Fama-French Model

In Estimation of Portfolio μ and Ω

By Sicheng Wang

Data

- 8 selected stock daily return from 2010-01-04 to 2022-08-19:
  "AAPL","AEP","BAX","ED”,"F","GE","GOOG","MCD"
- Fama-French 5 factors downloaded from Dartmouth Tuck Business School Data Library:

$$R_{i,t} - R_{ft} = \alpha_{(5-\text{factor})} + \beta_{i,MKT}(R_{m,t} - R_{ft}) + \beta_{i,SMB}SMB + \beta_{i,HML}HML + \beta_{i,\mathrm{RMW}}RMW + \beta_{i,\mathrm{CMA}}CMA + \varepsilon_{i,t} \tag{9}$$

Background of Fama French Model

$$R_{i,t} - R_{ft} = \alpha_{(5-\text{factor})} + \beta_{i,MKT}(R_{m,t} - R_{ft}) + \\ \beta_{i,SMB}SMB + \beta_{i,HML}HML + \beta_{i,\text{RMW}}RMW + \beta_{i,\text{CMA}}CMA + \varepsilon_{i,t} \tag{9}$$

- $\boldsymbol{R_m - R_f}$**(market excess return):** value weighted return of all CRSP firms incorporated in the U.S.
- **SMB(Small Minus Big):** the average return on three small-cap portfolios subtract the average return of three large-cap portfolios
- **HML(High Minus Low):** the average return on two value portfolios subtract the average return of two growth portfolios
- **RMW(Robust Minus Weak):** the average return on the two robust operating profitability portfolios subtract the average return on the two weak operating profitability portfolios
- **CMA(Conservative Minus Aggressive):** the average return on the two conservative investment portfolios minus the average return on the two aggressive investment portfolios

Training Set and Test Set

- Merged the stock return and the Fama-French 5 factors data
- Split the merged data into Training set(70%) and Test set(30%)

R Code:

```
# Split the Merged data set into training set and test set
sample <- sample(c(TRUE, FALSE), nrow(R), replace=TRUE, prob=c(0.7,0.3))
R_train <- R[sample, ]
R_test <- R[!sample, ]
```

Fitting Fama-French Models with Training Set

- Applying the training set data to fit the five factor Fama-French models
- Also fitting the three factor FF models to provide reference and comparison

<u>R code</u>

```
# Five Factor models
fitLM = lm(r~Mkt.RF + SMB + HML + RMW + CMA, data = R_train)
# Three Factor models
fitLM2 = lm(r~Mkt.RF + SMB + HML, data = R_train)
```

Model Outputs for AAPL excess returns:

Five Factor Model Outputs:

```
Call:
lm(formula = AAPL ~ Mkt.RF + SMB + HML + RMW + CMA, data = R_train)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.8579 -0.6451 -0.0300  0.6258  8.0296 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.04571    0.02624   1.742   0.0816 .  
Mkt.RF       1.17566    0.02659  44.209  < 2e-16 ***
SMB         -0.20622    0.04730  -4.360 1.36e-05 ***
HML         -0.50912    0.04503 -11.306  < 2e-16 ***
RMW          0.54568    0.06560   8.318  < 2e-16 ***
CMA          0.03071    0.08905   0.345   0.7302    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.234 on 2221 degrees of freedom
Multiple R-squared:  0.4989,	Adjusted R-squared:  0.4978 
F-statistic: 442.3 on 5 and 2221 DF,  p-value: < 2.2e-16
```

Three Factor Model Outputs:

```
Call:
lm(formula = AAPL ~ Mkt.RF + SMB + HML, data = R_train)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.1394 -0.6795 -0.0415  0.6490  8.3456 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.05294    0.02660   1.991   0.0467 *  
Mkt.RF       1.13134    0.02582  43.817  < 2e-16 ***
SMB         -0.32627    0.04562  -7.152 1.15e-12 ***
HML         -0.40362    0.03472 -11.626  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.253 on 2223 degrees of freedom
Multiple R-squared:  0.4833,	Adjusted R-squared:  0.4826 
F-statistic: 693.1 on 3 and 2223 DF,  p-value: < 2.2e-16
```

Model Outputs for AEP excess returns:

Five Factor Model Outputs:

```
Call:
lm(formula = AEP ~ Mkt.RF + SMB + HML + RMW + CMA, data = R_train)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.6778 -0.5447  0.0154  0.5685  7.8945 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.005006   0.021135   0.237    0.813    
Mkt.RF       0.616070   0.021421  28.760  < 2e-16 ***
SMB         -0.369145   0.038102  -9.688  < 2e-16 ***
HML          0.003075   0.036271   0.085    0.932    
RMW          0.388586   0.052841   7.354 2.69e-13 ***
CMA          0.550429   0.071729   7.674 2.48e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9942 on 2221 degrees of freedom
Multiple R-squared:  0.2914,	Adjusted R-squared:  0.2898 
F-statistic: 182.7 on 5 and 2221 DF,  p-value: < 2.2e-16
```

Three Factor Model Outputs:

```
Call:
lm(formula = AEP ~ Mkt.RF + SMB + HML, data = R_train)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.7351 -0.5372  0.0163  0.5796  8.1743 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.01670    0.02163   0.772     0.44    
Mkt.RF       0.54963    0.02100  26.177   <2e-16 ***
SMB         -0.47714    0.03710 -12.862   <2e-16 ***
HML          0.23818    0.02823   8.436   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.019 on 2223 degrees of freedom
Multiple R-squared:  0.2553,	Adjusted R-squared:  0.2543 
F-statistic:   254 on 3 and 2223 DF,  p-value: < 2.2e-16
```

Model Outputs for BAX excess returns:

Five Factor Model Outputs:

```
Call:
lm(formula = BAX ~ Mkt.RF + SMB + HML + RMW + CMA, data = R_train)

Residuals:
     Min       1Q   Median       3Q      Max
-12.2144  -0.5140   0.0320   0.5264   7.5174

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.03093    0.02421  -1.278  0.20141
Mkt.RF       0.76102    0.02453  31.019  < 2e-16 ***
SMB         -0.27402    0.04364  -6.279 4.08e-10 ***
HML         -0.11765    0.04154  -2.832  0.00467 **
RMW          0.12522    0.06052   2.069  0.03865 *
CMA          0.42912    0.08215   5.223 1.92e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.139 on 2221 degrees of freedom
Multiple R-squared:  0.3072,	Adjusted R-squared:  0.3056
F-statistic:   197 on 5 and 2221 DF,  p-value: < 2.2e-16
```

Three Factor Model Outputs:

```
Call:
lm(formula = BAX ~ Mkt.RF + SMB + HML, data = R_train)

Residuals:
     Min       1Q   Median       3Q      Max
-12.3959  -0.5112   0.0447   0.5218   7.7089

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.02405    0.02433  -0.988    0.323
Mkt.RF       0.72299    0.02362  30.605  < 2e-16 ***
SMB         -0.31954    0.04174  -7.656 2.84e-14 ***
HML          0.03431    0.03176   1.080    0.280
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.146 on 2223 degrees of freedom
Multiple R-squared:  0.2973,	Adjusted R-squared:  0.2964
F-statistic: 313.5 on 3 and 2223 DF,  p-value: < 2.2e-16
```

Model Outputs for GOOG excess returns:

Five Factor Model Outputs:

```
Call:
lm(formula = GOOG ~ Mkt.RF + SMB + HML + RMW + CMA, data = R_train)

Residuals:
    Min      1Q  Median      3Q     Max
-8.5596 -0.5042 -0.0263  0.4718 11.9483

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01651    0.02183   0.756  0.44967
Mkt.RF       1.03827    0.02212  46.928  < 2e-16 ***
SMB         -0.19701    0.03935  -5.006 5.99e-07 ***
HML         -0.17468    0.03746  -4.663 3.30e-06 ***
RMW          0.16961    0.05458   3.108  0.00191 **
CMA         -0.68701    0.07409  -9.273  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.027 on 2221 degrees of freedom
Multiple R-squared:  0.5642,	Adjusted R-squared:  0.5632
F-statistic:   575 on 5 and 2221 DF,  p-value: < 2.2e-16
```

Three Factor Model Outputs:

```
Call:
lm(formula = GOOG ~ Mkt.RF + SMB + HML, data = R_train)

Residuals:
    Min      1Q  Median      3Q     Max
-8.5050 -0.5164 -0.0320  0.4994 12.2401

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01013    0.02225   0.455    0.649
Mkt.RF       1.07046    0.02160  49.551  < 2e-16 ***
SMB         -0.20466    0.03817  -5.362 9.08e-08 ***
HML         -0.35272    0.02905 -12.142  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.048 on 2223 degrees of freedom
Multiple R-squared:  0.5454,	Adjusted R-squared:  0.5448
F-statistic: 889.1 on 3 and 2223 DF,  p-value: < 2.2e-16
```

Five Factor VS. Three Factor Fama-French

- The additional two predictors RMW (Robust Minus Weak) and CMA (Conservative Minus Aggressive) is not always significant in the estimation
- However, the five-factor model does provide a higher R^2 than the three-factor model on average, thus a better fit.

Coefficient Estimation of the Five Factor Models

R-code:

```
res = residuals(fitLM)
betas = fitLM$coefficients[2:6,]
Signif(betas, 2)
```

Output:

	AAPL	AEP	BAX	ED	F	GE	GOOG	MCD
Mkt.RF	1.200	0.6200	0.76	0.550	1.2000	1.100	1.00	0.730
SMB	-0.210	-0.3700	-0.27	-0.410	0.4500	0.120	-0.20	-0.140
HML	-0.510	0.0031	-0.12	-0.019	0.6100	0.710	-0.17	0.016
RMW	0.550	0.3900	0.13	0.400	0.1400	-0.065	0.17	0.430
CMA	0.031	0.5500	0.43	0.720	0.0056	0.260	-0.69	0.120

Prediction on Testing Set

- Use the estimated five factor model to make prediction on the testing set
- Calculate mean squared prediction error (MSPE) to evaluate model performance

R code:

```
# Prediction on testing set
pred <- predict(fitLM2, newdata = R_test)
# Calculating MSPE(mean squared prediction error)
c(MSPE_train = mean((R_train$r - predict.lm(fitLM2, R_train)) ^ 2),
  MSPE_test = mean((R_test$r - predict.lm(fitLM2, R_test)) ^ 2))
```

Output:

```
MSPE_train  MSPE_test
  1.998861   1.945220
```

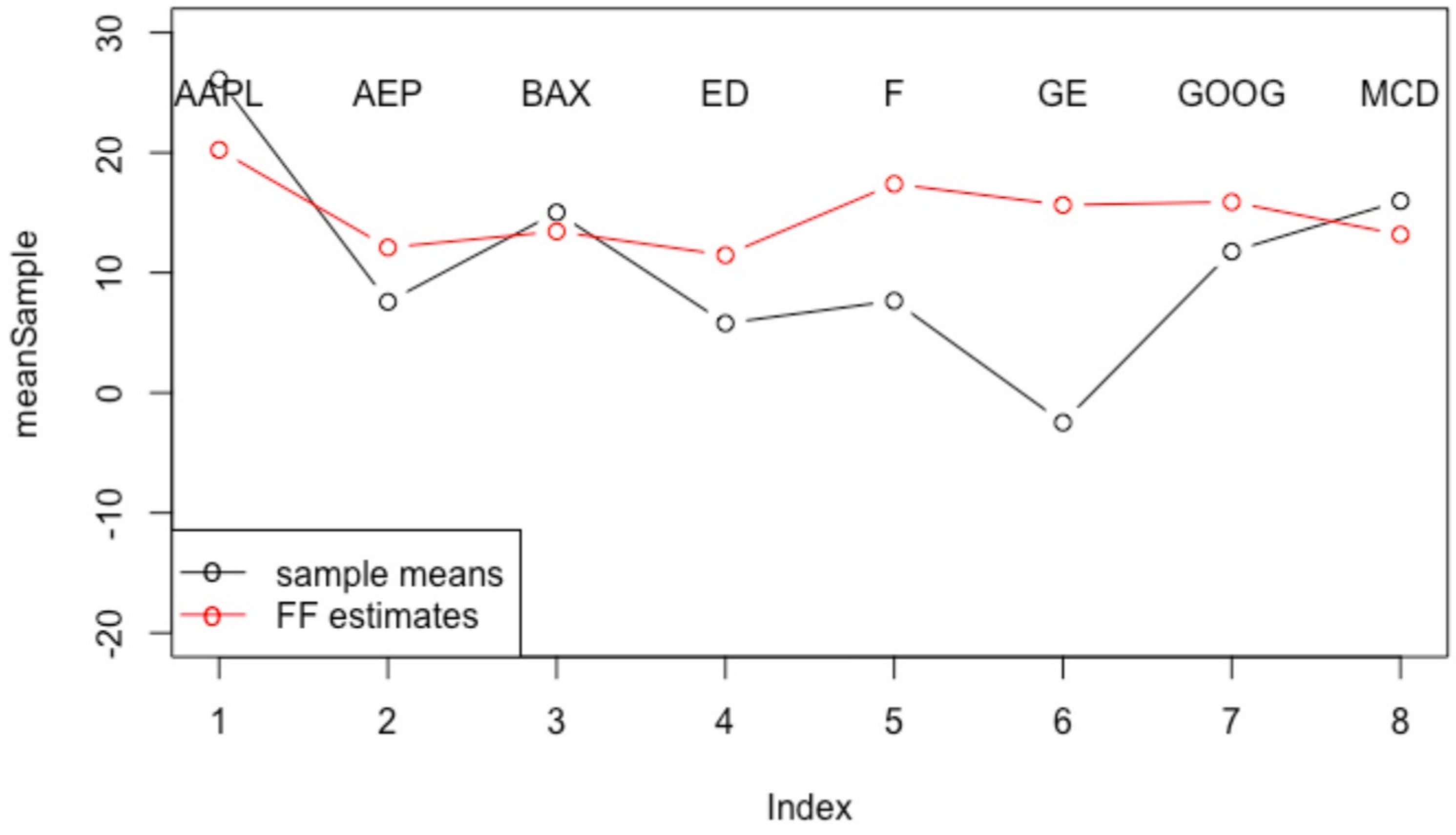
Model Performance

- From above we arrived at an MSPE output of :

```
MSPE_train  MSPE_test
  1.998861   1.945220
```

- MSPE summarize the predictive ability of a model, which the closer to the zero the better.
- Since MSPE of the test set is smaller than the MSPE of the training set, we can say that the model is well fitted and does not suffering from overfitting.

Sample Mean μ vs Estimated Mean $\hat{\mu}$



R-code:

```
# Finding sample mean excess returns
n = nrow(R)
meanSample = 253*colMeans(r,na.rm = TRUE)

# Finding estimated mean excess returns
meanFF = 253*(R$RF[n] + t(betas) %*% (colMeans(R[9:13])))
```

Sample Covariance Matrix Ω

	AAPL	AEP	BAX	ED	F	GE	GOOG	MCD
AAPL	3.1992918	0.5609036	0.8721414	0.4399083	1.4270972	1.2455059	1.6509382	0.8499815
AEP	0.5609036	1.4458856	0.5988618	1.1299393	0.6152814	0.6825282	0.5132121	0.5334708
BAX	0.8721414	0.5988618	1.9711668	0.5762858	0.8992229	0.9451165	0.8948237	0.5985095
ED	0.4399083	1.1299393	0.5762858	1.4152456	0.5227168	0.5826824	0.3969391	0.4980781
F	1.4270972	0.6152814	0.8992229	0.5227168	4.4626714	2.2334353	1.3518373	1.0312999
GE	1.2455059	0.6825282	0.9451165	0.5826824	2.2334353	4.0483832	1.2018547	0.9623700
GOOG	1.6509382	0.5132121	0.8948237	0.3969391	1.3518373	1.2018547	2.7845829	0.8146396
MCD	0.8499815	0.5334708	0.5985095	0.4980781	1.0312999	0.9623700	0.8146396	1.4221366

<u>R code:</u>

```
cov_sample <- cov(R$r)
```

Estimated Covariance Matrix $\widehat{\Omega}$

	AAPL	AEP	BAX	ED	F	GE	GOOG	MCD
AAPL	3.1167971	0.8916731	1.2232059	0.7490709	1.8087718	1.5862357	1.9117797	1.1581629
AEP	0.8916731	1.2293974	0.6228380	0.4911947	0.9526087	0.9153808	0.7878356	0.5958817
BAX	1.2232059	0.6228380	1.7128064	0.5435493	1.2569368	1.1637571	1.1317392	0.7515384
ED	0.7490709	0.4911947	0.5435493	1.1908642	0.8025233	0.7903540	0.6384393	0.5178257
F	1.8087718	0.9526087	1.2569368	0.8025233	4.2017729	2.3352057	1.7637896	1.2359961
GE	1.5862357	0.9153808	1.1637571	0.7903540	2.3352057	3.8037257	1.5415609	1.1298573
GOOG	1.9117797	0.7878356	1.1317392	0.6384393	1.7637896	1.5415609	2.5928558	1.0736509
MCD	1.1581629	0.5958817	0.7515384	0.5178257	1.2359961	1.1298573	1.0736509	1.3494977

R code:

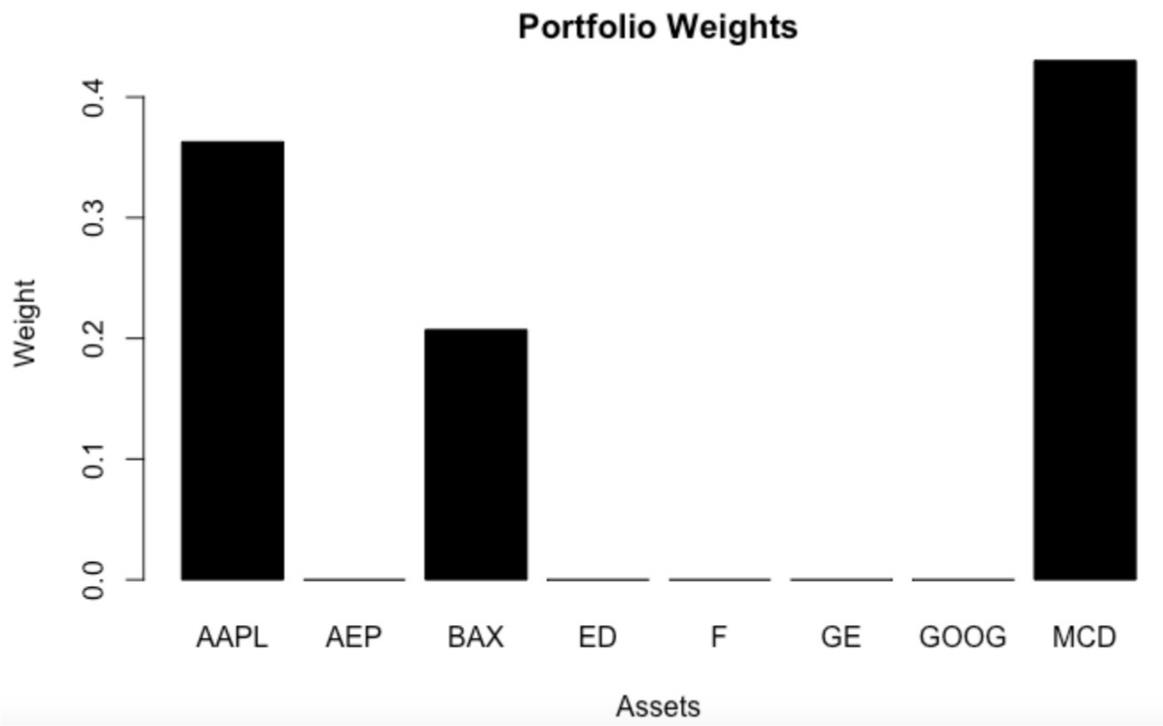
```
N = 3180
sigF5 = as.matrix(var(cbind(R_test$Mkt.RF, R_test$SMB, R_test$HML, R_test$RMW, R_test$CMA)))
sigeps5 = crossprod(res)/(N-6)
sigeps5 = diag(diag(sigeps5))
cov_FF = t(betas) %*% sigF5 %*% (betas) + sigeps5
```

Tangency Portfolio by sample μ and Ω

```
Call:
tangency.portfolio(er = meanSample, cov.mat = cov_sample, risk.free = mu_f,
    shorts = FALSE)

Portfolio expected return:     19.45221
Portfolio standard deviation:  1.127691
Portfolio weights:
  AAPL    AEP    BAX     ED      F     GE   GOOG    MCD
0.3628 0.0000 0.2071 0.0000 0.0000 0.0000 0.0000 0.4301
Call:
tangency.portfolio(er = meanSample, cov.mat = cov_sample, risk.free = mu_f,
    shorts = FALSE)

Portfolio expected return:     19.45221
Portfolio standard deviation:  1.127691
Portfolio Sharpe Ratio:        17.24795
Portfolio weights:
  AAPL    AEP    BAX     ED      F     GE   GOOG    MCD
0.3628 0.0000 0.2071 0.0000 0.0000 0.0000 0.0000 0.4301
```

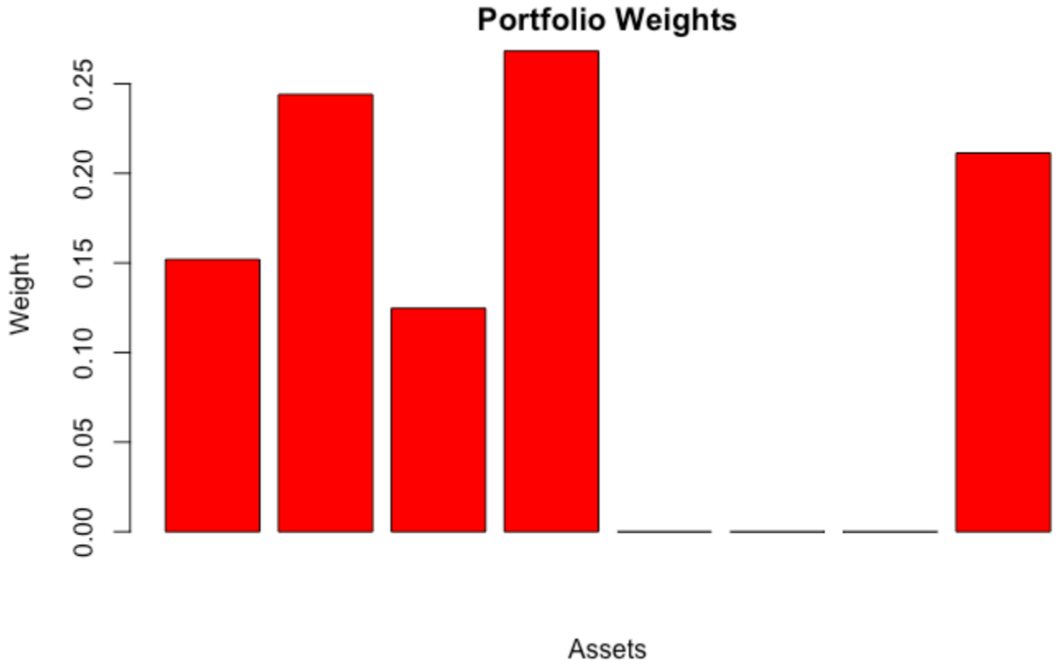


Tangency Portfolio by estimated $\hat{\mu}$ and $\widehat{\Omega}$

```
Call:
tangency.portfolio(er = meanFF, cov.mat = cov_FF, risk.free = mu_f,
    shorts = FALSE)

Portfolio expected return:     13.55308
Portfolio standard deviation:  0.9292993
Portfolio weights:
[1] 0.1519 0.2440 0.1246 0.2682 0.0000 0.0000 0.0000 0.2113
Call:
tangency.portfolio(er = meanFF, cov.mat = cov_FF, risk.free = mu_f,
    shorts = FALSE)

Portfolio expected return:     13.55308
Portfolio standard deviation:  0.9292993
Portfolio Sharpe Ratio:        14.58221
Portfolio weights:
[1] 0.1519 0.2440 0.1246 0.2682 0.0000 0.0000 0.0000 0.2113
```

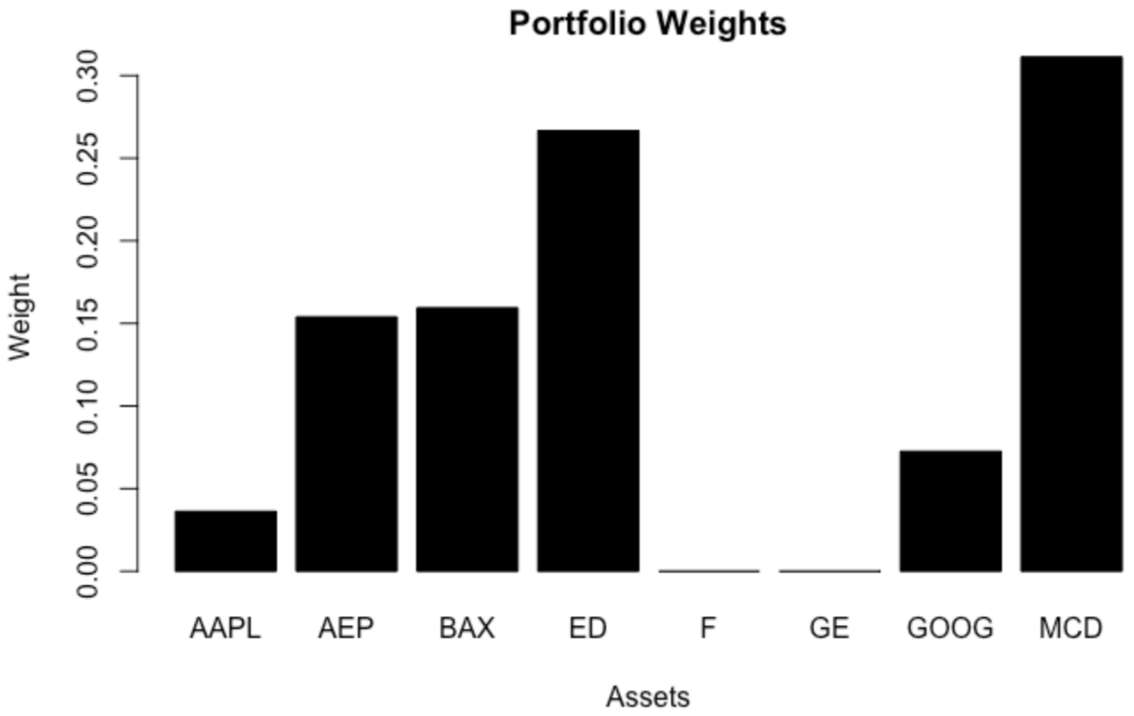


Mini-Variance Portfolio by sample μ and Ω

```
Call:
globalMin.portfolio(er = meanSample, cov.mat = cov_sample, shorts = FALSE)

Portfolio expected return:     11.87276
Portfolio standard deviation:  0.9180861
Portfolio weights:
  AAPL    AEP    BAX     ED      F     GE   GOOG    MCD
0.0362 0.1539 0.1594 0.2666 0.0000 0.0000 0.0726 0.3113
Call:
globalMin.portfolio(er = meanSample, cov.mat = cov_sample, shorts = FALSE)

Portfolio expected return:     11.87276
Portfolio standard deviation:  0.9180861
Portfolio Sharpe Ratio:        12.93007
Portfolio weights:
  AAPL    AEP    BAX     ED      F     GE   GOOG    MCD
0.0362 0.1539 0.1594 0.2666 0.0000 0.0000 0.0726 0.3113
```

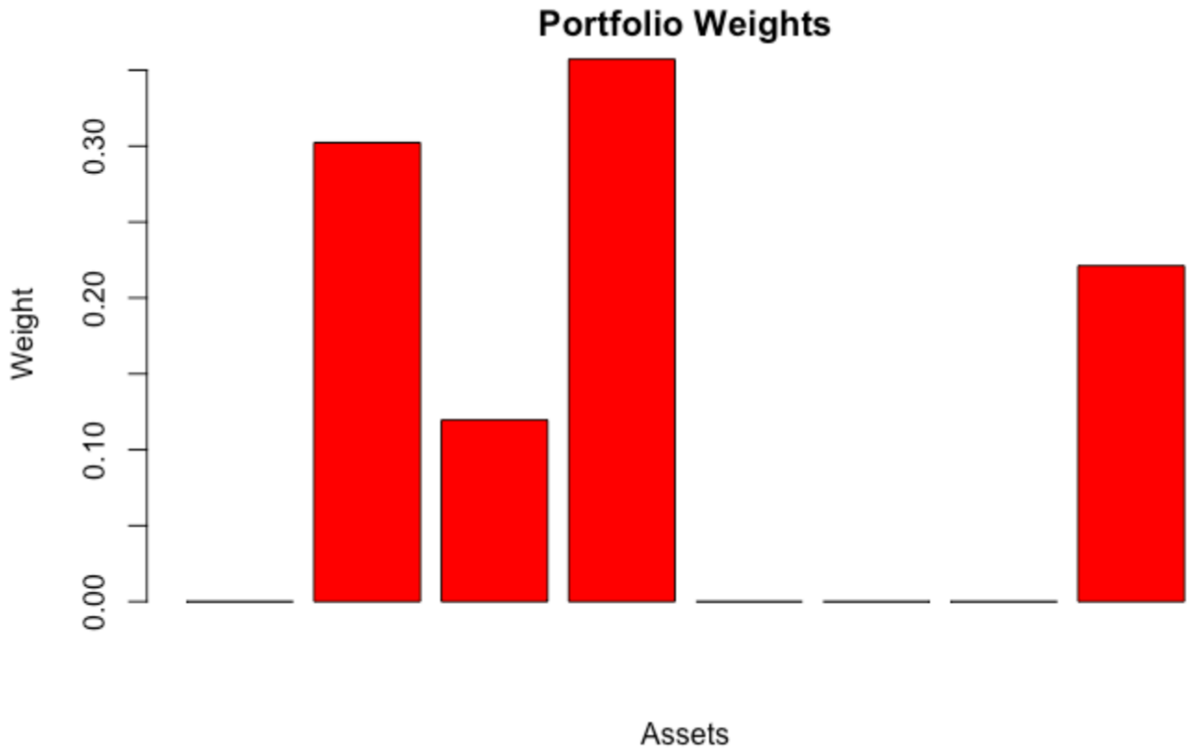


Mini-Variance Portfolio by estimated $\hat{\mu}$ and $\widehat{\Omega}$

```
Call:
globalMin.portfolio(er = meanFF, cov.mat = cov_FF, shorts = FALSE)

Portfolio expected return:     12.26455
Portfolio standard deviation:  0.8678809
Portfolio weights:
[1] 0.0000 0.3023 0.1195 0.3571 0.0000 0.0000 0.0000 0.2211
Call:
globalMin.portfolio(er = meanFF, cov.mat = cov_FF, shorts = FALSE)

Portfolio expected return:     12.26455
Portfolio standard deviation:  0.8678809
Portfolio Sharpe Ratio:        14.12947
Portfolio weights:
[1] 0.0000 0.3023 0.1195 0.3571 0.0000 0.0000 0.0000 0.2211
```



Closing Remark

- Estimated efficient portfolios tends to deviate from the true efficient portfolios, thus not idea for building trading strategy.
- However, CAPM based Fama-French 5 factor model does provide a useful estimation for μ and Ω consider the high level of randomness and lack stationarity of stock returns.
- For further research and study, one can investigate how limiting short selling impact the robustness of the estimation of Fama-French model.

Citations

- Shyam. (2021, September 14). *Factors, the famous 5*. StockViz. Retrieved December 1, 2022, from https://stockviz.biz/2020/08/23/factors-the-famous-5/
- French, K. R. (n.d.). *Description of Fama/French Factors*. Kenneth R. French - Home Page. Retrieved December 1, 2022, from https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_factors.html
- Zivot, E. (2021, June 11). *Introduction to computational finance and financial econometrics with R*. Home. Retrieved December 1, 2022, from https://bookdown.org/compfinezbook/introcompfinr/