A.Y.: 2023-24 **Class/Sem: B.E.B.Tech/ Sem-VII** **Sub:** Quantitative Portfolio Management

Experiment 5

Name: Umang Kirit Lodaya **SAP ID: 60009200032**

Batch: D11

Aim: Analyzing the plot EW and GMV on the Efficient Frontier on a given dataset.

Objective:

- Understand the concept of the efficient frontier.
- Learn how to plot the efficient frontier for a given dataset.
- Compare the performance of EW and GMV portfolios on the efficient frontier.

Theory:

The efficient frontier is a curve that shows the combinations of expected return and risk that can be achieved by an investor. The curve is efficient because it represents the best possible trade-off between return and risk.

There are two main types of portfolios that can be plotted on the efficient frontier: EW portfolios and GMV portfolios. EW portfolios are portfolios that are constructed by equally weighting the assets in a given dataset. GMV portfolios are portfolios that are constructed by maximizing the geometric mean return of the assets in a given dataset.

The efficient frontier is a curve that shows the combinations of expected return and risk that can be achieved by an investor. The curve is efficient because it represents the best possible trade-off between return and risk.

The formula for the efficient frontier is:

$$ef = np.minimum(er / sd, 2)$$

where:

- er is the vector of expected returns of the assets in the dataset
- sd is the vector of standard deviations of the assets in the dataset
- 2 is the risk-free rate

The efficient frontier is a downward-sloping curve because as the expected return of a portfolio increases, its risk must also increase. This is because there is no free lunch in investing: investors cannot expect to get higher returns without taking on more risk.

The EW and GMV portfolios are two specific portfolios that can be plotted on the efficient frontier. The EW portfolio is a portfolio that is constructed by equally weighting the assets in a given dataset. The GMV portfolio is a portfolio that is constructed by maximizing the geometric mean return of the assets in a given dataset.

The EW portfolio is typically located towards the bottom of the efficient frontier, while the GMV portfolio is typically located towards the top of the efficient frontier. This is because the EW portfolio has lower risk, but also lower expected return, than the GMV portfolio.

Investors can use the efficient frontier to choose the portfolio that best suits their risk tolerance and investment goals. Investors who are more risk-averse may choose a portfolio that is located towards the bottom of the efficient frontier, while investors who are more risk-seeking may choose a portfolio that is located towards the top of the efficient frontier.

Lab Experiment to be done by students:

1. Download a dataset of asset returns.
2. Calculate the expected returns and standard deviations of the assets in the dataset.
3. Plot the efficient frontier for the dataset.
4. Calculate the EW and GMV portfolios for the dataset.
5. Plot the EW and GMV portfolios on the efficient frontier.
6. Compare the performance of the EW and GMV portfolios on the efficient frontier.

NAME: UMANG KIRIT LODAYA

SAP ID: 60009200032

BATCH: D11

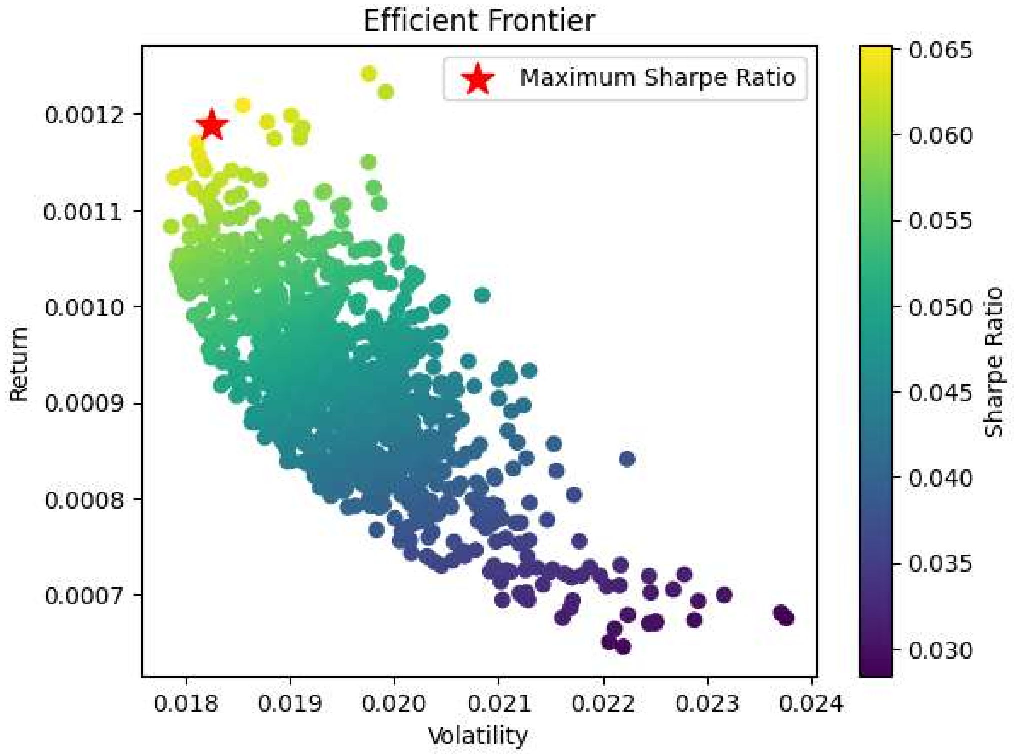
In [12]:
```
import yfinance as yf
import numpy as np
import matplotlib.pyplot as plt
```

In [13]:
```
# Step 1: Download a dataset of asset returns, import from yi
nance
tickers = ["AAPL", "MSFT", "GOOGL", "AMZN"]  # Replace with yo
ur desired tickers
data = yf.download(tickers, period="1y")["Adj Close"].pct_chan
ge().dropna()
```

```
[*********************100%***********************]  4 of 4 co
pleted
```

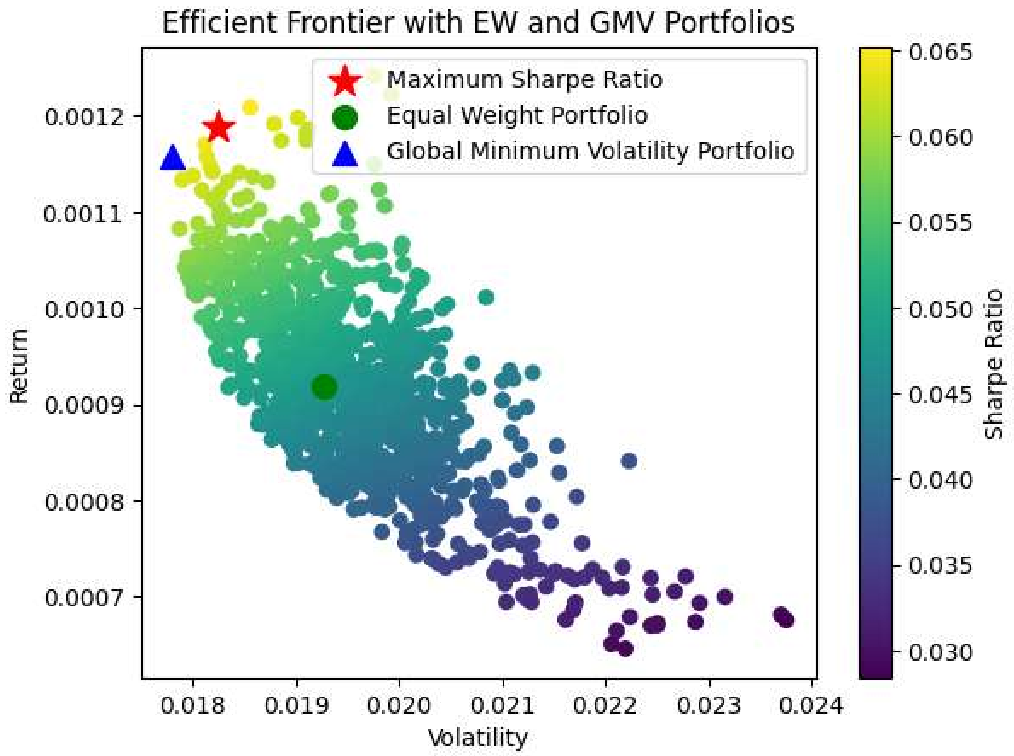
In [14]:
```
# Step 2: Calculate the expected returns and standard deviat
ns of the assets
returns = data.mean()
std_devs = data.std()
```

In [15]:
```python
# Step 3: Plot the efficient frontier for the dataset
num_portfolios = 1000
results = np.zeros((3, num_portfolios))
weights = np.zeros((num_portfolios, len(tickers)))

for i in range(num_portfolios):
    w = np.random.random(len(tickers))
    w /= np.sum(w)
    weights[i, :] = w
    port_return = np.sum(w * returns)
    port_std = np.sqrt(np.dot(w.T, np.dot(data.cov(), w)))
    results[0, i] = port_return
    results[1, i] = port_std
    results[2, i] = port_return / port_std

max_sharpe_idx = np.argmax(results[2])
max_sharpe_ret = results[0, max_sharpe_idx]
max_sharpe_std = results[1, max_sharpe_idx]

plt.scatter(results[1, :], results[0, :], c=results[2, :], cma
p='viridis')
plt.colorbar(label='Sharpe Ratio')
plt.xlabel('Volatility')
plt.ylabel('Return')
plt.title('Efficient Frontier')
plt.scatter(max_sharpe_std, max_sharpe_ret, marker='*', color
='r', s=200, label='Maximum Sharpe Ratio')
plt.legend()
plt.show()
```



In [16]:
```
# Step 4: Calculate the EW and GMV portfolios
ew_weights = np.ones(len(tickers)) / len(tickers)
ew_return = np.sum(ew_weights * returns)
ew_std = np.sqrt(np.dot(ew_weights.T, np.dot(data.cov(), ew_we
ights)))

inv_cov_matrix = np.linalg.inv(data.cov())
gmv_weights = inv_cov_matrix.dot(np.ones(len(tickers))) / np.s
um(inv_cov_matrix.dot(np.ones(len(tickers))))
gmv_return = np.sum(gmv_weights * returns)
gmv_std = np.sqrt(np.dot(gmv_weights.T, np.dot(data.cov(), gmv
_weights)))
```

In [17]:
```
# Step 5: Plot the EW and GMV portfolios on the efficient fr
tier
plt.scatter(results[1, :], results[0, :], c=results[2, :], cma
p='viridis')
plt.colorbar(label='Sharpe Ratio')
plt.xlabel('Volatility')
plt.ylabel('Return')
plt.title('Efficient Frontier with EW and GMV Portfolios')
plt.scatter(max_sharpe_std, max_sharpe_ret, marker='*', color
='r', s=200, label='Maximum Sharpe Ratio')
plt.scatter(ew_std, ew_return, marker='o', color='g', s=100, l
abel='Equal Weight Portfolio')
plt.scatter(gmv_std, gmv_return, marker='^', color='b', s=100,
label='Global Minimum Volatility Portfolio')
plt.legend()
plt.show()
```



In [18]:

```python
# Step 6: Compare the performance of the EW and GMV portfoli
on the efficient frontier
print("Equal Weight Portfolio:")
print("Return:", ew_return)
print("Volatility:", ew_std)

print("\nGlobal Minimum Volatility Portfolio:")
print("Return:", gmv_return)
print("Volatility:", gmv_std)
```

```
Equal Weight Portfolio:
Return: 0.0009188382183513977
Volatility: 0.01926681106352405

Global Minimum Volatility Portfolio:
Return: 0.0011583053277796736
Volatility: 0.017789908124995573
```