Homework 4, MATH 5261

Zongyi Liu

Wed, Oct 1, 2025

Github Repository Directory: `https:/github.com/zongyiliu/STAT5261/tree/main/Homework_4`

1 Question 1

Do problems 1, 2, and 3 from Section 16.10.1 of Chapter 16. Each problem is worth 5 points. To receive full credit, you must provide both:

- The R code you used to solve each problem.
- Your answers to each question.

1.1 Problem 1

This section uses daily stock prices in the data set `Stock Bond.csv` that is posted on the books website and in which any variable whose name ends with AC is an adjusted closing price. As the name suggests, these prices have been adjusted for dividends and stock splits, so that returns can be calculated without further adjustments. Run the following code which will read the data, compute the returns for six stocks, create a scatterplot matrix of these returns, and compute the mean vector, covariance matrix, and vector of standard deviations of the returns. Note that returns will be percentages.

Write an R program to find the efficient frontier, the tangency portfolio, and the minimum variance portfolio, and plot on reward-risk space the location of each of the six stocks, the efficient frontier, the tangency portfolio, and the line of efficient portfolios. Use the constraints that $0.1 \leq w_j \leq 0.5$ for each stock. The first constraint limits short sales but does not rule them out completely. The second constraint prohibits more than 50% of the investment in any single stock. Assume that the annual risk-free rate is 3% and convert this to a daily rate by dividing by 365, since interest is earned on trading as well as nontrading days.

Answer

Firstly to read the data as code given in textbook:

```
    dat = read.csv("Stock_Bond.csv", header = T)
    prices = cbind(dat$GM_AC, dat$F_AC, dat$CAT_AC, dat$UTX_AC,
    dat$MRK_AC, dat$IBM_AC)
    n = dim(prices)[1]
    returns =  100 * (prices[2:n, ] / prices[1:(n-1), ] - 1)
    pairs(returns)
    mean_vect = colMeans(returns)
    cov_mat = cov(returns)
    sd_vect = sqrt(diag(cov_mat))
```

For

```
    mean_vect
```

We get:

```
0.04089730 0.04786051 0.07989876 0.07284620 0.06240313 0.04616853
```

For

```
      cov_mat
```

We get:

```
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 4.2605957 2.6699572 1.4654812 1.2455745 0.8957578 1.2340664
[2,] 2.6699572 4.4391064 1.5316444 1.2740768 0.9780046 1.1875001
[3,] 1.4654812 1.5316444 3.9094617 1.3875721 0.8230401 1.1040855
[4,] 1.2455745 1.2740768 1.3875721 3.0467665 0.8154347 0.9491030
[5,] 0.8957578 0.9780046 0.8230401 0.8154347 3.0879666 0.7996968
[6,] 1.2340664 1.1875001 1.1040855 0.9491030 0.7996968 3.6498981
```

For

```
	sd_vect
```

We get:

```
2.064121 2.106919 1.977236 1.745499 1.757261 1.910471
```

The scatterplot can be generated as below:

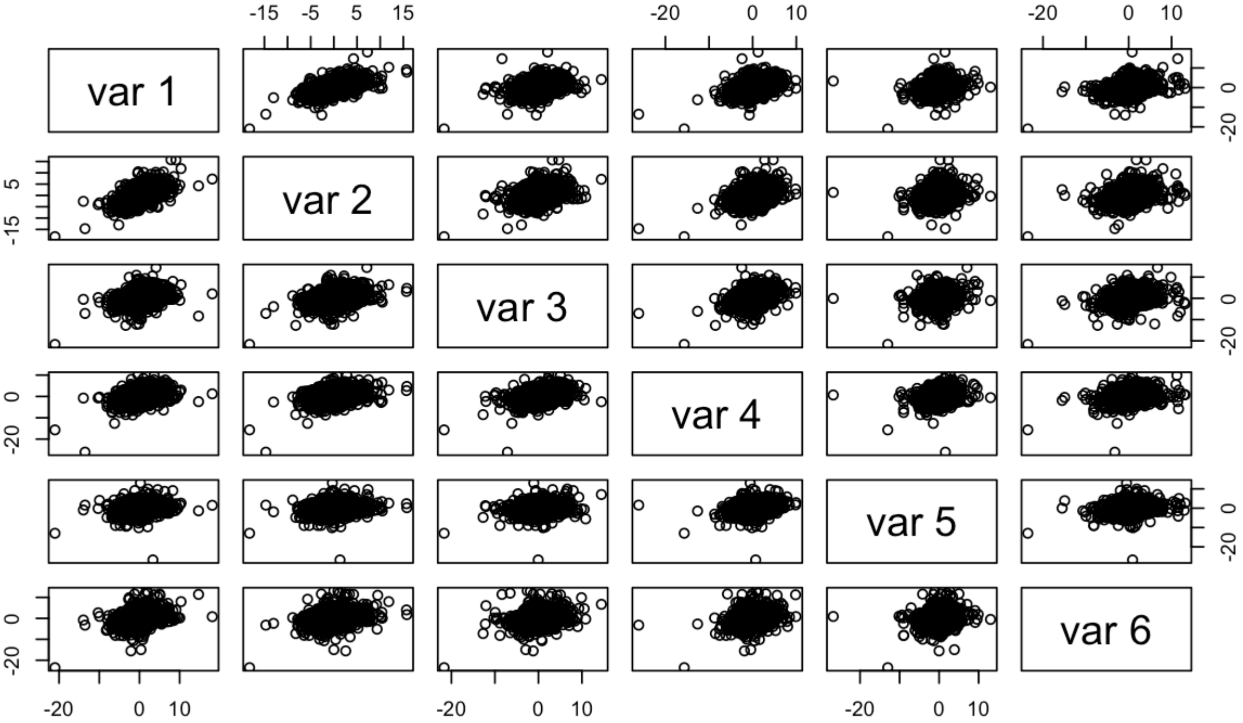


Figure 1: Scatterplot of returns

To find the efficient frontier and plot it in graph, I have R code as below; it was inspired by the sample code provided by the author to generate Figure 16.1 in the textbook.

```
	# dat <- read.csv("Stock_Bond.csv", header = TRUE)
	# read the library: library(quadprog)

	prices  <- cbind(dat$GM_AC, dat$F_AC, dat$CAT_AC, dat$UTX_AC, dat$MRK_AC, dat$IBM_AC)
	n       <- nrow(prices)
	returns <- 100 * (prices[2:n, ] / prices[1:(n-1), ] - 1)  # unit: % per day
	mean_vect  <- colMeans(returns)
	cov_mat    <- cov(returns)
	sd_vect    <- sqrt(diag(cov_mat))
	tickers    <- c("GM","F","CAT","UTX","MRK","IBM")

	## setting
	M   <- length(mean_vect)
```

```
rf  <- (0.03/365) * 100    # annual 3% converted to daily percentage to match 'returns'
lb  <- -0.10               # allow up to 10% short (w_j >= -0.1)
ub  <-  0.50               # cap any single weight at 50% (w_j <= 0.5)

# quadprog uses t(Amat) %*% w >= bvec, with the first 'meq' constraints as equalities
Amat <- cbind(rep(1, M),        # sum w = 1
mean_vect,        # target mean
diag(M),          # w_j >= lb
-diag(M))         # -w_j >= -ub  (i.e., w_j <= ub)

solve_frontier <- function(mu_grid, Sigma, mu) {
   k <- length(mu_grid)
   sdP <- numeric(k)
   W   <- matrix(NA_real_, nrow = k, ncol = M)
   Dmat <- Sigma + 1e-10 * diag(M)
   for (i in seq_len(k)) {
           bvec <- c(1, mu_grid[i], rep(lb, M), rep(-ub, M))
           res  <- solve.QP(Dmat = Dmat, dvec = rep(0, M), Amat = Amat, bvec = bvec, meq = 2)
           sdP[i] <- sqrt(2 * res$value)   # res$value = (1/2) * w' * Sigma * w
           W[i, ] <- res$solution
   }
   list(sd = sdP, W = W)
}

mu_grid <- seq(min(mean_vect) + 1e-4, max(mean_vect) - 1e-4, length.out = 300)
fr <- solve_frontier(mu_grid, cov_mat, mean_vect)
sdP <- fr$sd
W   <- fr$W

## global MVP
i_gmv  <- which.min(sdP)
w_gmv  <- W[i_gmv, ]
mu_gmv <- mu_grid[i_gmv]
sd_gmv <- sdP[i_gmv]

## tangency under the same bounds
sharpe <- (mu_grid - rf) / sdP
i_tan  <- which.max(sharpe)
w_tan  <- W[i_tan, ]
mu_tan <- mu_grid[i_tan]
sd_tan <- sdP[i_tan]

## print weights
cat("\n==␣GMV␣weights␣==\n")
print(setNames(round(w_gmv, 4), tickers))
cat("GMV␣mean␣=", round(mu_gmv, 5), "%␣␣sd␣=", round(sd_gmv, 5), "%\n")

cat("\n==␣Tangency␣weights␣(bounds␣-0.1..0.5,␣rf␣=", round(rf,5), "%/day)␣==\n")
print(setNames(round(w_tan, 4), tickers))
cat("Tangency␣mean␣=", round(mu_tan, 5), "%␣␣sd␣=", round(sd_tan, 5),
"%␣␣Sharpe␣=", round(max(sharpe), 4), "\n\n")

## plot
plot(sd_vect, mean_vect, pch = 19, cex = 1.1,
xlab = "Risk␣(sd␣per␣day)", ylab = "Expected␣return",
xlim = c(0, max(c(sd_vect, sdP)) * 1.05),
ylim = range(c(mean_vect, mu_grid, rf)))
text(sd_vect, mean_vect, labels = tickers, pos = 4, cex = 0.9)

# full (bounded) frontier
lines(sdP, mu_grid, col = "gray60", lwd = 1)
```

```

    # efficient part (to the right of GMV: higher mean than GMV)
    eff_idx <- mu_grid > mu_gmv
    lines(sdP[eff_idx], mu_grid[eff_idx], col = "red", lwd = 3)

    # risk-free point, tangency point, and CML
    points(0, rf, pch = 8, cex = 1.2, col = "blue")
    points(sd_tan, mu_tan, pch = 19, cex = 1.2, col = "purple")

    # Capital Market Line from rf through the tangency portfolio
    x_cml <- seq(0, max(sdP), length.out = 100)
    y_cml <- rf + (mu_tan - rf) / sd_tan * x_cml
    lines(x_cml, y_cml, col = "blue")

    legend("topleft",
    legend = c("Assets", "Frontier␣(feasible)", "Efficient␣frontier",
    "Risk-free", "Tangency", "CML"),
    pch = c(19, NA, NA, 8, 19, NA),
    lty = c(NA, 1, 1, NA, NA, 2),
    col = c("black", "gray60", "red", "blue", "purple", "blue"),
    pt.cex = c(1.1, NA, NA, 1.2, 1.2, NA),
    lwd = c(NA, 1, 3, NA, NA, 2), bty = "n")
```

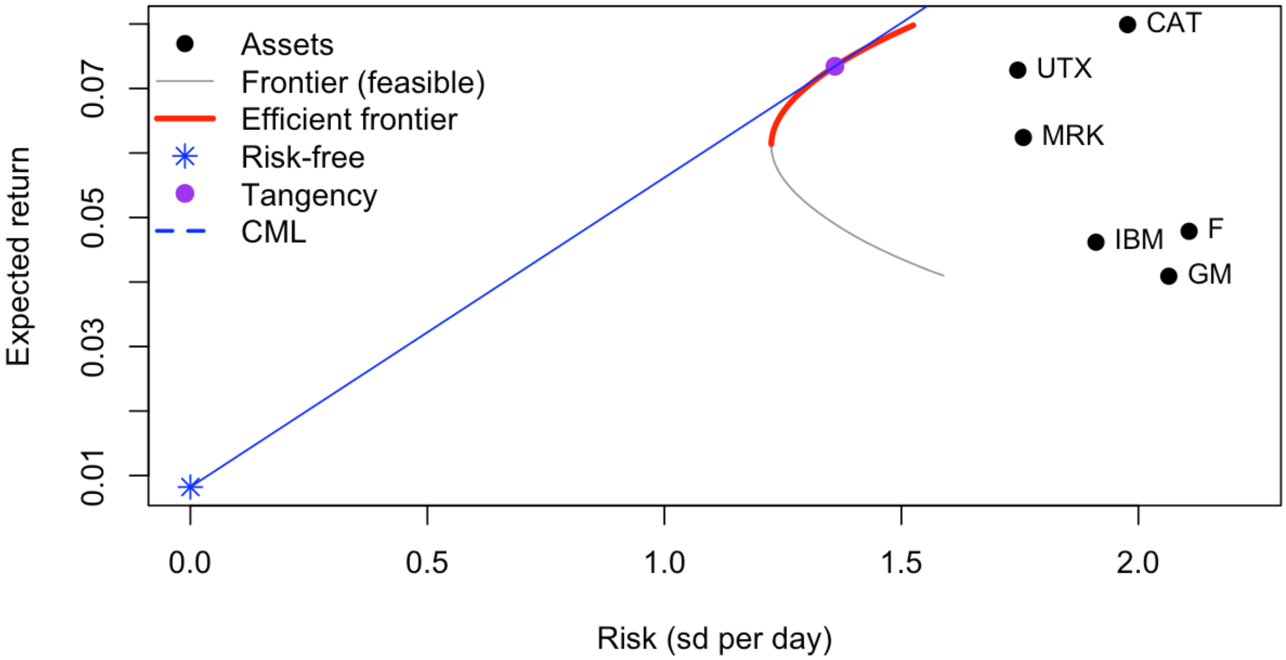


Figure 2: Expected return versus risk

1.2 Problem 2

If an investor wants an efficient portfolio with an expected daily return of 0.07%, how should the investor allocate his or her capital to the six stocks and to the risk-free asset? Assume that the investor wishes to use the tangency portfolio computed with the constraints $0.1 \leq w_j \leq 0.5$, not the unconstrained tangency portfolio.

Answer

Here we have:

```
target <- 0.07

y <- (target - rf) / (mu_tan - rf) # mix proportion in the tangency portfolio
w_final <- y * w_tan # weights in the 6 risky stocks
w_rf     <- 1 - y # weight in risk-free
sigma_final <- y * sd_tan # portfolio risk (sd, %/day)

cat("y␣=", round(y,7), "\n")
print(setNames(round(w_final,7), c("GM","F","CAT","UTX","MRK","IBM")))
cat("Risk-free␣weight␣=", round(w_rf,4), "\n")
cat("Target␣risk␣(sd,␣%/day)␣=", round(sigma_final,7), "\n")
```

The answer is:

```
y = 0.9472562
GM          F           CAT         UTX         MRK         IBM
-0.0872159 -0.0030514  0.3186343  0.3641956  0.3027479  0.0519458

Risk-free weight = 0.0527
```

We should buy each of six assets accordingly, and also need to buy 0.0527 of our portfolio of the risk-free asset.

1.3 Problem 3

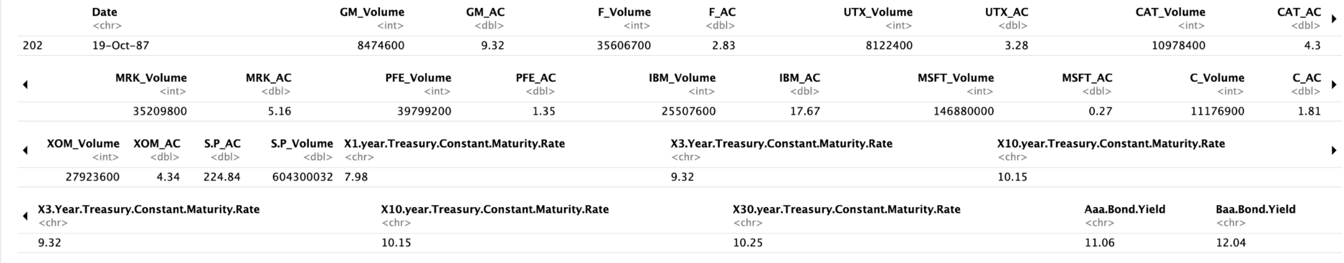
Does this data set include Black Monday?

Answer

Yes, it was included. Black Monday is October 19th 1987, we can check by selecting the date:

```
dat[ dat$Date == "19-Oct-87", ]
```

The print result is as below:



	Date <chr>	GM_Volume <int>	GM_AC <dbl>	F_Volume <int>	F_AC <dbl>	UTX_Volume <int>	UTX_AC <dbl>	CAT_Volume <int>	CAT_AC <dbl>
202	19-Oct-87	8474600	9.32	35606700	2.83	8122400	3.28	10978400	4.3

MRK_Volume <int>	MRK_AC <dbl>	PFE_Volume <int>	PFE_AC <dbl>	IBM_Volume <int>	IBM_AC <dbl>	MSFT_Volume <int>	MSFT_AC <dbl>	C_Volume <int>	C_AC <dbl>
35209800	5.16	39799200	1.35	25507600	17.67	146880000	0.27	11176900	1.81

XOM_Volume <int>	XOM_AC <dbl>	S.P_AC <dbl>	S.P_Volume <dbl>	X1.year.Treasury.Constant.Maturity.Rate <chr>	X3.Year.Treasury.Constant.Maturity.Rate <chr>	X10.year.Treasury.Constant.Maturity.Rate <chr>
27923600	4.34	224.84	604300032	7.98	9.32	10.15

X3.Year.Treasury.Constant.Maturity.Rate <chr>	X10.year.Treasury.Constant.Maturity.Rate <chr>	X30.year.Treasury.Constant.Maturity.Rate <chr>	Aaa.Bond.Yield <chr>	Baa.Bond.Yield <chr>
9.32	10.15	10.25	11.06	12.04

Figure 3: Row of Oct 19, 1987

To make more clear, we can also plot the graph of stock price during this period.

```
library(ggplot2)
library(tidyr)
library(readr)

dat_500 <- head(dat, 500)

# Select AC for 6 stocks
dat_prices <- dat_500[, c("Date", "GM_AC", "F_AC", "UTX_AC", "CAT_AC", "MRK_AC", "IBM_AC")]

# pivot longer
dat_long <- pivot_longer(dat_prices,
cols = -Date,
names_to = "Stock",
values_to = "Price")

dat_long$Date <- as.Date(dat_long$Date, format = "%d-%b-%y")
```

```

    ggplot(dat_long, aes(x = Date, y = Price, color = Stock)) +
    geom_line(size = 1) +
    labs(title = "Stock␣Price␣Changes␣1987-89",
    x = "Date", y = "Adjusted␣Closing␣Price") +
    theme_minimal()
```

We can see that there is a sharp decrease on October 19th, 1987, which is known as the Black Monday. On that day, there was a severe stock market crash, the worldwide losses were estimated at $1.71 trillion.

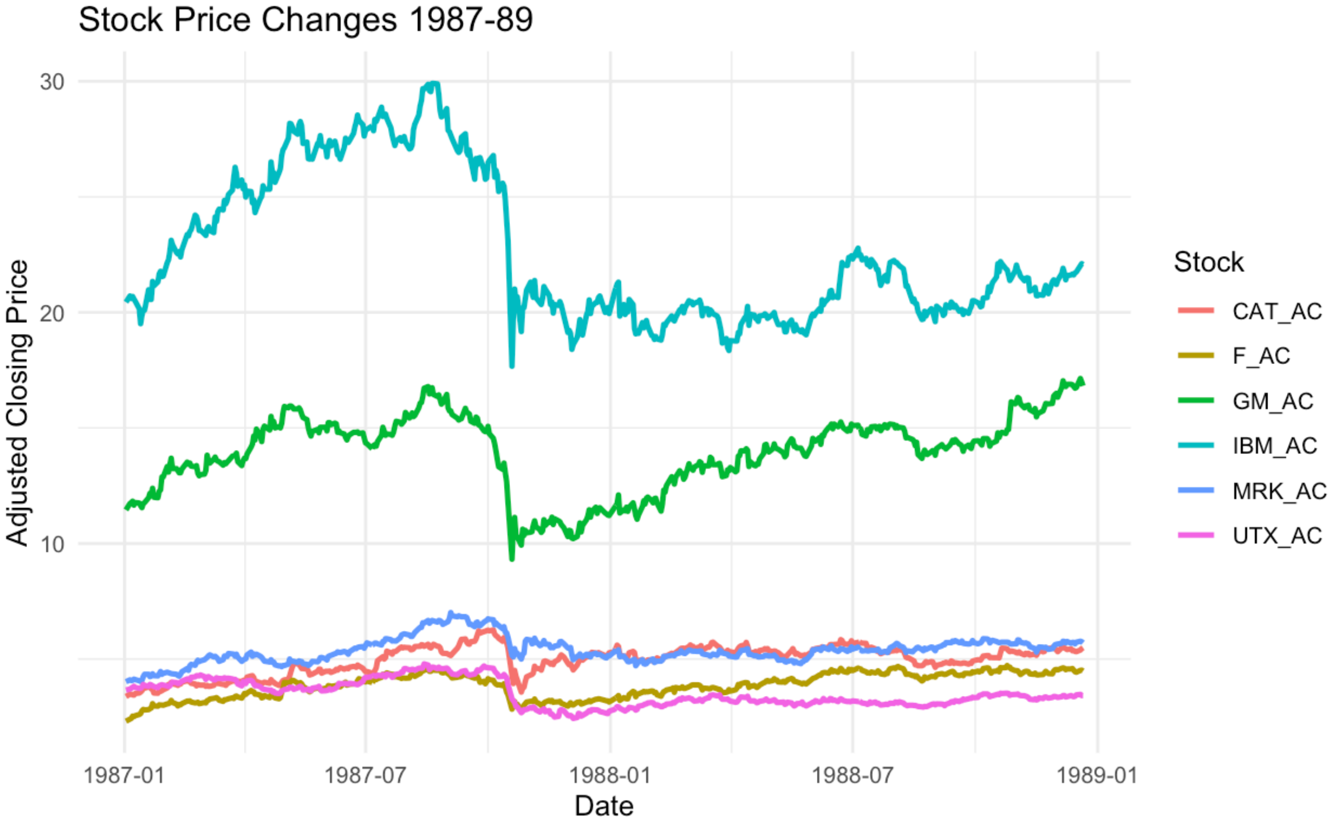


Figure 4: Stock price over time

2 Question 2

Suppose a firm is planning to invest \$1,000,000 in a combination of a risk-free asset and a risky asset A. Assume:

$$\mu_f = 5\%, \quad \mu_A = 12\%, \quad \sigma_A = 25\%$$

The company has capital reserves to cover losses up to \$200,000, and they want the probability of losing this amount or more to be at most 0.01 .

If the return of the portfolio is:

$$R = \omega\mu_A + (1-\omega)\mu_f$$

and the return is normally distributed, find the value of ω that satisfies the risk requirement.

Answer

Same as the mechanism of Example 16.2 in book, we have:

Here the portfolio return ($R \sim N(\mu_R, \sigma_R^2)$) with

$$\mu_R = \omega\mu_A + (1-\omega)\mu_f = 0.05 + 0.07, \omega, \qquad \sigma_R = \omega\sigma_A = 0.25, \omega.$$

Losing \$200,000 or more on \$1,000,000 means ($R \le -0.20$), then we have (in textbook $\Phi(0.01) = 2.33$, but the more precise estimation is 2.3263, which I used here):

$$\Phi!\left(\frac{-0.20-\mu_R}{\sigma_R}\right) \le 0.01; \Longrightarrow; \frac{-0.20-\mu_R}{\sigma_R} \le z_{0.01} = -2.3263.$$

Thus we got:

$$\mu_R; \ge; -0.20 + 2.3263, \sigma_R; \Longrightarrow; 0.05 + 0.07\omega; \ge; -0.20 + 2.3263(0.25\omega).$$

Solve for (ω):

$$0.25 \ge (0.581575 - 0.07)\omega; \Rightarrow; \omega \le \frac{0.25}{0.511575} \approx 0.4887.$$

So the largest risky weight satisfying the 0.01 loss constraint is:

$$\omega \approx 0.489.$$

3 Question 3

The table below gives sample statistics (monthly means, standard deviations, and covariances) for returns on Microsoft, Nordstrom, and Starbucks over the period January 1995 to January 2000.

Asset	μ_i	σ_i	Covariances
A (Microsoft)	0.0427	0.1000	$\sigma_{AB} = 0.0018$
B (Nordstrom)	0.0015	0.1044	$\sigma_{AC} = 0.0011$
C (Starbucks)	0.0285	0.1411	$\sigma_{BC} = 0.0026$

- (a) (4pt) Find the global minimum variance portfolio. What is its mean? What is its variance?
- (b) (4pt) Find the efficient portfolio of these assets with the same expected return as Microsoft. What is its risk?
- (c) (4pt) Assume a risk-free rate of 0.0001 per month (T-bill). What are the weights of the tangency portfolio?
- (d) (3pt) Find the portfolio made up of the risky assets and the risk-free asset that has the same expected return as Microsoft. What is its risk equal to?

Answer

3.1 Part a

Use Lagrange multipliers to find the global minimum variance portfolio (MVP)

$$\min_{w \in \mathbb{R}^n} \ \tfrac{1}{2}\, w^\top \Sigma w \quad \text{s.t. } \mathbf{1}^\top w = 1,$$

where Σ is the covariance matrix, $\mathbf{1} = (1, \dots, 1)^\top$.

$$\mathcal{L}(w, \lambda) = \tfrac{1}{2}\, w^\top \Sigma w - \lambda(\mathbf{1}^\top w - 1).$$

First-order conditions:

$$\nabla_w \mathcal{L} = \Sigma w - \lambda \mathbf{1} = 0, \qquad \mathbf{1}^\top w = 1.$$

Hence $w = \lambda\, \Sigma^{-1}\mathbf{1}$ and

$$1 = \mathbf{1}^\top w = \lambda\, \mathbf{1}^\top \Sigma^{-1} \mathbf{1} \implies \lambda = \frac{1}{\mathbf{1}^\top \Sigma^{-1} \mathbf{1}}.$$

Therefore we have:

$$w_{\text{GMV}} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^\top \Sigma^{-1} \mathbf{1}}, \qquad \sigma^2_{\text{GMV}} = \frac{1}{\mathbf{1}^\top \Sigma^{-1} \mathbf{1}}, \qquad \mu_{\text{GMV}} = w_{\text{GMV}}^\top \mu\,.$$

In this case, we have:

$$\Sigma = \begin{bmatrix} 0.01000000 & 0.00180000 & 0.00110000 \\ 0.00180000 & 0.01089936 & 0.00260000 \\ 0.00110000 & 0.00260000 & 0.01990921 \end{bmatrix}, \quad \mu = \begin{bmatrix} 0.0427 \\ 0.0015 \\ 0.0285 \end{bmatrix}.$$

Then

$$w_{\text{MVP}} \approx \begin{bmatrix} 0.44113 \\ 0.36569 \\ 0.19318 \end{bmatrix}, \qquad \mu_{\text{MVP}} \approx 0.02489, \qquad \sigma_{\text{MVP}} \approx 0.07268 \ (\sigma^2 \approx 0.005282).$$

3.2 Part b

To find the efficient portfolio with target mean equal to Microsoft's mean

$$\mu = \begin{bmatrix} 0.0427 \\ 0.0015 \\ 0.0285 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.01000000 & 0.00180000 & 0.00110000 \\ 0.00180000 & 0.01089936 & 0.00260000 \\ 0.00110000 & 0.00260000 & 0.01990921 \end{bmatrix}, \quad \mu_P = \mu_A = 0.0427, \ \mathbf{1} = (1, 1, 1)^\top.$$

Let:

$$a = \mathbf{1}^\top \Sigma^{-1} \mathbf{1}, \quad b = \mathbf{1}^\top \Sigma^{-1} \mu, \quad c = \mu^\top \Sigma^{-1} \mu.$$

Make the least variance of the targeted profit be:

$$w(\mu_P) = \lambda\,\Sigma^{-1}\mathbf{1} + \gamma\,\Sigma^{-1}\mu, \qquad \begin{bmatrix}\lambda\\ \gamma\end{bmatrix} = \frac{1}{ac-b^2}\begin{bmatrix}c & -b\\ -b & a\end{bmatrix}\begin{bmatrix}1\\ \mu_P\end{bmatrix}.$$

Then we have:

$$w^* = \begin{bmatrix}0.8275\\ -0.0907\\ 0.2632\end{bmatrix} \quad \text{(A,B,C)}, \qquad \mu(w^*) = \mu_P = 0.0427.$$

Its risk is:

$$\sigma(w^*) = \sqrt{{w^*}^\top \Sigma\, w^*} \;\approx\; 0.09166 \;\; (9.166\%),$$

The variance is:

$$\sigma^2(w^*) \approx 0.008401.$$

3.3 Part c

Let the mean vector be $\mu \in \mathbb{R}^3$, covariance matrix $\Sigma \in \mathbb{R}^{3\times 3}$ (symmetric positive definite), and riskfree rate r_f. Define the excessreturn vector

$$x := \mu - r_f\,\mathbf{1}, \qquad \mathbf{1} = (1,1,1)^\top.$$

The tangency portfolio maximizes the Sharpe ratio

$$S(w) = \frac{x^\top w}{\sqrt{w^\top \Sigma w}} \quad \text{(scale-invariant)}.$$

By Cauchy–Schwarz in the Σ-inner product,

$$(x^\top w)^2 \le (x^\top \Sigma^{-1} x)\,(w^\top \Sigma w),$$

with equality iff $w \propto \Sigma^{-1}x$. Imposing the budget constraint $\mathbf{1}^\top w = 1$ gives

$$w^\star = \frac{\Sigma^{-1}x}{\mathbf{1}^\top \Sigma^{-1}x} = \frac{\Sigma^{-1}(\mu - r_f\mathbf{1})}{\mathbf{1}^\top\Sigma^{-1}(\mu - r_f\mathbf{1})}.$$

With

$$\mu = \begin{bmatrix}0.0427\\ 0.0015\\ 0.0285\end{bmatrix}, \quad r_f = 0.0001, \quad x = \begin{bmatrix}0.0426\\ 0.0014\\ 0.0284\end{bmatrix},$$

and the given

$$\Sigma = \begin{bmatrix}0.01000000 & 0.00180000 & 0.00110000\\ 0.00180000 & 0.01089936 & 0.00260000\\ 0.00110000 & 0.00260000 & 0.01990921\end{bmatrix},$$

one finds

$$v := \Sigma^{-1}x \approx \begin{bmatrix}4.27639\\ -0.88941\\ 1.30635\end{bmatrix}, \qquad s := \mathbf{1}^\top v \approx 4.69333,$$

hence we have the weights:

$$w^\star = \frac{v}{s} \approx \begin{bmatrix}0.9112\\ -0.1895\\ 0.2783\end{bmatrix}.$$

3.4 Part d

$$\mu = \begin{bmatrix} 0.0427 \\ 0.0015 \\ 0.0285 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.01000000 & 0.00180000 & 0.00110000 \\ 0.00180000 & 0.01089936 & 0.00260000 \\ 0.00110000 & 0.00260000 & 0.01990921 \end{bmatrix}, \quad r_f = 0.0001.$$

Firstly we need to get the tangency portfolio, with $x = \mu - r_f \mathbf{1}$,

$$w^\star = \frac{\Sigma^{-1} x}{\mathbf{1}^\top \Sigma^{-1} x} \approx \begin{bmatrix} 0.911163 \\ -0.189505 \\ 0.278342 \end{bmatrix}, \qquad \mu_T = \mu^\top w^\star \approx 0.046555, \quad \sigma_T = \sqrt{w^{\star\top} \Sigma\, w^\star} \approx 0.099489.$$

On the CML, take

$$y = \frac{\mu_A - r_f}{\mu_T - r_f} = \frac{0.0427 - 0.0001}{0.046555 - 0.0001} \approx 0.917013,$$

i.e. invest y in $w^\star$ and $1 - y$ in the riskfree asset. Thus the weights are

$$w_{\text{risky}} = y\, w^\star \approx \begin{bmatrix} 0.835549 \\ -0.173779 \\ 0.255243 \end{bmatrix}, \qquad w_f = 1 - y \approx 0.082987.$$

Then we have the risk (also standard deviation) as

$$\sigma_{\text{portfolio}} = y\, \sigma_T \approx 0.917013 \times 0.099489 \approx 0.091233 \quad \text{(about 9.12\% per month).}$$