Problem1

To simulate price P_t at time t with three types of price returns:

1. Classical Brownian Motion

$$P_t = P_{t-1} + r_t$$

2. Arithmetic Return System

$$P_t = P_{t-1}\left(1 + r_t\right)$$

3. Log Return or Geometric Brownian Motion

$$P_t = P_{t-1} e^{r_t}$$

For Classical Brownian Motion, I take t = 1000, sigma = 0.02 which is the standard deviation, and make the initial price P_0 = 100. Initialize random normal sets r_t with mean of 0 and standard deviation of sigma. And then do 10000 times simulation of P_t. Then find the average and standard deviation of the P_t list.

The expected price after 1000 times motion is 100 because mean of r_t is zero
The expected standard deviation is calculated by sigma * sqrt(T), which match the result

We can see the Mean and Standard deviation all match that our expected.

```
Mean of P: 100.0106736464363
Expected Mean of P: 100
Standard deviation of P: 0.6317597302656832
Expected Standard deviation of P: 0.6324555320336759
```

For Arithmetic Return System, I take t = 1000, sigma = 0.02 which is the standard deviation, and make the initial price P_0 = 100. Initialize random normal sets r_t with mean of 0 and standard deviation of sigma. And then do 10000 times simulation of P_t. Then find the average and standard deviation of the P_t list.

The expected price after 1000 times motion is 100 because mean of r_t is zero.
The expected standard deviation is calculated by the mean of all standard deviation of P_(t-1).

We can see the Mean and Standard deviation all match that our expected.

```
Mean of P: 99.5979556116782
Standard deviation of P: 70.1740386041003
Expected Mean of P: 100
Expected Standard deviation of P: 70.0908271945894
```

- For Geometric Brownian Motion, I take t = 1000, sigma = 0.02 which is the standard deviation, and make the initial price P_0 = 100. Initialize random normal sets r_t with mean of 0 and standard deviation of sigma. And then do 10000 times simulation of P_t. Then find the average and standard deviation of the P_t list.

  The expected price is calculated by $P_0 * exp(sigma^2/2 * T)$ which is 122.14 in this case
  The expected standard deviation is calculated by $P_0 * sqrt((exp(sigma^2 * T) - 1) * exp(2 * mu * T))$, where mu = $(sigma^2/2)$

  We can see the Mean and Standard Deviation are roughly the same.

```
Mean of P: 122.55662658790327
Standard deviation of P: 86.57656685285664
Expected Mean of P: 122.14027581601698
Expected Standard deviation of P: 85.65723733877934
```

Overall, after our 10000 times simulation, P_t mean and standard deviation are both matching our expectations.

Problem 2

1. Implement function called return_calculate() that calculates returns from a Data Frame of prices. The function takes three arguments, prices, method, and date_column. prices is a Pandas Data Frame that contains columns of prices for different assets, method is a string that specifies the method to use for calculating returns, and date_column is the name of the column in the prices Data Frame that contains the dates.
   The function first checks that date_column is in prices and removes it from the list of asset columns. Then it calculates the returns for each asset using either the discrete or log method, depending on the value of method. Finally, it returns a new Data Frame with the dates and returns for each asset. The returns are calculated as the ratio of the current day's price to the previous day's price, with the discrete method subtracting 1 from the ratio to obtain the return, while the log method takes the natural log of the ratio.
2. Input DailyPrices.csv as data, using discrete method to get all returns rather than prices, in order to calculate portfolio later.
3. Take META’ s data as required, calculate the mean and remove the mean from the data
4. Calculate the VaR by the following 5 models
   1. Using a normal distribution.
   2. Using a normal distribution with an Exponentially Weighted variance ($\lambda = 0.94$)
   3. Using a MLE fitted T distribution.
   4. Using a fitted AR(1) model.
   5. Using a Historic Simulation.

   1. VaR using a normal distribution: This approach assumes that the returns are normally distributed. It calculates the standard deviation and mean of the returns and uses the inverse of the normal cumulative distribution function to find the VaR at a given alpha level. The output is stored in VaR_norm.
   2. VaR using a normal distribution with an Exponentially Weighted variance: This approach also assumes a normal distribution for the returns but with an Exponentially Weighted Moving Average (EWMA) of the variance. The EWMA is calculated using a smoothing factor of 0.06. The output is stored in var_ewm.
   3. VaR using a normal distribution with an Exponentially Weighted variance: This approach also assumes a normal distribution for the returns but with an Exponentially Weighted Moving Average (EWMA) of the variance. The EWMA is calculated using a smoothing factor of 0.06. The output is stored in var_ewm.
   4. VaR using a fitted AR(1) model : This approach fits an AR(1) model to the returns and uses it to find the VaR at a given alpha level. The output is stored in var_ar1.
   5. VaR using a Historic Simulation: This approach uses the historical data to calculate the VaR at a given alpha level. It sorts the returns in ascending order, calculates the percentile at the given alpha level and uses it to find the VaR. The output is stored in VaR_hs.

```
VaR using normal distribution:  -0.06546917484881122
VaR using normal distribution with EWM variance: -0.09019
VaR using T distribution:  -0.08222424439916937
VaR using fitted AR(1) model: -0.06560
VaR using historic simulation:  -0.0546200790823787
```

Conclusion

The normal distribution approach provides the highest VaR estimate, followed by the T distribution method, EWM variance normal distribution, AR(1) model and historic simulation method which provide the lowest VaR estimate. The results highlight the importance of the method used and the assumptions made in calculating VaR, and it is important to consider different methods to get a more comprehensive view of the potential risk.

Problem 3

Description

Using Portfolio.csv and DailyPrices.csv. Assume the expected return on all stocks is 0.

1. I created a script that reads in a portfolio and daily prices data in CSV format and calculates the value at risk (VaR) for each portfolio as well as the total VaR. The script uses NumPy and pandas libraries to manipulate the data.
2. To start, I retrieved the list of stocks and holdings for each portfolio from the portfolio data using pandas. I then used NumPy to calculate the daily returns for each portfolio based on the daily price data.
3. Next, I calculated the covariance matrix for each portfolio using an exponentially weighted covariance with a lambda value of 0.94. I also calculated the total covariance matrix using the same exponential weighting. The exponential weighting gives more weight to recent data, which is more relevant in financial markets.
4. Using the current stock holdings and the most recent stock prices, I calculated the value of each portfolio and the total portfolio value. Finally, I calculated the VaR for each portfolio and the total VaR using a confidence level of 0.95 and the inverse of the cumulative distribution function of the standard normal distribution. The VaR is a statistical measure of the potential loss in value of an investment over a certain period of time.
5. Overall, this script can be used to assess the risk of different portfolios and help investors make informed decisions based on their risk appetite.

Data

Portfolio A VaR: $5691.55
Portfolio B VaR: $4531.82
Portfolio C VaR: $3837.72
Total VaR: $13704.72

Conclusion

Based on the analysis conducted, we can conclude that the three portfolios have different levels of risk as indicated by their VaR values. Portfolio A has the highest VaR value of $5691.55, followed by portfolio B with a VaR of $4531.82, and portfolio C with a VaR of $3837.72. This implies that the risk associated with portfolio A is higher than the other two portfolios.

Moreover, the total VaR for all three portfolios combined is $13704.72, which represents the minimum loss that the portfolios may incur with a 95% confidence level. It is essential for investors to monitor the VaR of their portfolios regularly to manage their risk exposure effectively. This analysis can provide insights to investors in designing their investment strategies and making informed decisions based on their risk tolerance levels.

6. Calculate the model and returns by using AR(1) model instead of exponentially weighted covariance

We first need to fit an AR(1) model to the daily returns of each stock in the portfolios. This can be done using the ARMA class from the statsmodels library in Python.

Once we have the fitted AR(1) model, we can use it to generate the covariance matrix for each portfolio. This is done by first generating the autoregressive covariance matrix, which is a matrix where the (i, j)th entry is equal to the correlation between the ith and jth stock's returns, multiplied by the square root of the variance of the ith stock's returns and the square root of the variance of the jth stock's returns. We then use this autoregressive covariance matrix to generate the portfolio covariance matrix.

To calculate the VaR using the AR(1) model, we use the same formula as before, but substitute the AR(1) covariance matrix for the exponentially weighted covariance matrix.

Result:

Portfolio A VaR: $259.31091393274534
Portfolio B VaR: $220.67137374670634
Portfolio C VaR: $172.94982943584452
Portfolio total VaR: $652.9321171152961

Conclusion:

The VaR results from the AR(1) model are generally lower than the results from the exponentially weighted covariance method. This suggests that the AR(1) model is a more conservative approach to estimating VaR, which may be desirable for risk management purposes. However, it is also important to note that the AR(1) model may not capture all of the complex relationships between asset returns, which could lead to underestimation of risk in certain scenarios. Therefore, it is important to consider the strengths and limitations of both methods when choosing an approach for VaR estimation.