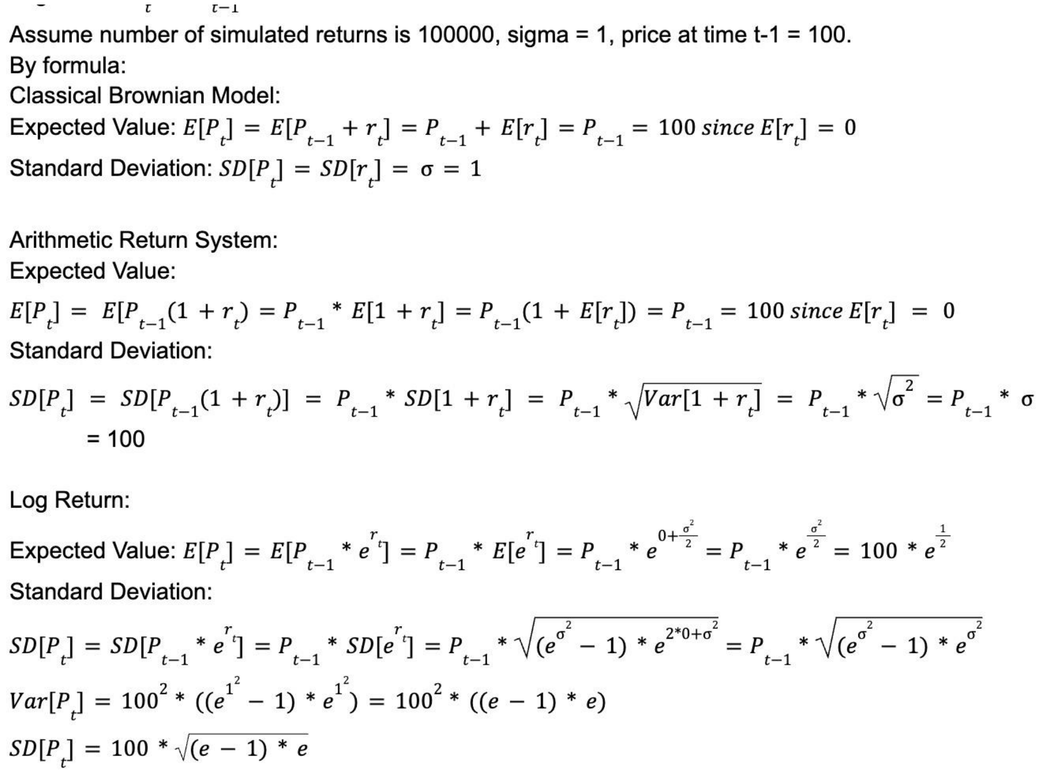
Kexin You(ky125)

Prob1



Assume number of simulated returns is 100000, sigma = 1, price at time t-1 = 100.

By formula:

Classical Brownian Model:

Expected Value: $E[P_t] = E[P_{t-1} + r_t] = P_{t-1} + E[r_t] = P_{t-1} = 100 \; since \; E[r_t] = 0$

Standard Deviation: $SD[P_t] = SD[r_t] = \sigma = 1$

Arithmetic Return System:

Expected Value:

$E[P_t] = E[P_{t-1}(1 + r_t) = P_{t-1} * E[1 + r_t] = P_{t-1}(1 + E[r_t]) = P_{t-1} = 100 \; since \; E[r_t] = 0$

Standard Deviation:

$SD[P_t] = SD[P_{t-1}(1 + r_t)] = P_{t-1} * SD[1 + r_t] = P_{t-1} * \sqrt{Var[1 + r_t]} = P_{t-1} * \sqrt{\sigma^2} = P_{t-1} * \sigma$ = 100

Log Return:

Expected Value: $E[P_t] = E[P_{t-1} * e^{r_t}] = P_{t-1} * E[e^{r_t}] = P_{t-1} * e^{0+\frac{\sigma^2}{2}} = P_{t-1} * e^{\frac{\sigma^2}{2}} = 100 * e^{\frac{1}{2}}$

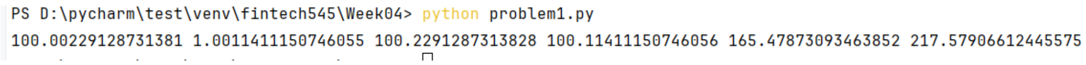
Standard Deviation:

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

$Var[P_t] = 100^2 * ((e^{1^2} - 1) * e^{1^2}) = 100^2 * ((e - 1) * e)$

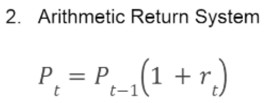
$SD[P_t] = 100 * \sqrt{(e - 1) * e}$

After run code, the outcome is as follows, which aligns with above calculation.



```
PS D:\pycharm\test\venv\fintech545\Week04> python problem1.py
100.00229128731381 1.0011411150746055 100.2291287313828 100.11411150746056 165.47873093463852 217.57906612445575
```

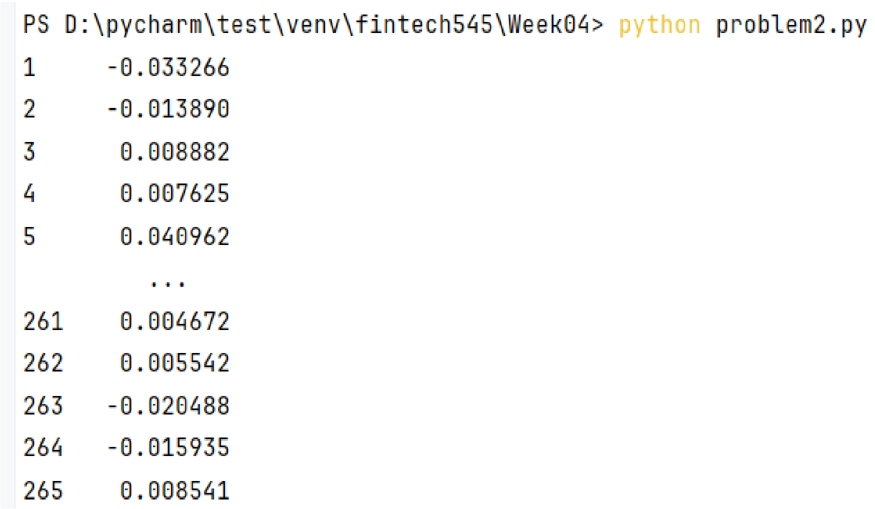
Prob2



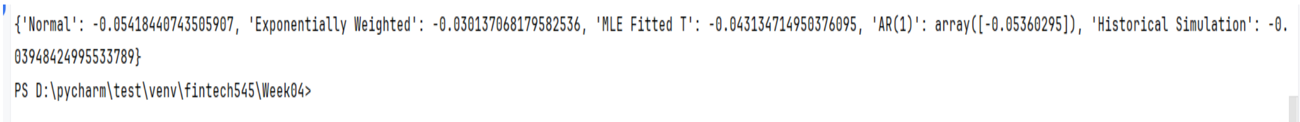
2. Arithmetic Return System

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

With “return_calculate()”, I use discrete to calculate as follows:



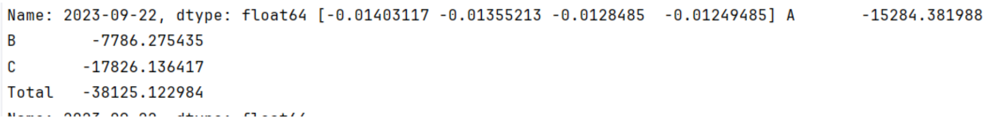
```
PS D:\pycharm\test\venv\fintech545\Week04> python problem2.py
1      -0.033266
2      -0.013890
3       0.008882
4       0.007625
5       0.040962
          ...
261     0.004672
262     0.005542
263    -0.020488
264    -0.015935
265     0.008541
```



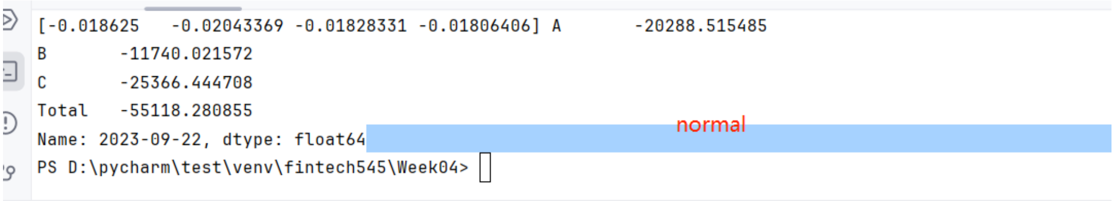
```
{'Normal': -0.05418440743505907, 'Exponentially Weighted': -0.030137068179582536, 'MLE Fitted T': -0.043134714950376095, 'AR(1)': array([-0.05360295]), 'Historical Simulation': -0.
03948424995533789}
PS D:\pycharm\test\venv\fintech545\Week04>
```

Prob3

I first calculate the portfolio value. Then use pro2..



```
Name: 2023-09-22, dtype: float64 [-0.01403117 -0.01355213 -0.0128485  -0.01249485] A       -15284.381988
B        -7786.275435
C       -17826.136417
Total   -38125.122984
Name: 2023-09-22, dtype: float64
```



```
[-0.018625   -0.02043369 -0.01828331 -0.01806406] A       -20288.515485
B       -11740.021572
C       -25366.444708
Total   -55118.280855
Name: 2023-09-22, dtype: float64
PS D:\pycharm\test\venv\fintech545\Week04>
```

I choose the normal model. The loss var is larger, and it is better model.