STOCK PORTFOLIO OPTIMIZATION WORKSHOP

2024 - 11 - 21

STOCK POFTFOLIO OPTIMIZATION WITH PYTHON

WORKSHOP HELD BY SUPERTYPE.AI - FULL CYCLE DATA SCINECE AND ANALYTICS CONSULTANCY

TABLE OF CONTENTS

WE'LL TAKE YOU THROUGH...

- **INTRODUCTION**
  - STOCK INVESTMENT MATRIX BASICS
  - STOCK INVESTMENT FRAMEWORK
- **SECTORS FINANCIAL API**
  - API OVERVIEW - DEMO
  - STOCK TICKERS & INDEX - DEMO
  - DATA COLLECTION - DEMO
  - RISK & DIVERSIFICATION STRATEGY
  - DIVERSIFICATION & EDA - DEMO
  - STOCK ANALYSIS SUMMARY
- **MEAN VARIANCE TRADE-OFF**
  - MEAN VARIANCE COVARIANCE
  - SINGLE STOCK RETURN - DEMO
  - MULTIPLE STOCK RETURN - DEMO
  - MULTIPLE STOCK VARIANCE - DEMO
  - VARIANCE - DIY
  - VARIANCE & VOLATILITY ANALYSIS
- **PORTFOLIO OPTIMIZATION**
  - EFFICIENT FRONTIER
  - INVESTOR INDIFFERENCE CURVE
  - COMBINE RISK FREE ASSETS
  - CAPITAL MARKET LINE
  - SHAPRE RATIO
  - SHARPE RATIO PRACTICE - DIY
  - MONTE CARLO SIMULATION - DEMO
  - SCIPY FUNCTION MINIMIZATION - DEMO
  - INTEGRATE UTILITY CURVE - DEMO
- **SUMMARY & CHALLENGE**

INTRODUCTION TO STOCK MARKET

INVESTMENT METRICS BASICS

UNDERSTANDING SOME BASIC CONCEPTS

Earnings per Share (EPS)

Indicates a **company's profitability** on a per-share basis, reflecting **income for shareholders**.

Price to Earnings (P/E)

Compares the stock's price to its earnings, helping assess whether the stock is **overvalued or undervalued** relative to its earnings.

Dividend Yield

Shows the **annualized** percentage of a company's earnings distributed as dividends, providing insight into **potential income** from the investment.

Debt-to-Equity Ratio (D/E)

Measures a company's **financial leverage**, indicating how much debt it uses to finance its operations compared to equity.

Market Capitalization

Represents the **total market value of a company's outstanding shares**, giving a sense of the company's size and stability.

Historical Price Volatility

Reflects the stock's **price fluctuations**, helping assess the risk and stability of the investment.

FRAMEWORKS

STOCK EVALUATION AND ANAYSIS FRAMEWORKS

FACTOR Framework

- **Fundamentals**: Focus on the company's financial health and profitability.
- **Assets & Liabilities**: Evaluate balance sheet strength
- **Cash Flow**: Analyze cash flow statement.
- **Timing**: Consider market timing and trends.
- **Outlook**: Examine industry trends and economic conditions.
- **Risks**: Evaluate internal and external market volatility.

SWOT Analysis

- **Strength**: Analyze company's competitive advantages.
- **Weakness**: Identify company's shortcomes.
- **Opportunity**: Consider growth potentials.
- **Threat**: Assess macroeconomic risks – PESTLE.

STOCK ANALYSIS

CAN SLIM Method – William O´Neil

A popular method for evaluating growth stocks based on **seven criterias**: current earnings, annual growth, new product/services, stock market supply and demand, leader or laggard, institutional sponsorship, market direction.

T.I.P.S Framework

- **Trends**: Analyze current industry and sector trends
- **Intrinsic Value**: Undervalued v.s. Overvalued
- **Profitability**: assess profit margins, RoE, Revenue Growth
- **Sustainability**: Evaluate ability to maintain growth.

SECTORS FINANCIAL API EXPLORATION

DEMO - API OVERVIEW

Sectors ⌄ Financial Markets ⌄ Stock Research ⌄

Sectors

Sectors Financial API

Making your first authenticated API request to Sectors

1. Sectors Standard or Professional

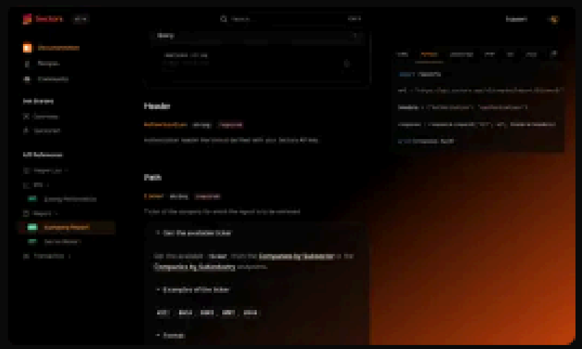
Through the Sectors API, you are two lines of code away from Indonesia's most comprehensive financial market data and stock research tools. Sectors API is available to our Standard and Professional plans subscribers.

2. Obtain Sectors API Key

Once logged in with your Sectors account (Standard or Professional plan), navigate to the API Keys section to generate a new API key. Include this API key in the header of your requests to authenticate and access the API (code snippets provided below).

Sectors Documentation

Sectors Financial API comes with a best-in-class API Documentation site that provides a comprehensive guide to integrating our financial data into your analysis and applications.



Sectors API Documentation

Data Analysts and Developers typically make their first API requests within minutes.

Sectors Docs

3. Code Example

Quick Start: Sectors API from your favorite programming language

Use one of the documented API endpoints along with your Sectors API Key to access data from Sectors. Explore Sectors API Documentation or Sectors API Playground to discover the available endpoints and sample responses.

Python | JavaScript | R

Copy code

```python
import requests

# Replace the URL with a URL from the Available Endpoints
url = "https://api.sectors.app/v1/.../"
api_key = "API key issued from the key management section"
headers = {"Authorization": api_key}

try:
    response = requests.get(url, headers=headers)
    response.raise_for_status()
    data = response.json()
except requests.exceptions.HTTPError as err:
    raise SystemExit(err)
```

DEMO - STOCK INDEX & TICKERS

Helper List

Companies by Index

Return list of companies from a stock index

GET / index / {index} /

Send

Header

Path

Available index: ftse, idx30, idxbumn20, idxesgl, idxg30, idxhidiv20, idxq30, idxv30, jii70, kompas100, lq45, sminfra18, srikehati, economic30

cURL | Python | JavaScript | PHP | Go | Java

```python
import requests

url = "https://api.sectors.app/v1/index/{index}/"

headers = {"Authorization": "<authorization>"}

response = requests.request("GET", url, headers=headers)

print(response.text)
```

200 | 429


```
[
  {
    "symbol": "NISP.JK",
    "company_name": "PT Bank OCBC NISP Tbk"
  },
  {
    "symbol": "PNBN.JK",
```

DEMO - DATA COLLECTION

Table of contents

*Quick Introduction *

Section 1 - Sectors Financial API & Stock Investment Overview

Section 2 - Where to start?

Section 3 - Data Collection from Sectors API

Section 3.1 - Stock Price Information

Section 3.2 - Company Report Information

Section 4 - Which Stock to choose?

Section 4.1 - Data Preprocessing

Section 4.2 - Exploratory Data Analysis (EDA)

Section 4.3 - Normalization & Scoring

Section 4.4 - Stock selection choice

Section 5 - MVO with Monte Carlos Simulation

Section 5.1 - Theoretical Context

Section 5.2 - Mean & Variance

Section 5.3 Monte Carlo Simulation

Section 5.4 - Sharpe Ratio

Section 5.5 - Efficient Frontier

+ Code + Text

Reconnect High RAM

Gemini

Section 3.1 - Stock Price Information

```
# Retrive Stock index from "Companies by Index" API

import time
import requests
from google.colab import userdata

# Retrieve the API key securely
api_key = userdata.get('SECTORS_API_KEY')

# Define the API URL
url = "https://api.sectors.app/v1/index/sminfra18/"

# Pass the API key in the header
headers = {"Authorization": api_key}

# Make the API request
response_company_index = requests.get(url, headers=headers)

print(response_company_index.text)
```

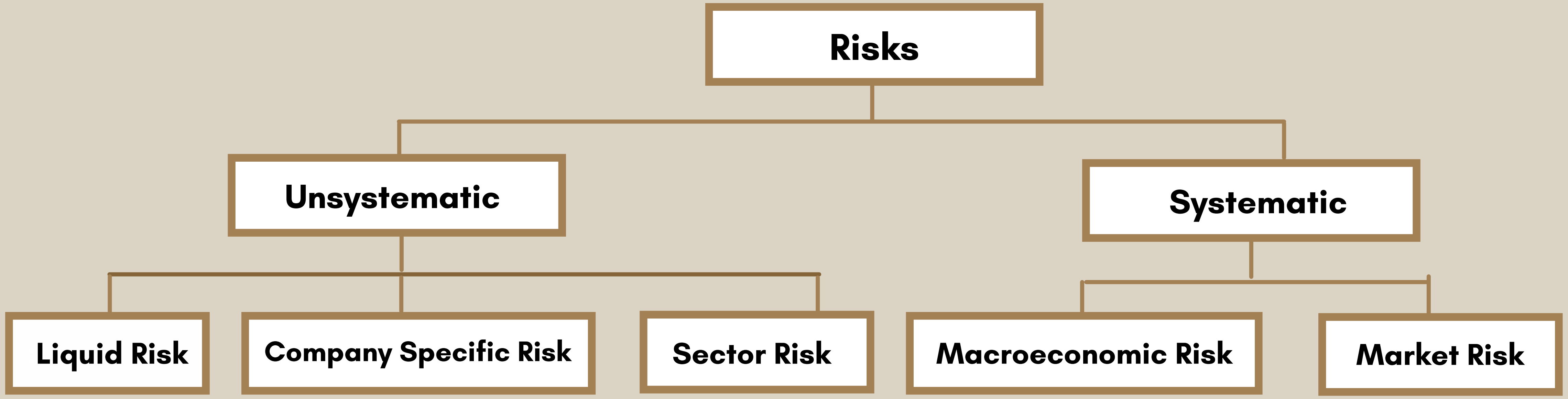
[{"symbol":"ADHI.JK","company_name":"PT Adhi Karya (Persero) Tbk."},{"symbol":"AKRA.JK","company_name":"PT AKR Corporindo Tbk."},{"symbol":"BBNI.JK","company_na

```
# Retrieve date and price from "Daily Transaction Data" API

from datetime import datetime, timedelta

# Function to calculate the date 90 days ago from today
def calculate_start_date(days_ago=90):
    return (datetime.now() - timedelta(days=days_ago)).strftime('%Y-%m-%d')

# Calculate the start date 90 days ago
start_date = calculate_start_date()

# Looping API info
history_sminfra18 = []
```

RISKS & DIVERSIFICATION

USE DIVERSIFICATION STRATEGY TO ELIMINATE UNSYSTEMATIC RISKS

- **Systematic risks**: “market risk”, can not be eliminated through diversification effect.
- **Unsystematic risks:** Individual companies or industries specific issues can be eliminated through diversification strategy.



Diversification strategy reduces volatility, smooth out **Unsystematic Risks** and protect investors from unpredictable and unexpected market shocks.

Sector and Industry
Spread investments across different industries to reduce sector specific risk.

Returns & Correlation
Selecting stocks that offer varied returns and are not too closely correlated.

Geography
Invest in stocks from different geographical regions to reduce regional risks.

Diversify Assets
Invest in different assets classes, tangible & intangible, bonds, stocks, commodities...

DEMO - APPLY DIVERSIFICATION STRATEGY

Table of contents

Stock Investment Overview
Section 2 - Where to start?
Section 3 - Data Collection from Sectors API
Section 3.1 - Stock Price Information
Section 3.2 - Company Report Information
Section 4 - Which Stock to choose - Showcase of Diverisifaction Strategy and Simple EDA Process
Section 4.1 - Data Preprocessing
Section 4.2 - Exploratory Data Analysis (EDA)
Section 4.3 - Normalization & Scoring
Section 4.4 - Stock selection choice
Section 5 - MVO with Monte Carlos Simulation
Section 5.1 - Theoretical Context
Section 5.2 - Mean & Variance
Section 5.3 Monte Carlo Simulation
Section 5.4 - Sharpe Ratio
Section 5.5 - Efficient Frontier
Section 6 - MVO with Scipy Minimized Function

+ Code + Text Reconnect High RAM Gemini

Section 4 - Which Stock to choose - Showcase of Diverisifaction Strategy and Simple EDA Process

Section 4.1 - Data Preprocessing

How to select stocks?

For our client Indra, who's focused on stocks from the SMInfra18 index and doesn't want to invest in the entire index, it's time to carefully select the right stocks.

Several key economic factors should guide stock selection from an index:

- Diversification by Industry and Sector: Ensuring exposure across different sectors to reduce risk.
- Diversification by Correlation and Returns: Selecting stocks that offer varied returns and are not too closely correlated.
- Company Growth and Dividend Information: Choosing companies with strong growth potential and reliable dividend payouts.

Other factors, like recent projects, mergers, acquisitions, and broader macroeconomic or geopolitical policies, are also worth considering, though we won't delve into them here.

In this section, we will employ a range of data analysis techniques, along with our expertise in stock selection, to identify two suitable stocks from a pool of 18 for our client, Indra, to invest in.

Step 1: Create a Dataframe for historical stock performance

```
[ ] # Import all useful libraries
    import numpy as np
    import pandas as pd
```

```
[ ] # Flatten the list of lists into a single list of dictionaries
    flattened_data = [item for sublist in history_sminfra18 for item in sublist]

    # Convert to a pandas DataFrame
    df_history_sminfra18 = pd.DataFrame(flattened_data)
```

STOCK ANALYSIS SUMMARY

WORKFLOW WITH SECTORS FINANCIAL API

Explore Sectors Financial API Documentation and indentify available data endpoints

Choose specific endpoint that aligns your investment objectives: Daily Transaction Data, Companies by Index, Company Report

Understand the characteristics of each stock index and align these with your investor profile

Use modellings to select stocks: fundamental analysis, ML, **Portfolio Optimization**

Forecasting and Tracking profit and loss performance - possible topics for next workshop

IMPORTANT FACTORS TO CONSIDER

- **Alignment with your investment goals** to ensure risk tolerance over time.
- **Diversification Strategy** to spread risks and reduce impact of poor performance in single area.
- **Macroeconomic Factors** impact market sentiment and directly affect stock valuations.

MEAN VARIANCE TRADEOFF

MEAN VARIANCE COVARIANCE

THE MEAN VARIANCE TRADE OFF BALANCING RISK AND RETURN

Overview of Mean Variance Tradeoff

- The mean-variance trade-off is **the balance between expected return (mean) and risk (variance) in a portfolio**. Introduced by **Harry Markowitz**, this concept forms the basis of Modern Portfolio Theory.
- Key Idea is that investors want the **highest possible return for the lowest possible risk**. This trade-off guides investors to select portfolios that meet their desired balance of risk and return.

Mean, Variance, Covariance

- **Expected Return (Mean)**: This is **the weighted average** of the returns of the assets in a portfolio. The expected return gives investors an **estimate of the portfolio's potential return** based on historical data or projected growth.
- **Risk (Variance)**: Variance measures how much each asset's return deviates from the expected return, giving an **indication of volatility**. A high variance means the returns are spread out, resulting in greater risk.
- **Covariance**: Covariance assesses **how two assets move in relation** to each other. If the covariance is positive, the assets tend to move in the same direction, increasing risk.

DEMO - SINGLE STOCK EXPECTED RETURN

```python
import pandas as pd

# Define a dataset of stock prices for 3 stocks over 5 days
data = {
    "Stock A": [100, 102, 101, 103, 105],
    "Stock B": [50, 51, 52, 51, 53],
    "Stock C": [200, 198, 202, 205, 210],
}
prices = pd.DataFrame(data, index=["Day 1", "Day 2", "Day 3", "Day 4", "Day 5"])

# Calculate daily returns for each stock
daily_returns = prices.pct_change().iloc[1:]  # Drop the first row (NaN returns)

# Calculate the average daily return (expected return) for each stock
average_daily_returns = daily_returns.mean()

# Annualize the returns (assuming 252 trading days in a year)
annualized_returns = ((1 + average_daily_returns) ** 252 - 1) * 100  # Convert to percentage

# Display the results
results = pd.DataFrame({
    "Average Daily Return (%)": average_daily_returns * 100,
    "Annualized Return (%)": annualized_returns
})

print(results)
```

- **price_df.pct_change()**: Calculates the daily percentage change (daily returns) for each stock based on historical prices.
- **daily_returns.mean()**: Calculates the mean (average) of daily returns, providing the expected daily return for each stock.
- **Annualization**: To annualize, we compound the daily rate (exponential) by 252 (assuming there are 252 trading days in a year).

DEMO - PORTFOLIO EXPECTED RETURN

```
[1] # Expected Return = Sum (weights of assets in the porfolio Wi * Expected return of asset ri)

    import numpy as np

    # Portfolio weights for 3 assets
    weights = np.array([0.5, 0.3, 0.2])
    # Expected returns for each asset
    returns = np.array([0.08, 0.10, 0.12])

    # Calculate expected return of the portfolio
    expected_return = np.dot(weights, returns) #np.dot() use for matrix multiplication
    print("Expected Return:", expected_return)

    Expected Return: 0.094
```

In this example, we have a portfolio of three assets with weights 0.5, 0.3, and 0.2 and expected returns of 8%, 10%, and 12%, respectively. The expected return of the portfolio is the weighted sum of these returns, which is 9.6%.

DEMO - PORTFOLIO VARIANCE & COVARIANCE

```python
# Example price data for 5 stocks over 5 days
data = {
    'Stock_A': [100, 102, 101, 104, 103],
    'Stock_B': [50, 51, 49, 52, 50],
    'Stock_C': [200, 202, 198, 205, 204],
    'Stock_D': [75, 76, 74, 78, 77],
    'Stock_E': [300, 298, 301, 305, 302]
}
price_df_v = pd.DataFrame(data)

weights = np.array([0.2, 0.2, 0.2, 0.2, 0.2])

# Calculate daily returns
daily_returns_v = price_df_v.pct_change().dropna()

# Calculate the covariance matrix of daily returns
cov_matrix = daily_returns_v.cov()

# Calculate portfolio variance using the formula
# Portfolio Variance = weights.T * cov_matrix * weights
portfolio_variance = np.dot(weights.T, np.dot(cov_matrix, weights))

# Display results
print("Daily Returns:\n", daily_returns)
print("\nCovariance Matrix:\n", cov_matrix)
print("\nPortfolio Variance:", portfolio_variance)
```

Portfolio Variance = weights.T * cov_matrix * weights

Inner product: np.dot(cov_matrix, weights)

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{n} \sigma_{1j} \cdot w_j \\ \sum_{j=1}^{n} \sigma_{2j} \cdot w_j \\ \vdots \\ \sum_{j=1}^{n} \sigma_{nj} \cdot w_j \end{bmatrix}$$

Outer product: np.dot(cov_matrix, weights)

$$\text{portfolio\_variance} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix} \begin{bmatrix} \sum_{j=1}^{n} \sigma_{1j} \cdot w_j \\ \sum_{j=1}^{n} \sigma_{2j} \cdot w_j \\ \vdots \\ \sum_{j=1}^{n} \sigma_{nj} \cdot w_j \end{bmatrix} = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i \cdot w_j \cdot \sigma_{ij}$$

VARIANCE PRACTICE

- **Portfolio variance** = $w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2Cov_{1,2}$

Where:

- w_1 = the portfolio weight of the first asset
- w_2 = the portfolio weight of the second asset
- σ_1 = the standard deviation of the first asset
- σ_2 = the standard deviation of the second asset
- $Cov_{1,2}$ = the co-variance of the two assets, which can thus be expressed as $p_{(1,2)}\sigma_1\sigma_2$, where $p_{(1,2)}$ is the correlation co-efficient between the two assets

This is the formula for two stocks, for more stocks, we showed you how to do it through python in the previous slide.

Try it yourself

Assume there is a portfolio that consists of two stocks. Stock A is worth $50,000 and has a standard deviation of 20%. Stock B is worth $100,000 and has a standard deviation of 10%. The covariance between the two stocks is 0.85.

What is the variance? How do you interpret the result? Is this a “good” variance?

VARIANCE PRACTICE EXPLANATION

Variance = (33.3%^2 × 20%^2) + (66.7%^2 × 10%^2) + (2 × 33.3% × 20% × 66.7% × 10% × 0.85) = **1.64%**

What does Variance Represent?

- Variance quantifies how much the portfolio's returns deviate from its average (expected) return.
- In practice, it's often easier to interpret standard deviation (Square root of 1.64% is around 12.8%)

How to interpret variance?

- Low variance – less than 1%: low volatility, low risk very stable assets such as bonds.
- **Moderate variance**: 1% to 5%: Shows in balanced portfolios.
- High Variance is above 5%: High volatile, but also potential for high return – growth oriented stocks

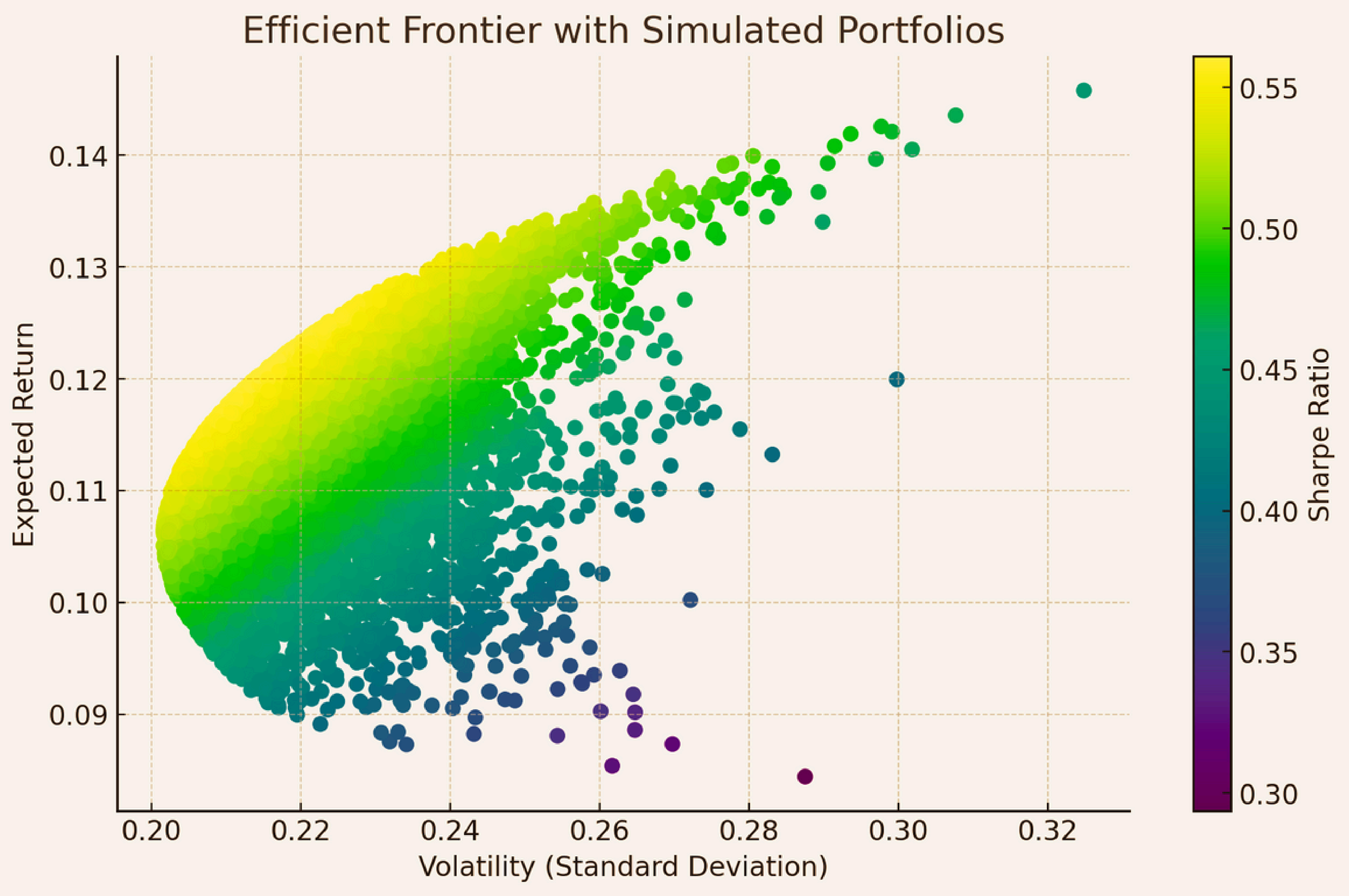
Is 1.64% a "good" variance?

- Variance alone is not enough, depends on the return. What if the return is only 1%? **MEAN VARIANCE TRADE OFF**
- To answer this question better, we will use **Sharpe Ratio for Risk Adjusted Return**

PORTFOLIO OPTIMIZATION

EFFICIENT FRONTIER

OPTIMIZING PORTFOLIOS TO MAXIMIZE RETURNS FOR GIVEN LEVELS OF RISK



Construct Efficient Frontier

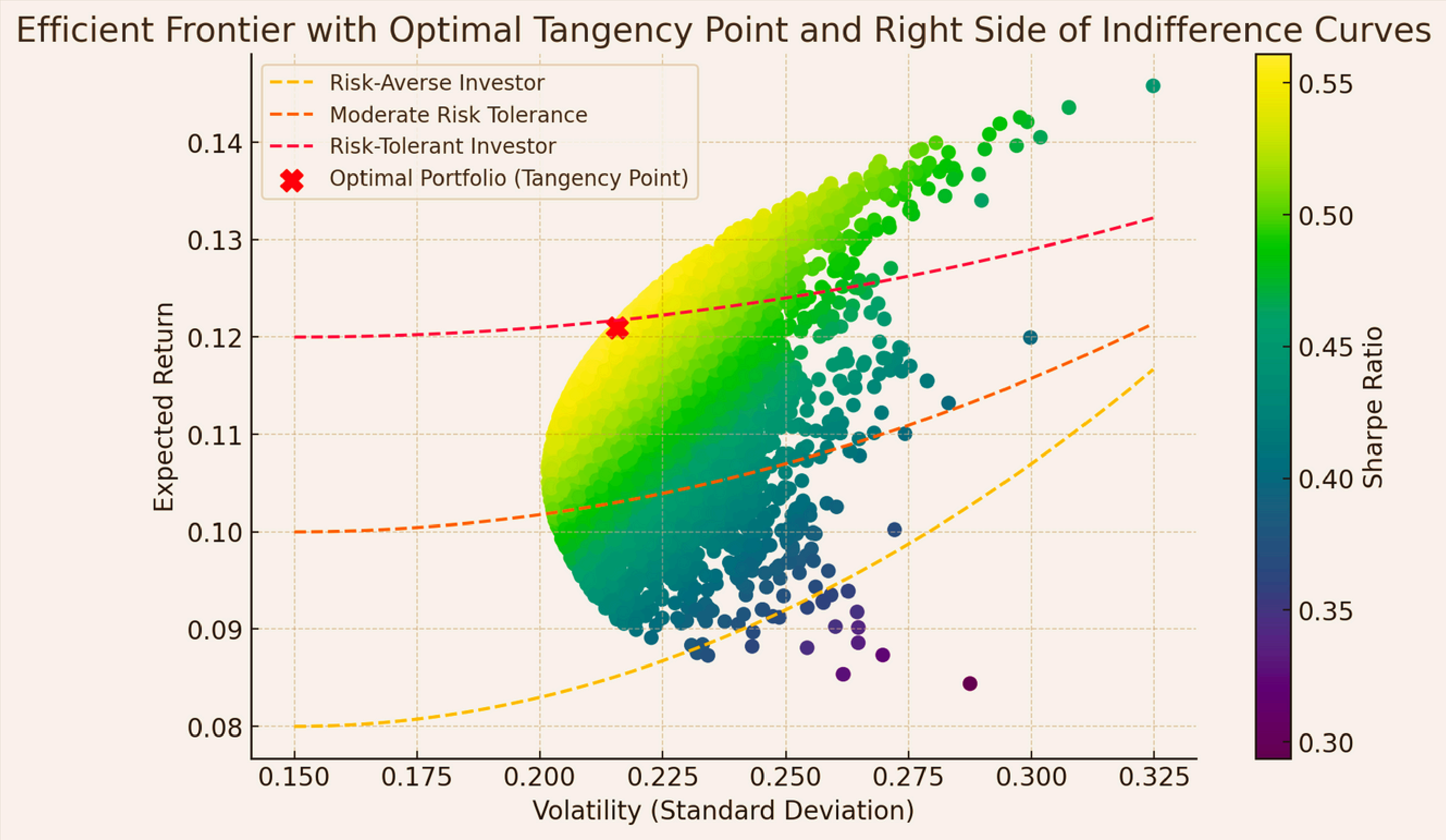
A portfolio is considered "efficient" if it provides the highest possible expected return for a specific level of risk or, conversely, the lowest possible risk for a desired level of return.

Markowitz Efficient Frontier is the set of all efficient portfolios achievable given a set of assets and their characteristics (expected returns, variances, and covariances).

- By **changing the weights** assigned to each asset, we can generate various combinations of expected returns and risks - Monte Carlo Simulation
- MVO is applied iteratively to find the weight sets that minimize risk for each return level or maximize return for each risk level.

INVESTOR INDIFFERENCE CURVE

ALIGNING INDIVIDUAL RISK PREFERENCES MAXIMIZING EFFICIENCY AND SATISFACTION



Investor Indifference Curve

Indifference curves are **upward-sloping** and **convex**, indicating that as risk increases, an investor requires a higher expected return to remain equally satisfied. Convexity shows **risk aversion**, as each additional unit of risk requires a disproportionately higher return to maintain the same satisfaction level.

Finding the optimal portfolio

The optimal portfolio for an investor lies at the tangent point where the highest possible indifference curve touches the Efficient Frontier. This tangency point represents the portfolio that **maximizes the investor's utility** by **offering the best risk-return tradeoff** aligned with their preferences.

COMBINE RISK FREE ASSETS

UNDERSTANDING AND CHOOSING RISK-FREE ASSETS IN PORTFOLIO ANALYSIS

1 Why Risk Free Assets

Customizing Risk Levels for Different Investors based on their individual risk level:

- **Risk-averse investors**: Allocate more to the risk-free asset.
- **Aggressive investors**: Allocate more to the Market Portfolio or even use leverage.
- Combining a risk-free asset with the Market Portfolio creates a portfolio that maximizes risk-adjusted return at any desired risk level.

2 Types of Risk Free Assets

- **Short-Term Government Bonds** (Treasury Bills) – Low Risk – Short Maturities
- **Long-Term Government Bonds** – minimal credit risk – higher interest rate risk – Long maturity
- **Bank Certificates of Deposits** (CDs) – fixed interest rates & fixed maturities . low risk
- Saving Accounts in Stable Currencies
- Repurchase Agreements

3 How to use for Analysis

- **Time Horizon Management**: short-term investments, a short-term Treasury bill or an equivalent low-risk asset is usually appropriate
- **Currency Alignment:** Ensure risk-free asset's currency matches the currency of the portfolio

Consider Market Convention!

SHARPE RATIO

MAXIMIZING RISK-ADJUSTED RETURNS THROUGH OPTIMAL PORTFOLIO ALLOCATION

THINK! WHICH ONE IS MORE EFFICIENT?

- Investment manager A

  Expected Return 15%, Standard Deviation 8%
- Investment manager B

  Expected Return 12%, Standard Deviation 5%

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

where:

R_p = return of portfolio

R_f = risk-free rate

σ_p = standard deviation of the portfolio's excess return

- The **Sharpe Ratio** is **a measure of risk-adjusted return**, indicating **how much excess return** an investor earns per unit of risk.
- **A higher Sharpe Ratio suggests a more efficient portfolio**, providing better compensation for each unit of risk.
- The risk-free rate provides a **baseline return** that investors can achieve with minimal risk. **Sharpe Ratio focuses on additional percentage returns earned from taking on risk of each unit of stock** (– Rf).

SHARPE RATIO - DIY

MAXIMIZING RISK-ADJUSTED RETURNS THROUGH OPTIMAL PORTFOLIO ALLOCATION

THINK! WHICH ONE IS MORE EFFICIENT? HOW TO INTERPRET SHARPE RATIO RESULT?

- **Investment manager A**: Expected Return 15%, Standard Deviation 8%
- **Investment manager B**: Expected Return 12%, Standard Deviation 5%
- **Assume a Risk Free Rate is 5%**
- Let's ignore the time, currency, Rf type, normal distribution for returns and other assumptions for now

- Normally a SHAPRE RATIO **above 1.0** is considered "good", **above 2.0** is excellent - depends on market conditions
- a SHARPE RATIO **below 1.0 may** indicate the porfolio's retunrs are not adequately compensating for its risk when comparing with risk-free returns

CAPITAL MARKET LINE

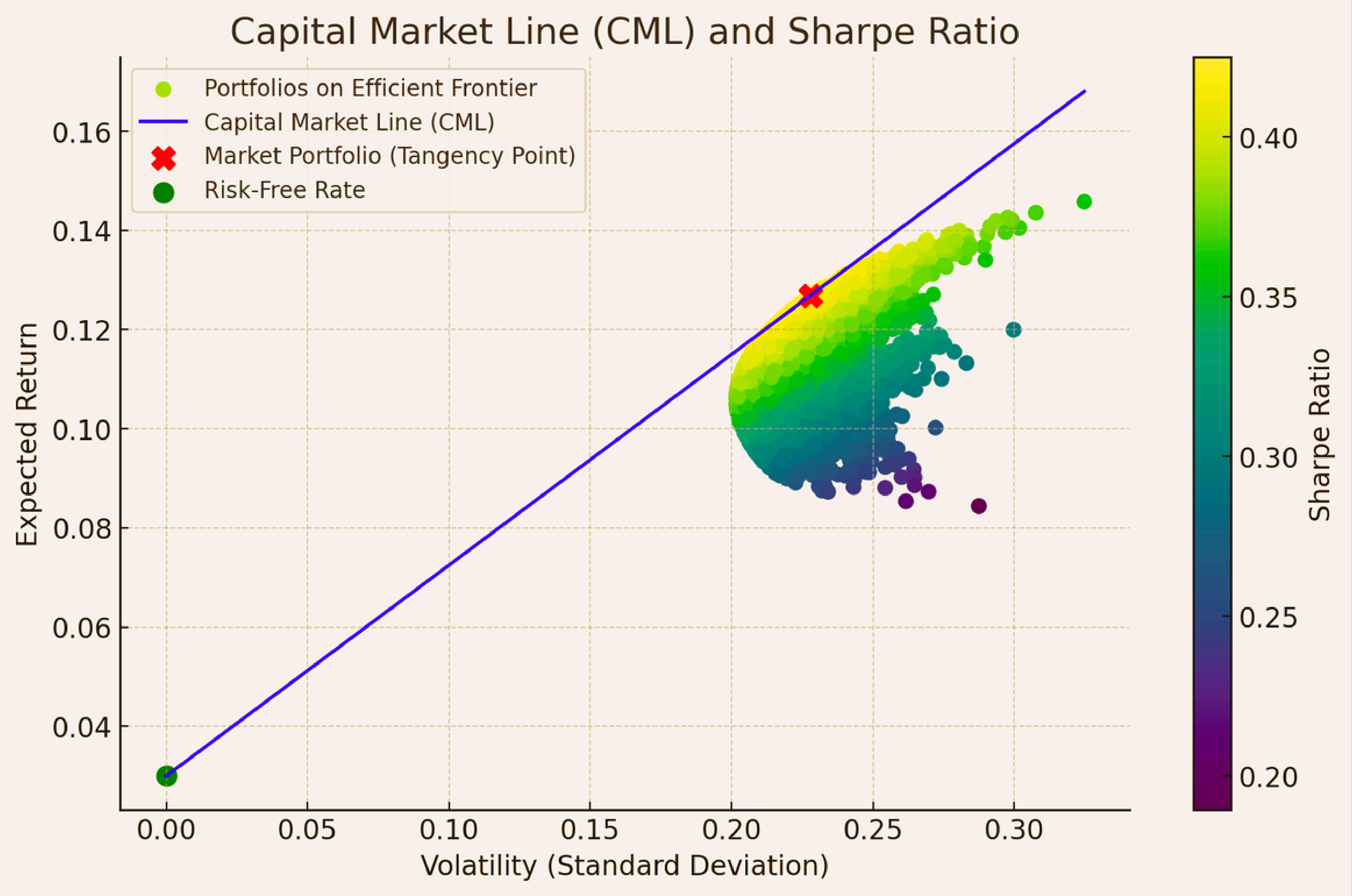
ACHIEVING OPTIMAL RISK-RETURN PORTFOLIOS WITH THE CML

- The Capital Market Line (CML) is **a line that represents the optimal combination** of **a risk-free asset and the Market Portfolio**.
- This **slope** is **the maximum achievable Sharpe Ratio** for risky assets, guiding investors to portfolios with optimal risk-return tradeoffs.

CML Formula:

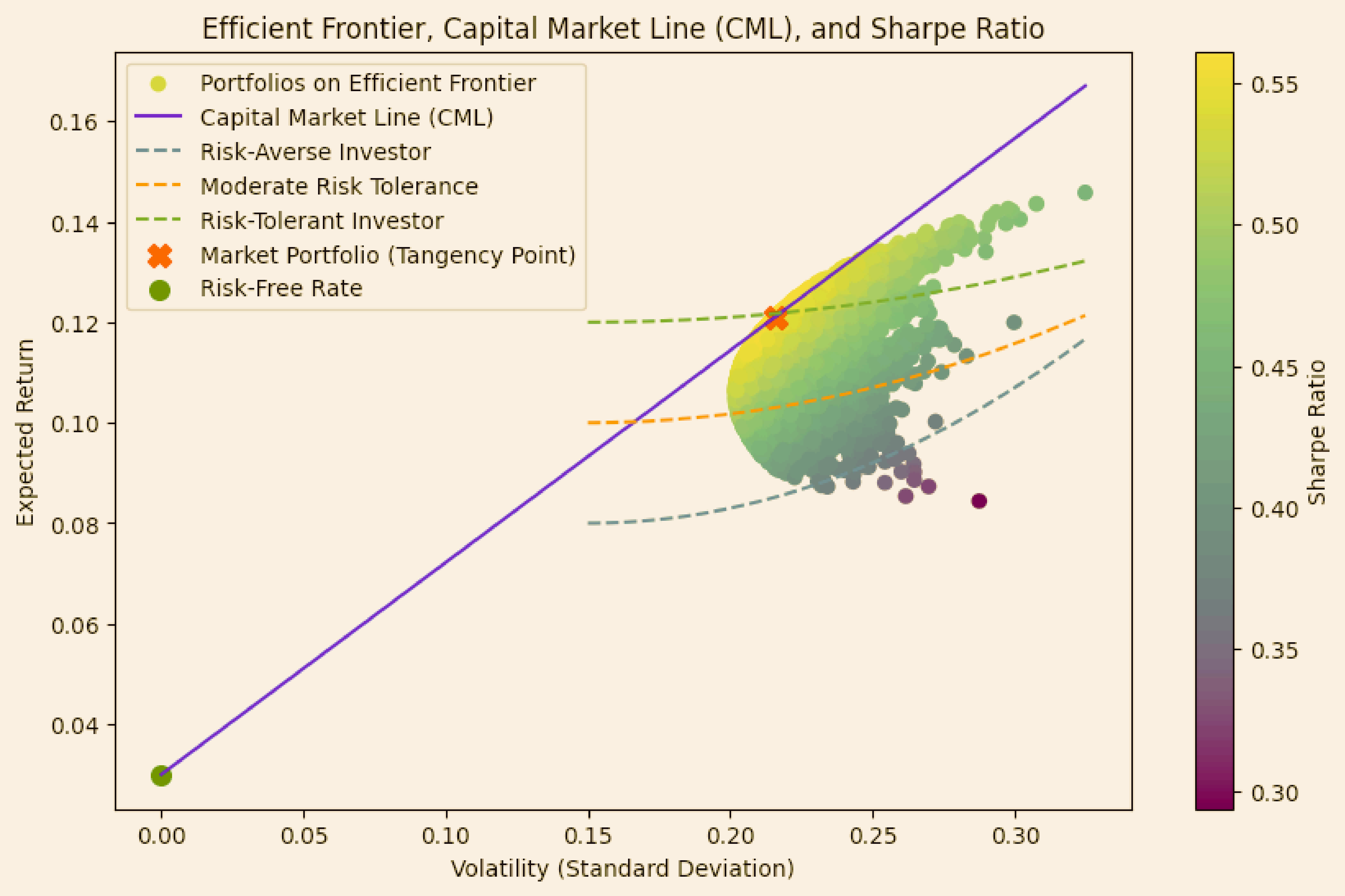
$$\text{Expected Return on Portfolio (CML)} = R_f + \frac{E(R_m) - R_f}{\sigma_m} \times \sigma_p$$

- R_f: Risk-free rate of return
- $E(R_m)$: Expected return of the Market Portfolio (the tangency point on the Efficient Frontier)
- σ_m: Standard deviation of the Market Portfolio
- σ_p: Standard deviation of the investor's portfolio on the CML



DEMO - INTEGRATING INVESTOR UTILITY

NTEGRATING THE EFFICIENT FRONTIER, CML, SHARPE RATIO AND INVESSTOR INDIFFERENCE CURVE



DEMO - MONTE CARLO SIMULATION MVO

Section 5.3 Monte Carlo Simulation

Monte Carlo simulation is a powerful technique used in portfolio optimization to assess the potential outcomes of different investment strategies or different allocations under varying conditions. It involves generating multiple scenarios based on statistical models and random sampling.

Implementing Monte Carlo simulation in Python involves combining statistical analysis, simulation, and optimization techniques to gain insights into portfolio performance under different allocations. For our analysis, we will run a simulation on different allocations of the same stocks to find the optimum allocation. A single run of the simulation is shown in the code below.

```
# Define Risk Free rate
# Assume RF as 6.64% as the yield of 10 year Indonesian Government bond
# Source link https://tradingeconomics.com/indonesia/government-bond-yield
rf = 0.0664

# Monte Carlo simulation
#Single Run
np.random.seed(101)
print(selected_stocks)

# Generates an array of random numbers representing initial weights for each asset in the portfolio
weights = np.array(np.random.random(4))
print('\nRandom Weights')
print(weights)

# Normalizing the randomly generated weights to ensure they sum up to 1, representing a fully invested portfolio.
print('\nRebalanced Weights')
weights = weights / np.sum(weights)
print(weights)
```

DEMO - SCIPY MINIMIZATION MVO

Section 6 - MVO with Scipy Minimized Function

Section 6.1 - Overview

In our next analysis method, we will optimize the same portfolio allocation mathematically using the minimize function in Scipy (a library in Python) and Sharpe ratio.

Portfolio optimization using Scipy's minimize function and the Sharpe ratio involves using mathematical optimization to find the optimal asset allocation that maximizes the Sharpe ratio—a measure of risk-adjusted returns.

The basic principle is to find the Sharpe ratio for a random allocation and then multiply it by -1 to make it negative and then minimize it to obtain the allocation weights that gives the highest Sharpe ratio.

```
[ ] def ret_vol_sr(weights):
      weights = np.array(weights)

      # Calculate Annualized Expected Returns
      ret = np.sum(returns.mean() * weights * 252)

      # Calculate Portfolio Volaticity
      vol = np.sqrt(np.dot(weights.T, np.dot(returns.cov() * 252, weights)))

      # Calculate Sharp Ratio, Rist Free rate 6.4%
      sr = (ret - rf) / vol

      return np.array([ret, vol, sr])
```

CHALLENGES

CHALLENGES

- **Implement the CML in MVO**: Incorporate the CML by adding a risk-free asset (e.g., short-term Treasury rate) to your MVO simulation. Identify the Market Portfolio as the tangency point with the Efficient Frontier and show how CML enhances portfolio choices – Combine Risk Free Assets in your optimization model

- **Create a Sensitivity Analysis for Mean and Variance Estimates**: Perform a sensitivity analysis by varying the expected returns and covariances in your Monte Carlo MVO. Analyze how small changes affect portfolio composition and efficiency – understand the influence of estimation error

- **Create Alternative Risk Measures**: Implement an alternative to the Sharpe Ratio, such as the **Sortino Ratio**, which considers downside risk only. Compare portfolios' performances using both ratios – Explore other performance matrix