Problem 1:

Model Classical Brownian Motion, arithmetic return system, and log return (Gerometric Brownian Motion) and compare theoretical and simulated mean and standard deviation. `r_t` is assumed to follow `N(0, σ^2)`

```python
import numpy as np
import matplotlib.pyplot as plt
import os
path = os.getcwd()
```

[57] ✓ 0.0s Python

Approach

1. Calculation of expected mean and standard deviation is done by manual calculation using the knowledge of normal distribution of r_t.
2. For the estimated price, I simulated 10,000 iterations for each of the three methods of calculating the price at time `t`, using a standard deviation (`sigma`) of 0.1 for the normal distribution of returns `r_t` and an initial price (`p_{t-1}`) of 100.
3. Mean and standard deviation are calculated from the saved list.

Expected Results

We assume that (P_{t-1} = m), where (P_{t-1}) is known as today's price and is a constant.

Given (r_t \sim N(0, \sigma^2)), the following are the derivations for each model:

let $P_{t-1} = m$ (constant)

1. **Classical Brownian Motion**:
   - Expected Mean: `\( E[P_t] = E[P_{t-1} + r_t] = P_{t-1} + E[r_t] = m \)`
   - Standard Deviation: `\( SD(P_t) = SD(P_{t-1} + r_t) = \sigma \)`

$E(P_t) = P_{t-1} + E(r_t) = m$

$SD(P_t) = SD(r_t + P_{t-1}) = \sigma$

2. **Arithmetic Return System**:
   - Expected Mean: `\( E[P_t] = E[P_{t-1} \cdot (1 + r_t)] = P_{t-1} + E[r_t] \cdot P_{t-1} = m \)`
   - Standard Deviation: `\( SD(P_t) = SD(P_{t-1} * (1 + r_t)) = P_{t-1} \cdot SD(r_t) = m * \sigma \)`

$E(P_t) = E(P_{t-1}(1+r_t)) = P_{t-1} + E(r_t) P_{t-1} = m$

$SD(P_t) = SD(P_{t-1}(1+r_t)) = P_{t-1} \cdot SD(1+r_t) = m\sigma$

3. **Log Return or Geometric Brownian Motion**:
   - Expected Mean: `\( E[P_t] = E[P_{t-1} * e^{r_t}] = P_{t-1} * E[e^{r_t}] = m * e^{\frac{\sigma^2}{2}} \)`
   - Standard Deviation: `\( SD(P_t) = SD(P_{t-1} * e^{r_t}) = P_{t-1} * SD(e^{r_t}) = m * \sqrt{e^{\sigma^2} - 1} \)`

$E(P_t) = E(P_{t-1} e^{r_t}) = P_{t-1} E(e^{r_t}) = m e^{\frac{1}{2}\sigma^2}$

$SD(P_t) = SD(P_{t-1} e^{r_t}) = P_{t-1} SD(e^{r_t}) = m\sqrt{e^{\sigma^2}-1}$

In the case of the log return or geometric Brownian motion, the expected mean is multiplied by `\( e^{\frac{\sigma^2}{2}} \)` due to the properties of the log-normal distribution when exponentiating a normally distributed variable. The standard deviation is the initial price multiplied by the square root of `\( (e^{\sigma^2} - 1) \)`, reflecting the variance of the log-normal distribution.

Simulated Results and Analysis

1. For the classical Brownian motion (using the formula `P_t = P_{t-1} + r_t`):
   - Mean price: Approximately 100.00 (Expected 100.00)
   - Standard Deviation of the price: Approximately 0.10 (Expected 0.10)
2. For the arithmetic return system (using the formula `P_t = P_{t-1} * (1 + r_t)`):
   - Mean price: Approximately 99.82 (Expected 100.00)
   - Standard Deviation of the price: Approximately 9.88 (Expected 10.00)
3. For the log return (geometric Brownian motion using the formula `P_t = P_{t-1} * exp(r_t)`):
   - Mean price: Approximately 100.30 (Expected 100.50)
   - Standard Deviation of the price: Approximately 9.94 (Expected 10.03)

As expected, the mean prices are around the starting price of 100, and the standard deviations reflect the volatility introduced by the returns `r_t` simulated from the normal distribution. The arithmetic and log return methods yield a higher standard deviation due to the multiplicative effect on the initial price, while the classical Brownian motion reflects only the standard deviation of the return since it's added directly to the initial price.

Problem 2: VaR

Allowing the user to specify the method of return calculation, calculate the arithmetic returns of all prices of META in DailyPrices.csv

Approach

First, I calculate the arithmetic returns of the stocks by using the pct_change() method, which computes the percentage change between the current and a prior element. Then, I substrct the mean of the META returns from each return value to normalize the META returns and this centers the mean of the returns at 0. For the 95% confidence level, I calculated VaR using five different methods:

- Normal Distribution: I use the scipy.stats norm function to calculate VaR based on a normal distribution using the mean and standard deviation of the META returns.
- Normal Distribution with EWMA: I apply an exponentially weighted moving average to the variance of the META returns with a lambda parameter of 0.94 and then calculate VaR using the adjusted variance.
- MLE Fitted T Distribution: I fit a T distribution to the META returns using maximum likelihood estimation and calculate VaR from this fitted distribution.
- AR(1) Model: I fit an autoregressive model of order 1 to the returns, simulate future returns based on this model, and then compute VaR from these simulated values.
- Historic Simulation: I directly calculate VaR from the historical distribution of META returns without any distributional assumptions.

Results

Normal Distribution: -5.43%,
Normal EWMA: -3.01%,
MLE T Distribution:-4.31%,
AR(1) Model: -5.26%,
Historic Simulation: -3.95%

Analysis

When comparing the VaR values obtained from each method, the Normal Distribution and AR(1) model yielded the highest VaR, indicating highest risk, while the Normal EWMA yield the lowest VaR. The differences are due to the various assumptions each method makes about the distribution of stock returns and the weight they place on historical data.

Problem 3

Approach

For this problem, I used an exponentially weighted moving covariance (EWMC) method to account for the fact that recent returns have more impact on the current risk profile than older returns. This is reflected in the choice of `LAMDA = 0.94`, which implies that recent observations have a higher weight in the calculation. I've used the covariance matrix to calculate the portfolio variance, which is then used to calculate the standard deviation and VaR. The results represent the maximum expected loss over a specified period at a 95% confidence level:

Portfolio 'A': VaR = $15,206.39
Portfolio 'B': VaR = $7,741.25
Portfolio 'C': VaR = $17,877.73
Total Portfolio VaR = $40,825.38

Choice of Alternative Model: Historical Simulation

In the second part, I chose historical simulation because I believe actual historical return distribution is a better measure of risk than the assumption of a normal distribution as the data exhibits skewness after calculating the third moment. For example, NVDA shows a skewness of 1.602, which is quite high and indicates a distribution with a pronounced right tail, implying the presence of extreme positive returns. On the other hand, ZTS shows a slight negative skewness, implying a distribution with a longer left tail. These skewness metrics can inform the risk management process by highlighting the potential for asymmetric risk or extreme values that would not be captured by models assuming a normal distribution of returns.

Analysis of Results

In historical method, the actual historical returns are used to simulate potential future outcomes, and VaR is determined based on the percentile of these outcomes. The results are:

Portfolio 'A': VaR = $16,987.48
Portfolio 'B': VaR = $10,980.36
Portfolio 'C': VaR = $22,143.33
Total Portfolio VaR = $50111.17

These values represent the maximum expected loss at a 95% confidence level, based on historical data.

This result is different from that generated by EWMC because the latter assumes that recent returns are more indicative of future risk while the former assumes equal weighting.