Security and Portfolio Performance Analysis

Elie Diwambuena,

July 7, 2023

Introduction

Nowadays, investing in stocks has become increasingly popular, particularly in developed nations. However, despite the growing interest in investing, many individuals still struggle to determine the most suitable approach. Typically, people tend to follow prevailing trends such as investing in cryptocurrency or high-growth technology stocks. Unfortunately, this strategy has proven to be suboptimal and risky. While we acknowledge that conducting comprehensive securities analysis is a skill possessed by only a few, this article aims to provide essential concepts that we believe every investor should be familiar with before selecting specific stocks or a group of stocks.

The objective of this article is to examine sets of securities (stocks) and determine the optimal portfolio consisting of technology and healthcare companies, while also comparing their performance. To achieve this, Section A focuses on analyzing the technology portfolio and optimizing it to identify the tangency portfolio. Similarly, Section B repeats the same analysis for the healthcare portfolio. In Section C, we compare the performance of these two portfolios. Throughout the analysis, we utilize Jupyter Notebook and Python, which involves the inclusion of relevant code snippets.

---

[1]:
```python
# Importing librairies
import yfinance as yf
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
import seaborn as sb
import random as rdm
import statsmodels.api as sm
```

[2]:
```python
# Loading data
data = yf.Tickers(['AAPL','TSLA','MSFT','NVDA','GOOGL']).
↪history(start="2018-06-30", end="2023-06-30").loc[:,"Close"]
```

```
[*********************100%***********************]  5 of 5 completed
```

A. Tech Portfolio Analysis

Let's consider our interest in five technology firms, namely Apple, Google, Microsoft, Nvidia, and Tesla. Before making any investments in these companies, it is important to address the following inquiries:

1. What has been the historical performance of these stocks in terms of earnings?
2. What can we anticipate in terms of future earnings from these stocks?
3. If we decide to invest in all five stocks, how should we distribute our funds among them?

Answering the first question is relatively straightforward; we can simply examine the average historical returns. However, the second and third questions necessitate a deeper understanding of security pricing models and portfolio optimization techniques. To estimate the expected future return of a stock, it is common to employ the Capital Asset Pricing Model (CAPM). According to the CAPM, the expected return can be determined using the following expression:

$$E_{r_A} = rf + \beta_A(r_M - rf)$$

In the CAPM formula, the expected return on security A (E_{r_A}) is determined by several factors. These factors include the risk-free rate (rf), the systematic risk of security A (β_A), and the market risk premium ($(r_M - rf)$). Essentially, the CAPM suggests that investors have the option to invest all their funds in a risk-free asset and earn a risk-free rate, rather than investing in a volatile asset like stocks. Therefore, if an investor chooses to invest in a risky asset, their return will depend on the asset's sensitivity to the overall market (measured by β_A) and the excess return they can potentially earn from the market ($r_M - rf$). Without the possibility of earning an excess return, there would be little incentive for an investor to invest in stocks.

Government bonds issued by countries such as the US or Germany are considered risk-free assets due to their low probability of default. The yield to maturity of these bonds is commonly used as the risk-free rate in calculations. Market indices like the S&P 500, Nasdaq, Euro Stoxx, DAX, and others serve as proxies for the overall market. Therefore, the average return on these indices is used as the market rate of return (r_M). It's important to note that the risk premium is typically estimated at around 6% empirically, although this figure can vary depending on the country or region. Damodaran is a reliable source for obtaining the components of the CAPM model and further information on this topic.(here).

The estimation of beta involves regressing the returns of a company (y) against the returns of the market (x), with beta representing the coefficient of x. However, there are also platforms like Google Finance or Yahoo Finance that provide pre-estimated beta values. Addressing the third question can be a somewhat arduous task, as it requires exploring various combinations of stocks to identify the optimal portfolio. Fortunately, leveraging the computational power of Python allows us to swiftly generate several combinations for evaluation. Selecting the optimal portfolio from these combinations is not a straightforward decision and heavily depends on an investor's risk tolerance. For instance, an investor who prioritizes risk avoidance may opt for the portfolio with the lowest risk, commonly referred to as the "global minimum variance portfolio." Conversely, some investors may prefer the portfolio that offers the highest return per unit of risk, known as the "optimal portfolio" or "tangency portfolio." The latter can be determined using the following ratio:

$$\text{Risk-adjusted return} = \frac{r_P}{\theta_P}$$

Here, r_P represents the portfolio return, and θ_P denotes the portfolio risk or volatility. By subtracting the risk-free rate from the portfolio return, we obtain the excess return from the portfolio.

Dividing this excess return by the portfolio risk yields the risk-adjusted excess return, often referred to as the "Sharpe ratio," expressed as:

$$\text{Sharpe ratio} = \frac{r_P - rf}{\theta_P}$$

Now, let's proceed with the actual analysis. We will begin by examining the historical return, expected return, volatility, and risk-adjusted return of each security.

---

1. Return and Risk Analysis

[3]:
```python
print(data.round(3).head(3))
print(data.round(3).tail(3))
```

```
              AAPL   GOOGL    MSFT    NVDA    TSLA
Date
2018-06-29  44.226  56.459  93.265  58.684  22.863
2018-07-02  44.721  57.105  94.589  60.007  22.338
2018-07-03  43.942  55.814  93.681  58.669  20.724
              AAPL   GOOGL    MSFT    NVDA    TSLA
Date
2023-06-27  188.06  118.33  334.57  418.76  250.21
2023-06-28  189.25  120.18  335.85  411.17  256.24
2023-06-29  189.59  119.10  335.05  408.22  257.50
```

[4]:
```python
# daily stock returns
d_returns = data.pct_change()
```

- **Historical return**

[5]:
```python
# average daily return in %
avg_dreturn = d_returns.mean() * 100
```

[6]:
```python
# annualized returnrs in %
avg_yreturn = avg_dreturn * 250
avg_yreturn
```

[6]:
```
AAPL     34.472812
GOOGL    19.916400
MSFT     30.277129
NVDA     52.848022
TSLA     70.113075
dtype: float64
```

[7]:
```python
# historical return
hr_df = pd.DataFrame(avg_yreturn).reset_index().rename(columns={"index":
 ↪"stock", 0:"return"})
```

- **Expected return: CAPM**

[8]:
```
# nasdaq 100(NDX),sp500,stoxx50 and nasdaq composite(IXIC)
Mkt = yf.Tickers(["^NDX","^IXIC","^GSPC","^STOXX50E", "^GDAXI"]).
 ↪history(start="2018-06-30", end="2023-06-30").loc[:,"Close"]
print(Mkt.tail(3))
```

```
[*********************100%***********************]  5 of 5 completed
                  ^GDAXI        ^GSPC         ^IXIC          ^NDX    ^STOXX50E
Date
2023-06-27  15846.860352  4378.410156  13555.669922  14945.910156  4305.259766
2023-06-28  15949.000000  4376.859863  13591.750000  14964.570312  4344.750000
2023-06-29  15946.719727  4396.439941  13591.330078  14939.950195  4354.689941
```

[9]:
```
# risk premium
rf = 2.61 # german 5 years bond
market_ret = round(Mkt.pct_change().mean()*250*100,3)
mkt_ret = market_ret.mean()
rp = round(mkt_ret - rf,3)
rp
```

[9]:
```
9.132
```

[10]:
```
# market sensitivity beta from Google finance
beta = {
    "AAPL":1.28,
    "GOOGL":1.06,
    "MSFT":0.91,
    "NVDA":1.74,
    "TSLA":2.04,
}
beta = pd.Series(beta,index=beta.keys(),name="beta")
```

[11]:
```
# capm return
capm_return = (rf + beta * rp).round(3)
capm_return
```

[11]:
```
AAPL     14.299
GOOGL    12.290
MSFT     10.920
NVDA     18.500
TSLA     21.239
Name: beta, dtype: float64
```

[12]:
```
# expected return
er_df = pd.DataFrame(capm_return).reset_index().rename(columns={"index":
 ↪"stock", "beta":"return"})
```

- **Risk Analysis**

[13]:
```
# daily stock volatility
std_dreturn = d_returns.std() * 100
std_dreturn
```

[13]:
```
AAPL     2.098939
GOOGL    2.013563
MSFT     1.966072
NVDA     3.378298
TSLA     4.192667
dtype: float64
```

[14]:
```
# annualized stock volatility
std_yreturn = std_dreturn * math.sqrt(250)
std_yreturn
```

[14]:
```
AAPL     33.187146
GOOGL    31.837231
MSFT     31.086334
NVDA     53.415581
TSLA     66.291882
dtype: float64
```

[15]:
```
# volatility
vol_df = pd.DataFrame(std_yreturn).reset_index().rename(columns={"index":
 ↪"stock", 0:"volatility"})
```

- **Risk-Adjusted Expected Return**

[16]:
```
raj = (capm_return/std_yreturn).round(3)
raj
```

[16]:
```
AAPL     0.431
GOOGL    0.386
MSFT     0.351
NVDA     0.346
TSLA     0.320
dtype: float64
```

[17]:
```
raj_df = pd.DataFrame(raj).reset_index().rename(columns={"index":"stock", 0:
 ↪"return"})
```

[18]:
```
# plot all graphs
fig, ax = plt.subplots(1,4,sharey=True, figsize=(15,5))
sb.barplot(ax=ax[3],data = raj_df,x="return", y="stock")
ax[3].set_title("Risk Adjusted Return of Stocks")
sb.barplot(ax=ax[2],data = er_df,x="return", y="stock")
```

```
ax[2].set_title("Expected Return of Stocks")
sb.barplot(ax=ax[1],data = hr_df,x="return", y="stock")
ax[1].set_title("Historical Return of Stocks")
sb.barplot(ax=ax[0],data = vol_df,x="volatility", y="stock")
ax[0].set_title("Volatility of Stocks")
plt.show()
```

Based on the analysis, the historical returns for the stocks from June 2018 to June 2023 are as follows: Tesla had the highest return at 70.11% per year, followed by Nvidia at 52.85% per year, Apple at 34.47% per year, Microsoft at 30.27% per year, and Google at 19.92% per year. These figures represent the average returns for investors who held these stocks during the specified period. In terms of risk, Tesla had the highest volatility at 66.29% per year, followed by Nvidia at 53.42%, Apple at 33.19%, Google at 31.84%, and Microsoft at 31.09%. This implies that Tesla investors experienced higher returns but also faced greater risks. Investors with a higher risk appetite may find this appealing, while those with a lower risk appetite may prefer Google or Microsoft stocks.

However, relying solely on historical return and volatility does not provide a complete understanding. Past performance does not guarantee future earnings, so it is necessary to estimate the expected return of the stocks based on the CAPM. In terms of expected return, Tesla had the highest at 21.24% per year, followed by Nvidia at 18.50%, Apple at 14.30%, Google at 12.29%, and Microsoft at 10.92%. It may be tempting to conclude that one should invest in Tesla due to its higher expected return. However, comparing stocks based solely on expected returns can be misleading. Despite Tesla's profitability, it carries higher risk, and it is debatable whether the return is commensurate with the risk. Therefore, it is important to assess how much return each stock yields per unit of risk. To do this, we compute the risk-adjusted return for each stock. In terms of risk-adjusted return, Apple ranks first with 0.43, followed by Google with 0.38, Microsoft with 0.35, Nvidia with 0.34, and Tesla with 0.32.

At this point, it becomes evident that Tesla has the lowest return per unit of risk among the five stocks. Notably, the ranking based on risk-adjusted returns contradicts the ranking based on expected returns. This is a crucial observation to consider when making investment decisions.

Nevertheless, all stocks show profitability with an expected rate of return of approximately 10% or higher. Therefore, an investor with a lower required rate of return may choose to invest in all of them. However, the allocation of funds across these stocks will be explored in the next section.

---

2. Portfolio Analysis

When an investor decides to allocate their funds across multiple assets, the return of the portfolio will be equal to the weighted average return of the individual assets. For example, if an investor decides to invest 20% of their funds in each stock, with an equal distribution of funds, the portfolio's expected return can be calculated as follows:

$$E_{r_P} = 20\% \cdot r_{Apple} + 20\% \cdot r_{Tesla} + 20\% \cdot r_{Google} + 20\% \cdot r_{Nvidia} + 20\% \cdot r_{Microsoft}$$

It is important to note that the sum of all the percentages should always equal 100%. Similarly, when an investor holds multiple stocks, the portfolio's risk is determined by the square of the sum of weighted standard deviations adjusted for correlation. For instance, if an investor holds stocks A and B, the portfolio's risk can be calculated as:

$$\theta_P = (W_A \cdot \theta_A + W_B \cdot \theta_B)^2 = (W_A \cdot \theta_A)^2 + (W_B \cdot \theta_B)^2 + 2W_A \cdot \theta_A \cdot W_B \cdot \theta_B \cdot \mathrm{Corr}(A, B)$$

Here, W represents the weight or percentage invested, θ denotes the standard deviation, and $\mathrm{Corr}(A, B)$ is the correlation between stocks A and B. The same formula is applicable when considering more than two stocks in the portfolio. Generally, portfolio risk tends to be lower than individual stock risk due to the benefits of diversification. When two stocks are perfectly positively correlated ($\mathrm{Corr}(A, B) = 1$), the portfolio variance is equal to the square of the sum of weighted standard deviations, indicating no diversification benefit. Conversely, when two stocks are perfectly negatively correlated ($\mathrm{Corr}(A, B) = -1$), the portfolio variance is equal to the square of the difference of the weighted standard deviations or $(W_A \cdot \theta_A - W_B \cdot \theta_B)^2$, which is equivalent to $(W_A \cdot \theta_A)^2 + (W_B \cdot \theta_B)^2 - 2W_A \cdot \theta_A \cdot W_B \cdot \theta_B \cdot \mathrm{Corr}(A, B)$ representing full diversification. When two stocks are uncorrelated ($\mathrm{Corr}(A, B) = 0$), the portfolio variance is the sum of the squared weighted variances or $(W_A \cdot \theta_A)^2 + (W_B \cdot \theta_B)^2$.

Introducing the correlation term allows us to adjust $(W_A \cdot \theta_A + W_B \cdot \theta_B)^2$ to accommodate those different scenarios. Now we will proceed with the analysis of portfolio return and volatility. Subsequently, we will optimize the portfolio using the Markowitz approach to determine the optimal allocation.

- **Portfolio return**

[19]:
```
# stock weights
weights = {
    "APPL": 0.20,
    "GOOGL":0.20,
    "MSFT": 0.20,
    "NVDA": 0.20,
    "TSLA": 0.20,
}
```

[20]:
```python
weights = pd.Series(weights.values(), index=weights.keys(),name="weights")
```

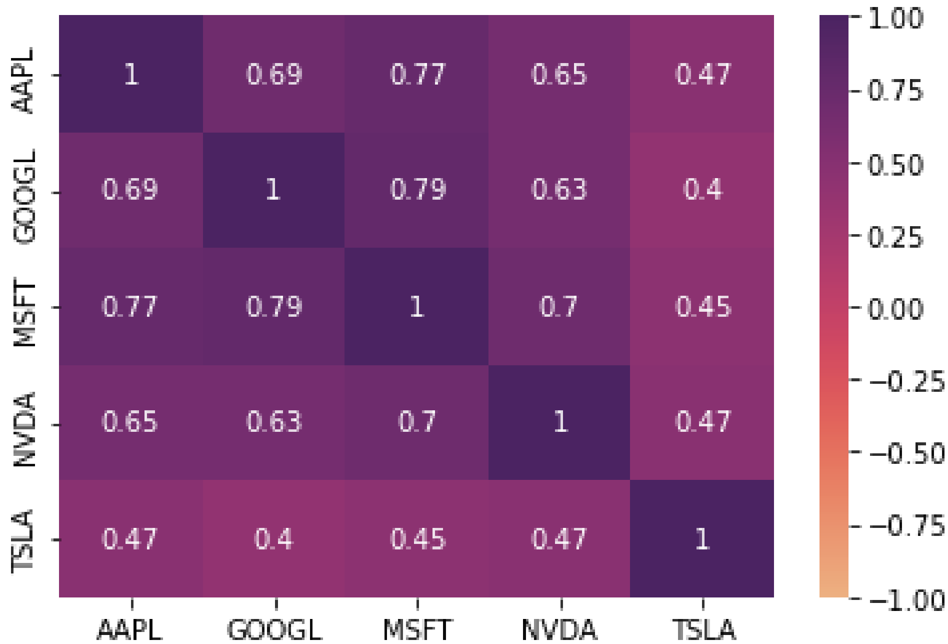
[21]:
```python
ptf_Eyreturn = round(np.dot(capm_return, weights),3)
ptf_Eyreturn
```

[21]: 15.45

- **Portfolio risk**

[22]:
```python
# Correlation b/w stocks
corr_matrix = d_returns.corr().round(3)
```

[23]:
```python
sb.heatmap(corr_matrix,vmin=-1, vmax=1,cmap='flare', annot=True)
plt.show()
```



[24]:
```python
ptf_Eyvol = np.sqrt(np.dot(np.dot(d_returns.cov()*250, weights),weights)).
  ↪round(5)*100
ptf_Eyvol
```

[24]: 35.047

- **Risk-Adjusted Return**

[25]:
```python
rar = (ptf_Eyreturn/ptf_Eyvol).round(3)
rar
```

[25]: 0.441

- **Portfolio Optimization**

[26]:
```
# weights combination
w_combo = []
ret_combo = []
risk_combo = []
for _ in np.arange(8000):
    weights = (np.random.random(len(avg_yreturn))).round(4)
    weights /= np.sum(weights)
    w_combo.append(weights)
    ret_combo.append(np.dot(weights,capm_return).round(3))
    risk_combo.append(np.dot(np.dot(d_returns.cov()*250, weights),weights)**(1/
↪2)*100)
```

[27]:
```
# global minimum variance portfolio
gmv_vol = min(risk_combo)
index = risk_combo.index(gmv_vol)
gmv_ret = ret_combo[index]
gmv_weight = w_combo[index]*100
```

[28]:
```
display_gmv = f"The global minimum variance portfolio has a return rate of␣
↪{gmv_ret:.2f}% with a volatility of {gmv_vol:.2f}% (RAR={gmv_ret/gmv_vol:.
↪2f}). This portfolio is achieved by allocating {gmv_weight[0]:.2f}% to␣
↪Apple, {gmv_weight[1]:.2f}% to Google, {gmv_weight[2]:.2f}% to Microsoft,␣
↪{gmv_weight[3]:.2f}% to Nvidia, and {gmv_weight[4]:.2f}% to Tesla."
```

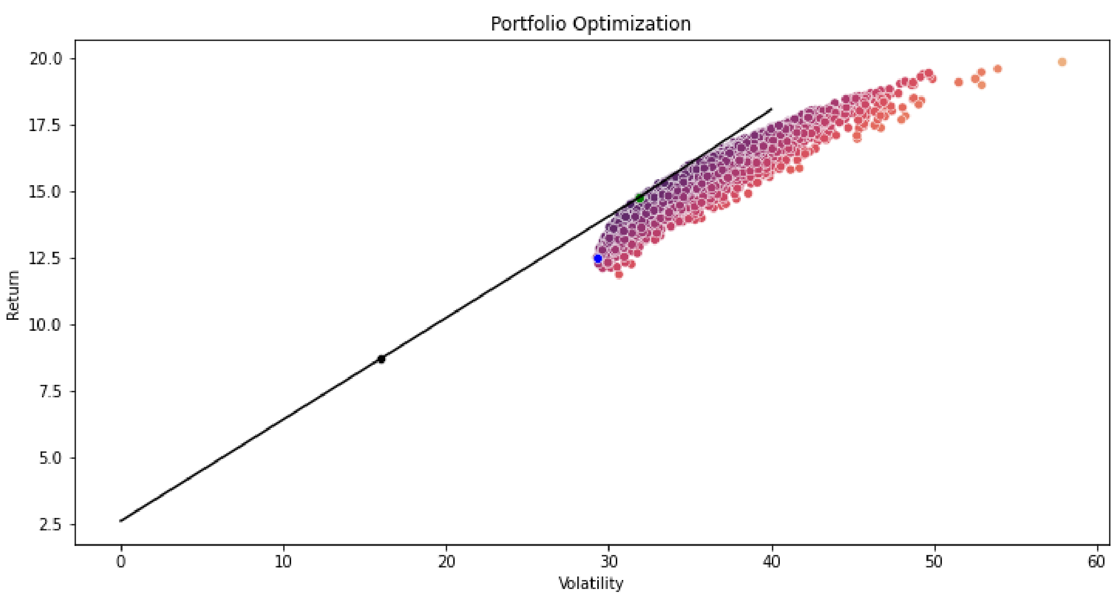
[29]:
```
# optimal portfolio
sharpe = ret_combo/pd.Series(risk_combo)
index_s = sharpe[sharpe==sharpe.max()].index[0]
sharpe_vol = risk_combo[index_s]
sharpe_ret = ret_combo[index_s]
sharpe_w = w_combo[index_s]*100
```

[30]:
```
display_op = f"The optimal portfolio has a return rate of {sharpe_ret:.2f}%␣
↪with a volatility of {sharpe_vol:.2f}% (RAR={sharpe_ret/sharpe_vol:.2f}).␣
↪This portfolio is obtained by investing {sharpe_w[0]:.2f}% in Apple,␣
↪{sharpe_w[1]:.2f}% in Google, {sharpe_w[2]:.2f}% in Microsoft, {sharpe_w[3]:.
↪2f}% in Nvidia, and {sharpe_w[4]:.2f}% in Tesla."
```

[31]:
```
# highest return portfolio
high_ret = max(ret_combo)
index = ret_combo.index(high_ret)
high_vol = risk_combo[index]
high_w = w_combo[index]*100
```

[32]:
```
display_high = f"The portfolio with the highest return rate has a return of␣
↪{high_ret:.2f}% with a volatility of {high_vol:.2f}% (RAR={high_ret/high_vol:
↪.2f}). This portfolio is achieved by allocating {high_w[0]:.2f}% to Apple,␣
↪{high_w[1]:.2f}% to Google, {high_w[2]:.2f}% to Microsoft, {high_w[3]:.2f}%␣
↪to Nvidia, and {high_w[4]:.2f}% to Tesla."
```

[33]:
```
# Portfolio optimization
slope= (ret_combo[index_s]-rf)/(risk_combo[index_s]-0)
mkt_lineX =np.array([0,risk_combo[index_s],40])
mkt_lineY =np.array([rf,ret_combo[index_s],(40*slope+rf)+.2])
plt.figure(figsize=(12,6))
#palette=sb.color_palette("Blues"))
sb.scatterplot(x=risk_combo,y=ret_combo,
               hue=sharpe,
               palette= sb.color_palette("flare",as_cmap=True),legend=False)
sb.scatterplot(x= [gmv_vol], y=gmv_ret,color="blue")
sb.scatterplot(x= [sharpe_vol], y=sharpe_ret,color="green")
sb.lineplot(x = mkt_lineX, y=mkt_lineY, color="black")
sb.scatterplot(x= [sharpe_vol*0.5], y=[(sharpe_ret+rf)*0.5],color="black")
plt.title("Portfolio Optimization")
plt.xlabel("Volatility")
plt.ylabel("Return")
plt.savefig('Markowitz.png')
plt.show()
```



[34]:

```python
a=f"Please note that when we invest an equal amount in all stocks, the expected␣
↪portfolio return is {ptf_Eyreturn}%, with a volatility of {ptf_Eyvol}% and a␣
↪RAR of {ptf_Eyreturn/ptf_Eyvol:.2f}."
print(display_gmv,display_op, display_high,a)
```

```
The global minimum variance portfolio has a return rate of 12.49% with a
volatility of 29.26% (RAR=0.43). This portfolio is achieved by allocating 29.20%
to Apple, 38.01% to Google, 32.02% to Microsoft, 0.50% to Nvidia, and 0.27% to
Tesla. The optimal portfolio has a return rate of 14.76% with a volatility of
31.87% (RAR=0.46). This portfolio is obtained by investing 46.15% in Apple,
35.09% in Google, 0.53% in Microsoft, 3.12% in Nvidia, and 15.10% in Tesla. The
portfolio with the highest return rate has a return of 19.84% with a volatility
of 57.90% (RAR=0.34). This portfolio is achieved by allocating 4.89% to Apple,
4.75% to Google, 5.91% to Microsoft, 1.11% to Nvidia, and 83.34% to Tesla.
Please note that when we invest an equal amount in all stocks, the expected
portfolio return is 15.45%, with a volatility of 35.047% and a RAR of 0.44.
```

The RAR obtained by investing an equal amount in all stocks is inferior to the RAR of the optimal portfolio. This suggests that an equal allocation may not always be the optimal choice. In the provided graph, the blue dot represents the global minimum variance portfolio, which has the lowest volatility. On the other hand, the green dot represents the tangency portfolio, also known as the optimal portfolio, which offers the highest risk-adjusted return. The term "tangency portfolio" refers to its position where it intersects the tangent line connecting the risk-free rate and the efficient frontier (the curve above the global minimum variance portfolio). The line connecting the risk-free rate and the tangency portfolio is called the "capital allocation line" (CAL), illustrating the various combinations an investor can create between the risk-free asset and the risky asset (portfolio of stocks). For example, an investor who allocates 50% to the risk-free asset and 50% to the optimal portfolio would have a risk-return profile similar to the dark dot on the CAL. The remaining dots represent the risk-return profiles of over 5000 combinations of the same portfolio. All the dots below the upper frontier of the curve represent suboptimal portfolios and should be avoided, whereas all the dots on the frontier of the curve represents efficient portfolios.

Now let's move on to the analysis of the health-portfolio. However, to keep it concise, we will provide a brief overview as we have already covered important concepts in Part A.

B. Health Portfolio Analysis

1. Risk and Return Analysis

Suppose we are equally interested in five health-firms namely Amgen, Johnson&Johnson, Astrazeneca, Pfizer and Sanofi.

[35]:
```python
data2 = yf.Tickers(["JNJ","AZN","SNY","AMGN","PFE"]).
↪history(start="2018-06-30", end="2023-06-30").loc[:,"Close"]
```

```
[*********************100%***********************]  5 of 5 completed
```

[36]:
```
print(data2.round(3).head(3))
print(data2.round(3).tail(3))
```

```
               AMGN     AZN      JNJ     PFE     SNY
Date
2018-06-29  158.767  30.596  106.216  28.668  33.090
2018-07-02  159.369  30.344  106.426  28.707  33.082
2018-07-03  159.730  30.291  107.415  28.723  33.636
              AMGN    AZN     JNJ    PFE    SNY
Date
2023-06-27  222.61  71.67  163.29  36.42  53.67
2023-06-28  221.31  70.96  162.96  36.29  53.74
2023-06-29  221.16  70.85  164.10  36.12  52.99
```

- **Historical return**

[37]:
```
# Historical return
avg_yreturn2 = (data2.pct_change().mean()*250*100).round(3)
d_returns2 = data2.pct_change()
hr_df2 = pd.DataFrame(avg_yreturn2).reset_index().rename(columns={"index":
↪"stock", 0:"return"})
avg_yreturn2
```

[37]:
```
AMGN     9.789
AZN     20.098
JNJ     10.720
PFE      8.045
SNY     12.179
dtype: float64
```

- **Risk premium**

[38]:
```
# risk premium
rf = 2.61
market_ret = round(Mkt.pct_change().mean()*250*100,3)
mkt_ret = market_ret.mean()
rp = round(mkt_ret - rf,3)
rp
```

[38]: 9.132

[39]:
```
beta2 = {
    "AMGN":0.63,
    "AZN":0.5,
    "JNJ":0.55,
    "PFE":0.61,
    "SNY":0.71
}
```

```
beta2 = pd.Series(beta2, index=beta2.keys())
```

- **Expected return**

[40]:
```
# expected return
capm_return2 = (rf + beta2 * rp).round(3)
er_df2 = pd.DataFrame(capm_return2).reset_index().rename(columns={"index":
↪"stock", 0:"return"})
capm_return2
```

[40]:
```
AMGN    8.363
AZN     7.176
JNJ     7.633
PFE     8.181
SNY     9.094
dtype: float64
```

- **Risk-Adjusted Return**

[41]:
```
raj2 = capm_return2/(d_returns2.std()*math.sqrt(250)*100)
raj2 = pd.DataFrame(raj2).reset_index().rename(columns={"index":"stock", 0:
↪"return"})
print(raj2)
```

```
  stock    return
0  AMGN  0.329786
1   AZN  0.275389
2   JNJ  0.375752
3   PFE  0.310966
4   SNY  0.383270
```

[42]:
```
fig, ax = plt.subplots(1,3,sharey=True, figsize=(12,5))
sb.barplot(ax=ax[0],data = er_df2,x="return", y="stock")
ax[0].set_title("Expected Return of Stocks")
sb.barplot(ax=ax[1],data = hr_df2,x="return", y="stock")
ax[1].set_title("Historical Return of Stocks")
sb.barplot(ax=ax[2],data = raj2,x="return", y="stock")
ax[2].set_title("Risk Adjusted Return of Stocks")
plt.show()
```

Based on the analysis, Astrazeneca demonstrates the highest historical return of 20.01% per year, followed by Sanofi with 12.18% per year, Johnson & Johnson with 10.72% per year, Amgen with 9.72% per year, and Pfizer with 8.05% per year. These figures represent the average returns for investors who held these stocks from June 2018 to June 2023.

In terms of expected return, Sanofi has the highest return at 9.09% per year, followed by Amgen with 8.36%, Pfizer with 8.18%, Johnson & Johnson with 7.63%, and Astrazeneca with 7.18%.

When considering risk-adjusted return, Sanofi exhibits the highest Risk-Adjusted Return (RAR) at 0.38, followed by Johnson & Johnson with 0.37, Amgen with 0.33, Pfizer with 0.31, and Astrazeneca with 0.28.

---

2. Portfolio Analysis

[43]:
```
# weights combination
w_combo2 = []
ret_combo2 = []
risk_combo2 = []
for _ in range(8000):
    weights = (np.random.random(len(avg_yreturn2))).round(4)
    weights /= np.sum(weights)
    w_combo2.append(weights)
    ret_combo2.append(np.dot(weights,capm_return2).round(3))
    risk_combo2.append(np.dot(np.dot(d_returns2.cov()*250,␣
↪weights),weights)**(1/2)*100)
```

[44]:
```
# global minimum variance portfolio
gmv_vol2 = min(risk_combo2)
index = risk_combo2.index(gmv_vol2)
gmv_ret2 = ret_combo2[index]
```

```
gmv_weight2 = w_combo2[index]*100
```

[45]:
```
display_gmv = f"The global minimum variance portfolio has a return rate of {gmv_ret2:.2f}% with a volatility of {gmv_vol2:.2f}% (RAR={gmv_ret2/gmv_vol2:.2f}). This portfolio is obtained by allocating {gmv_weight2[0]:.2f}% to Amgen, {gmv_weight2[1]:.2f}% to AstraZeneca PLC, {gmv_weight2[2]:.2f}% to J&J, {gmv_weight2[3]:.2f}% to Pfizer, and {gmv_weight2[4]:.2f}% to Sanofi."
```

[46]:
```
# optimal portfolio
sharpe = ret_combo2/pd.Series(risk_combo2)
index = sharpe[sharpe==sharpe.max()].index[0]
sharpe_vol2 = risk_combo2[index]
sharpe_ret2 = ret_combo2[index]
sharpe_w2 = w_combo2[index]*100
```

[47]:
```
display_op = f"The optimal portfolio has a return rate of {sharpe_ret2:.2f}% with a volatility of {sharpe_vol2:.2f}% (RAR={sharpe_ret2/sharpe_vol2:.2f}). This portfolio is achieved by investing {sharpe_w2[0]:.2f}% in Amgen, {sharpe_w2[1]:.2f}% in AstraZeneca PLC, {sharpe_w2[2]:.2f}% in J&J, {sharpe_w2[3]:.2f}% in Pfizer, and {sharpe_w2[4]:.2f}% in Sanofi."
```

[48]:
```
print(display_gmv, display_op)
```

```
The global minimum variance portfolio has a return rate of 8.05% with a
volatility of 17.91% (RAR=0.45). This portfolio is obtained by allocating 9.66%
to Amgen, 12.10% to AstraZeneca PLC, 45.24% to J&J, 8.29% to Pfizer, and 24.72%
to Sanofi. The optimal portfolio has a return rate of 8.35% with a volatility of
18.26% (RAR=0.46). This portfolio is achieved by investing 15.14% in Amgen,
2.19% in AstraZeneca PLC, 34.33% in J&J, 10.30% in Pfizer, and 38.04% in Sanofi.
```

C. Portfolio Performance Analysis

Having identified the optimal portfolios for both the tech-stocks and health-stocks, we now aim to compare their performance by considering various portfolio performance criteria. One such criterion is the Sharpe ratio, which we introduced in Part A. Additionally, there are several other measures such as the Treynor measure, Jensen alpha, information ratio, and Modigliani-square, among others. Each criterion has its advantages and disadvantages and may be more suitable in different situations. The mathematical expressions for these measures are provided below.

The Treynor measure is defined as:

$$\text{Treynor measure} = \frac{r_p - rf}{\beta_p}$$

The Jensen alpha is given by:

$$\text{Jensen alpha} = r_p - [rf + \beta_A(r_M - rf)]$$

The information ratio is calculated as:

$$\text{Information ratio} = \frac{\text{Jensen alpha}}{\theta_{e_p}}$$

And the Modigliani-square ratio is expressed as:

$$\text{Modigliani-square} = [r_P \cdot \frac{\theta_M}{\theta_P} + rf \cdot (1 - \frac{\theta_M}{\theta_P})] - r_M = r_{P^*} - r_M$$

In simple terms, the Treynor measure is similar to the Sharpe ratio but uses the systematic risk β instead of the total risk θ. The Jensen alpha represents the difference between the actual portfolio return and the expected portfolio return. The information ratio is the excess realized return divided by the nonsystematic risk $\theta_{e_p} = \theta_p - \beta_p$. The Modigliani-square ratio is conceptually similar to the Sharpe ratio. It assumes that the portfolio return should be a combination of the portfolio return weighted by $\frac{\theta_M}{\theta_P}$ and the risk-free return weighted by $1 - \frac{\theta_M}{\theta_P}$. Therefore, it is obtained by taking the difference between the adjusted portfolio return r_{P^*} and the market return.

The choice of which measure to use depends on the specific situation. The Sharpe ratio is commonly employed when an investor holds a single portfolio. Because such portfolio does not compute with others for fund allocation, the sharpe ratio is used because its consider the toal risk of the holder. In other words, Sharpe ratio is useful when the goal is to evaluate the performance of a single portfolio compared to a benchmark such as the market index or when choosing one portfolio out of two or several others. In cases where an investor has multiple portfolios, the Treynor measure is preferred as it considers systematic risk rather than total risk due to diversification. The jensen alpha (α) is related to the Sharpe and Treynor measures by the following relation:

$$\text{Sharpe ratio} = \frac{r_P - rf}{\theta_P} = \frac{\alpha_P}{\theta_P} + \text{Corr}(P, M)\frac{\alpha_M}{\theta_M}$$

$$\text{Treynor measure} = \frac{r_P - rf}{\beta_P} = \frac{\alpha_M}{\beta_P} + \frac{r_M - rf}{\beta_M}$$

Thus, superior performance requires a positive alpha. However, a higher alpha alone does not guarantee a better Sharpe ratio, as an investor can simultaneously increase their risk. For practical purposes, we only use the first 3 ratios since computing the information ratio will required additional estimation of the idiosyncratic risk, which can be otained by calculating the standard deviations of error terms from the regression:

$$\overline{r}_{Portfolio} = \alpha + \beta \cdot r_{Market} + e$$

where

$$e = \overline{r}_{Portfolio} - r_{Portfolio}$$

and $\overline{r}$ is the regression-estimated return and r is the actual return.

[49]:
```
# portfolios summary
Op_ptf = np.array([[sharpe_ret, round(sharpe_vol,3)],
                   [sharpe_ret2, round(sharpe_vol2,3)]])
Op_ptf = pd.DataFrame(Op_ptf, columns=["ret","risk"], index=["tech","health"])
```

[50]:
```
print(Op_ptf)
```

```
           ret    risk
tech    14.756  31.868
health   8.346  18.257
```

- **Sharpe ratios**

[51]:
```
((Op_ptf.ret-rf)/Op_ptf.risk).round(2)
```

[51]:
```
tech      0.38
health    0.31
dtype: float64
```

- **Treynor measure**

[52]:
```
TechPtf_beta = round(np.dot(beta/100,sharpe_w),2)
HealthPtf_beta = round(np.dot(beta2/100,sharpe_w2),2)
ptf_beta = pd.Series([TechPtf_beta,HealthPtf_beta], index=["tech","health"])
```

[53]:
```
ptf_beta
```

[53]:
```
tech      1.33
health    0.63
dtype: float64
```

[54]:
```
((Op_ptf.ret-rf)/ptf_beta).round(2)
```

[54]:
```
tech      9.13
health    9.10
dtype: float64
```

- **Jensen Alpha and Information ratio**

[55]:
```
annualized_rrT = ((data.iloc[-1]/data.iloc[0])**(250/len(data))-1)*100
annualized_rrH = ((data2.iloc[-1]/data2.iloc[0])**(250/len(data2))-1)*100
```

[56]:
```
realized_TechR = round(np.dot(annualized_rrT,sharpe_w/100),3)
realized_HealthR = round(np.dot(annualized_rrH,sharpe_w/100),3)
realized_ptfR = pd.Series([realized_TechR, realized_HealthR],␣
 ↪index=["tech","health"])
```

[57]:
```python
# jensen
jensen_alpha = realized_ptfR - Op_ptf.ret
jensen_alpha
```

[57]:
```
tech      17.293
health     2.845
dtype: float64
```

Considering the Sharpe ratios, it can be concluded that the Tech-portfolio outperforms the Health-portfolio. This indicates that, for investors who hold a single portfolio, choosing the Tech-portfolio would result in higher returns per unit of total risk. However, when analyzing the Treynor measures, the Health-portfolio surpasses the Tech-portfolio. Therefore, for investors with multiple portfolios, the Health-portfolio would be a more favorable option as it provides higher returns per unit of systematic risk. This highlights that the evaluation of portfolio performance is dependent on the specific situation and the investor's preferences.

Both portfolios exhibit positive alphas, but the Tech-portfolio has generated a significantly higher alpha compared to the Health-portfolio (approximately 5 times higher). This implies that the Tech-portfolio has performed exceptionally well and has exceeded expectations over the past five years.

Conclusion

The objective of this article is to present essential concepts in security analysis and portfolio selection. We have conducted an analysis of two investment portfolios: one consisting of technology firms and the other composed of health firms. By applying the Markowitz optimization model, we have identified the optimal portfolio for each category. Additionally, we have addressed the challenges associated with comparing portfolios using historical and expected returns, and have introduced more robust measures for portfolio evaluation and selection.

END