Computational Investing with Python

Alexandre Landi[*]
IBM
Skema Business School
Georgia Institute of Technology
Balanced Research

January 8, 2024

Abstract

This lecture series, "Computational Investing with Python," is designed to provide a comprehensive overview of essential concepts in modern investment strategies and portfolio management, all through the lens of Python programming. The course begins by introducing foundational topics such as arithmetic and logarithmic return measurements, alongside various risk and reward measures including annualized returns, volatility, and ratios like Sharpe and Sortino. It delves into the Capital Asset Pricing Model (CAPM) to lay the groundwork for understanding asset pricing and risk management.

Building upon these basics, the series progresses into Modern Portfolio Theory (MPT), exploring diverse portfolio construction techniques. From the classic Equally Weighted and Minimum Variance portfolios to more sophisticated approaches like Mean-Variance, Black-Litterman, and Multi-Factor Models, the lectures provide both theoretical knowledge and practical Python coding examples. Special attention is given to advanced strategies like Risk-Parity and Beta-Neutral portfolios, including a unique take on optimizing the covariance matrix.

The final section focuses on back-testing methodologies, crucial for evaluating the performance of investment strategies. It covers techniques like Rolling and Expanding Windows, providing insights into their application in real-world scenarios. The series concludes with practical applications, tying together theory and computation, to equip students with the skills necessary to implement and assess their investment strategies in the dynamic world of finance.

Overall, this lecture series is tailored for those looking to blend financial theory with practical Python-based applications, offering a deep dive into computational investing strategies for both academic and professional advancement.

[*] Emails: alexandre.landi1@ibm.com, alexandre.landi@skema.edu, alandi3@gatech.edu, alexandre.landi@balanced-research.com

Contents

1 Measuring Returns

Imagine you are a portfolio manager entrusted with managing $10 millions. Your primary goal is to grow this amount through strategic investments, but how will you measure your performance? This is where the concept of returns comes into play, and understanding different methods of calculating returns is crucial for your success as a portfolio manager [Bodie et al., 2018].

1.1 Introducing Arithmetic Returns

Arithmetic returns, also known as simple returns, are the most straightforward way to measure the performance of your investments. Let's say you invest the entire $10 millions in one stock, and after a year, its value increases to $11 millions. Your arithmetic return would be calculated as follows:

$$\begin{aligned}\text{Arithmetic Return} &= \frac{\text{End Value} - \text{Beginning Value}}{\text{Beginning Value}} \\ &= \frac{\$11\text{M} - \$10\text{M}}{\$10\text{M}} \\ &= 0.1 \text{ or } 10\%\end{aligned}$$

This calculation seems intuitive and straightforward, right? However, arithmetic returns have limitations, especially when you are dealing with multiple time periods [Fabozzi et al., 2010].

1.2 The Problem with Arithmetic Returns

Arithmetic returns are not time-additive. This means that if you want to calculate the total return over a multi-year period, simply adding annual returns can lead to incorrect results. This limitation becomes apparent in volatile markets where investment values fluctuate significantly [Lo and MacKinlay, 1988].

For example, consider a scenario where your investment grows by 10% in the first year but then falls by 10% in the second year. The arithmetic return for each year would be +10% and -10%, respectively. You might think that your total return over these two years is 0% (since +10% - 10% = 0%). However, this is not the case because the 10% loss in the second year is on the increased amount from the first year, not the original investment.

Example:
Assume you start with a capital of $10 million. Your investment experiences different returns over two years:

- Year 1: The investment grows by 10%.
- Year 2: The investment falls by 10%.

Calculation Using Arithmetic Returns:

- **End of Year 1:**

$$\text{Capital} = \$10\text{M} \times (1 + 10\%) = \$11\text{M}$$

- **End of Year 2:**

$$\text{Capital} = \$11\text{M} \times (1 - 10\%) = \$9.9\text{M}$$

- **Total Arithmetic Return:**

$$10\% + (-10\%) = 0\%$$

Despite the arithmetic return suggesting a total return of 0% over the two years, the actual capital at the end of Year 2 is \$9.9 million, not the original \$10 million. This example clearly demonstrates that simply adding up annual arithmetic returns can lead to incorrect conclusions about the overall investment performance, particularly in volatile markets [Campbell and Shiller, 1988].

1.3 Addressing the Issue with Logarithmic Returns

Unlike arithmetic returns, logarithmic returns provide a more accurate representation of investment performance over multiple periods, especially in volatile markets. Let's revisit the previous example using logarithmic returns.

Example Revisited with Logarithmic Returns:
Assume the same investment scenario with a starting capital of \$10 million:

- Year 1: The investment grows by 10%.
- Year 2: The investment falls by 10%.

Calculation Using Logarithmic Returns:

- **End of Year 1:**

$$\text{Log Return Year 1} = \ln(1 + 10\%) = \ln(1.10)$$

- **End of Year 2:**

$$\text{Log Return Year 2} = \ln(1 - 10\%) = \ln(0.90)$$

- **Total Logarithmic Return:**

$$\begin{aligned}\text{Total} &= \ln(1.10) + \ln(0.90)\\ &\approx 0.095 + (-0.105)\\ &\approx -0.01 \text{ or } -1\%\end{aligned}$$

This looks much better, because $10 millions minus 1% of $10 millions equals $9.9 millions, which is the actual capital that we expected to have at the end of year 2. Eureka!

This approach accurately reflects the compounded effect of returns over time. Unlike the arithmetic method, logarithmic returns take into account the sequence of returns and the impact of gains and losses on the evolving investment value. As a result, the total logarithmic return provides a more realistic picture of the investment performance across multiple periods, especially in markets with significant fluctuations.

1.3.1 Another Example: Arithmetic vs. Logarithmic Returns

Now let's say that our $10 million portfolio changes over three years as follows:

- Year 1: $12 million
- Year 2: $9 million
- Year 3: $13.5 million

1.3.2 Arithmetic Returns Calculation

$$\begin{aligned}
\text{Year 1 Arithmetic Return} &= \frac{\$12M - \$10M}{\$10M} = 0.20\,(\text{or } 20\%)\\
\text{Year 2 Arithmetic Return} &= \frac{\$9M - \$12M}{\$12M} = -0.25\,(\text{or -25\%})\\
\text{Year 3 Arithmetic Return} &= \frac{\$13.5M - \$9M}{\$9M} = 0.50\\
\text{Total Arithmetic Return} &= 20\% - 25\% + 50\% = 45\%
\end{aligned}$$

Again, this doesn't look right. Our $10 million portfolio did not grow by 45%, but rather 35%. So now let's try with logarithmic returns:

1.3.3 Logarithmic Returns Calculation

$$\begin{aligned}
\text{Year 1 Logarithmic Return} &= \ln\left(\frac{\$12M}{\$10M}\right) = \ln(1.20)\\
\text{Year 2 Logarithmic Return} &= \ln\left(\frac{\$9M}{\$12M}\right) = \ln(0.75)\\
\text{Year 3 Logarithmic Return} &= \ln\left(\frac{\$13.5M}{\$9M}\right) = \ln(1.5)
\end{aligned}$$

$$\begin{aligned}\text{Total Logarithmic Return} &= \ln(1.20) + \ln(0.75) + \ln(1.5)\\ &\approx 0.18 + (-0.29) + 0.41\\ &\approx 0.30 \text{ or } 30\%\end{aligned}$$

That looks a little bit closer, but not quite right yet. We had \$10 millions at year 0 and we have \$13.5 millions at year 3, hence we would expect a cumulative return of 35%, not 30%.

1.3.4 Calculating Final Investment Value Using Logarithmic Returns

Logarithmic returns provide a nuanced way to measure investment performance over multiple periods. To convert the total logarithmic return to an actual final investment value, we need to follow specific steps. Let's see how this is done:

Steps to Calculate Final Value:
Assuming an initial investment value and a calculated total logarithmic return over multiple periods, the final investment value is computed as follows:

1. **Exponentiate the Total Logarithmic Return:** This step reverses the logarithmic operation, translating the compounded growth rate back to a multiplicative factor.

$$\text{Growth Factor} = e^{\text{Total Logarithmic Return}}$$

2. **Multiply by Initial Investment Value:** The growth factor is then applied to the initial value to get the final investment value.

$$\text{Final Investment Value} = \text{Initial Value} \times \text{Growth Factor}$$

Why Exponentiate the Total Logarithmic Return?
Logarithmic returns convert multiplicative growth rates into additive values, making them ideal for analyzing compounded growth over time. To revert these additive values back to a growth factor, we use exponentiation. Here's the algebraic rationale:

1. **Inverting the Logarithm:** Logarithmic returns represent the natural logarithm (ln) of growth ratios. The inverse of taking a logarithm is exponentiation (raising e to the power of the logarithmic return).

$$\begin{aligned}\text{If Log Return} &= \ln(\text{Growth Ratio})\\ \text{Then Growth Ratio} &= e^{\text{Log Return}}\end{aligned}$$

2. **Applying to Total Logarithmic Return:** The total logarithmic return over multiple periods represents the cumulative compounded growth in logarithmic terms. Exponentiating this value converts it back to a real growth factor.

$$\text{Growth Factor} = e^{\text{Total Logarithmic Return}}$$

Calculating the Final Investment Value:
Given the total logarithmic return and the initial investment value, the final value is calculated as follows:

1. **Exponentiate the Total Logarithmic Return:**

$$\text{Growth Factor} = e^{\text{Total Logarithmic Return}}$$

2. **Multiply by Initial Investment Value:**

$$\text{Final Investment Value} = \text{Initial Investment Value} \times \text{Growth Factor}$$

Example:
Let's apply these steps to our previous example:

- Initial Investment Value: $10 million
- Total Logarithmic Return over 3 Years: Approximately 0.30

1. Calculate the growth factor:

$$\text{Growth Factor} = e^{0.30}$$

2. Calculate the final investment value:

$$\begin{aligned}\text{Final Investment Value} &= \text{Initial Value} \times \text{Growth Factor}\\ &= \$10M \times e^{0.30}\\ &\approx \$10M \times 1.35\\ &\approx \$13.5M\end{aligned}$$

Yay! This calculation provides the actual value of the investment at the end of the period, reflecting the compounded growth captured by the logarithmic returns.

If you just wanted to know the percentage cumulative return, we would simply subtract 1 from the growth factor.

$$\begin{aligned}\text{Cumulative Return} &= \text{Growth Factor} - 1\\ &= e^{0.30} - 1\\ &\approx 1.35 - 1\\ &\approx 0.35 \text{ or } 35\%\end{aligned}$$

Eureka!

1.3.5 Comparison and Conclusion

While the arithmetic method suggests a total return of 45% over three years, the logarithmic method, which accurately accounts for compounding, indicates a different (and more accurate) cumulative return of 35%. This example highlights the importance of using logarithmic returns for a more accurate assessment of portfolio performance over multiple periods.

1.4 Python Exercise: Calculating and Visualizing Returns

You are given a dataset representing the annual closing values of an investment portfolio over a five-year period with more volatile values. Calculate both the arithmetic and logarithmic returns for each year, the total returns over the entire period, and visualize the cumulative returns to see the difference visually.

Annual closing values of the portfolio (in millions USD):

- Year 0: $10.0
- Year 1: $12.5
- Year 2: $8.0
- Year 3: $13.5
- Year 4: $7.5
- Year 5: $15.0

1.4.1 Calculate and Plot Annual Arithmetic and Logarithmic Returns

Requirements:

- Python
- Libraries: pandas, numpy, matplotlib

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

# Updated portfolio values
values = np.array([10.0, 12.5, 8.0, 13.5, 7.5, 15.0])

# Calculating arithmetic and logarithmic returns
arithmetic_returns = (values[1:] - values[:-1]) / values[:-1]
log_returns = np.log(values[1:] / values[:-1])

# Plotting the returns
plt.figure(figsize=(10, 6))
plt.plot(arithmetic_returns, label='Arithmetic Returns', marker='o')
plt.plot(log_returns, label='Logarithmic Returns', marker='x')
plt.title('Annual Arithmetic vs. Logarithmic Returns')
plt.xlabel('Year')
plt.ylabel('Returns')
plt.legend()
plt.grid(True)
plt.show()
```

1.4.2 Calculate and Plot Cumulative Returns

```python
# Calculating cumulative returns
cumulative_arithmetic_return = np.cumsum(arithmetic_returns)
cumulative_logarithmic_return = np.exp(np.cumsum(log_returns)) - 1

# Plotting cumulative returns
plt.figure(figsize=(10, 6))
plt.plot(cumulative_arithmetic_return, label='Cumulative Arithmetic Returns', marker='o')
plt.plot(cumulative_logarithmic_return, label='Cumulative Logarithmic Returns', marker='x')
plt.title('Cumulative Arithmetic vs. Logarithmic Returns')
plt.xlabel('Year')
plt.ylabel('Cumulative Returns')
plt.legend()
plt.grid(True)
plt.show()
```

1.4.3 Compare and Discuss

After plotting the returns, compare the graphs of arithmetic and logarithmic returns, as well as their cumulative returns. Discuss the visual differences and the implications of these differences for investment performance measurement.

1.4.4 Expected Outcomes

By now, you should be able to:

- Write a Python script to calculate, compare, and plot arithmetic and logarithmic returns.
- Visualize the difference in cumulative returns using matplotlib.
- Articulate the insights gained from the visual comparison of the two methods.

1.5 Python Exercise: S&P 500 Futures Data Analysis

In this exercise, you will apply the concepts of calculating and visualizing returns to real-world financial data. You will download daily data of S&P 500 front month futures from Yahoo Finance and perform a similar analysis as above.

1.5.1 Downloading the Data

Requirements:

- Python
- Libraries: pandas, numpy, matplotlib, yfinance

yfinance

To get started, you need to download the daily price data for S&P 500 futures. This can be done using Python's `pandas` and `yfinance` libraries. Here is a step-by-step guide to download the data and put it into a pandas dataframe:

```python
import yfinance as yf
import pandas as pd

# Downloading S&P 500 futures data
ticker = "ES=F"  # S&P 500 front-month futures ticker symbol
data = yf.download(ticker)

# Creating a pandas dataframe
sp500_data = pd.DataFrame(data)
print(sp500_data.head())  # Displaying the first few rows
```

1.5.2 Your Task

Using this data, calculate and plot the annual arithmetic and logarithmic returns of the S&P 500 futures. Then, analyze the cumulative returns over the entire period. You should:

1. **Calculate and Plot Cumulative Returns:** Similar to the previous exercise, calculate the cumulative arithmetic and logarithmic returns for the S&P 500 futures data.
2. **Compare and Discuss:** Analyze the plots and discuss the visual differences. What insights can you gain about the performance of the S&P 500 futures over this period?

1.5.3 Expected Learning Outcome

This exercise will help you to:

- Apply the concepts of returns calculation to real-world financial data.
- Develop skills in data handling and analysis using Python.
- Gain insights into the practical implications of different return measurement methods in financial markets.

Note: This exercise is exploratory and intended to deepen your understanding of financial data analysis. No solution is provided; you are encouraged to explore and interpret the data independently.

1.6 Quiz

For each question, select one or more correct answers.

1. How are arithmetic returns calculated for an investment?
   (a) $\frac{P_{\text{final}}}{P_{\text{initial}}} - 1$, where P_{initial} is the initial investment value and P_{final} is the value at the end of the period.
   (b) $\ln(\frac{P_{\text{final}}}{P_{\text{initial}}})$, where P_{initial} and P_{final} are the initial and final prices of the asset, respectively.
   (c) $\frac{P_{\text{final}} - P_{\text{initial}}}{P_{\text{initial}}}$, where P_{initial} is the initial investment value and P_{final} is the value at the end of the period.
   (d) The sum of the natural logarithm of the final price minus the natural logarithm of the initial price.
   (e) $\frac{P_{\text{final}}}{P_{\text{initial}}}$, where P_{final} and P_{initial} are the final and initial values of the investment.

2. Which of the following statements are true about the properties of logarithms?
   (a) $\ln(a \times b) = \ln(a) + \ln(b)$ for any positive numbers a and b.
   (b) $\ln\left(\frac{a}{b}\right) = \ln(a) - \ln(b)$ for any positive numbers a and b.
   (c) $\ln(a^b) = b \times \ln(a)$ for any positive number a and any number b.
   (d) $\ln(a + b) = \ln(a) + \ln(b)$ for any positive numbers a and b.
   (e) The natural logarithm of a negative number is undefined.

3. How are logarithmic returns calculated for an investment?
   (a) $\frac{P_{\text{final}} - P_{\text{initial}}}{P_{\text{initial}}}$, where P_{initial} and P_{final} are the initial and final prices of the asset, respectively.
   (b) $\ln(\frac{P_{\text{final}}}{P_{\text{initial}}})$, where P_{initial} is the initial investment value and P_{final} is the value at the end of the period.
   (c) $\ln(P_{\text{final}}) - \ln(P_{\text{initial}})$, where P_{initial} and P_{final} are the initial and final prices of the asset, respectively.
   (d) $\frac{P_{\text{final}}}{P_{\text{initial}}} - 1$, where P_{initial} is the initial investment value and P_{final} is the value at the end of the period.
   (e) $\frac{P_{\text{final}}}{P_{\text{initial}}}$, where P_{final} and P_{initial} are the final and initial values of the investment.

4. Based on the advantages and disadvantages of logarithmic and arithmetic returns, which of the following statements are true?
   (a) Arithmetic returns are more accurate than logarithmic returns for multi-period analysis due to their simplicity in calculation.

   (b) Logarithmic returns inherently account for the compounding effects and are more suited for analyzing multiple-period returns.
   (c) Arithmetic returns do not consider the compounding effect, which can lead to misleading results in volatile markets over multiple periods.
   (d) Logarithmic returns cannot be used for risk assessment and volatility analysis in financial markets.
   (e) The total logarithmic return for multiple periods is the product of the logarithmic returns of each period.

5. Given a dataframe of daily prices, which Python code snippet is most appropriate to compute the total logarithmic return over a multi-year period?
   (a) `total_log_return = df[’Price’].pct_change().sum()`
   (b) `total_log_return = np.log(df[’Price’] / df[’Price’].shift(1)).sum()`
   (c) `total_log_return = (df[’Price’].iloc[-1] / df[’Price’].iloc[0]) - 1`
   (d) `total_log_return = np.log(df[’Price’]).diff().sum()`
   (e) `total_log_return = (df[’Price’].pct_change() + 1).prod() - 1`

6. Consider an investment with the following annual closing values over three years, starting with a $10 million portfolio:
   - Year 1: $11 million
   - Year 2: $8 million
   - Year 3: $12 million

   Using logarithmic returns, calculate the total logarithmic return and the final investment value. Which of the following statements are correct?
   (a) The total logarithmic return is approximately 0.20, and the final investment value is about $12 million.
   (b) The total logarithmic return is approximately 0.18, and the final investment value is about $11.8 million.
   (c) The growth factor calculated from the total logarithmic return is about 1.22.
   (d) The final investment value is approximately $12.2 million.
   (e) The cumulative return over the three years is approximately 20%.

7. Given the following annual closing values of an investment portfolio (in millions USD):
   - Year 0: $10.0

- Year 1: \$13.0
- Year 2: \$7.0
- Year 3: \$14.0
- Year 4: \$8.0
- Year 5: \$16.0

Which of the following lines of code will correctly calculate the cumulative logarithmic return?

(a)
```
values = np.array([10.0, 13.0, 7.0, 14.0, 8.0, 16.0])
cumulative_arithmetic_return = np.sum((values[1:]  - values[:-1])
/ values[:-1])
```

(b)
```
values = np.array([10.0, 13.0, 7.0, 14.0, 8.0, 16.0])
cumulative_logarithmic_return = np.exp(np.sum(np.log(values[1:]
/ values[:-1])))
```

(c)
```
values = np.array([10.0, 13.0, 7.0, 14.0, 8.0, 16.0])
cumulative_arithmetic_return = np.prod((values[1:]  / values[:-1])
- 1)
```

(d)
```
values = np.array([10.0, 13.0, 7.0, 14.0, 8.0, 16.0])
cumulative_logarithmic_return = np.exp(np.sum(np.log(values[1:]
/ values[:-1]))) - 1
```

(e)
```
values = np.array([10.0, 13.0, 7.0, 14.0, 8.0, 16.0])
cumulative_arithmetic_return = np.cumsum((values[1:]  - values[:-1])
/ values[:-1])[-1]
```

1.7 Harmonizing Performance Measurement

1.7.1 The Problem: Comparing Returns Over Different Time Periods

Consider three different investments, each with returns reported over different time frames. How can you know which one performed best or worst?

Investment Scenarios

- **Investment A:** Reported a return of 6% in one month.
- **Investment B:** Reported a return of 15% over three months.
- **Investment C:** Reported a return of 25% over a year.
- **Investment D:** Reported a return of 0.2% in one day.
- **Investment E:** Reported a return of 1.5% over one week.

1.7.2 The Solution: Annualizing Logarithmic Returns

To fairly compare the performance of these investments over different time frames, we utilize the concept of annualizing logarithmic returns. This method allows us to standardize returns to a common time frame (one year), taking into account the effect of compounding.

Understanding Logarithmic Returns

Logarithmic returns, represented as $\ln(1+r)$ where r is the return, are particularly useful when dealing with compound interest and fluctuating investment values. They allow for additive properties over time, making them ideal for aggregation over different periods.

Annualizing the Returns

The annualization of logarithmic returns involves scaling the log returns to a yearly basis. This is done by multiplying the log return by the number of respective periods in a year.

Calculating Annualized Returns for Each Investment

Investment A (Monthly):

$$\begin{aligned}\text{Logarithmic Return} &= \ln(1+0.06)\\ \text{Annualized Return} &= \ln(1+0.06)\times 12 \quad \text{(12 months in a year)}\end{aligned}$$

Investment B (Quarterly):

$$\begin{aligned}\text{Logarithmic Return} &= \ln(1+0.15)\\ \text{Annualized Return} &= \ln(1+0.15)\times 4 \quad \text{(4 quarters in a year)}\end{aligned}$$

Investment C (Annually):

$$\text{Logarithmic Return} = \ln(1 + 0.25) \quad \text{(Already annual)}$$

Investment D (Daily):

$$\begin{aligned}\text{Logarithmic Return} &= \ln(1 + 0.002)\\ \text{Annualized Return} &= \ln(1 + 0.002) \times 252 \quad \text{(252 trading days in a year)}\end{aligned}$$

Investment E (Weekly):

$$\begin{aligned}\text{Logarithmic Return} &= \ln(1 + 0.015)\\ \text{Annualized Return} &= \ln(1 + 0.015) \times 52 \quad \text{(52 weeks in a year)}\end{aligned}$$

Interpreting the Results

The annualized logarithmic returns provide a standardized way to compare the investments. A higher annualized return indicates better performance over the year, irrespective of the original investment period. This method accounts for the compound nature of returns, offering a more accurate measure of performance, especially for volatile or inconsistent returns over time.

1.8 Conclusion

The annualization of logarithmic returns is a powerful tool in performance measurement, harmonizing the comparison of returns across different investment time frames. It ensures a level playing field, allowing investors and analysts to make informed decisions based on standardized and comparable data.

1.9 Python Exercise: Downloading S&P500 Futures Prices and Calculating Annualized Mean Logarithmic Return

Objective: Download the front month S&P500 futures prices, compute the logarithmic returns, and then annualize the mean return.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance

Steps:

1. **Install Required Libraries:** Ensure you have the necessary libraries installed. You can install them using pip:

```
pip install pandas numpy yfinance
```

```python
# Import libraries
import pandas as pd
import numpy as np
import yfinance as yf

# Use the ‘yfinance‘ library to download the front month S&P500 futures price data.
sp500_futures = yf.download('ES=F')

# Calculate the daily logarithmic returns of the futures prices.
sp500_futures['Log_Returns'] = np.log(sp500_futures['Adj Close'] / sp500_futures['Adj Close'].shift(1))

# Annualize the mean of the logarithmic returns.
annualized_return = sp500_futures['Log_Returns'].mean() * 252
print("annualized_return:", annualized_return)

```

1.10 Limitations of the Annualized Mean Return as a Metric

The annualized mean return is a key metric in finance for assessing the average performance of an investment over a standardized one-year period. Despite its prevalence, it comes with several notable limitations:

- **Risk Ignorance:** The metric solely focuses on returns, disregarding the associated risks or volatility. Investments with identical annualized mean returns could have vastly different risk profiles.

- **Past Performance and Future Returns:** It is based on historical data, assuming continuity in performance. However, past returns are not always indicative of future outcomes.

- **Uniform Return Distribution:** The annualized mean return does not account for the distribution of returns, neglecting the possibility of varying skewness and kurtosis.

- **Time Frame Sensitivity:** The return is highly sensitive to the chosen analysis period. Different start and end dates can lead to disparate results due to market fluctuations and economic conditions.

- **Inflation and Real Returns:** Inflation impacts are typically not considered, which can significantly affect the real return on an investment.

In conclusion, while the annualized mean return provides a baseline for investment performance evaluation, it should be complemented with other metrics that consider risk, inflation, and specific investment characteristics for a holistic analysis.

1.11 Python Exercise: Visualizing S&P 500 Futures Logarithmic Returns Distribution

Objective: Download the front month S&P500 futures prices using the ‘yfinance‘ library, compute the daily logarithmic returns, and visualize the distribution of these returns in a histogram. Additionally, plot a vertical dashed line to indicate the mean of the daily logarithmic returns.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance, matplotlib

Steps:

1. **Install Required Libraries:** Ensure you have the necessary libraries installed. Install them using pip if needed:

```
pip install pandas numpy yfinance matplotlib
```

```python
# Import libraries
import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker

# Download front month S&P500 futures price data
sp500_futures = yf.download('ES=F')

# Calculate daily logarithmic returns of the futures prices
sp500_futures['Log_Returns'] = np.log(sp500_futures['Adj Close']).diff()

# Determine the number of bins for the histogram
max_daily_return = np.round(sp500_futures['Log_Returns'].max(), 2)
min_daily_return = np.round(sp500_futures['Log_Returns'].min(), 2)
n_bins = int((max_daily_return - min_daily_return) * 100)

# Plot a histogram of the logarithmic returns
plt.hist(sp500_futures['Log_Returns'], bins=n_bins, color="orange", label="Return Distribution")

# Add a vertical dashed line to indicate the mean of the returns
plt.axvline(sp500_futures['Log_Returns'].mean(), color='black', linestyle='dashed', linewidth=2, label="Daily Mean Return")

```

```python
    # Configure the plot
    plt.title('Histogram of S&P 500 Futures Logarithmic Returns')
    plt.xlabel('Logarithmic Returns')
    plt.ylabel('Frequency (log)')
    plt.yscale("log")
    plt.gca().yaxis.set_major_formatter(ticker.StrMethodFormatter('{x:,.0f}'))  # Format y-axis labels as integers
    plt.legend()

    # Display the plot
    plt.show()

```

1.12 Quiz

For each question, select one or more correct answers.

1. Given the following investment scenarios with different time frames, calculate the annualized logarithmic returns and identify which investments have the highest and lowest annualized returns.

    - Investment F: Reported a return of 7% in one month.
    - Investment G: Reported a return of 10% over two months.
    - Investment H: Reported a return of 18% over six months.
    - Investment I: Reported a return of 0.1% in one day.
    - Investment J: Reported a return of 2% over one week.

    Which of the following statements are correct?

    (a) Investment F has the highest annualized return.
    (b) Investment I has the lowest annualized return.
    (c) Investment G has a higher annualized return than Investment H.
    (d) Investment J has a higher annualized return than Investment F.
    (e) Investment H has the highest annualized return.

2. When calculating the annualized mean return using logarithmic returns for a dataset of daily returns, which of the following statements are correct?

    (a) The annualized mean return can be calculated by summing the logarithmic returns and multiplying by 252.
    (b) To calculate the logarithmic return for each day, you divide the price at the end of the day by the price at the beginning of the day.
    (c) The annualized mean return provides an average rate of return per month.

(d) Logarithmic returns are preferred as they can be easily summed over time.

(e) The annualized mean return is calculated using simple returns and does not need any scaling factor.

3. Which of the following Python code snippets correctly calculates and annualizes the logarithmic returns of an investment based on daily price data?

    (a) `annualized_return = np.log(df[’Close’] / df[’Close’].shift(1)).mean() * 252`

    (b) `annualized_return = df[’Close’].pct_change().mean() * 252`

    (c) `annualized_return = np.sum(np.log(df[’Close’] / df[’Close’].shift(1))) / len(df) * 252`

    (d) `annualized_return = np.log(df[’Adj Close’] / df[’Adj Close’].shift(1)).mean() * 252`

    (e) `annualized_return = (df[’Close’].iloc[-1] / df[’Close’].iloc[0] - 1) * 252`

4. Which of the following are limitations of using the annualized mean return as a metric for assessing investment performance?

    (a) It does not consider the risk or volatility associated with the investment.

    (b) It assumes that the future performance of an investment will mirror its past performance.

    (c) It accounts for the distribution of returns, including skewness and kurtosis.

    (d) The metric is sensitive to the specific time frame chosen for analysis.

    (e) It adjusts the returns for inflation to provide a real return measure.

5. Which of the following lines of code are correct for visualizing the distribution of S&P 500 Futures Logarithmic Returns?

    (a) `plt.hist(sp500_futures[’Log_Returns’])`

    (b) `plt.axvline(sp500_futures[’Log_Returns’].mean(), linestyle=’dashed’, linewidth=2)`

    (c) `plt.ylabel(’Frequency (log)’); plt.yscale("log")`

    (d) `sp500_futures[’Log_Returns’] = np.log(sp500_futures[’Adj Close’] / sp500_futures[’Adj Close’].shift(1))`

    (e) `plt.legend()`

2 Measuring Risk

Now, suppose you are the Head of Portfolio Management at an asset management firm with \$2 billions in assets under management (AUM), and you need to hire a portfolio manager to launch a new fund. From a head hunter, you received the resumes of two candidates, along with a plot of their past performance.

How will you choose between the two portfolio manager candidates? Most likely, it will depend on the risk appetite of the investors who are subscribing to the new fund. In the previous sections, we discussed how to measure returns, but how do we measure risk?

2.1 Volatility

A common method to measure risk in finance is by computing the standard deviation of returns, also known as volatility. Volatility measures the degree of variation in the returns of an investment over a period. High fluctuations in returns indicate higher risk. Here is how volatility is calculated:

1. **Historical Return Data:** Collect the historical return data of the investment over the desired period.

2. **Calculate the Mean Return:** Compute the average return by summing up the individual returns and then dividing by the number of returns. For a specific example with 5 periods, the mean return is calculated as follows:

$$\text{Mean Return} = \frac{\text{Return}_1 + \text{Return}_2 + \text{Return}_3 + \text{Return}_4 + \text{Return}_5}{5}$$

For a more general case with n periods, where n could be any number of periods, the formula would be expanded as:

$$\text{Mean Return} = \frac{\text{Return}_1 + \text{Return}_2 + \ldots + \text{Return}_n}{n}$$

Here, the ellipsis $(\ldots)$ represents the continuation of the summation process up to the n-th term.

Alternatively, using sigma notation for this general case, the mean return is expressed as:

$$\text{Mean Return} = \frac{1}{n}\sum_{i=1}^{n} \text{Return}_i$$

Where:

- $\sum$ represents the sum.
- $i = 1$ to n indicates that the summing is done from the first period to the n-th period.

- Return_i is the return for the i-th period.
- n is the total number of periods.

3. **Calculate Deviations from the Mean:** For each period's return, calculate its deviation from the mean return. In the context of computing volatility, calculating the deviation of each period's return from the mean return is essential for measuring how much the returns fluctuate around their average, providing a foundational basis for quantifying investment risk.

$$\text{Deviation} = \text{Return}_i - \text{Mean Return}$$

4. **Square Each Deviation:** Square the deviations to eliminate negative values. In the context of computing volatility, we square each deviation to ensure all values are positive (as negative and positive deviations would otherwise cancel each other out), providing a consistent measure of how far returns deviate from the mean, regardless of direction.

$$\text{Squared Deviation} = (\text{Return}_i - \text{Mean Return})^2$$

5. **Average of Squared Deviations:** Compute the average of these squared deviations, known as variance. In computing volatility, we calculate the average of squared deviations (variance) to determine the typical squared deviation from the mean, providing a consistent measure of spread in returns.

$$\text{Variance} = \frac{\sum(\text{Return}_i - \text{Mean Return})^2}{\text{Number of Returns} - 1}$$

Dividing by $n - 1$ instead of n (known as Bessel's correction) provides a more accurate estimate of a population's variance from a sample by compensating for the fact that we're using the sample's mean (not the true population mean), which can otherwise underestimate the actual variability.

6. **Standard Deviation (Volatility):** Take the square root of the variance to convert the measure back to the same unit as the original returns, thus providing a more interpretable and directly comparable measure of variability.

$$\text{Volatility} = \sqrt{\text{Variance}}$$

Intuitive Explanation: Imagine hitting a target with several attempts. If most hits are close to the bullseye (the average), the volatility is low, indicating less risk. If the hits are widely scattered, the volatility is high, implying higher risk. In the context of hiring a portfolio manager, you would assess each manager's past portfolio performance by calculating the volatility of their returns. The choice between a manager with lower volatility and one with higher volatility will align with the risk preferences of your investors.

2.1.1 Annualizing Volatility

Annualizing volatility is a process of scaling the calculated standard deviation of returns to an annual basis, regardless of the original period of the returns. This standardization allows for a consistent comparison across investments with different time frames. Here's why and how we scale by the square root of the number of periods:

1. **Scaling Process:** To annualize volatility, we multiply the standard deviation (volatility) of the returns by the square root of the number of periods in a year. The formula for annualizing volatility is given by:

   $$\text{Annualized Volatility} = \text{Volatility} \times \sqrt{\text{Number of Periods in a Year}}$$

   Where:

   - Volatility is the standard deviation of the returns calculated over the original time period.
   - Number of Periods in a Year depends on the frequency of the data (e.g., 252 for daily, 52 for weekly, 12 for monthly).

   Wait, wait, wait! Where did that square root come from? Why are we multiplying by the square root of the number of periods in a year rather than simply multiplying by the plain number of periods in a year, just like we did when we were annualizing returns in the previous section?

2. **Explanation of the Square Root in Annualizing Volatility:** The reason we multiply by the square root of the number of periods in a year when annualizing volatility is because we are effectively taking the square root of the annualized variance. Recall that volatility is the standard deviation, which is the square root of variance. When we annualize the variance, we multiply the variance by the number of periods in a year. To convert this annualized variance back to volatility (standard deviation), we must take its square root.

   - The annualized variance is given by:

     $$\text{Annualized Variance} = \text{Variance} \times \text{Number of Periods in a Year}$$

   - Applying the square root to both sides of the equation, we use the property of square roots that $\sqrt{a \times b} = \sqrt{a} \times \sqrt{b}$:

     $$\sqrt{\text{Annualized Variance}} = \sqrt{\text{Variance} \times \text{Number of Periods in a Year}}$$
     $$\sqrt{\text{Annualized Variance}} = \sqrt{\text{Variance}} \times \sqrt{\text{Number of Periods in a Year}}$$
     $$\text{Annualized Volatility} = \text{Volatility} \times \sqrt{\text{Number of Periods in a Year}}$$

- This is why the square root of the number of periods in a year is used in the formula to annualize volatility. It ensures that the scaling of variance to an annual basis is correctly converted back to the standard deviation, providing an accurate measure of annualized volatility.

This explanation clarifies the mathematical reasoning behind the use of the square root in the process of annualizing volatility, ensuring that the scaling aligns with the statistical properties of variance and standard deviation.

3. **Examples:**

   - **Daily:** For daily returns, there are approximately 252 trading days in a year. Therefore, the annualized volatility is:

   $$\text{Annualized Volatility} = \text{Volatility} \times \sqrt{252}$$

   - **Weekly:** With about 52 weeks in a year, the annualized volatility for weekly returns is:

   $$\text{Annualized Volatility} = \text{Volatility} \times \sqrt{52}$$

   - **Monthly:** Given 12 months in a year, for monthly returns, the annualized volatility is:

   $$\text{Annualized Volatility} = \text{Volatility} \times \sqrt{12}$$

   - **Annual:** For annual returns, the volatility is already on an annual basis, so no scaling is necessary:

   $$\text{Annualized Volatility} = \text{Volatility}$$

This scaling allows us to compare the risk (volatility) of different investments on an equal footing, regardless of the time frame of their returns. It's particularly useful in portfolio management for making informed decisions about risk management and asset allocation.

2.1.2 Example: Computing Annualized Volatility from Monthly Returns

In this example, we will compute the annualized volatility using a set of monthly returns. The process involves several steps, each detailed below:

1. **Convert Percentage Returns to Decimal**: Given the monthly returns in percentages: [5, -3, 7, 6, -4, 2, 3, 7, -5, 4, -6, 8], we convert them to decimal form:

   $$\text{Decimal Returns} = [0.05, -0.03, 0.07, 0.06, -0.04, 0.02, 0.03, 0.07, -0.05, 0.04, -0.06, 0.08]$$

2. **Calculate the Mean Return**: The mean return is calculated as the average of these decimal returns:

$$\text{Mean Return} = \frac{0.05 - 0.03 + 0.07 + 0.06 - 0.04 + 0.02 + 0.03 + 0.07 - 0.05 + 0.04 - 0.06 + 0.08}{12}$$

$$\text{Mean Return} = \frac{0.24}{12} = 0.02$$

3. **Calculate Deviations from the Mean**: We calculate the deviation of each return from the mean:

$$\text{Deviation} = \text{Return}_i - 0.02$$

Resulting in deviations: [0.03, -0.05, 0.05, 0.04, -0.06, 0.00, 0.01, 0.05, -0.07, 0.02, -0.08, 0.06]

4. **Square Each Deviation**: Each deviation is squared:

$$\text{Squared Deviation}_i = (\text{Deviation}_i)^2$$

Resulting in squared deviations: [0.0009, 0.0025, 0.0025, 0.0016, 0.0036, 0.0000, 0.0001, 0.0025, 0.0049, 0.0004, 0.0064, 0.0036]

5. **Calculate the Variance**: The variance is the average of these squared deviations:

$$\text{Variance} = \frac{\sum \text{Squared Deviations}}{11}$$

$$\text{Variance} = \frac{0.0009 + 0.0025 + 0.0025 + ... + 0.0004 + 0.0064 + 0.0036}{11} \approx 0.00263636$$

6. **Calculate the Standard Deviation (Volatility)**: The standard deviation (volatility) is the square root of the variance:

$$\text{Volatility} = \sqrt{0.00263636} \approx 0.05135$$

7. **Annualize the Volatility**: The annualized volatility is then calculated by multiplying the monthly volatility by the square root of 12:

$$\text{Annualized Volatility} = 0.05135 \times \sqrt{12} \approx 0.17787$$

Therefore, the annualized volatility for these larger monthly returns is approximately 0.17787 or 17.787%.

This detailed step-by-step computation provides an accurate measure of the annualized volatility based on monthly returns, offering a standardized metric for comparing risk across various investments.

2.2 Python Exercise: Downloading S&P500 Futures Prices and Calculating Annualized Volatility

Objective: Download the front month S&P500 futures prices, compute the logarithmic returns, and then calculate the annualized volatility.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance

Steps:

1. **Install Required Libraries:** Ensure you have the necessary libraries installed. Install them using pip if needed:

```
pip install pandas numpy yfinance
```

```python
# Import libraries
import pandas as pd
import numpy as np
import yfinance as yf

# Downloading S&P 500 front month futures price data using yfinance
sp500_futures = yf.download(’ES=F’)

# Calculate the daily logarithmic returns
sp500_futures[’Log_Returns’] = np.log(sp500_futures[’Adj Close’] / sp500_futures[’Adj Close’].shift(1))

# Calculate the standard deviation (volatility) of the logarithmic returns
volatility = sp500_futures[’Log_Returns’].std()

# Annualize the volatility
# There are approximately 252 trading days in a year
annualized_volatility = volatility * np.sqrt(252)
print("Annualized Volatility:", annualized_volatility)

```

Note: This exercise teaches you how to calculate the annualized volatility, a key measure of risk, of S&P 500 futures. Understanding volatility is crucial in financial markets for assessing the risk associated with investments.

2.3 Limitations of the Annualized Volatility as a Metric

The annualized volatility is widely used in finance to quantify the variability or risk associated with the return of an investment over a one-year period. While it is a valuable tool for assessing risk, this metric has several inherent limitations:

- **Assumption of Normal Distribution:** When using annualized volatility as a metric, too many investors and academics often rely on the assumption that returns are normally distributed. However, financial markets frequently exhibit non-normal characteristics, such as skewness and leptokurtosis (heavy tails), leading to an underestimation of extreme market movements.

- **Focus on Total Volatility:** This metric considers both upside and downside volatility equally, which may not align with the perspectives of risk-averse investors who are more concerned about downside risk.

- **Ignores Asymmetry in Returns:** Annualized volatility does not distinguish between periods of high positive returns and periods of high negative returns, treating all deviations from the mean the same way.

- **Time Period Sensitivity:** Similar to annualized mean returns, the annualized volatility is also sensitive to the chosen time frame for analysis. Volatility can vary greatly over different market cycles or economic conditions.

In conclusion, while annualized volatility serves as a critical indicator of risk in investment analysis, it is important to recognize its limitations. Investors should ideally use it in conjunction with other risk metrics and analyses that take into account the asymmetry of returns, non-normal distribution characteristics, and the specific nature of the investment instruments.

2.4 Python Exercise: Analyzing S&P 500 Futures Logarithmic Returns Distribution with Standard Deviation Lines

Objective: Use Python to download the front month S&P500 futures prices, calculate the daily logarithmic returns, and create a histogram to visualize these returns. Enhance the histogram by adding vertical lines to represent the mean and standard deviations (one, two, and three standard deviations away from the mean) on both sides.

Requirements:

- Python

- Libraries: pandas, numpy, yfinance, matplotlib

Instructions:

1. First, ensure you have the necessary Python libraries (pandas, numpy, yfinance, and matplotlib). Install them using pip if they're not already installed.

2. Import the required libraries in your Python script.

3. Utilize the ‘yfinance‘ library to download the front month S&P500 futures price data. Store the downloaded data in a DataFrame.

4. Calculate the daily logarithmic returns of the S&P500 futures prices. To do this, use the natural logarithm (‘np.log()‘) of the ’Adj Close’ price divided by its previous day’s value. The ‘diff()‘ method may be helpful here.

5. Compute the mean of the logarithmic returns using the ‘.mean()‘ method, and store it in a variable named ‘mean_return‘.

6. Calculate the standard deviation of the logarithmic returns using the ‘.std()‘ method. The code snippet for this and the following steps is provided:

```
std_return = sp500_futures[’Log_Returns’].std()
# Adding vertical lines for mean and standard deviations
plt.axvline(mean_return, color=’black’, linestyle=’dashed’, linewidth=2, label="Mean Return")
for i in range(1, 4):
    plt.axvline(mean_return + i * std_return, color=’green’, linestyle=’dashed’, linewidth=1, label=f"+{i} STD")
    plt.axvline(mean_return - i * std_return, color=’red’, linestyle=’dashed’, linewidth=1, label=f"-{i} STD")

```

7. Determine the number of bins for the histogram based on the range of the logarithmic returns. Consider using ‘np.round()‘ to round the maximum and minimum log returns and then calculate the number of bins.

8. Create a histogram of the logarithmic returns using ‘plt.hist()‘. Ensure to label it appropriately.

9. Add the vertical lines for the mean and standard deviations on the histogram, using the provided code snippet.

10. Configure your plot with a title, labels for the x-axis and y-axis, and a legend. Set the y-axis scale to logarithmic and format the y-axis labels to display integers.

11. Finally, display the plot using ‘plt.show()‘.

In this exercise, you will visually analyze the distribution of the S&P 500 Futures Logarithmic Returns and understand how the returns deviate from the mean, providing insight into the volatility of the asset.

2.5 Quiz

For each question, select one or more correct answers.

1. Which of the following statements about the properties of square roots are correct?

(a) $\sqrt{a \times b} = \sqrt{a} \times \sqrt{b}$ for any positive numbers a and b.

(b) $\sqrt{a + b} = \sqrt{a} + \sqrt{b}$ for any positive numbers a and b.

(c) $\sqrt{a^2} = a$ for any number a.

(d) $\sqrt{a} \times \sqrt{a} = a$ for any positive number a.

(e) $\sqrt{\frac{a}{b}} = \frac{\sqrt{a}}{\sqrt{b}}$ for any positive numbers a and b.

2. What is the general formula for computing annualized volatility from periodic returns?

   (a) `annualized_volatility = volatility * number_of_periods`

   (b) `annualized_volatility = volatility * sqrt(number_of_periods)`

   (c) `annualized_volatility = sqrt(volatility * number_of_periods)`

   (d) `annualized_volatility = volatility / sqrt(number_of_periods)`

   (e) `annualized_volatility = volatility * sqrt(252)`

3. Consider a portfolio with a standard deviation of monthly returns (volatility) of 3%. How is the annualized volatility computed?

   (a) `annualized_volatility = 3% * 12`

   (b) `annualized_volatility = 3% * sqrt(12)`

   (c) `annualized_volatility = 3% / sqrt(12)`

   (d) `annualized_volatility = 3% * sqrt(252)`

   (e) `annualized_volatility = sqrt(3% * 12)`

4. Consider a short series of monthly returns for an investment: 2%, -3%, 6%, and -4%. Calculate the annualized volatility of these returns.

   (a) 5%

   (b) 10%

   (c) 16%

   (d) 20%

   (e) None of the above

5. Based on the Python exercise to calculate the annualized volatility of S&P 500 futures, which of the following lines of code correctly computes the annualized volatility?

   (a) `annualized_volatility = sp500_futures['Log_Returns'].std() * np.sqrt(252)`

   (b) `annualized_volatility = sp500_futures['Log_Returns'].std() / np.sqrt(252)`

   (c) `annualized_volatility = sp500_futures['Log_Returns'].std() * 252`

(d) `annualized_volatility = sp500_futures['Log_Returns'].mean() * np.sqrt(252)`

(e) `annualized_volatility = sp500_futures['Log_Returns'].var() * np.sqrt(252)`

6. Which of the following statements correctly identify the limitations of using annualized volatility as a metric for assessing investment risk?

(a) Annualized volatility keeps into account that investment returns are generally not normally distributed.

(b) The metric considers only downside volatility, ignoring any positive fluctuations in investment returns.

(c) Annualized volatility treats periods of high positive returns and high negative returns equally, without distinguishing between them.

(d) The metric is not sensitive to different time frames and provides consistent risk assessments across various market conditions.

(e) Annualized volatility is helpful to measure the likelihood of extreme events.

2.6 Skewness

Skewness is a statistical measure used to describe the asymmetry of the distribution of returns in finance. It helps in understanding the tendency of returns to lean towards either higher or lower values than the average. Here is a detailed yet intuitive explanation of how skewness is calculated:

1. **Historical Return Data:** Gather the historical return data of the investment over the desired period.

2. **Calculate the Mean Return:** Compute the average (mean) return. For n periods, it is calculated as:

$$\text{Mean Return} = \frac{1}{n}\sum_{i=1}^{n} \text{Return}_i$$

   Where Return_i is the return for the i-th period.

3. **Calculate Deviations from the Mean:** Calculate the deviation of each period's return from the mean return. This step is crucial for understanding how far and in which direction each return deviates from the average.

$$\text{Deviation} = \text{Return}_i - \text{Mean Return}$$

4. **Cube Each Deviation:** Cube the deviations to accentuate the impact of extreme values and to retain the sign (positive or negative) of the deviations. This is essential in skewness calculation as it amplifies the effect of asymmetrical tails in the distribution.

$$\text{Cubed Deviation} = (\text{Return}_i - \text{Mean Return})^3$$

5. **Average of Cubed Deviations:** Compute the average of these cubed deviations. This average represents the central tendency of the cubed deviations and is a key component in measuring skewness.

$$\text{Cubed Deviation Average} = \frac{\sum(\text{Return}_i - \text{Mean Return})^3}{n}$$

6. **Standard Deviation (Volatility):** Compute the standard deviation of the returns, as done in volatility calculation.

$$\text{Volatility} = \sqrt{\frac{\sum(\text{Return}_i - \text{Mean Return})^2}{n-1}}$$

7. **Skewness:** Finally, skewness is calculated by dividing the average of cubed deviations by the cube of volatility (standard deviation). This normalizes the skewness value, allowing it to be compared across different datasets.

$$\text{Skewness} = \frac{\text{Cubed Deviation Average}}{\text{Volatility}^3}$$

Intuitive Explanation: Think of skewness as the measure of the symmetry, or lack thereof, in your investment returns. If returns frequently fall far below the average, skewness will be negative, indicating a left-leaning tail. Conversely, if returns often exceed the average by a large margin, skewness will be positive, showing a right-leaning tail. In financial analysis, understanding skewness is like knowing if an archer's missed shots tend to be grouped to one side of the target rather than evenly around it. This information is crucial for an investor to understand the nature of risks in their investment, especially the likelihood of extreme gains or losses.

2.6.1 Example: Computing Skewness from Monthly Returns

This example demonstrates a detailed computation of skewness from a set of 12 monthly returns:

1. **Convert Percentage Returns to Decimal**: Monthly returns given in percentages: [5, -3, 7, 6, -4, 2, 3, 7, -5, 4, -6, 8]. Conversion to decimal:

   $$\text{Decimal Returns} = [0.05, -0.03, 0.07, 0.06, -0.04, 0.02, 0.03, 0.07, -0.05, 0.04, -0.06, 0.08]$$

2. **Calculate the Mean Return**: The mean of these decimal returns is computed:

   $$\text{Mean Return} = \frac{0.05 - 0.03 + 0.07 + \ldots + 0.04 - 0.06 + 0.08}{12} = 0.02$$

3. **Calculate Deviations from the Mean**: Deviation of each return from the mean is calculated:

   $$\text{Deviation}_1 = 0.05 - 0.02 = 0.03$$

   $$\text{Deviation}_2 = -0.03 - 0.02 = -0.05$$

   $$\ldots$$

   $$\text{Deviation}_{12} = 0.08 - 0.02 = 0.06$$

   Deviations: [0.03, -0.05, 0.05, 0.04, -0.06, 0.00, 0.01, 0.05, -0.07, 0.02, -0.08, 0.06]

4. **Cube Each Deviation**: Cubing each deviation:

   $$\text{Cubed Deviation}_1 = (0.03)^3 = 0.000027$$

   $$\text{Cubed Deviation}_2 = (-0.05)^3 = -0.000125$$

   $$\ldots$$

   $$\text{Cubed Deviation}_{12} = (0.06)^3 = 0.000216$$

   Cubed Deviations: [0.000027, -0.000125, 0.000125, 0.000064, -0.000216, 0.000000, 0.000001, 0.000125, -0.000343, 0.000008, -0.000512, 0.000216]

5. **Calculate the Average of Cubed Deviations**: Computing the average of these cubed deviations:

$$\text{Cubed Deviation Average} = \frac{0.000027 - 0.000125 + \ldots - 0.000512 + 0.000216}{12} = -0.0000525$$

6. **Calculate the Standard Deviation (Volatility)**: The standard deviation is the square root of the variance:

$$\text{Volatility} = \sqrt{\frac{(0.05 - 0.02)^2 + (-0.03 - 0.02)^2 + \ldots + (0.08 - 0.02)^2}{11}} \approx 0.05135$$

7. **Calculate Skewness**: Skewness is calculated by dividing the cubed deviation average by the cube of volatility:

$$\text{Skewness} = \frac{-0.0000525}{(0.05135)^3} \approx -0.39$$

The resulting skewness of -0.38784 indicates a left-leaning distribution of returns, suggesting a higher frequency or extremity of negative returns compared to positive ones. This detailed calculation helps in understanding the asymmetry in the distribution of investment returns.

2.7 Quiz

For each question, select one or more correct answers.

1. Based on the explanation of skewness provided, identify which of the following statements are correct:

   (a) Skewness measures the degree to which returns are spread out or concentrated on either side of the mean.

   (b) A positive skewness indicates a distribution with a long left tail, suggesting a greater likelihood of extremely low returns.

   (c) Skewness is calculated by taking the average of the cubed deviations from the mean and dividing it by the cube of the standard deviation.

   (d) If the skewness is zero, it implies that the returns distribution is perfectly symmetric.

   (e) Positive skewness in financial returns indicates a higher frequency of extreme losses compared to extreme gains.

2. Determine the skewness of the following set of 12 monthly returns (in percentages): [2, 5, -10, -7, -7, -3, -1, 9, 8, -6, -4, 2].

   (a) 0.15

   (b) -0.25

   (c) 0.23

   (d) -0.30

   (e) None of the above

2.8 Kurtosis

Kurtosis is a statistical measure used in finance to describe the 'tailedness' of the distribution of returns. It provides insights into the extremity of returns, whether they are unusually large or small compared to the average. Here's how kurtosis is typically calculated:

1. **Historical Return Data:** Collect the historical return data for the investment over the desired period.

2. **Calculate the Mean Return:** Determine the average (mean) return for the period. For n periods, the mean return is:

$$\text{Mean Return} = \frac{1}{n}\sum_{i=1}^{n} \text{Return}_i$$

   Here, Return_i is the return for the i-th period.

3. **Calculate Deviations from the Mean:** Find the deviation of each period's return from the mean return. This step is important to measure the distance of each return from the average.

$$\text{Deviation} = \text{Return}_i - \text{Mean Return}$$

4. **Fourth Power of Each Deviation:** Raise each deviation to the fourth power. This step emphasizes the impact of extreme values on the distribution.

$$\text{Fourth Power Deviation} = (\text{Return}_i - \text{Mean Return})^4$$

5. **Average of Fourth Power Deviations:** Compute the average of these fourth power deviations. This average is a key component in measuring kurtosis.

$$\text{Fourth Power Deviation Average} = \frac{\sum(\text{Return}_i - \text{Mean Return})^4}{n}$$

6. **Standard Deviation (Volatility):** Compute the standard deviation (volatility) of the returns.

$$\text{Volatility} = \sqrt{\frac{\sum(\text{Return}_i - \text{Mean Return})^2}{n-1}}$$

7. **Kurtosis:** Finally, kurtosis is calculated by dividing the average of fourth power deviations by the fourth power of volatility.

$$\text{Kurtosis} = \frac{\text{Fourth Power Deviation Average}}{\text{Volatility}^4}$$

   It is common to subtract 3 from this result to center the normal distribution kurtosis at 0 (excess kurtosis).

Intuitive Explanation: Consider kurtosis as a measure of the 'weight' of the tails in the distribution of investment returns. High kurtosis (leptokurtic) suggests a distribution with heavy tails and a sharp peak, indicating a higher likelihood of extreme returns (both positive and negative) compared to a normal distribution. Low kurtosis (platykurtic), on the other hand, implies lighter tails and a flatter peak, suggesting fewer extreme returns. Understanding kurtosis helps investors gauge the risk of extreme market movements that could significantly impact their investments.

2.8.1 Example: Computing Kurtosis from Monthly Returns

This example illustrates the calculation of kurtosis from a set of 12 monthly returns:

1. **Convert Percentage Returns to Decimal**: Given monthly returns in percentages: [5, -3, 7, 6, -4, 2, 3, 7, -5, 4, -6, 8], conversion to decimal is:

   $$\text{Decimal Returns} = [0.05, -0.03, 0.07, 0.06, -0.04, 0.02, 0.03, 0.07, -0.05, 0.04, -0.06, 0.08]$$

2. **Calculate the Mean Return**: The mean return is computed as the average of these decimal returns:

   $$\text{Mean Return} = \frac{0.05 - 0.03 + 0.07 + \ldots + 0.04 - 0.06 + 0.08}{12} = 0.02$$

3. **Calculate Deviations from the Mean**: Each return's deviation from the mean is calculated:

   $$\text{Deviation}_1 = 0.05 - 0.02 = 0.03$$

   $$\text{Deviation}_2 = -0.03 - 0.02 = -0.05$$

   $$\ldots$$

   $$\text{Deviation}_{12} = 0.08 - 0.02 = 0.06$$

   Deviations: [0.03, -0.05, 0.05, 0.04, -0.06, 0.00, 0.01, 0.05, -0.07, 0.02, -0.08, 0.06]

4. **Fourth Power Each Deviation**: Each deviation is raised to the fourth power:

   $$\text{Fourth Power Deviation}_1 = (0.03)^4 = 0.00000081$$

   $$\text{Fourth Power Deviation}_2 = (-0.05)^4 = 0.00000625$$

   $$\ldots$$

   $$\text{Fourth Power Deviation}_{12} = (0.06)^4 = 0.00001296$$

   Fourth Power Deviations: [0.00000081, 0.00000625, 0.00000625, 0.00000256, 0.00001296, 0.0, 0.00000001, 0.00000625, 0.00002401, 0.00000016, 0.00004096, 0.00001296]

5. **Calculate the Average of Fourth Power Deviations**: The average of these fourth power deviations is:

$$\text{Fourth Power Deviation Average} = \frac{0.00000081 + \ldots + 0.00001296}{12} \approx 0.0000094317$$

6. **Calculate the Standard Deviation (Volatility)**: The standard deviation is calculated as the square root of the variance:

$$\text{Volatility} = \sqrt{\frac{(0.05 - 0.02)^2 + (-0.03 - 0.02)^2 + \ldots + (0.08 - 0.02)^2}{11}} = 0.05135$$

7. **Calculate Kurtosis**: Kurtosis is calculated by dividing the fourth power deviation average by the fourth power of volatility:

$$\text{Kurtosis} = \frac{0.0000094317}{(0.05135)^4} \approx 1.36$$

The calculated kurtosis of 1.36 indicates a less peaked distribution with lighter tails compared to a normal distribution. It suggests a lower likelihood of extreme returns (both positive and negative) compared to a normal distribution.

2.9 Quiz

For each question, select one or more correct answers.

1. Based on the explanation of kurtosis, select the correct statements:

   (a) High kurtosis in a distribution implies a higher likelihood of extreme returns, both positive and negative.

   (b) A distribution with low kurtosis (platykurtic) will have a sharper peak and heavier tails than a normal distribution.

   (c) Kurtosis is calculated by dividing the average of the fourth power deviations from the mean by the fourth power of the standard deviation (volatility).

   (d) If the kurtosis is greater than 3 (leptokurtic), it indicates a flatter peak and lighter tails in the distribution.

   (e) Subtracting 3 from the kurtosis result adjusts it to compare with a normal distribution, which has a kurtosis of 3.

2. Using the method described for calculating kurtosis, determine the kurtosis of a new set of 12 monthly returns. The returns (in percentages) are as follows: [-3, 3, 4, 0, 1, 3, -3, 7, -8, 8, -7, 4]. Which of the following is the correct kurtosis value for these returns?

   (a) 0.87

   (b) 1.45

   (c) 1.75

   (d) 2.30

   (e) None of the above

2.10 Python Exercise: Comparing S&P 500 Futures Logarithmic Returns with a Normal Distribution

Objective: Analyze the distribution of S&P 500 front-month futures logarithmic returns and compare it with a normal distribution curve. The exercise will involve downloading data, calculating returns, and creating a histogram with a normal distribution overlay.

Requirements:

- Python
- Libraries: numpy, yfinance, matplotlib, scipy

Instructions:

1. **Import Necessary Libraries:** Begin by importing the required libraries: numpy (as np), yfinance (as yf), matplotlib.pyplot (as plt), and norm from scipy.stats.

2. **Download S&P 500 Futures Data:** Use yfinance’s ‘download‘ function to retrieve the ’Adj Close’ prices of S&P 500 front-month futures (’ES=F’).

3. **Calculate Daily Logarithmic Returns:** Compute the daily logarithmic returns of the S&P 500 futures. Use numpy’s ‘log‘ function and the ‘shift‘ method on the ’Adj Close’ prices. Remember to drop any NaN values that result from the calculation.

4. **Calculate Mean and Standard Deviation of Returns:** Determine the mean and standard deviation of the logarithmic returns. Store these values in variables named ‘mean‘ and ‘std‘.

```
    # Calculate mean and standard deviation of returns
    mean, std = log_returns.mean(), log_returns.std()

```

5. **Generate Range of Values for Normal Distribution:** Create a range of values (linspace) between the minimum and maximum of the logarithmic returns. Use numpy’s ‘linspace‘ function and multiply the difference between max and min returns by 10000 to determine the number of bins (‘n_bins‘).

```
    # Generate a range of values for the normal distribution
    n_bins = int(max(log_returns) - min(log_returns) * 10000)
    norm_dist = np.linspace(min(log_returns), max(log_returns)
    , n_bins)

```

6. **Calculate Normal Distribution Curve:** Compute the normal distribution curve with the same mean and standard deviation as the logarithmic returns. Use the ‘pdf‘ function from the ‘norm‘ module of scipy.stats.

```
    # Calculate the normal distribution with the same mean and standard deviation
    normal_curve = norm.pdf(norm_dist, mean, std)

```

7. **Plot Histogram and Normal Distribution:** Create a histogram of the logarithmic returns and overlay the normal distribution curve on it. Use matplotlib’s ‘plt.hist‘ for the histogram and ‘plt.plot‘ for the normal distribution curve. Ensure the histogram uses the same number of bins as calculated earlier and is colored orange. Label the histogram as ’Actual Distribution’ and the curve as ’Normal Distribution’.

8. **Add Vertical Lines for Mean and Standard Deviations:** Draw vertical dashed lines at the mean and at one, two, and three standard deviations away from the mean on both sides. Color the lines differently for positive (green) and negative (red) standard deviations and label them accordingly.

9. **Finalize the Plot:** Add a title, labels for the x-axis (’Logarithmic Returns’) and y-axis (’Frequency/Density’), and a legend to the plot. Then, display the plot using ‘plt.show()‘.

10. **Compute and Print Measures of Return and Risk:** Print the annualized return, the annulaized volatility, the skewness and the kurtosis.

This exercise will help you visualize how closely the logarithmic returns of S&P 500 futures follow a normal distribution, providing insights into the nature of return distributions in financial markets.

2.11 Drawdown

As a portfolio manager, a client entrusted you with $10 millions of their net worth and gave you the instruction to allocate 60% of their capital in stocks and 40% in bonds. One day, they come to your office and ask: "By how much our capital will go down in the worst case scenario?". What will you respond?

There are a couple of radical answers, none of which you are likely to tell your client, unless you really want to lose them as a client: 1) "I have no idea", 2) "In the real worst case scenario, you will lose it all". Instead, you may offer some educated estimate through the analysis of historical drawdowns.

Drawdown in finance refers to the peak-to-trough decline during a specific record period of an investment, fund, or trading account. It's a measure of downside risk, highlighting the potential loss an investor might face. Calculating drawdown involves determining the percentage loss from a peak to a subsequent trough. It's important because it helps assess the risk of a portfolio or a specific asset, providing an estimate of the maximum potential loss an investor might experience. This information is crucial for understanding the risk tolerance and managing the investment strategy accordingly. In the context of portfolio management, presenting drawdown to clients provides a realistic view of potential risks and helps in setting expectations.

2.11.1 Example: Computing Maximum Drawdown

Let's stick to the example of our client's $10 million portfolio. Drawdown measures the peak-to-trough decline during a specific period for this investment. Let's see how it's calculated:

1. **Establish the Peak:** The portfolio reaches its highest value. Let's say it peaked at $12 million.
2. **Identify a Trough:** The value then drops to a low point, say $9 million.
3. **Calculate Maximum Drawdown:** Maximum Drawdown is the percentage loss from the peak to the trough.

$$\begin{aligned}\text{Maximum Drawdown} &= \frac{\text{Peak Value} - \text{Trough Value}}{\text{Peak Value}}\\ &= \frac{\$12\text{M} - \$9\text{M}}{\$12\text{M}}\\ &= \frac{\$3\text{M}}{\$12\text{M}}\\ &= 25\%\end{aligned}$$

This calculation reveals a 25% drawdown, indicating the maximum percentage decline from the peak during the observed period.

2.11.2 Computing Drawdown Over Time

For a more detailed analysis of portfolio drawdown, we would need to compute the drawdown on an ongoing basis. To illustrate the computation of maximum drawdown over time, let's consider a portfolio with monthly returns across a year. We'll calculate the drawdown for each month and identify the maximum drawdown. Here's an example:

1. **Monthly Returns:** Suppose the portfolio has the following monthly returns (in percentages):

Month	**Return (%)**
January	2
February	-3
March	4
April	-1
May	3
June	-5
July	6
August	-2
September	2
October	-4
November	5
December	-3

Table 1: Monthly Portfolio Returns

2. **Cumulative Returns:** Calculate the cumulative return each month. For example, the cumulative return after two months would be the sum of the returns for the first and second months. With the monthly returns mentioned in the example above, our monthly cumulative returns would be the following:

Month	**Return (%)**	**Previous Return (%)**	**Cumulative Return (%)**
January	2	0	2
February	-3	2	-1
March	4	-3	3
April	-1	4	2
May	3	-1	5
June	-5	3	0
July	6	-5	6
August	-2	6	4
September	2	-2	6
October	-4	2	2
November	5	-4	7
December	-3	5	4

Table 2: Monthly Portfolio Returns with Previous and Cumulative Returns

3. **Tracking Peaks:** Identify the highest cumulative return at each step as the new peak.

Month	**Cumulative Return (%)**	**Cumulative Max (%)**
January	2	2
February	-1	2
March	3	3
April	2	3
May	5	5
June	0	5
July	6	6
August	4	6
September	6	6
October	2	6
November	7	7
December	4	7

Table 3: Monthly Cumulative Returns and Cumulative Maximum

4. **Calculating Drawdown:** For each month, calculate the drawdown from the peak. The drawdown is the difference between the current cumulative return and the highest cumulative return to that point.

Month	Cumulative Return (%)	Cumulative Max (%)	Drawdown (%)
January	2	2	0
February	-1	2	3
March	3	3	0
April	2	3	1
May	5	5	0
June	0	5	5
July	6	6	0
August	4	6	2
September	6	6	0
October	2	6	4
November	7	7	0
December	4	7	3

Table 4: Monthly Cumulative Returns, Cumulative Maximum, and Ongoing Drawdown

5. **Maximum Drawdown:** The maximum drawdown is the largest of these drawdowns over the 12 months.

This approach allows the portfolio manager to track how the portfolio's value has decreased from its peak over a period, providing an insight into the risk and volatility of the investment.

2.12 Python Exercise: Calculating and Plotting Ongoing Drawdown of S&P 500 Futures

Objective: Download front month S&P 500 futures prices, compute the logarithmic returns, calculate the cumulative return, and plot the ongoing drawdown with the ongoing drawdown represented as a red area.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance, matplotlib

Steps:

1. **Install Required Libraries:** Ensure the necessary libraries are installed. Use pip to install them if needed:

```
pip install pandas numpy yfinance matplotlib
```

2. **Download Data and Calculate Returns:**

```python
import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt

# Download S&P 500 front month futures data
sp500_futures = yf.download('ES=F')

# Calculate logarithmic returns
sp500_futures['Log_Returns'] = np.log(
    sp500_futures['Adj Close'] / sp500_futures['Adj Close'].shift(1)
    )

```

3. **Calculate Cumulative Returns and Ongoing Drawdown:**

```python
# Calculate cumulative returns
sp500_futures['Cumulative_Returns'] = np.exp(sp500_futures['Log_Returns'].cumsum()) - 1

# Calculate ongoing drawdown
rolling_max = sp500_futures['Cumulative_Returns'].cummax()
sp500_futures['Drawdown'] = rolling_max - sp500_futures['Cumulative_Returns']

# Plotting the results
fig, ax = plt.subplots()
ax.fill_between(sp500_futures.index, sp500_futures['Drawdown'],
                color='red', alpha=0.3)
ax.plot(sp500_futures['Cumulative_Returns'], label='Cumulative Returns')
ax.set_title('Cumulative Returns and Ongoing Drawdown')
ax.set_xlabel('Date')
ax.set_ylabel('Returns/Drawdown')
ax.legend()
plt.show()

```

Note: This exercise teaches you how to calculate and visualize the ongoing drawdown in conjunction with cumulative returns of S&P 500 futures. The drawdown is represented as a red area in the plot, offering a visual understanding of risk in financial markets.

2.13 Python Exercise: Calculating and Plotting Ongoing Drawdown of a Mixed Portfolio

Objective: Construct a portfolio with 60% weight in S&P 500 futures (ES=F) and 40% weight in US 10-year Treasury futures (ZN=F), calculate the logarithmic returns of the portfolio, compute the cumulative return, and plot the ongoing drawdown with the drawdown represented as a red area. Then, print the maximum drawdown, as well the other measures of return and risk we studied so far: annualized return, annualized volatility, skewness and kurtosis.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance, matplotlib

Steps:

1. **Install Required Libraries:** Ensure the necessary libraries are installed. Use pip to install them if needed:

```
pip install pandas numpy yfinance matplotlib
```

```python
# Import libraries
import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt

# Download S&P 500 and US 10-year Treasury futures data
sp500_futures = yf.download('ES=F')['Adj Close']
treasury_futures = yf.download('ZN=F')['Adj Close']

# Align the datasets
data = pd.DataFrame({
    'SP500': sp500_futures,
    'Treasury': treasury_futures
}).dropna()

# Calculate daily logarithmic returns
data['SP500_Returns'] = np.log(data['SP500'] / data['SP500'].shift(1))
data['Treasury_Returns'] = np.log(data['Treasury'] / data['Treasury'].shift(1))

# Calculate portfolio returns (60% SP500, 40% Treasury)
data['Portfolio_Returns'] = 0.6 * data['SP500_Returns'] + 0.4 * data['Treasury_Returns']

# Calculate cumulative returns for SP500, Treasury, and Portfolio
data['Cumulative_SP500_Returns'] = data['SP500_Returns'].cumsum()
data['Cumulative_Treasury_Returns'] = data['Treasury_Returns'].cumsum()
data['Cumulative_Returns'] = data['Portfolio_Returns'].cumsum()

# Calculate ongoing drawdown
rolling_max = data['Cumulative_Returns'].cummax()
data['Drawdown'] = rolling_max - data['Cumulative_Returns']
```

```

    # Plotting the results
    fig, ax = plt.subplots()
    ax.fill_between(data.index, -data[’Drawdown’], color=’red’, alpha=0.3)
    ax.plot(data[’Cumulative_Returns’], label=’Portfolio Cumulative Returns’)
    ax.plot(data[’Cumulative_SP500_Returns’], label=’S&P 500 Cumulative Returns’, linestyle=’--’, linewidth=0.5)
    ax.plot(data[’Cumulative_Treasury_Returns’], label=’10-Year Treasury Cumulative Returns’, linestyle=’--’, linewidth=0.5)
    ax.set_title(’Portfolio Cumulative Returns and Ongoing Drawdown’)
    ax.set_xlabel(’Date’)
    ax.set_ylabel(’Returns/Drawdown’)
    ax.legend()
    plt.show()

```

2. **Compute and Print Measures of Return and Risk:** Print the annualized return, the annualized volatility, the skewness, the kurtosis and the maximum drawdown.

Note: This exercise guides you through the process of constructing a mixed asset portfolio, calculating its returns, and visualizing the risk through drawdown analysis. The plot will display the cumulative returns of the portfolio and highlight the drawdown periods in red, offering insights into the risk-return profile of the portfolio.

2.14 Quiz

For each question, select one or more correct answers.

1. A portfolio manager is analyzing the monthly returns of a portfolio over a year. The monthly returns are as follows: [2%, -3%, 4%, -1%, 3%, -5%, 6%, -2%, 2%, -4%, 5%, -3%]. Calculate the maximum drawdown over this 12-month period.

   (a) 5%

   (b) 7%

   (c) 8%

   (d) 10%

   (e) None of the above

2. A portfolio manager wants to calculate the maximum drawdown of a portfolio using Python. Given a DataFrame ’data’ with a column ’Cumulative_Returns’, which line of code correctly computes the maximum drawdown?

(a) `max_drawdown = data[’Cumulative_Returns’].min() - data[’Cumulative_Returns’].max()`

(b) `max_drawdown = (data[’Cumulative_Returns’].cummax() - data[’Cumulative_Returns’]).max()`

(c) `max_drawdown = (data[’Cumulative_Returns’].max() - data[’Cumulative_Returns’]).min()`

(d) `max_drawdown = data[’Cumulative_Returns’].max() - data[’Cumulative_Returns’].min()`

(e) `max_drawdown = (data[’Cumulative_Returns’] - data[’Cumulative_Returns’].cummin()).max()`

3 Measuring Risk-Adjusted Returns

Once again, suppose you are the Head of Portfolio Management at an asset management firm with $2 billions in assets under management (AUM), and you need to hire a portfolio manager to launch a new fund. From a head hunter, you received the resumes of several candidates, along with a plot of their past performance.

How will you choose between the portfolio manager candidates? Most likely, it will depend on the risk appetite of the investors who are subscribing to the new fund. In the previous sections, we discussed how to measure returns and how to measure risk separately, but wouldn't it be helpful to be able to measure both?

3.1 Sharpe Ratio

The Sharpe Ratio is an essential tool for investors to evaluate the performance of an investment relative to its risk. It addresses a fundamental dilemma in investing: the need to balance potential returns with associated risks.

The Problem: Balancing Return and Risk
Investors often grapple with choosing between higher-return, higher-risk investments and lower-return, lower-risk alternatives. Opting for investments based solely on high returns disregards the risks involved. Conversely, focusing only on minimizing risk might result in underwhelming returns. The crucial challenge is striking an optimal balance between risk and return.

The Solution: Sharpe Ratio
The Sharpe Ratio offers a resolution to this challenge by comparing the performance of an investment to a risk-free asset, while adjusting for the investment's risk. It is calculated using the formula:

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p} \tag{1}$$

where R_p represents the return of the portfolio, R_f denotes the risk-free rate (such as the return on Treasury bills), and σ_p is the standard deviation of the portfolio's excess return, a measure of risk.

Why Sharpe Ratio Works

- *Risk Adjustment:* The Sharpe Ratio incorporates risk by considering the standard deviation of the portfolio's returns, allowing for a risk-adjusted comparison of different investments.
- *Risk-Free Benchmarking:* It references a risk-free asset, showing the additional return earned per unit of risk taken over a risk-free investment.

- *Standardized Measurement:* The Sharpe Ratio provides a uniform metric to compare varied investments or portfolios, irrespective of their risk.
- *Reward-to-Variability:* A higher Sharpe Ratio suggests a more favorable risk-adjusted return, quantifying the 'reward-to-variability' ratio.

Practical Application

Investors utilize the Sharpe Ratio to assess the performance of different funds, portfolios, or individual investments. Generally, a higher Sharpe Ratio is preferred, indicating a more efficient return per unit of risk. It is widely used in finance for portfolio optimization, investment decision-making, and financial planning.

In essence, the Sharpe Ratio provides a comprehensive measure that encapsulates both expected returns and associated risks, thereby aiding investors in making well-informed investment decisions.

3.1.1 Understanding the Risk-Free Rate

The risk-free rate is a critical component in financial calculations like the Sharpe Ratio. It plays a pivotal role in investment decision-making by providing a benchmark against which the performance of risky assets can be evaluated.

The Problem: Assessing Return Against a Safe Benchmark

Investors face the challenge of gauging whether the returns from an investment are sufficient to compensate for the risk undertaken. To make this assessment, one needs a baseline or a benchmark that signifies a 'safe' return, with minimal to no risk involved.

The Solution: Risk-Free Rate

The risk-free rate serves as this benchmark. It is theoretically the rate of return of an investment with zero risk of financial loss. In practice, the risk-free rate is the return on the safest possible investment in the financial market. This rate is crucial because:

- It represents the minimum return an investor would expect for any risk taken.
- It serves as a baseline to compare against returns from risky investments.
- It is used to calculate the excess return over a risk-free investment.

Common Proxies for Risk-Free Rate

In practice, certain assets are typically used as proxies for the risk-free rate:

- *Treasury Bills:* Often, short-term government securities like U.S. Treasury bills are used, as they are backed by the government's promise and are thus considered free of credit risk.

- *Government Bonds:* In some cases, long-term government bonds are also used, though they may carry some risk (e.g., interest rate risk).

Limitations of the Risk-Free Rate Concept

While the notion of a risk-free rate is widely accepted, it has certain limitations:

- *Theoretical Construct:* In reality, no investment is entirely risk-free. Even government securities carry minimal risks, such as inflation risk or interest rate risk.
- *Variability Across Countries:* The risk-free rate varies across countries, depending on their economic stability and creditworthiness.
- *Impact of Economic Conditions:* The risk-free rate can fluctuate with changing economic conditions, impacting investment decision metrics like the Sharpe Ratio.

In summary, the risk-free rate is a fundamental concept in finance, providing a benchmark for evaluating investment returns. However, its theoretical nature and the variability in its practical proxies necessitate careful consideration when using it in financial models and calculations.

3.2 Python Exercise: Calculating the Sharpe Ratio of S&P 500 Futures with Annualized Data

Objective: Download front month S&P 500 futures prices, compute annualized returns and volatility, download a risk-free rate series, and calculate the Sharpe Ratio.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance

Steps:

1. **Install Required Libraries:** Ensure you have the necessary libraries installed. Use pip to install them if needed:

```
pip install pandas numpy yfinance
```

2. **Download S&P 500 Futures Data and Calculate Returns:**

```python
import pandas as pd
import numpy as np
import yfinance as yf

```

```
    # Download S&P 500 front month futures data
    sp500_futures = yf.download(’ES=F’)[’Adj Close’]

    # Calculate daily logarithmic returns
    sp500_futures[’Log_Returns’] = np.log(
        sp500_futures / sp500_futures.shift(1)
    )

```

3. **Annualize Returns and Volatility:**

```
    # Annualize returns and volatility
    annualized_return = sp500_futures[’Log_Returns’].mean() *
    252
    annualized_volatility = sp500_futures[’Log_Returns’].std()
     * np.sqrt(252)

```

4. **Download Risk-Free Rate Series:**

```
    # Download the 3-month Treasury bill rate as the risk-free
     rate
    sp500_start_date = str(sp500_futures.index[0])[:10]
    risk_free_rate_series = yf.download(’^IRX’, start=
    sp500_start_date)[’Adj Close’]
    risk_free_rate = risk_free_rate_series.mean() / 100

```

5. **Calculate the Sharpe Ratio:**

```
    # Calculate the annual Sharpe Ratio
    sharpe_ratio = (annualized_return - risk_free_rate) /
    annualized_volatility
    print("annualized_return:", annualized_return)
    print("risk_free_rate_annual:", risk_free_rate)
    print("annualized_volatility:", annualized_volatility)
    print("sharpe_ratio:", sharpe_ratio)

```

Note: This exercise guides you through the process of calculating the annual Sharpe Ratio for S&P 500 futures using real market data for the risk-free rate. The annual Sharpe Ratio helps assess the risk-adjusted performance of an investment relative to its annualized risk.

3.3 Python Exercise: Calculating the Sharpe Ratio of 10-Year Treasury Futures with Annualized Data

Objective: In this exercise, students will calculate the annualized returns, volatility, and Sharpe Ratio for the 10-year Treasury futures (ZN=F), following the steps previously taken for S&P 500 futures.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance

Task:

1. **Download 10-Year Treasury Futures Data:** Utilize yfinance to download the 10-year Treasury futures data (ZN=F). Focus on selecting an appropriate time period.
2. **Calculate Daily Logarithmic Returns:** Compute daily logarithmic returns for the 10-year Treasury futures. This step is similar to what was done for the S&P 500 futures.
3. **Annualize Returns and Volatility:** Annualize both the returns and the volatility for the 10-year Treasury futures, akin to the procedure followed for the S&P 500 futures.
4. **Use the Same Risk-Free Rate Series:** Apply the 3-month Treasury bill rate as the risk-free rate, ensuring proper alignment of data by date.
5. **Calculate the Sharpe Ratio:** Use the formula Sharpe Ratio = (Annualized Return - Risk-Free Rate) / Annualized Volatility for the 10-year Treasury futures.
6. **Interpretation:** Analyze and contrast the calculated Sharpe Ratio for the 10-year Treasury futures with that of the S&P 500 futures. Discuss the implications for risk-adjusted performance.

Considerations:

- Ensure correct alignment of data series by date.
- Reflect on the distinct risk and return profiles of equity (S&P 500) and debt (10-year Treasury) instruments.
- Pay attention to consistency in the annualization process.

Note: This exercise aims to enhance students' understanding of calculating and interpreting the Sharpe Ratio for different financial instruments, providing insights into the comparative risk-adjusted performance of equity and debt investments.

3.4 Python Exercise: Calculating the Sharpe Ratio of a Mixed Portfolio with S&P 500 and 10-Year Treasury Futures

Objective: This exercise focuses on constructing a mixed portfolio with 60% weight in S&P 500 futures (ES=F) and 40% weight in 10-year Treasury futures (ZN=F), and then calculating the annualized returns, volatility, and Sharpe

Ratio for this portfolio.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance

Task:

1. **Download Futures Data:** Use yfinance to download both the S&P 500 futures (ES=F) and the 10-year Treasury futures (ZN=F). Be mindful of selecting a consistent time frame for both datasets.

2. **Calculate Daily Logarithmic Returns for Each Asset:** Compute daily logarithmic returns separately for the S&P 500 futures and the 10-year Treasury futures.

3. **Create a Mixed Portfolio:** Construct a mixed portfolio with 60% allocated to S&P 500 futures and 40% to 10-year Treasury futures. Calculate the portfolio's daily returns based on these weights.

4. **Annualize Portfolio Returns and Volatility:** Calculate the annualized returns and volatility of the mixed portfolio.

5. **Utilize the Risk-Free Rate:** Apply the 3-month Treasury bill rate as the risk-free rate. Ensure the data is synchronized with the portfolio's timeline.

6. **Compute and Print Measures of Return and Risk:** Print the annualized return, the annualized volatility, the skewness, the kurtosis and the maximum drawdown.

7. **Compute and Print the Sharpe Ratio:** Using the formula Sharpe Ratio = (Annualized Return - Risk-Free Rate) / Annualized Volatility, compute and print the Sharpe Ratio for the mixed portfolio.

8. **Plot the Cumulative Returns and Drawdowns:** Plot the cumulative returns and drawdowns of the mixed portfolio, as well a dashed line for the S&P 500 futures and a dashed line for the 10-year Treasury futures cumulative profits.

```python
    plt.plot(portfolio_data['Cumulative'], label='Mixed Portfolio')
    plt.plot(sp500_data['Cumulative'], label='S&P 500 Futures', linestyle='dashed')
    plt.plot(treasury_data['Cumulative'], label='10-Year Treasury Futures', linestyle='dashed')
    plt.plot(portfolio_data['Drawdown'], label='Drawdowns')

    plt.title('Cumulative Returns and Drawdowns')
    plt.xlabel('Date')
    plt.ylabel('Cumulative Returns')
    plt.legend()
    plt.show()

```

9. **Analysis and Comparison:** Analyze the Sharpe Ratio of the mixed portfolio and compare it to the individual Sharpe Ratios of the S&P 500 futures and the 10-year Treasury futures. Discuss the implications for diversification and risk-adjusted performance.

Considerations:

- Ensure accurate calculation and alignment of the returns for both components of the mixed portfolio.
- Reflect on the impact of diversification on the risk-return profile of the portfolio.
- Emphasize the importance of the Sharpe Ratio in assessing the efficiency of a mixed portfolio.

Note: This exercise aims to provide a practical understanding of constructing and analyzing a mixed portfolio, focusing on the concepts of diversification, risk-adjusted returns, and the significance of the Sharpe Ratio in portfolio management.

3.5 Quiz

For each question, select one or more correct answers.

1. Why is the Sharpe Ratio considered an essential tool in evaluating investment performance?
   (a) The Sharpe Ratio only accounts for the return of an investment, ignoring its risk.
   (b) It provides a standardized measure to compare different investments, considering both risk and return.
   (c) The Sharpe Ratio allows investors to evaluate the additional return earned per unit of risk taken over a risk-free investment.
   (d) It helps in determining the highest returning portfolio without considering its volatility.
   (e) The ratio incorporates the risk-free rate, enabling a comparison of excess returns over a safe benchmark.

2. What is the significance of the risk-free rate in financial calculations, and what are its limitations?
   (a) The risk-free rate represents the guaranteed return that an investor can expect from any investment in the market.

(b) It is used as a benchmark to assess whether the returns from a risky investment are sufficient to compensate for the risk undertaken.

(c) Treasury Bills are often used as a proxy for the risk-free rate due to their government backing and perceived absence of credit risk.

(d) The risk-free rate remains constant and unaffected by economic conditions or interest rate changes.

(e) The concept of the risk-free rate has limitations as no investment is truly risk-free, and it varies across countries based on economic stability.

3. When considering the risk-free rate in financial calculations, which of the following can be deemed appropriate proxies, and why?

   (a) Short-term U.S. Treasury bills, because they are backed by the U.S. government and have minimal credit risk.

   (b) Corporate bonds, as they offer higher yields compared to government securities.

   (c) Low volatility stocks that pay high dividends.

   (d) High-yield junk bonds, due to their potential for higher returns.

   (e) Money market funds, as they are considered stable and have low risk.

4. Given annualized returns, annualized volatility, and the risk-free rate from S&P 500 futures data, which Python line(s) of code correctly compute the Sharpe Ratio?

   (a) `sharpe_ratio = (annualized_return - risk_free_rate) / annualized_volatility`

   (b) `sharpe_ratio = annualized_return / annualized_volatility`

   (c) `sharpe_ratio = (annualized_return + risk_free_rate) / annualized_volatility`

   (d) `sharpe_ratio = annualized_volatility / (annualized_return - risk_free_rate)`

   (e) `sharpe_ratio = (annualized_return - risk_free_rate) / np.sqrt(annualized_volatility)`

5. For a mixed portfolio comprising 60% S&P 500 futures and 40% 10-year Treasury futures, which Python line(s) of code correctly contribute to computing the annual Sharpe Ratio? Assume that ‘sp500_returns‘, ‘treasury_returns‘, and ‘risk_free_rate‘ are already defined as the daily logarithmic returns of S&P 500 futures, 10-year Treasury futures, and the risk-free rate, respectively.

   (a) `mixed_portfolio_returns = 0.6 * sp500_returns + 0.4 * treasury_returns`

   (b) `annualized_return = mixed_portfolio_returns.mean() * 252`

   (c) `annualized_volatility = mixed_portfolio_returns.std() * np.sqrt(252)`

   (d) `sharpe_ratio = (annualized_return - risk_free_rate) / annualized_volatility`

   (e) `sharpe_ratio = (annualized_return + risk_free_rate) / annualized_volatility`

3.6 Inflation-Adjusted Sharpe Ratio

Now, another client is considering entrusting you with \$20 millions of their capital. On one condition, though: they want their capital to maintain its purchasing power over time, despite inflation. Hence, your portfolio returns need to beat inflation. This client shows absolutely no interest in the notion of risk-free rate. All they care about is their money to maintain its purchasing power. How will you know if a given portfolio meets your client's requirement?

In a scenario where a client is primarily concerned with maintaining the purchasing power of their capital in the face of inflation, the concept of Inflation-Adjusted Sharpe Ratio becomes particularly relevant. This approach addresses a different aspect of investment performance, focusing on the impact of inflation on the value of returns.

3.6.1 The Problem: Maintaining Purchasing Power Against Inflation

For investors like the client mentioned, the primary concern is not just generating returns but ensuring that these returns are sufficient to outpace inflation. The challenge is to ensure that the investment grows enough to not only provide returns but also preserve or increase the real value of the capital, considering the eroding effect of inflation.

3.6.2 The Solution: Inflation-Adjusted Sharpe Ratio

The Inflation-Adjusted Sharpe Ratio measures the investment return adjusted for the effects of inflation, providing a clearer picture of risk-adjusted performance in real terms. The Inflation-Adjusted Sharpe Ratio is calculated as follows:

3.6.3 Formula and Purpose

$$\text{Inflation-Adjusted Sharpe Ratio} = \frac{R_p - R_i}{\sigma_p} \tag{2}$$

where:

- R_p is the return of the portfolio.
- R_i is the rate of inflation.
- σ_p is the standard deviation (volatility) of the portfolio's returns.

This metric substitutes the risk-free rate (R_f) in the traditional Sharpe Ratio with the rate of inflation (R_i), making it more relevant for investors whose primary goal is to maintain or increase their purchasing power over time.

3.6.4 Relevance and Application

- *Focus on Inflation:* This metric is particularly useful in environments with high inflation, where maintaining purchasing power becomes a significant concern for investors.
- *Comparative Analysis:* It allows investors to compare how effectively different investments or portfolios are performing in terms of real, inflation-adjusted returns.
- *Risk-Adjusted Returns:* By considering volatility (σ_p), it provides a risk-adjusted measure of how much an investment outperforms inflation.

3.7 Python Exercise: Calculating the Inflation-Adjusted Sharpe Ratio of a Mixed Portfolio with S&P 500 and 10-Year Treasury Futures

Objective: This exercise focuses on constructing a mixed portfolio with 60% weight in S&P 500 futures (ES=F) and 40% weight in 10-year Treasury futures (ZN=F), and then calculating the annualized returns, volatility, and Inflation-Adjusted Sharpe Ratio for this portfolio.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance, pandas_datareader

Task:

1. **Download Futures Data:** Use yfinance to download both the S&P 500 futures (ES=F) and the 10-year Treasury futures (ZN=F). Be mindful of selecting a consistent time frame for both datasets.
2. **Calculate Daily Logarithmic Returns for Each Asset:** Compute daily logarithmic returns separately for the S&P 500 futures and the 10-year Treasury futures.
3. **Create a Mixed Portfolio:** Construct a mixed portfolio with 60% allocated to S&P 500 futures and 40% to 10-year Treasury futures. Calculate the portfolio's daily returns based on these weights.
4. **Annualize Portfolio Returns and Volatility:** Calculate the annualized returns and volatility of the mixed portfolio.
5. **Utilize the Inflation Rate:** Import inflation rate data from FRED (Federal Reserve Economic Data) using the 'pandas_datareader' library. Specify the series ID for the Consumer Price Index for All Urban Consumers (CPIAUCSL) and choose an appropriate time frame.

```
# Specify the data source as ’fred’ (Federal Reserve Economic Data)
data_source = ’fred’

# Specify the series ID for the Consumer Price Index for All Urban Consumers (CPIAUCSL)
series_id = ’CPIAUCSL’

# Define the start and end dates for the data
start_date = ’YYYY-MM-DD’
end_date = ’YYYY-MM-DD’

# Fetch the inflation data
inflation_data = pdr.get_data_fred(series_id, start_date, end_date)
# Calculate the annualized inflation rate
inflation_rate = inflation_data[’CPIAUCSL’].pct_change(periods=12)
# Resample monthly inflation data to daily frequency (forward fill)
inflation_rate = inflation_rate.resample(’D’).ffill()

```

6. **Calculate the Inflation-Adjusted Sharpe Ratio:** Using the formula

$$\text{Inflation-Adjusted Sharpe Ratio} = \frac{\text{Annualized Return} - R_i}{\text{Annualized Volatility}} \tag{3}$$

compute the Inflation-Adjusted Sharpe Ratio for the mixed portfolio, where R_i is the inflation rate.

7. **Analysis and Comparison:** Analyze the Inflation-Adjusted Sharpe Ratio of the mixed portfolio and compare it to the individual Inflation-Adjusted Sharpe Ratios of the S&P 500 futures and the 10-year Treasury futures. Discuss the implications for diversification and risk-adjusted performance, considering the goal of maintaining purchasing power.

Considerations:

- Ensure accurate calculation and alignment of the returns for both components of the mixed portfolio.

- Reflect on the impact of diversification on the risk-return profile of the portfolio, specifically in the context of preserving purchasing power.

- Emphasize the importance of the Inflation-Adjusted Sharpe Ratio in assessing the efficiency of a mixed portfolio for a client primarily concerned with maintaining purchasing power.

Note: This exercise aims to provide a practical understanding of constructing and analyzing a mixed portfolio, focusing on the concepts of diversification, risk-adjusted returns, and the significance of the Inflation-Adjusted Sharpe Ratio in portfolio management for clients with an inflation-preservation goal.

3.8 Python Exercise: Calculating the Inflation-Adjusted Sharpe Ratio of a Mixed Portfolio

Objective: This exercise focuses on constructing a mixed portfolio with a specific asset allocation and then calculating the annualized returns, volatility, and Inflation-Adjusted Sharpe Ratio for this portfolio.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance, pandas_datareader

Task:

1. **Download Asset Data:** Use yfinance to download historical price data for the following ETFs:

   - US Equity (30%): SPDR S&P 500 ETF Trust (SPY) or iShares Core S&P 500 ETF (IVV)
   - Foreign Developed Equity (15%): iShares MSCI EAFE ETF (EFA) or Vanguard FTSE Developed Markets ETF (VEA)
   - Emerging Market Equity (5%): iShares MSCI Emerging Markets ETF (EEM) or Vanguard FTSE Emerging Markets ETF (VWO)
   - US REITs (20%): Vanguard Real Estate ETF (VNQ) or iShares U.S. Real Estate ETF (IYR)
   - US Treasury Bonds (15%): iShares 7-10 Year Treasury Bond ETF (IEF) or iShares 20+ Year Treasury Bond ETF (TLT)
   - US Treasury Inflation-Protected Securities (TIPS - 15%): iShares TIPS Bond ETF (TIP) or Schwab U.S. TIPS ETF (SCHP)

   Be mindful of selecting a consistent time frame for all ETFs.

2. **Calculate Daily Logarithmic Returns:** Compute daily logarithmic returns separately for each ETF in the portfolio.

3. **Create the Mixed Portfolio:** Construct the mixed portfolio with the specified asset allocation percentages. Calculate the portfolio's daily returns based on these weights.

4. **Annualize Portfolio Returns and Volatility:** Calculate the annualized returns and volatility of the mixed portfolio.

5. **Utilize the Inflation Rate:** Import inflation data from FRED (Federal Reserve Economic Data) using the 'pandas_datareader' library. Specify the series ID for the Consumer Price Index for All Urban Consumers (CPIAUCSL) and choose an appropriate time frame.

6. **Calculate the Inflation-Adjusted Sharpe Ratio:** Using the formula

$$\text{Inflation-Adjusted Sharpe Ratio} = \frac{\text{Annualized Return} - R_i}{\text{Annualized Volatility}} \quad (4)$$

   compute the Inflation-Adjusted Sharpe Ratio for the mixed portfolio, where R_i is the inflation rate.

7. **Analysis and Comparison:** Analyze the Inflation-Adjusted Sharpe Ratio of the mixed portfolio and compare it to the individual Inflation-Adjusted Sharpe Ratios of the ETFs representing each asset class. Discuss the implications for diversification and risk-adjusted performance, considering the goal of maintaining purchasing power.

This exercise allows you to implement a specific asset allocation strategy and assess its performance in the context of inflation-adjusted returns. Remember to conduct your research and due diligence when selecting specific ETFs, and consult with a financial advisor or investment professional for personalized guidance.

3.9 Quiz

For each question, select one or more correct answers.

1. When is the Inflation-Adjusted Sharpe Ratio particularly relevant for investors?

   (a) In low-inflation environments

   (b) In high-volatility environments

   (c) In environments with high inflation

   (d) In environments with low volatility

   (e) In environments with declining interest rates

2. Which library should be used to download inflation rate data from FRED (Federal Reserve Economic Data) in Python?

   (a) `pandas`

   (b) `numpy`

   (c) `yfinance`

   (d) `pandas_datareader`

   (e)

3. Which of the following lines of code correctly calculates the Inflation-Adjusted Sharpe Ratio for a mixed portfolio? Which of the following lines of code correctly calculates the Inflation-Adjusted Sharpe Ratio for a mixed portfolio? Assume that `portfolio_return` contains daily returns, `interest_rate` contains the interest rate on any given day, `portfolio_volatility` contains daily volatility and `R_i` contains the mean interest rate.

(a) `inflation_adjusted_sharpe_ratio = (annualized_return - inflation_rate.mean())`
`/ annualized_volatility`

(b) `inflation_adjusted_sharpe_ratio = (portfolio_return.mean()`
`* 252 - inflation_rate) / portfolio_volatility`

(c) `inflation_adjusted_sharpe_ratio = (annualized_return - R_i)`
`/ annualized_volatility`

(d) `inflation_adjusted_sharpe_ratio = (annualized_return - inflation_rate)`
`/ np.sqrt(252)`

(e) `inflation_adjusted_sharpe_ratio = (portfolio_return.mean()`
`* 252 - inflation_rate.mean()) / (portfolio_volatility * np.sqrt(252))`

3.10 Information Ratio

Now, let's envision another scenario: you are a portfolio manager at an asset management firm and a high-net-worth prospective client comes to you for information. The client is a successful entrepreneur who amassed a $13-million net worth through their business. They were good at making money but they do not understand much about investing in capital markets. They come from an industrial engineering background, did not study much finance in engineering school and spent most of their time building their business. As a result, they accumulated a sizeable amount of money but do not really know what to do with it. They just read and heard stories about Warren Buffett and his legendary Berkshire Hathaway and how it ”outperformed the market”. Your client comes from an engineering background, so they have an analytical mind, and wants to know what it means exactly that Berkshire Hathaway ”outperformed the market” and - most importantly - how Berkshire Hathaway's overperformance is measured with numbers. How would you answer your prospective client's question?

This is where the Information Ratio (IR) might be of help. The Information Ratio (IR) is another metric in the field of investment management, particularly for assessing the performance of active fund managers relative to a specific benchmark. In the case of the client mentioned above, the benchmark may be the S&P 500 Index.

While similar to the Sharpe Ratio in evaluating risk-adjusted returns, the Information Ratio has distinct attributes that make it a valuable tool.

3.10.1 Formula and Comparison of Information Ratio to Sharpe Ratio

The Information Ratio and the Sharpe Ratio are both critical metrics in finance, but they serve different purposes and are based on different formulas. Let's delve into their formulas and understand how they differ.

Information Ratio Formula: The Information Ratio (IR) is calculated using the formula:

$$\text{Information Ratio} = \frac{R_p - R_b}{\sigma_{\text{excess}}} \tag{5}$$

where:

- R_p is the return of the portfolio.
- R_b is the return of the benchmark.
- σ_{excess} is the standard deviation of the excess return of the portfolio over the benchmark, often referred to as the 'tracking error'.

Sharpe Ratio Formula: In contrast, the Sharpe Ratio is calculated as:

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p} \tag{6}$$

where:

- R_f represents the risk-free rate.
- σ_p is the standard deviation of the portfolio's return.

Computing the Tracking Error The standard deviation of the excess return of a portfolio over its benchmark, also known as the 'tracking error', is a key measure in portfolio management. It quantifies the consistency and magnitude of the portfolio's performance relative to the benchmark. Here is how it is computed:

1. **Calculate Excess Returns:**
   - Determine the excess returns for each period by subtracting the benchmark return from the portfolio return.
   - Formula: $\text{Excess Return}_t = \text{Portfolio Return}_t - \text{Benchmark Return}_t$.
2. **Compute Standard Deviation of Excess Returns:**
   - The standard deviation of these excess returns is calculated to assess the volatility of the portfolio's performance compared to the benchmark.
   - Formula:

   $$\sigma_{\text{excess}} = \sqrt{\frac{1}{N-1}\sum_{t=1}^{N}(\text{Excess Return}_t - \overline{\text{Excess Return}})^2}$$

   where N is the number of periods, Excess Return_t is the excess return at time t, and $\overline{\text{Excess Return}}$ is the mean excess return.

3. **Interpretation of Tracking Error:**
   - A higher tracking error indicates a greater divergence from the benchmark, signifying higher active risk.
   - A lower tracking error implies that the portfolio closely follows its benchmark.

Understanding the tracking error is crucial for assessing the risk taken by a portfolio manager in attempting to outperform the benchmark. It reflects the consistency and scale of over- or underperformance relative to the benchmark.

Key Differences:

1. **Benchmark Reference:** The most significant difference is in the benchmark used. The Information Ratio uses a specific benchmark or index (R_b) as a reference, while the Sharpe Ratio uses a risk-free rate (R_f).

2. **Focus on Excess Return:** The IR focuses specifically on the excess return over a benchmark, making it ideal for evaluating active management. In contrast, the Sharpe Ratio looks at the excess return over a risk-free asset, assessing overall risk-adjusted return.

3. **Risk Assessment:** The IR uses the standard deviation of excess returns (tracking error), which measures the consistency of outperforming or underperforming the benchmark. The Sharpe Ratio uses the standard deviation of the portfolio's total returns, measuring overall investment risk.

These differences make the Information Ratio particularly useful for comparing the performance of active managers relative to their benchmarks, while the Sharpe Ratio is better suited for evaluating the risk-adjusted returns of an investment relative to a risk-free asset.

3.10.2 The Purpose of Computing the Information Ratio

1. **Benchmark Comparison:** The IR measures the excess return of an investment relative to a chosen benchmark, adjusted for the variability of these excess returns. This is in contrast to the Sharpe Ratio, which compares returns to a risk-free rate, making the IR more suitable for strategies aiming to outperform a market index.

2. **Active Management Evaluation:** The IR is particularly useful for evaluating the skill of active fund managers. It quantifies the additional return generated over the benchmark, per unit of additional risk, highlighting the manager's ability to make effective investment decisions.

3.10.3 Unique Insights Offered by the Information Ratio

1. **Relative Performance:** The IR captures the performance of a portfolio or fund manager relative to a benchmark, emphasizing the results of active management strategies.

2. **Consistency of Outperformance:** A high Information Ratio indicates consistent outperformance of the benchmark, highlighting the manager’s skill in generating returns.

3. **Focus on Active Risk:** The IR considers the ’active risk’ or ’tracking error’, which is the standard deviation of the excess returns over the benchmark. This provides insights into the manager’s ability to generate returns above the benchmark, adjusted for the volatility of those excess returns.

In conclusion, the Information Ratio offers a nuanced perspective on a manager’s performance against a benchmark. It complements the Sharpe Ratio by focusing on the effectiveness of active portfolio management, providing a more detailed picture of a manager’s ability to generate superior risk-adjusted returns.

3.11 Python Exercise: Calculating the Information Ratio of Berkshire Hathaway with S&P 500 as Benchmark

Objective: Download Berkshire Hathaway stock prices and S&P 500 front month futures prices, compute annualized excess returns over the benchmark, and calculate the Information Ratio.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance

Steps:

1. **Install Required Libraries:** Ensure you have the necessary libraries installed. Use pip to install them if needed:

```
pip install pandas numpy yfinance
```

2. **Download Data:**

```python
    import pandas as pd
    import numpy as np
    import yfinance as yf

    # Download Berkshire Hathaway stock data
    berkshire = yf.download(’BRK-A’)[’Adj Close’]
    # Download S&P 500 front month futures data
    sp500_futures = yf.download(’ES=F’)[’Adj Close’]

```

3. **Calculate Daily Logarithmic Returns:**

```
    # Compute latest starting date
    berkshire_start_date = berkshire.index[0]
    sp500_start_date = sp500_futures.index[0]
    latest_start_date = max(berkshire_start_date, sp500_start_date)

    # Use only data starting from the latest starting data,
    # so that we have the same data points for both series
    berkshire = berkshire.loc[latest_start_date:]
    sp500_futures = sp500_futures.loc[latest_start_date:]

    # Calculate daily logarithmic returns
    berkshire[’Log_Returns’] = np.log(berkshire / berkshire.shift(1))
    sp500_futures[’Log_Returns’] = np.log(sp500_futures / sp500_futures.shift(1))

```

4. **Compute Excess Returns over the Benchmark:**

```
    # Compute excess returns
    excess_returns = berkshire[’Log_Returns’] - sp500_futures[’Log_Returns’]

```

5. **Annualize Excess Returns and Calculate Information Ratio:**

```
    # Annualize excess returns
    annualized_excess_return = excess_returns.mean() * 252
    tracking_error = excess_returns.std() * np.sqrt(252)

    # Calculate the Information Ratio
    information_ratio = annualized_excess_return / tracking_error
    print("Information Ratio:", information_ratio)

```

Note: This exercise is focused on calculating the Information Ratio of Berkshire Hathaway’s returns compared to the S&P 500 futures as a benchmark. The Information Ratio is a crucial metric for evaluating the performance of an investment relative to a benchmark, offering insights into the risk-adjusted excess returns achieved by the portfolio.

3.12 Python Exercise: Calculating the Information Ratio of an Equally-Weighted FAANG Portfolio with NASDAQ as Benchmark

Objective: Create an equally-weighted portfolio of FAANG companies (Facebook, Apple, Amazon, Netflix, Google), download their stock prices and NASDAQ index prices, and guide students to compute the annualized excess returns over the NASDAQ benchmark, and calculate the Information Ratio.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance

Task Overview:

1. **Install Required Libraries:** Ensure the necessary libraries are installed. Use pip to install them if needed:

```
pip install pandas numpy yfinance
```

2. **Download Data:**

```python
import pandas as pd
import numpy as np
import yfinance as yf

# Download stock data for FAANG companies
faang_tickers = ['META', 'AAPL', 'AMZN', 'NFLX', 'GOOGL']
faang_data = yf.download(faang_tickers, start='YYYY-MM-DD')['Adj Close']

# Download NASDAQ index data
nasdaq_data = yf.download('NQ=F', start='YYYY-MM-DD')['Adj Close']

```

3. **Instructions for the Rest of the Exercise:**

   (a) Compute daily logarithmic returns for each FAANG company and the NASDAQ index.

   (b) Create an equally-weighted portfolio of FAANG companies by averaging their daily returns.

   (c) Calculate the excess returns of the FAANG portfolio over the NASDAQ benchmark.

   (d) Annualize the excess returns and compute the standard deviation of these excess returns (tracking error).

(e) Calculate the Information Ratio using the formula Information Ratio = $\frac{\text{Annualized Excess Return}}{\text{Tracking Error}}$.

(f) Interpret the results, focusing on what the Information Ratio indicates about the FAANG portfolio's performance relative to the NASDAQ.

Guidance:

- Ensure accurate alignment of the time frames and data points for both the FAANG portfolio and the NASDAQ index.
- Pay attention to the calculation of the tracking error, as it is crucial for the accurate computation of the Information Ratio.
- Reflect on the implications of the Information Ratio in the context of the FAANG portfolio's risk-adjusted performance compared to the NASDAQ benchmark.

Note: This exercise aims to develop an understanding of the Information Ratio as a measure of risk-adjusted performance relative to a benchmark. It also emphasizes the significance of portfolio construction and benchmark selection in performance evaluation.

Objective for Students: Students should complete the exercise by following the instructions provided after downloading the data. They will need to apply their understanding of Python programming and financial analysis concepts to calculate and interpret the Information Ratio of the FAANG portfolio compared to the NASDAQ.

3.13 Limitations of Sharpe Ratio and Information Ratio

While the Sharpe Ratio and Information Ratio can be helpful tools in investment analysis and portfolio management, they do have certain limitations:

3.13.1 Limitations of the Sharpe Ratio

1. Sensitivity to Risk-Free Rate: The Sharpe Ratio is sensitive to changes in the risk-free rate. During periods of low interest rates, the Sharpe Ratio can be artificially inflated, suggesting better risk-adjusted performance than is actually the case.

2. Single Period Measure: It typically uses a single-period model of risk and return, which may not accurately capture the investment's risk over multiple periods or in different market conditions.

3. No Distinction Between Positive and Negative Volatility: The Sharpe Ratio treats all volatility as bad, not differentiating between upside and downside volatility. This could be misleading for investments that have high positive volatility.

3.13.2 Limitations of the Information Ratio

1. Benchmark Dependency: The effectiveness of the Information Ratio is highly dependent on the choice of benchmark. An inappropriate benchmark can lead to misleading IR values.

2. Tracking Error as a Risk Measure: The IR uses tracking error as a measure of risk, which only reflects the deviation from the benchmark, not the total risk of the portfolio.

3. Not Suitable for Passive Management: The IR is less relevant for passively managed funds or portfolios that aim to replicate a benchmark, as these strategies inherently have low tracking errors.

4. Time Period Sensitivity: Like the Sharpe Ratio, the Information Ratio can be sensitive to the time period chosen for analysis. Different periods can yield significantly different IR values.

3.13.3 General Limitations

1. Risk Measure Limitations: Both ratios use standard deviation as a measure of risk, which may not fully capture the true risk of an investment, especially for asymmetrical return distributions..

2. Not Accounting for Tail Risk: Neither ratio adequately captures tail risk (the risk of extreme loss), which can be critical in assessing the safety of an investment.

3. Historical Data Dependency: Both ratios are based on historical data, which may not be a reliable indicator of future performance, especially in rapidly changing market conditions.

4. No Consideration for Investment Objectives: These metrics do not take into account the specific investment objectives or constraints of an investor.

In summary, while the Sharpe Ratio and Information Ratio provide valuable insights into risk-adjusted returns, they should be used in conjunction with other metrics and qualitative analysis to get a comprehensive understanding of an investment's performance and risk profile.

3.14 Quiz

For each question, select one or more correct answers.

1. Consider the scenario where an investor seeks to evaluate the performance of active fund managers. Given the purpose and characteristics of the Information Ratio (IR) and the Sharpe Ratio, which of the following statements are correct regarding the use of the Information Ratio for this purpose?

(a) The Information Ratio should be preferred over the Sharpe Ratio because it specifically measures the manager’s ability to generate returns above a specified benchmark.

(b) The Sharpe Ratio is more suitable than the Information Ratio for evaluating active fund managers since it focuses on the total risk-adjusted returns.

(c) The Information Ratio is appropriate for assessing active management as it takes into account the tracking error, reflecting the manager’s skill in outperforming the benchmark.

(d) The Information Ratio should not be used for active fund managers because it only considers the risk-free rate.

(e) The Information Ratio is preferred in this context because it provides insights into the consistency of the manager’s outperformance relative to the benchmark.

2. In the context of evaluating a portfolio manager’s performance, consider the Information Ratio (IR) and Sharpe Ratio. Which of the following statements correctly reflect the computation of the Information Ratio and its differences from the Sharpe Ratio?

   (a) The Information Ratio is calculated as $\frac{R_p - R_b}{\sigma_{\text{excess}}}$, where R_p is the portfolio return, R_b is the benchmark return, and σ_{excess} is the standard deviation of the portfolio’s excess returns over the benchmark.

   (b) The Sharpe Ratio, unlike the Information Ratio, uses the risk-free rate instead of a benchmark return in its formula.

   (c) The Information Ratio uses the standard deviation of the portfolio’s total returns, similar to the Sharpe Ratio.

   (d) The Information Ratio focuses on the excess return over a benchmark, while the Sharpe Ratio assesses the excess return over a risk-free asset.

   (e) The standard deviation in the Information Ratio formula (σ_{excess}) is the standard deviation of the portfolio’s return, not the excess return over the benchmark.

3. In the context of calculating the Information Ratio for Berkshire Hathaway against the S&P 500 as a benchmark, consider the following lines of Python code. Which of these correctly contribute to computing the Information Ratio?

   (a) `excess_returns = berkshire[’Log_Returns’] - sp500_futures[’Log_Returns’]`

   (b) `annualized_excess_return = excess_returns.mean() * 252`

   (c) `information_ratio = annualized_excess_return / excess_returns.std()`

   (d) `tracking_error = np.sqrt(np.var(excess_returns) * 252)`

(e) `information_ratio = (berkshire[’Log_Returns’].mean() - sp500_futures[’Log_Returns’].mean()) / tracking_error`

4. Which of the following lines of Python code are necessary for computing the Information Ratio of an equally-weighted FAANG portfolio against the NASDAQ benchmark? Select all that apply.

   (a) `faang_portfolio_returns = faang_data.mean(axis=1)`

   (b) `excess_returns = faang_portfolio_returns - nasdaq_data`

   (c) `annualized_excess_return = excess_returns.mean() * 252`

   (d) `tracking_error = excess_returns.std() * np.sqrt(252)`

   (e) `information_ratio = annualized_excess_return / tracking_error`

5. What are the limitations of the Sharpe Ratio?

   (a) It assumes that returns are normally distributed.

   (b) It is sensitive to changes in the risk-free rate.

   (c) It treats all volatility as bad, not differentiating between upside and downside volatility.

   (d) It is sensitive to the time period chosen for analysis.

   (e) It is highly dependent on the choice of benchmark.

6. What are the limitations of the Information Ratio?

   (a) It assumes that returns are normally distributed.

   (b) It is highly dependent on the choice of benchmark.

   (c) It uses tracking error as a measure of risk, which only reflects deviation from the benchmark.

   (d) It is suitable for passively managed funds or portfolios that aim to replicate a benchmark.

   (e) It is sensitive to changes in the risk-free rate.

3.15 Sortino Ratio

The Sortino Ratio, similar to the Sharpe Ratio, is a tool for evaluating the risk-adjusted return of an investment or portfolio. However, it addresses a specific limitation of the Sharpe Ratio.

Main Limitation of Sharpe Ratio Addressed by Sortino Ratio:
The primary limitation of the Sharpe Ratio that the Sortino Ratio seeks to address is its treatment of all volatility as equally risky. The Sharpe Ratio does not differentiate between upside and downside volatility, treating both as indicative of risk. However, investors are typically more concerned with downside risk – the risk of losses – rather than volatility associated with above-average returns.

The Solution: Sortino Ratio
The Sortino Ratio modifies the concept of the Sharpe Ratio by focusing only on the downside risk. It is calculated using the formula:

$$\text{Sortino Ratio} = \frac{R_p - R_f}{\sigma_d} \tag{7}$$

where R_p represents the return of the portfolio, R_f denotes the risk-free rate, and σ_d is the standard deviation of the portfolio's negative returns, known as downside deviation.

Why Sortino Ratio Works

- *Focus on Downside Risk:* The Sortino Ratio provides a more relevant measure of risk for investors who are primarily concerned about losses rather than fluctuations in positive returns.
- *Improved Risk Assessment:* By isolating downside volatility, the Sortino Ratio offers a more precise assessment of the risk an investor is taking in pursuit of excess returns.
- *Better Suitability for Asymmetrical Distributions:* For portfolios that exhibit skewed or non-normal return distributions, the Sortino Ratio can provide a more accurate picture of risk-adjusted performance.

Practical Application
The Sortino Ratio is particularly useful for evaluating investments or strategies that aim to minimize downside risk, such as retirement portfolios or funds with loss aversion objectives. It allows investors to understand not just the return per unit of risk, but the return per unit of downside risk, aligning more closely with many investors' risk concerns.

In conclusion, the Sortino Ratio enhances the concept of risk-adjusted return by focusing specifically on the risk that matters most to investors: the risk of losing money. It serves as a complementary metric to the Sharpe Ratio, particularly valuable when assessing investments where downside risk is a primary concern.

3.15.1 Limitations of the Sortino Ratio

While addressing certain limitations of the Sharpe Ratio, the Sortino Ratio may have at least a couple of limitations:

1. **Data Sensitivity:** Like the Sharpe Ratio, the Sortino Ratio is sensitive to the time period of data used. Different time periods can yield significantly different Sortino Ratios, affecting comparability and consistency.
2. **Lack of Context for Negative Returns:** The ratio does not differentiate between frequent small losses and infrequent large losses. Two investments with the same Sortino Ratio could have very different risk profiles if one has frequent small losses and the other has rare but large losses.

In summary, while the Sortino Ratio is a valuable tool for assessing risk-adjusted returns with a focus on downside risk, it should be used in conjunction with other metrics and a comprehensive understanding of an investment's risk profile.

3.16 Quiz

For each question, select one or more correct answers.

1. How is the Sortino Ratio computed?
   (a) Sortino Ratio = $\frac{R_p - R_f}{\sigma_p}$ where R_p is the return of the portfolio, R_f is the risk-free rate, and σ_p is the standard deviation of the portfolio's returns.
   (b) Sortino Ratio = $\frac{R_p - R_f}{\sigma_d}$ where R_p is the return of the portfolio, R_f is the risk-free rate, and σ_d is the standard deviation of the portfolio's negative returns (downside deviation).
   (c) Sortino Ratio = $\frac{R_p}{\sigma_d}$ where R_p is the return of the portfolio and σ_d is the standard deviation of the portfolio's negative returns.
   (d) Sortino Ratio = $\frac{R_f}{\sigma_p}$ where R_f is the risk-free rate and σ_p is the standard deviation of the portfolio's returns.
   (e) Sortino Ratio = $\frac{R_p}{\sigma_p}$ where R_p is the return of the portfolio and σ_p is the standard deviation of the portfolio's returns.
2. Which statements are correct about the similarities and differences between Sortino Ratio and Sharpe Ratio?

(a) The Sortino Ratio uses the standard deviation of the portfolio's returns (σ_p) whereas the Sharpe Ratio uses the downside deviation (σ_d).

(b) The Sortino Ratio and the Sharpe Ratio both use the standard deviation of the portfolio's returns (σ_p).

(c) The Sortino Ratio uses the downside deviation (σ_d) of negative returns, while the Sharpe Ratio uses the standard deviation of all returns (σ_p).

(d) Both ratios adjust returns by subtracting the risk-free rate (R_f) from the portfolio returns (R_p).

(e) The Sharpe Ratio focuses on total risk including both upside and downside volatility, whereas the Sortino Ratio focuses only on downside risk.

3. Which are limitations of the Sortino Ratio?

(a) The Sortino Ratio assumes that returns are normally distributed.

(b) The Sortino Ratio is sensitive to the time period of data used.

(c) The Sortino Ratio does not differentiate between upside and downside volatility.

(d) The Sortino Ratio does not differentiate between frequent small losses and infrequent large losses.

(e) The Sortino Ratio requires a complex calculation method for downside deviation that involves complex numbers.

3.17 Calmar Ratio

The Calmar Ratio, like the Sortino Ratio, is a tool for evaluating the risk-adjusted return of an investment or portfolio. However, it addresses different limitations compared to the Sortino Ratio, particularly focusing on the risk associated with significant drawdowns.

Limitations of Sortino Ratio Addressed by Calmar Ratio:
While the Sortino Ratio improves upon the Sharpe Ratio by focusing on downside volatility, it still does not directly address the impact of large drawdowns, which can be a significant concern for investors. The Sortino Ratio also does not differentiate between different types of negative returns, treating frequent small losses similarly to infrequent large losses. The Calmar Ratio addresses these limitations by specifically focusing on the maximum drawdown experienced by a portfolio or investment.

The Solution: Calmar Ratio
The Calmar Ratio measures the relationship between the annual return of an investment and its maximum drawdown over a specified period. It is defined using the formula:

$$\text{Calmar Ratio} = \frac{\text{Annualized Return}}{\text{Maximum Drawdown}} \tag{8}$$

where the Maximum Drawdown is the largest peak-to-trough decline in the value of the portfolio over a specified period.

Why Calmar Ratio Works

- *Focus on Drawdown Risk:* The Calmar Ratio specifically addresses the risk of significant drawdowns, which are a primary concern for many investors.
- *Better Assessment of Extreme Risk:* By comparing returns to the maximum drawdown, the Calmar Ratio provides a clear picture of how much return an investment is generating per unit of the most extreme risk taken.
- *Useful for Long-Term Investments:* The Calmar Ratio is particularly relevant for long-term investments where the impact of large drawdowns can be more pronounced and potentially more damaging.

Practical Application
The Calmar Ratio is most useful for evaluating the performance of investment strategies over longer periods, particularly those that aim to minimize the impact of significant market downturns. It is an important tool for retirement portfolios and long-term investment plans where recovery from large drawdowns is a critical concern.

In conclusion, the Calmar Ratio complements the Sortino Ratio by adding a focus on the risk of large drawdowns. While the Sortino Ratio offers a measure of risk-adjusted performance considering downside volatility, the Calmar Ratio provides insights into the performance relative to the most significant declines, thereby offering a more complete view of risk-adjusted returns.

3.17.1 Limitations of the Calmar Ratio

Similar to other risk-adjusted performance metrics, the Calmar Ratio also has its limitations:

1. **Period Sensitivity:** The Calmar Ratio can be highly sensitive to the time period chosen for analysis, as it depends on the occurrence and magnitude of the maximum drawdown during that period.

2. **Less Relevant for Short-Term Analysis:** For short-term investments or trading strategies, the Calmar Ratio may not be as relevant, given that maximum drawdowns are more impactful over longer time horizons.

Therefore, while the Calmar Ratio is a valuable tool for assessing the performance of long-term investments against extreme risks, it should be used in conjunction with other metrics to gain a comprehensive understanding of an investment's performance and risk profile.

3.18 Python Exercise: Calculating and Visualizing Comprehensive Metrics for a Diversified Portfolio

Objective: This exercise involves calculating and visualizing a range of metrics, including the Sortino Ratio, Inflation-Adjusted Sortino Ratio, and Calmar Ratio, to assess the risk-adjusted performance of a diversified portfolio.

Requirements:

- Python
- Libraries: pandas, numpy, yfinance, pandas_datareader, matplotlib

Task:

1. **Download Asset Data:** Use yfinance to download historical price data for the specified ETFs, ensuring data consistency over the selected time frame.

2. **Calculate Daily Logarithmic Returns:** Compute the daily logarithmic returns for each ETF.

3. **Construct the Portfolio:** Create the portfolio based on the given weights and calculate its daily returns.

4. **Annualize Portfolio Returns:** Calculate the annualized returns of the portfolio.

5. **Calculate Annualized Volatility:** Compute the annualized volatility (standard deviation) of the portfolio's returns.

6. **Determine Downside Deviation:** Calculate the downside deviation. The code snippet for this is as follows:

```
downside_returns = portfolio_returns[portfolio_returns < 0]
annualized_downside_deviation = np.sqrt(np.sum(downside_returns**2) / len(downside_returns)) * np.sqrt(252)

```

7. **Compute Skewness and Kurtosis:** Use pandas to calculate skewness and kurtosis of the portfolio's returns.

8. **Calculate Maximum Drawdown:** Determine the maximum drawdown of the portfolio over the entire period.

9. **Calculate Sharpe, Sortino, and Calmar Ratios:** Compute the Sharpe Ratio, Sortino Ratio, and Calmar Ratio for the portfolio.

10. **Inflation Data Acquisition:** Import inflation data using the ‘pandas_datareader‘ library.

11. **Calculate Inflation-Adjusted Sharpe, Sortino, and Calmar Ratios:** Adjust the Sharpe, Sortino, and Calmar Ratios for inflation.

12. **Print Metrics:** Output the annualized return, annualized volatility, skewness, kurtosis, maximum drawdown, Sharpe ratio, inflation-adjusted Sharpe ratio, Sortino ratio, inflation-adjusted Sortino ratio, Calmar ratio, and inflation-adjusted Calmar ratio.

13. **Plot Drawdown:** Plot the ongoing drawdown of the portfolio to visualize the dips and recoveries over time.

14. **Analysis and Interpretation:** Analyze the results, focusing on the risk-adjusted performance, the impact of inflation adjustment, and the implications of the drawdown plot. Compare the different ratios to understand the portfolio's performance from various perspectives.

This comprehensive exercise is designed to provide a thorough understanding of portfolio performance metrics, emphasizing downside risk assessment, the impact of inflation, and visualizing risk through drawdown plots. It also introduces the Calmar Ratio as an additional measure for evaluating the portfolio's performance relative to its maximum drawdown, offering valuable insights into the diversified portfolio's behavior and its risk-return characteristics.

3.19 Quiz

For each question, select one or more correct answers.

1. What are the differences between the Calmar Ratio and other risk-adjusted return metrics such as the Sharpe Ratio and the Sortino Ratio?

   (a) The Calmar Ratio measures return relative to the maximum drawdown, while the Sharpe and Sortino Ratios measure return relative to standard deviation.

   (b) The Calmar Ratio is calculated using downside deviation, whereas the Sharpe and Sortino Ratios use maximum drawdown.

   (c) The Calmar Ratio is not sensitive to the period of analysis, as opposed to the Sharpe and Sortino Ratios.

   (d) The Sharpe Ratio considers all volatility, whereas the Calmar Ratio focuses on the risk of large drawdowns.

   (e) The Sortino Ratio and the Calmar Ratio both focus exclusively on downside risk, but the Sortino Ratio uses downside deviation while the Calmar Ratio uses maximum drawdown.

2. What are the strengths of the Calmar Ratio compared to other risk-adjusted return metrics such as the Sharpe Ratio and the Sortino Ratio?

   (a) The Calmar Ratio provides a more comprehensive view of risk by considering the maximum drawdown, which is not directly addressed by the Sharpe or Sortino Ratios.

   (b) The Calmar Ratio is less sensitive to the time period of analysis compared to the Sharpe and Sortino Ratios.

   (c) The Calmar Ratio is particularly relevant for long-term investments because it provides insight on how long it will take to recover from worst-case losses, unlike the Sharpe and Sortino Ratios.

   (d) The Calmar Ratio specifically focuses on the risk of significant drawdowns, which is a primary concern for many investors, whereas the Sharpe and Sortino Ratios focus on volatility and downside volatility, respectively.

   (e) The Calmar Ratio's emphasis on maximum drawdown makes it a valuable metric during periods of market stress or downturns, offering a different perspective compared to the Sharpe and Sortino Ratios.

3. Which of the following lines of Python code correctly compute the Sortino Ratio?

   (a)

   ```
   sortino_ratio = (portfolio_return - risk_free_rate)
   / np.std(portfolio_returns)
   ```

(b)

```
downside_returns = portfolio_returns[portfolio_returns < 0]
sortino_ratio = (portfolio_return - risk_free_rate)
/ np.std(downside_returns)
```

(c)

```
downside_returns = portfolio_returns[portfolio_returns < 0]
sortino_ratio = (portfolio_return - risk_free_rate)
/ (np.sqrt(np.sum(downside_returns**2) /
len(downside_returns)) * np.sqrt(252))
```

(d)

```
sortino_ratio = (portfolio_return - risk_free_rate) /
np.std(portfolio_returns[portfolio_returns < 0])
```

(e)

```
annualized_downside_deviation =
portfolio_returns[portfolio_returns < 0].std() * np.sqrt(252)
sortino_ratio = (portfolio_return - risk_free_rate) /
annualized_downside_deviation
```

4. Which of the following lines of Python code correctly compute the Calmar Ratio?

(a)

```
calmar_ratio = annualized_return / max_drawdown
```

(b)

```
calmar_ratio = annualized_return / np.std(portfolio_returns)
```

(c)

```
calmar_ratio = (annualized_return + risk_free_rate) / max_drawdown
```

(d)

```
max_drawdown = np.min(rolling_max - portfolio_value) / rolling_max
calmar_ratio = annualized_return / max_drawdown
```

(e)

```
calmar_ratio = annualized_return / np.min(portfolio_returns)
```

References

[Bodie et al., 2018] Bodie, Z., Kane, A., and Marcus, A. J. (2018). *Investments*. McGraw-Hill Education.

[Campbell and Shiller, 1988] Campbell, J. Y. and Shiller, R. J. (1988). The dividend-price ratio and expectations of future dividends and discount factors. *The Review of Financial Studies*, 1(3):195–228.

[Fabozzi et al., 2010] Fabozzi, F. J., Modigliani, F., and Jones, F. J. (2010). *Foundations of Financial Markets and Institutions*. Pearson Education.

[Lo and MacKinlay, 1988] Lo, A. W. and MacKinlay, A. C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *The Review of Financial Studies*, 1(1):41–66.