Question 1

By setting the mean of initializing price p0=100 and its standard deviation sigma equals to 1, we can calculate the expected value of pt and the standard deviation of pt

Brownian Motion Return:

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

Expected Value: E[Pt] = E[P(t-1)}+E(rt) = E[P(t-1)] = 100
Standard Deviation: SD[Pt] = SD[rt] = sigma = 1
Code generated Pt value: 99.98156627984173
Code generated Pt standard deviation: 0.9875565681761207

Arithmetic Return:

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

Expected Value: E[Pt] = E[P(t-1)*(1+rt)] = 100 as E[rt] = 0
Standard Deviation: SD[Pt] = SD[P(t-1)*(1+rt)] = P(t-1)* sqrt(Var[1+rt]) = P(t-1)*sigma = 100
Code generated Pt value: 98.15662798417343
Code generated Pt standard deviation: 98.75565681761208

Log Return:

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

Expected Value: E[Pt]=P(t-1)* e^(sigma^2/2) = 100* e^1/2
Standard Deviation:

$$SD[P_t] = SD[P_{t-1} * e^{r_t}] = P_{t-1} * SD[e^{r_t}] = P_{t-1} * \sqrt{(e^{\sigma^2} - 1) * e^{2*0+\sigma^2}} = P_{t-1} * \sqrt{(e^{\sigma^2} - 1) * e^{\sigma^2}}$$

=100*sqrt[(e-1)*e]
Code generated Pt value: 160.28465399880852
Code generated Pt standard deviation: 204.03015934540528

From the analysis shown above, we can see the difference between calculated value and code generated value are pretty small. Below is the screen shot of the code generated value.

```
brownian Motion Return Mean: 99.98156627984173 Brownian Motion Return std: 0.9875565681761207
Arithmetic Return Mean: 98.15662798417343 Arithmetic Return std: 98.75565681761208
Log Return Mean: 160.28465399880852 Log Return std: 204.03015934540528
```

Question 2

For all the methods below, I used the embedded function ppf() in scipy.stats to calculate the inverse pdf model for various distribution. Using the confidence level of 5% and lambda= 0.94 for exponential weighted average variance.

Before processing to the calculation, we need to first extract the META column and calculate its return using arithmetic return. Then calculate its return mean and remove it to make the return centered in zero. After the preprocessing, META's return looks like this:

```
1   -0.033266
2   -0.013890
3    0.008882
4    0.007625
5    0.040962
```

Method 1: Normal distribution

By calculating the return mean and return standard deviation of META, we could directly get the VaR using ppf. The percentage VaR at 5% is 5.43%.

Method 2: Normal distribution with exponential weighted average

Using the ewm function to calculate its exponentially weighted variance. By using the ppf function, we can get the percentage VaR at 5% equals to 2.87%

Method 3: MLE fitted t distribution

Using the t.fit function, we could get the parameters needed for t-distribution. The percentage VaR at 5% is 4.31%

Method 4: fitter AR(1) model

By defined the AR1 model using the ARIMA in statesmodel followed by fit() function, we could obtain the parameters needed for ppf. The percentage VaR at 5% equals to 5.37%

Method 5: Historical simulation

By using the quantile function on historical return, we get the percentage VaR at 5% equals to 3.95%

From the results above, we can see that the VaR for different distribution varies a lot, the largest VaR is almost as twice as the smallest one, which indicates that choosing the return distribution model could strongly affect the VaR. Comparing the absolute values in this situation, we can conclude that normal> AR1>T>historical>exponential weighted variance.

Below is the screenshot of code generated value:

```
VaR using normal distribution is: -0.05428693242254699
VaR using normal distribution with exponentially weighted variance is: -0.028695021689623904
VaR using T distribution is: -0.04313471495037609
VaR using AR1 is: -0.053730339133240025
VaR using historical distribution is: -0.03948424995533789
```

Question 3

The methodology for question 3 is similar to question 2, except that we need to calculate the portfolio value and portfolio return.

Method: Exponential weighted variance

Step 1: construct the portfolio value, by using holding (from portfolio,csv) * stock price (from dailyprices.csv) joined by stock symbol. Besides portfolio A B and C, we also need to construct the total portfolio which equals to A+B+C. Below is the 5 head rows from portfolio value

```
portfolio value:
 Portfolio              A              B             C  Total Portfolio
Date
2022-09-01  864511.243155  524138.141658  1.095803e+06     2.484453e+06
2022-09-02  855196.252739  519074.251166  1.081036e+06     2.455307e+06
2022-09-06  854111.899156  517844.336430  1.074900e+06     2.446856e+06
2022-09-07  868575.869512  527874.300107  1.100873e+06     2.497323e+06
2022-09-08  875384.565354  532238.524863  1.134243e+06     2.541866e+06
```

Step 2: generate the portfolio return using the return_calculate function in question 2 by setting Method = discrete. Below is the 5 head rows from portfolio return

```
portfolio return:
         Date         A         B         C  Total Portfolio
1  2022-09-02 -0.010775 -0.009661 -0.013476        -0.011731
2  2022-09-06 -0.001268 -0.002369 -0.005677        -0.003442
3  2022-09-07  0.016935  0.019369  0.024163         0.020625
4  2022-09-08  0.007839  0.008268  0.030313         0.017837
5  2022-09-09  0.014413  0.016863  0.020010         0.017424
```

Step 3: remove the mean of return for each portfolio

Step 4: Calculate percentage value using the calculate_VaR function in question 2. indicating method = Normal with exponential weighted variance, we can get the percentage VaR as shown below

```
Percentage VaR for exponential weighted average:
 {'A': -0.01403117161908723, 'B': -0.01355213360200832, 'C': -0.012848503862796005, 'Total Portfolio': -0.012494845446776872}
```

Step 5: Transferring the percentage VaR into dollar VaR. We could use current portfolio value, initial portfolio value and average portfolio value etc. to transfer percentage VaR into dollar VaR. I used the last day portfolio value for transferring. The last day portfolio value is

```
 Portfolio
A                  1.089316e+06
B                  5.745424e+05
C                  1.387410e+06
Total Portfolio    3.051268e+06
```

And dollar VaR is

```
Dollar VaR for exponential weighted average:
 {'A': -15284.381987563216, 'B': -7786.275434611994, 'C': -17826.136416650625, 'Total Portfolio': -38125.1229839467}
```

To sum up, the 5% dollar VaR for portfolios are listed as
A: $15284.38
B: $7786.28
C: $17826.14

Total: $38125.12

Method 2: Normal Distribution

By changing the method to Normal, we can get the percentage VaR

```
Percentage VaR for normal distribution:
 {'A': -0.018660243125807538, 'B': -0.02047235389415768, 'C': -0.018317909507458752, 'Total Portfolio': -0.018098237046485232}
```

And corresponding dollar VaR is:

```
Dollar VaR for normal distribution:
 {'A': -20326.904385351452, 'B': -11762.235445431323, 'C': -25414.44180853927, 'Total Portfolio': -55222.5728704679}
```

To sum up, the 5% dollar VaR for portfolios under normal distribution are listed as

A: $20326.98

B: $11762.24

C: $25414.44

Total: $55222.57

The reason why I choose this method is because from question 2, normal VaR is the largest and exponential weighted VaR is the smallest. By calculating these two vars, we can get a interval of the portfolio VaR under different assumption.