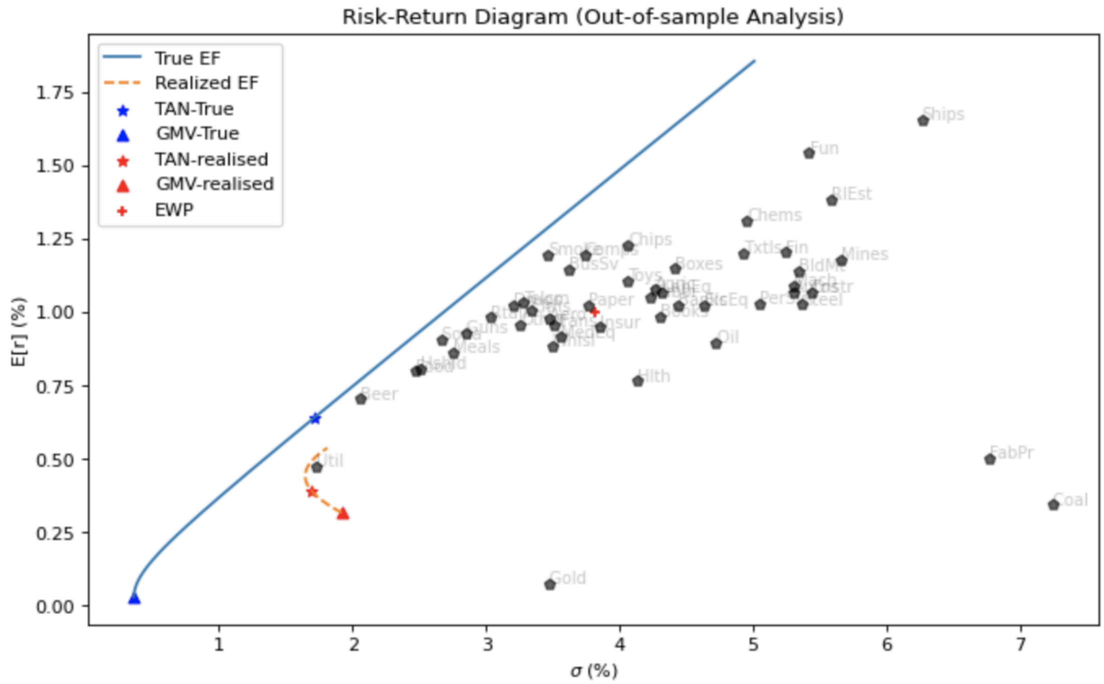

```
******
The tangency portfolio constructed from the testing data is:  [ 0.104  0.502 -0.544  0.277  0.145 -0.16  -0.079 -0.024 -0.216  0.092
  0.055 -0.382  0.011 -0.207  0.425  0.184 -0.182 -0.255 -0.471 -0.113
  0.989 -0.34   0.01  -0.429 -0.302  0.63   0.052  0.004  0.079  0.067
 -0.09   0.399 -0.112  0.338 -0.451  0.461  0.323 -0.233 -0.376  0.133
  0.564  0.309  0.108  0.8   -0.877  0.119 -0.407  0.068]
The global minimum variance portfolio constructed from the testing data  is:  [ 0.108  0.058 -0.215  0.413  0.109 -0.335 -0.092 -0.072  0.089  0.091
  0.103 -0.231 -0.523 -0.083 -0.311  0.102 -0.268 -0.174 -0.266  0.042
  0.369 -0.228 -0.11  -0.141 -0.259  0.328  0.145  0.131  0.066 -0.078
  0.24   0.136  0.032  0.409 -0.277 -0.352  0.698  0.328 -0.29  -0.112
  1.042  0.224  0.065  0.334 -0.05   0.028 -0.213 -0.012]
******
The tangency portfolio constructed from the training data is:  [ 0.11   0.043  0.008 -0.036 -0.032  0.01  -0.042  0.098  0.255 -0.107
 -0.046  0.098 -0.039 -0.27  -0.074  0.028 -0.115 -0.16  -0.074  0.086
 -0.048 -0.21  -0.101 -0.163 -0.009  0.125  0.074  0.007 -0.019  0.151
  0.505  0.084 -0.025 -0.203  0.106  0.062 -0.016  0.199  0.12   0.325
  0.063  0.205  0.067 -0.093 -0.094  0.15   0.031 -0.035]
The global minimum variance portfolio constructed from the training data  is:  [ 0.129  0.044  0.005 -0.003 -0.053  0.004 -0.069  0.143  0.297 -0.094
 -0.06   0.115 -0.003 -0.308 -0.103 -0.011 -0.147 -0.22  -0.096  0.112
 -0.042 -0.239 -0.131 -0.2   -0.018  0.14   0.085 -0.005 -0.021  0.172
  0.56   0.097 -0.032 -0.173  0.133  0.082 -0.032  0.228  0.156  0.354
  0.07   0.237  0.099 -0.205 -0.117  0.158 -0.007 -0.03 ]
******
Realised portfolio sigma:  1.9544177325078511
Realised portfolio Er:  0.47799353754519924
Realised portfolio Sharpe:  0.2445708149259637
True portfolio sigma:  1.7323399472422727
True portfolio Er:  0.6617739421667457
True portfolio Sharpe:  0.3820115925977632
Difference in Sharpe:  0.1374407776717995
```

Lambda = 0.85




```
******
The tangency portfolio constructed from the testing data is:  [ 0.103  0.497 -0.54   0.28   0.144 -0.162 -0.079 -0.024 -0.211  0.092
  0.056 -0.381  0.003 -0.206  0.416  0.183 -0.184 -0.254 -0.468 -0.112
  0.981 -0.338  0.009 -0.425 -0.301  0.626  0.054  0.006  0.079  0.066
 -0.087  0.396 -0.11   0.337 -0.448  0.451  0.327 -0.227 -0.374  0.13
  0.57   0.307  0.107  0.795 -0.867  0.118 -0.404  0.069]
The global minimum variance portfolio constructed from the testing data  is:  [ 0.108  0.049 -0.207  0.415  0.109 -0.339 -0.092 -0.073  0.095
  0.104 -0.227 -0.534 -0.081 -0.326  0.1   -0.27  -0.172 -0.262  0.046
  0.357 -0.226 -0.112 -0.134 -0.258  0.321  0.147  0.134  0.065 -0.082
  0.247  0.131  0.035  0.412 -0.274 -0.37   0.707  0.341 -0.289 -0.118
  1.052  0.222  0.065  0.322 -0.031  0.026 -0.209 -0.014]
******
The tangency portfolio constructed from the training data is:  [ 0.119  0.043  0.008 -0.031 -0.039  0.011 -0.049  0.11   0.27  -0.106
 -0.049  0.104 -0.036 -0.29  -0.081  0.021 -0.126 -0.176 -0.08   0.095
 -0.048 -0.226 -0.112 -0.175 -0.011  0.131  0.078  0.004 -0.019  0.156
  0.528  0.086 -0.026 -0.204  0.114  0.066 -0.018  0.21   0.13   0.34
  0.069  0.215  0.076 -0.124 -0.104  0.158  0.022 -0.035]
The global minimum variance portfolio constructed from the training data  is:  [ 0.134  0.044  0.005 -0.004 -0.056  0.006 -0.071  0.147  0.305
 -0.061  0.118 -0.006 -0.321 -0.104 -0.01  -0.153 -0.226 -0.099  0.116
 -0.043 -0.25  -0.137 -0.206 -0.018  0.143  0.087 -0.006 -0.021  0.173
  0.573  0.096 -0.031 -0.179  0.136  0.083 -0.031  0.234  0.16   0.364
  0.075  0.242  0.102 -0.217 -0.123  0.164 -0.009 -0.031]
******
Realised portfolio sigma:  1.8259725564293205
Realised portfolio Er:  0.4374596633565925
Realised portfolio Sharpe:  0.23957625311304923
True portfolio sigma:  1.7300776104407871
True portfolio Er:  0.6540845413123457
True portfolio Sharpe:  0.3780665892472297
Difference in Sharpe:  0.13849033613418046
```

Lambda = 0.9




```
******
The tangency portfolio constructed from the testing data is:  [ 0.103  0.487 -0.532  0.285  0.143 -0.166 -0.078 -0.025 -0.204  0.093
  0.058 -0.379 -0.01  -0.203  0.398  0.18  -0.186 -0.252 -0.463 -0.108
  0.965 -0.334  0.006 -0.418 -0.3    0.619  0.056  0.009  0.079  0.064
 -0.079  0.39  -0.107  0.337 -0.444  0.431  0.335 -0.215 -0.371  0.124
  0.582  0.305  0.105  0.785 -0.848  0.116 -0.399  0.068]
The global minimum variance portfolio constructed from the testing data  is:  [ 0.108  0.039 -0.2    0.418  0.108 -0.343 -0.093 -0.075  0.101
  0.105 -0.223 -0.545 -0.078 -0.342  0.098 -0.272 -0.17  -0.257  0.049
  0.344 -0.224 -0.115 -0.128 -0.257  0.315  0.149  0.137  0.065 -0.086
  0.254  0.125  0.037  0.415 -0.27  -0.388  0.715  0.354 -0.287 -0.123
  1.063  0.22   0.064  0.311 -0.013  0.024 -0.205 -0.017]
******
The tangency portfolio constructed from the training data is:  [ 0.129  0.043  0.007 -0.024 -0.046  0.012 -0.057  0.124  0.287 -0.105
 -0.053  0.111 -0.031 -0.313 -0.088  0.014 -0.139 -0.196 -0.088  0.105
 -0.048 -0.244 -0.124 -0.189 -0.013  0.137  0.084  0.001 -0.02   0.162
  0.554  0.088 -0.027 -0.204  0.123  0.072 -0.02   0.223  0.142  0.358
  0.076  0.227  0.086 -0.161 -0.114  0.166  0.011 -0.036]
The global minimum variance portfolio constructed from the training data  is:  [ 0.14   0.044  0.005 -0.004 -0.06   0.008 -0.073  0.152  0.314
 -0.062  0.122 -0.01  -0.336 -0.106 -0.009 -0.159 -0.233 -0.101  0.121
 -0.044 -0.262 -0.143 -0.213 -0.019  0.147  0.09  -0.006 -0.021  0.174
  0.588  0.096 -0.031 -0.186  0.14   0.084 -0.029  0.241  0.164  0.375
  0.08   0.247  0.105 -0.23  -0.129  0.171 -0.012 -0.033]
******
Realised portfolio sigma:  1.6941902959318886
Realised portfolio Er:  0.39041574300446324
Realised portfolio Sharpe:  0.23044385506276038
True portfolio sigma:  1.7091732993025728
True portfolio Er:  0.639667186729034
True portfolio Sharpe:  0.37425531219686725
Difference in Sharpe:  0.14381145713410687
```

Summary of the 7 Experiments

Lambda	Difference in Sharpe Ratio (Between Realized and True Portfolios)
0.50	0.1574149065220632
0.30	0.17821568609493266
0.10	0.20312892774164615
0.75	0.13870302829530767
0.80	0.1374407776717995
0.85	0.13849033613418046
0.90	0.14381145713410687

From our 7 experiments, we found that the optimal lambda is 0.80 where the sharpe ratio difference between the realized and the true portfolios is the lowest among the 7.

It has the lowest difference of 0.1374 as compared to the next lowest of 0.1384 where lambda is 0.75. As the lambda moves towards 0.1 from 0.5, we can see the sharpe ratio difference increase from 0.1574 to >0.20.

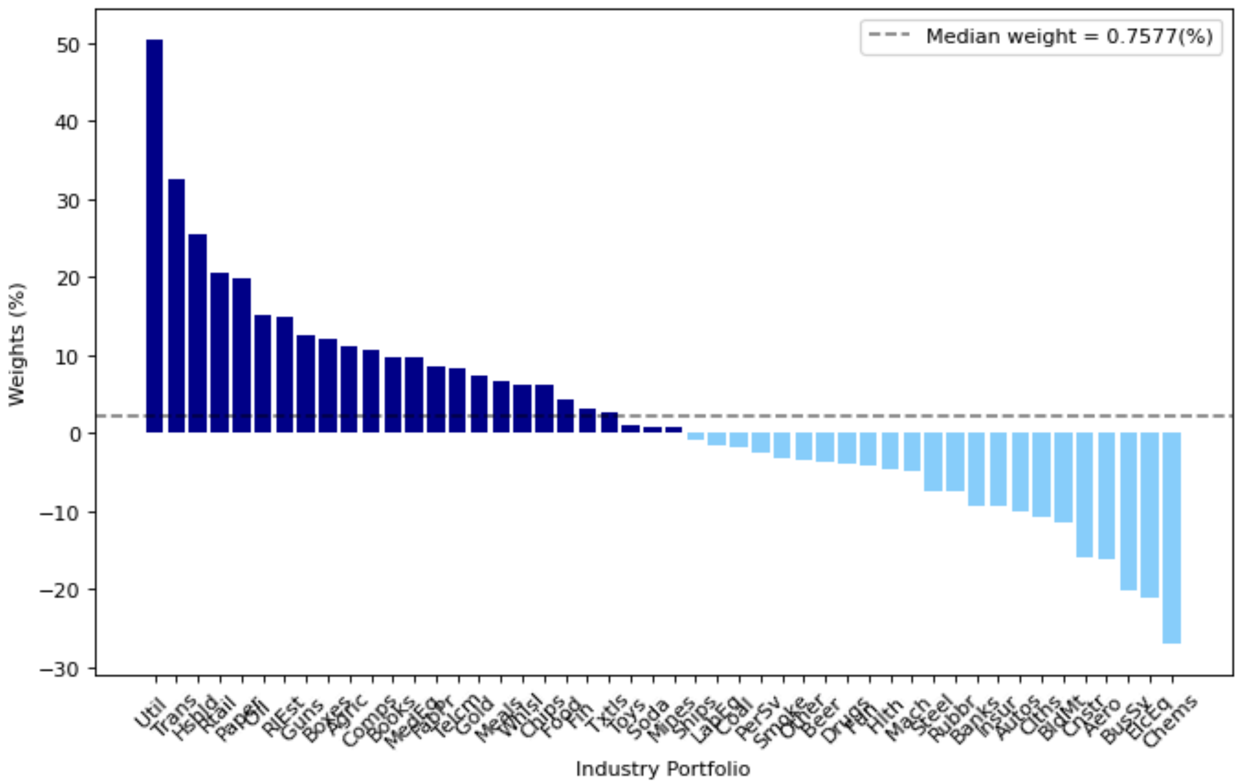
Hence, moving into the opposite direction towards 1, we can see the sharpe ratio difference decrease to 0.1374 when lambda is 0.8. However, the sharpe ratio difference increases back to 0.1384 when lambda is 0.85. Hence, this suggests that the lambda of 0.80 provides us with the most robust portfolio construction model.

Therefore, we would be choosing the weights suggested by our portfolio where lambda is 0.80.

2.5. Final Recommendation

Having decided to use the portfolio where lambda is 0.80, we would use the trained weights from the training data as the final investment allocation. The portfolio performance would then be evaluated on the test data.

Listed below is our allocation for our portfolio:



<u>Individual Weights</u>

	Industry	Weights
30	Util	0.505060
39	Trans	0.325324
8	Hshld	0.254572
41	Rtail	0.205047
37	Paper	0.198772
29	Oil	0.151167
45	RlEst	0.150035
25	Guns	0.124778
38	Boxes	0.120314
0	Agric	0.110150
34	Comps	0.106371
7	Books	0.097819
11	MedEq	0.097514
19	FabPr	0.086245
31	Telcm	0.084131
26	Gold	0.073668
42	Meals	0.067413
40	Whlsl	0.062869
35	Chips	0.061870
1	Food	0.043137
46	Fin	0.031148
15	Txtls	0.027516
5	Toys	0.009654
2	Soda	0.008422
27	Mines	0.006732
24	Ships	-0.008655
36	LabEq	-0.016202
28	Coal	-0.018625
32	PerSv	-0.025095
4	Smoke	-0.032071

47	Other	-0.035181
3	Beer	-0.036179
12	Drugs	-0.038817
6	Fun	-0.041698
10	Hlth	-0.045715
20	Mach	-0.048350
18	Steel	-0.074219
14	Rubbr	-0.074314
43	Banks	-0.093134
44	Insur	-0.093950

22	Autos	-0.100850
9	Clths	-0.106523
16	BldMt	-0.115159
17	Cnstr	-0.159598
23	Aero	-0.162510
33	BusSv	-0.202568
21	ElcEq	-0.210092
13	Chems	-0.270221

Looking at the weights, we see most of the weights go to the Utilities and Transportation while our portfolio provided the most shorts on Chemicals and Electrical Equipment. Looking at the breakdown of the respective industries we can observe most weights go to public-service related industries but the biggest shorts go to the private sector manufacturing and services.

	Mkt-RF	SMB	HML
Chems	1.089083	-0.048251	0.399824
ElcEq	1.267342	-0.097931	0.046660
BusSv	1.179516	0.145292	-0.651832
Util	0.518279	-0.205407	0.396131
Trans	0.992084	0.071349	0.438956
Hshld	0.774732	-0.162940	0.133165

Looking deeper into the 3 factor betas of the top 3 longs (Util, Trans, Hshld) and top 3 shorts (Chems, ElcEq, BusSv), we can see that the recommended portfolio prefers lower beta in market excess returns while preferring value companies as shown by the positive beta in HML (high minus low).

<u>Final Portfolio Performance Metrics</u>

On the test data:

	Monthly	Annually
Excess Returns (%)	0.478047	5.889823
Standard Deviation (%)	1.954714	6.771328
Sharpe Ratio	0.244561	0.244561

With that, we recommend this portfolio with a sharpe ratio of 0.244 of an annual excess returns of 5.89% and standard deviation of 6.77%.

2.6. Conclusion

We adopted the Fama-French 3-Factor Model as we believe that it could better explain the market prices better than CAPM by adjusting to size-risk and value factors.

After considering numerical stability and the model's performance replicability on out-of-sample datasets, we have chosen a shrinkage intensity/constant of 0.80 for our 3 factor model's covariance matrix. This model performs the best among other shrinkage constants and other methods of deriving the covariance matrix with the smallest difference in sharpe ratio between the 'realised' and 'true' portfolios.

From the selected model, we would recommend the weights above where we could achieve a sharpe ratio of 0.245 and excess annual returns 5.89%.