Student Name: LI JIAYI
Student ID: 1155188890

RMSC 5101 Financial Modeling(2)

Asset allocation

---

1. **Risk and Return**

   The trade-off is actually between risk and *expected return*, not between risk and actual return.

   Denote the return of a stock r is a Random Variable.

   Table 1: One period return with J states

Probability	Return
p_1	r_1
p_2	r_2
$\vdots$	$\vdots$
p_J	r_J

   Theoretically, *expected return* is when we look ahead one period.

   $$\mathbb{E}[r] = \sum_{j=1}^{J} p_j r_j$$

   In practice, we estimate the expected return using the past n period observations.

   $$\hat{\mathbb{E}}[r] = \frac{1}{n}\sum_{t=1}^{n} r_t$$

   Risk is measured by standard deviation σ of the return over one period.

   $$\sigma = \sqrt{\mathbb{E}[r^2] - \mathbb{E}^2[r]}$$

   Now consider invest 2 stocks with weight w_1 and w_2

   $$r_p = w_1 r_1 + w_2 r_2$$

   denote $\mu_i = \mathbb{E}[r_i]$

   $$\begin{aligned} \mathbb{E}[r_p] &= w_1\mathbb{E}[r_1] + w_2\mathbb{E}[r_2] \\ \Rightarrow \mu_p &= w_1\mu_1 + w_2\mu_2 \end{aligned}$$

   denote $\sigma_i^2 = Var(r_i)$

   $$\begin{aligned} Var(r_i) &= w_1^2 Var(r_1) + w_2^2 Var(r_2) + 2w_1 w_2 Cov(r_1, r_2) \\ \Rightarrow \sigma_p^2 &= w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2\rho_{12} w_1 w_2 \sigma_1 \sigma_2 \\ \sigma_p &= \sqrt{w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2\rho_{12} w_1 w_2 \sigma_1 \sigma_2} \end{aligned}$$

where $\rho_{12} = \dfrac{Cov(r_1, r_2)}{\sqrt{Var(r_1)Var(r_2)}}$

Demonstrate the μ_p and σ_p relationship with numerical example. $\rho_{12} = 0$ in this example.

Time t	**Statistics**	
Assets	$\mathbb{E}[\cdot]$	$\sqrt{Var(\cdot)}$
1	$\mu_1 = 0.05$	$\sigma_1 = 0.15$
2	$\mu_2 = 0.10$	$\sigma_2 = 0.20$

The smallest standard deviation is: 0.120
The corresponding portfolio weights are: 0.667 for Asset 1 and 0.333 for Asset 2
The corresponding expected portfolio return is: 0.067

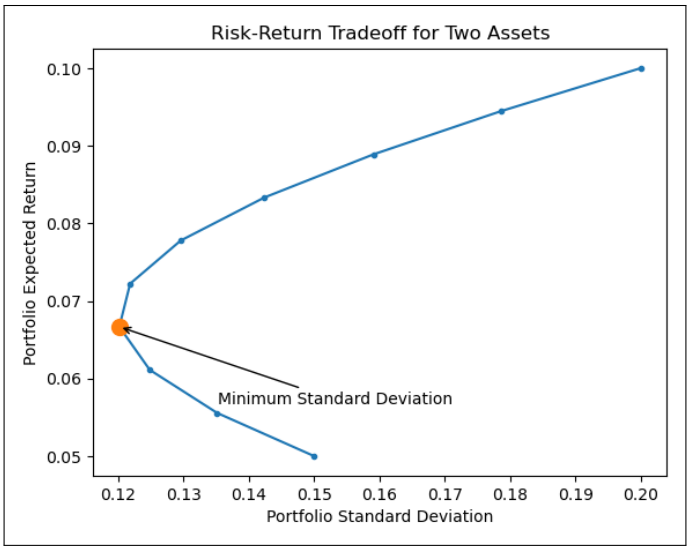


Figure 1: Risk and Return with 2 stocks

Most investors are risk-averse. They want to increase expected return while reducing the standard deviation of return. This means that they want to move as far as they can in a "northwest" direction in Figure 1.

2. **Modern Portfolio Theory**

Consider a portfolio with N stocks. Let r_i represent the individual returns and w_i their weights of each stock in the portfolio.

$$\boldsymbol{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

Denote r_p the portfolio return

$$r_p = \sum_{i=1}^{N} w_i r_i = \boldsymbol{w}^\top \boldsymbol{r}$$

Let $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{r}]$ and $\boldsymbol{\Sigma} = Var(\boldsymbol{r})$

$$\mathbb{E}[r_p] = \sum_{i=1}^{N} w_i \mathbb{E}[r_i] = \boldsymbol{w}^\top \boldsymbol{\mu}$$
$$Var(r_p) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij} = \boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w}$$

where σ_{ij} is the $(i,j)^{th}$ element of covariance matrix $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N1} & \sigma_{N2} & \cdots & \sigma_{NN} \end{bmatrix} \in \mathbb{R}^{N \times N}$$

Demonstrate with equal weighted portfolio

Ticker(.HK)	Stock name	weight
0388	HKEX	0.2
1211	BYD	0.2
1810	Xiaomi	0.2
3690	Meituan	0.2
6862	Haidilao	0.2

Created snapshot consists of cumulative portfolio returns, the underwater plot depicting the drawdown periods and daily returns.

(a) *Cumulative returns plot:* : presents the evolution of the portfolio's worth over time.

(b) *Underwater plot*: presents the investment from a pessimistic point of view, as it focuses on losses. It plots all the drawdown periods and how long they lasted until the value rebounded to a new high.

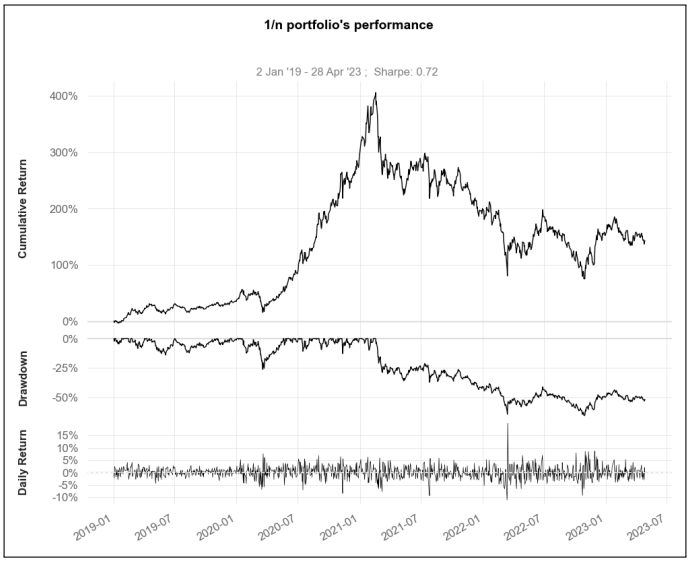


Figure 2: Evaluation metrics of the $1/n$ portfolio

3. **Markowitz Model**

According to the Modern Portfolio Theory, the Markowitz Model is to find the portfolio with a minimised variance given a pre-specified return μ.

Minimise $\frac{1}{2}Var(r_p)$ to facilitate the calculation

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \tfrac{1}{2}\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} \\ \text{s.t.} \quad & \boldsymbol{w}^\top \boldsymbol{\mu} = \mu \\ & \boldsymbol{w}^\top \mathbf{1} = 1 \end{aligned} \tag{1}$$

When there are only equality constraints and $\boldsymbol{\Sigma}$ is positive semidefinite, the solution (weights $\boldsymbol{w}$ of the minimum variance portfolio) to this constrained optimisation problem can be found using Lagrange multipliers.

Lagrange Multipliers

Single constraint In order to find the maximum or minimum of a function $f(x)$ subjected to the equality constraint $g(x)$

$$\mathcal{L}(x, \lambda) \triangleq f(x) + \lambda g(x)$$

find the stationary points of $\mathcal{L}$ by setting all partial derivatives to zero

$$\frac{\partial \mathcal{L}}{\partial x} = 0, \quad \frac{\partial \mathcal{L}}{\partial \lambda} = 0$$

Multiple constraints

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \triangleq f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{g}(\boldsymbol{x})$$

solve the linear equations

$$\begin{aligned} \nabla f(\boldsymbol{x}) - \boldsymbol{\lambda}^\top \nabla \boldsymbol{g}(\boldsymbol{x}) &= \mathbf{0} \\ \boldsymbol{g}(\boldsymbol{x}) &= 0 \end{aligned}$$

Rewrite the equation 1 with Lagrange multipliers setting

$$f(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w}, \quad \boldsymbol{g}(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{w}^\top \boldsymbol{\mu} - \mu \\ \boldsymbol{w}^\top \mathbf{1} - 1 \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

Solve the Minimization problem

$$\begin{aligned} \boldsymbol{\Sigma} \boldsymbol{w} - \lambda_1 \boldsymbol{\mu} - \lambda_2 \mathbf{1} &= \mathbf{0} \\ \boldsymbol{w}^\top \boldsymbol{\mu} &= \mu \\ \boldsymbol{w}^\top \mathbf{1} &= 1 \end{aligned}$$

also can be rewriten as

$$\begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\mu} & \mathbf{1} \\ \boldsymbol{\mu}^\top & 0 & 0 \\ \mathbf{1}^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ -\lambda_1 \\ -\lambda_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mu \\ 1 \end{bmatrix}$$

where $\boldsymbol{w}$, λ_1, λ_2 are unknowns.

4. **Efficient Frontier**

Efficient Set Theorem: An investor will choose a portfolio from the set of portfolios that

(a) (Non-satiation) offers maximum levels of expected return for varying levels of risk.

(b) (Risk aversion) offers minimum levels of risk for varying levels of expected return.

The minimum variance set includes all the portfolios satisfy the second condition of risk aversion. The efficient frontier (or efficient set) includes all the portfolios that satisfy the two conditions above. The minimum variance point (global minimum variance portfolio) represents the portfolio with the smallest variance in the feasible set.

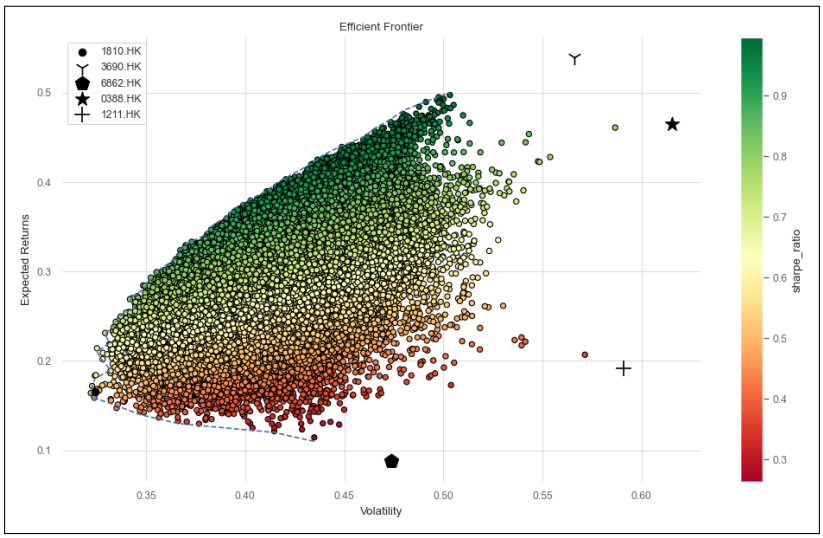


Figure 3: Efficient frontier identified using Monte Carlo simulations

Ideally, we should search for a portfolio offering exceptional returns but with a combined standard deviation that is lower than the standard deviations of the individual assets. For example, we should not consider a portfolio consisting only of Haidilao(6862.HK) stock (it is not efficient), but the one that lies on the frontier directly above. That is because the latter offers a much better expected return for the same level of expected volatility.

5. **Capital Market Line and Tangency Portfolio**

Two-Fund Theorem: any efficient fund (or portfolio) can be replicated, in terms of mean and variance, as a combination of two efficient funds. Mathematically, if $\boldsymbol{w}^1$ and $\boldsymbol{w}^2$ are (the weights of) two efficient portfolios, then $\tilde{\boldsymbol{w}} = \alpha \boldsymbol{w}^1 + (1-\alpha)\boldsymbol{w}^2$ is also form an efficient portfolio.

Now consider the include the risk-free asset with r_f and $\sigma = 0$ and the previous portfolio with mean μ_M and risk σ_M.

$$r_p = \alpha r_f + (1-\alpha) r_M$$

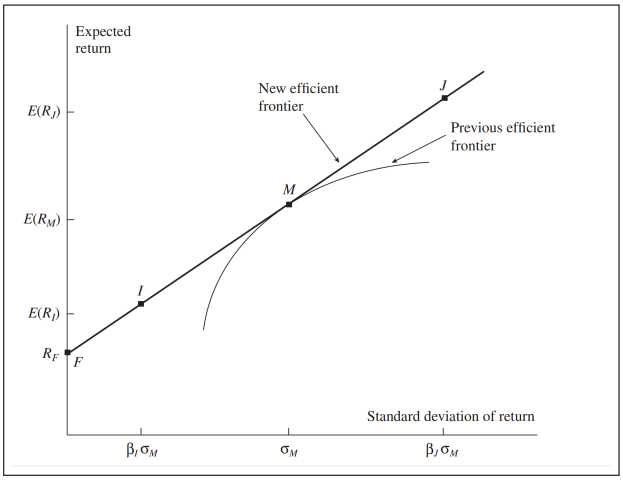
The mean return and variance of the new portfolio are

$$\begin{aligned} \mu_p &= \alpha r_f + (1-\alpha)\mu_M \\ \sigma_p^2 &= (1-\alpha)^2 \sigma_M^2 \end{aligned}$$

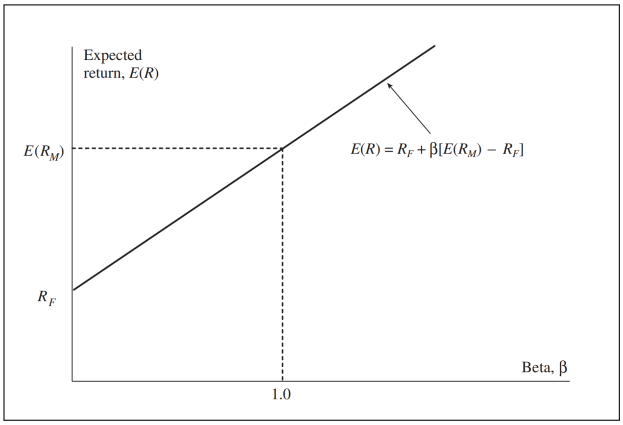
From the latter equation $\alpha = 1 - \frac{\sigma_p}{\sigma_M}$, substitute into former one

$$\begin{aligned} \mu_p &= \Big(1 - \frac{\sigma_p}{\sigma_M}\Big) r_f + \frac{\sigma_p}{\sigma_M}\mu_M \\ &= r_f + \frac{\sigma_p}{\sigma_M}\Big(\mu_M - r_f\Big) \\ \mu_p &= r_f + \Big(\frac{\mu_M - r_f}{\sigma_M}\Big)\sigma_p \end{aligned}$$

One-Fund Theorem: There is a single fund M of risky assets, called the Market(Tangency) portfolio such that any efficient portfolio can be constructed as a combination of this fund M and a risk-free asset.



(a) CML



(b) SML

Figure 4: CML & SML

The tangent portfolio is the one that maximizes θ or $tan(\theta)$.

$$tan(\theta) = \frac{\mu_p - r_f}{\sigma_p} = \frac{\boldsymbol{w}^\top \Big(\boldsymbol{\mu} - \boldsymbol{r_f}\Big)}{\Big(\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w}\Big)^{\frac{1}{2}}}$$

Formulate the problem

$$\begin{aligned} \max_{\boldsymbol{w}} \quad & tan(\theta) \\ \text{s.t.} \quad & \boldsymbol{w}^\top \mathbf{1} = 1 \end{aligned} \tag{2}$$

6. **Capital Asset Pricing Model** (CAPM)

Investors decide on the expected returns they require for individual investments based on the market portfolio. A common procedure is to use historical data and regression to determine a best-fit linear relationship between returns from an investment r and returns from the market portfolio r_M.

$$r = \alpha + \beta r_M + \varepsilon \tag{3}$$

Based on the Figure 4, $F = (0, r_f)$, $M = (\sigma_M, \mu_M)$

$$\mathbb{E}[r] = r_f + \beta\Big(\mathbb{E}[r_M] - r_f\Big) \tag{4}$$

Now suppose the portfolio manager invest in stock with realized return r. During the same period the market yield the realized return r_M. Define α as superior return for the amount of systematic risk being taken.

$$\alpha = r - r_f - \beta(r_M - r_f)$$

Assumptions

(a) We assumed that investors care only about the expected return and the standard deviation of return of their portfolio. Another way of saying this is that investors look only at the first two moments of the return distribution. If returns are normally , it is reasonable for investors to do this. However, the returns from many assets are non-normal. They have skewness and excess kurtosis.

(b) We assumed that the ε variables for different investments in equation are independent. Equivalently we assumed the returns from investments are correlated with each other only because of their correlation with the market portfolio. However stocks in the same industry share the correlation.

(c) We assumed that investors focus on returns over just one period and the length of this period is the same for all investors. This is also clearly not true. Some investors such as pension funds have very long time horizons. Others such as day traders have very short time horizons.

(d) We assumed that investors can borrow and lend at the same risk-free rate. This is approximately true in normal market conditions for a large financial institution that has a good credit rating. But it is not true for small investors.

(e) We did not consider tax. In some jurisdictions, capital gains are taxed differently from dividends and other sources of income. Some investments get special tax treatment and not all investors are subject to the same tax rate.

(f) we assumed that investors have *homogeneous expectations* that all investors make the same estimates of expected returns, standard deviations of returns, and correlations between returns for available investments. If we lived in a world of homogeneous expectations there would be no trading.

7. **Arbitrage Pricing Theory** (APT)

Model 1: Single Factor

$$r_i = a_i + b_i F \tag{5}$$

To form an arbitrage portfolio with weight w_i on asset i, the following three conditions are required:

(a) *Self-financing*: The portfolio is without any exogenous infusion or withdrawal of money. The purchase of a new asset must be financed by the sale of the old ones.

$$\sum_{i=1}^{n} w_i = 0$$

(b) *Risk-free*: the only risk of any asset comes from the factor F. To be risk-free, the sensitivity of the portfolio to the factor has to be equal to zero, that is, $b_p = 0$

$$\sum_{i=1}^{n} w_i b_i = 0$$

(c) *Positive Return*: The arbitrage portfolio has to earn a positive return

$$\sum_{i=1}^{n} w_i \mathbb{E}[r_i] > 0$$

Model 2: Multi-factor Model

Suppose that there are n assets whose returns are governed by m factors, with $m < n$

$$r_i = a_i + \sum_{j=1}^{m} b_{ij} F_j \tag{6}$$

and there $\exists \lambda_j$ for $j \in [0, m]$ s.t.

$$\mu_i = \mathbb{E}[r_i] = \lambda_0 + \sum_{j=1}^{m} b_{ij}\lambda_j$$

Now consider there exist error term ε_i

$$r_i = a_i + \sum_{j=1}^{m} b_{ij}F_j + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2_{\varepsilon_i})$$

for the whole portfolio r_p

$$r_p = \sum_{i=1}^{n} w_i r_i = a_p + \sum_{j=1}^{m} b_{pj}F_j + \varepsilon_p$$

Equal weighted portfolio r_p

Suppose that $w_i = \frac{1}{n}$ and $max_i\, \sigma^2_{\varepsilon_i} = C < \infty$

$$\sigma^2_{\varepsilon_p} = \sum_{i=1}^{n} w_i^2 \sigma^2_{\varepsilon_i} \leq \frac{1}{n^2} \sum_{i=1}^{n} C = \frac{C}{n}$$

$$\Rightarrow \lim_{n\to\infty} \sigma^2_{\varepsilon_p} = 0$$

As a result, when the number of assets n is sufficiently large, the Multi-factor Model r_p is get rid of the error term ε_p

8. **Appendix**

Code for Figure 1: : Risk and Return with 2 stocks

```
import numpy as np
import matplotlib.pyplot as plt

# Define the expected returns and standard deviations for two assets
exp_return_1 = 0.05
exp_return_2 = 0.10
std_dev_1 = 0.15
std_dev_2 = 0.20

# Generate a range of possible portfolio weights for the two assets
w = np.linspace(0, 1, 10)

# Calculate the expected portfolio return and standard deviation for each weight combinati
port_return = w * exp_return_1 + (1 - w) * exp_return_2
port_std_dev = np.sqrt((w * std_dev_1)**2 + ((1 - w) * std_dev_2)**2)

# Find the smallest standard deviation and the corresponding portfolio weights and expecte
min_std_dev = np.min(port_std_dev)
min_std_dev_idx = np.argmin(port_std_dev)
min_std_dev_weight_1 = w[min_std_dev_idx]
min_std_dev_weight_2 = 1 - min_std_dev_weight_1
min_std_dev_return = port_return[min_std_dev_idx]
```

```
# Plot the risk-return tradeoff for the two assets
plt.plot(port_std_dev, port_return, marker = ".")
plt.plot(min_std_dev, min_std_dev_return, 'o', markersize=10)
plt.annotate('Minimum Standard Deviation', xy=(min_std_dev, min_std_dev_return),
             xytext=(min_std_dev + 0.015, min_std_dev_return - 0.01),
             arrowprops=dict(facecolor='black', arrowstyle='->'))
plt.title('Risk-Return Tradeoff for Two Assets')
plt.xlabel('Portfolio Standard Deviation')
plt.ylabel('Portfolio Expected Return')
plt.show()

print('The smallest standard deviation is:', f'{min_std_dev:.3f}')
print('The corresponding portfolio weights are:', f'{min_std_dev_weight_1:.3f}',
'for Asset 1 and', f'{min_std_dev_weight_2:.3f}', 'for Asset 2')
print('The corresponding expected portfolio return is:', f'{min_std_dev_return:.3f}')
```

Code for Figure 2: Evaluation metrics

```
import pandas as pd
import yfinance as yf
import quantstats as qs

ASSETS = ["1810.HK", "3690.HK", "6862.HK", "0388.HK", "1211.HK"]
START_DATE = "2019-01-01"
END_DATE = "2023-05-01"

#Define the weights
n_assets = len(ASSETS)
portfolio_weights = n_assets * [1 / n_assets]

#Download the assets price infomation
asset_df = yf.download(ASSETS,
                       start=START_DATE,
                       end=END_DATE,
                       progress=False)

#Calculate the assets return matrix
returns = asset_df["Adj Close"].pct_change().dropna()
#Calculate the portfolio return
portfolio_returns = pd.Series(
    np.dot(portfolio_weights, returns.T),
    index=returns.index
)

#Generate the performance plot
qs.plots.snapshot(portfolio_returns,
                  title="1/n portfolio's performance",
                  grayscale=True)
```

Code for Figure 3: Efficient frontier using Monte Carlo simulations

```
#Set up the simulation numbers
N_PORTFOLIOS = 10 ** 5
N_DAYS = 252
avg_returns = returns.mean() * N_DAYS
cov_mat = returns.cov() * N_DAYS

np.random.seed(7)
#Generate the simulated weight matrix
weights = np.random.random(size=(N_PORTFOLIOS, n_assets))
#Normalize each row weight
#The newaxis() is to increase the dimensions of an array by adding new axes.
weights /=  np.sum(weights, axis=1)[:, np.newaxis]

#Calculate portfolio metrix
portf_rtns = np.dot(weights, avg_returns)

portf_vol = []
#The len() function returns rows number when pass a two-dimensional array
for i in range(0, len(weights)):
    vol = np.sqrt(
        np.dot(weights[i].T, np.dot(cov_mat, weights[i]))
    )
    portf_vol.append(vol)
portf_vol = np.array(portf_vol)

portf_sharpe_ratio = portf_rtns / portf_vol

#Create a pandas DataFrame to contain all the data
portf_results_df = pd.DataFrame(
    {"returns": portf_rtns,
     "volatility": portf_vol,
     "sharpe_ratio": portf_sharpe_ratio}
)

N_POINTS = 100

ef_rtn_list = []
ef_vol_list = []

possible_ef_rtns = np.linspace(
    portf_results_df["returns"].min(),
    portf_results_df["returns"].max(),
    N_POINTS
)
possible_ef_rtns = np.round(possible_ef_rtns, 2)
portf_rtns = np.round(portf_rtns, 2)

for rtn in possible_ef_rtns:
    if rtn in portf_rtns:
```

```
        ef_rtn_list.append(rtn)
        matched_ind = np.where(portf_rtns == rtn)
        ef_vol_list.append(np.min(portf_vol[matched_ind]))

#Locate the points creating the efficient frontier
N_POINTS = 100

ef_rtn_list = []
ef_vol_list = []

possible_ef_rtns = np.linspace(
    portf_results_df["returns"].min(),
    portf_results_df["returns"].max(),
    N_POINTS
)
possible_ef_rtns = np.round(possible_ef_rtns, 2)
portf_rtns = np.round(portf_rtns, 2)

for rtn in possible_ef_rtns:
    if rtn in portf_rtns:
        ef_rtn_list.append(rtn)
        matched_ind = np.where(portf_rtns == rtn)
        ef_vol_list.append(np.min(portf_vol[matched_ind]))

#Plot the efficient frontier
MARKERS = [".", "1", "p", "*", "+"]

with sns.plotting_context("paper"):
    fig, ax = plt.subplots()
    portf_results_df.plot(kind="scatter", x="volatility",
                          y="returns", c="sharpe_ratio",
                          cmap="RdYlGn", edgecolors="black",
                          ax=ax)
    ax.set(xlabel="Volatility",
           ylabel="Expected Returns",
           title="Efficient Frontier")
    ax.plot(ef_vol_list, ef_rtn_list, "b--")
    for asset_index in range(n_assets):
        ax.scatter(x=np.sqrt(cov_mat.iloc[asset_index, asset_index]),
                   y=avg_returns[asset_index],
                   marker=MARKERS[asset_index],
                   s=150, color="black",
                   label=ASSETS[asset_index])
    ax.legend()

    sns.despine()
    plt.tight_layout()
```