Week04 Project Report

Problem 1

Problem

Calculate and compare the expected value and standard deviation of price at time t (P_t) , given each of the 3 types of price returns, assuming $r_t \sim \mathcal{N}(0, \sigma^2)$. Simulate each return equation using $r_t \sim \mathcal{N}(0, \sigma^2)$ and show the mean and standard deviation match your expectations.

Answer

1.Calculate the expected value and standard deviation of price at time t

To calculate the expected value and standard deviation, we first assume that (P_t) satisfies Markov Process, that is (P_t) is only influenced by (P_{t-1}) and r_t, but not be influenced by the previous prices. We will calculate the expected value and standard deviation of (P_t) on the condition that (P_{t-1}) is known.

For Classical Brownian Motion, $P_t = P_{t-1} + r_t$, since $r_t \sim \mathcal{N}(0, \sigma^2)$, $E[P_t] = E[P_{t-1}] + E[r_t] = P_{t-1} + 0$, the expected value of (P_t) is the value of (P_{t-1}). Meanwhile, r_t is independent to P_{t-1}, $Var[P_t] = Var[P_{t-1}] + Var[r_t] = \sigma^2$.

For Arithmetic Return System, $P_t = P_{t-1}(1 + r_t)$, since r_t is independent to P_{t-1}, $E[P_t] = E[P_{t-1}] \times E[1 + r_t] = E[P_{t-1}] = P_{t-1}$, $Var[P_t] = E[P_{t-1}^2]E[(1 + r_t)^2] - E^2[P_{t-1}]E^2[1 + r_t]] = P_{t-1}^2 \times (1 + \sigma^2) - P_{t-1}^2 = P_{t-1}^2\sigma^2$.

For Geometric Brownian Motion, $P_t = P_{t-1}e^{r_t}$, since r_t is independent to P_{t-1}, $E[P_t] = E[P_{t-1}]E[e^{r_t}] = P_{t-1} \times e^{\sigma^2/2}$, $Var[P_t] = E[P_{t-1}^2]E[(e^{r_t})^2] - E^2[P_{t-1}]E^2[e^{r_t}]] = P_{t-1}^2 \times e^{2\sigma^2} - P_{t-1}^2 \times e^{\sigma^2} = P_{t-1}^2 e^{\sigma^2}(e^{\sigma^2} - 1)$.

2.Simulate each return equation and compare the results

To simulate each return equation, I assumed that $r_t \sim \mathcal{N}(0, 1)$, which means that $\sigma = 1$. Since we assume that P_t will not be influenced by the previous prices, except from P_{t-1}, I first assumed that `P_t_pre = 2` in my simulation, that is, $P_{t-1} = 2$. The simulated results are as follows.

	Expectation(Simulation/Calculation)	Standard Deviation(Simulation/Calculation)
BM	1.9882926998628376 / 2	1.0086224833809354 / 1
ARITH	1.9765853997256753 / 2	2.0172449667618704 / 2
GBM	3.260016948818 / 3.3	4.114918951889528 / 4.322

If I assumed that `P_t_pre = 0.0001`, which is very small, that is nearly 0, the simulation is as follows.

	Expectation(Simulation/Calculation)	Standard Deviation(Simulation/Calculation)

	Expectation(Simulation/Calculation)	Standard Deviation(Simulation/Calculation)
BM	-0.011607300137162298 / 0.0001	1.0086224833809354 / 1
ARITH	9.882926998628378e-05 / 0.0001	0.00010086224833809354 / 0.0001
GBM	0.0001630008474409 / 0.000165	0.0002057459475944764 / 0.000216

Now I compare the results with my calculation above. For the two P_{t-1}, the simulation and calculation are all very close. The simulation matches my calculation. I changed the P_{t-1} to compare the three methods of calculating returns.

We can see that the Classical Brownian Motion is possible to make the price negative, when the previous price is close to 0, so it is not very commonly used.

Arithmetic Return System is also possible to give negative price, but since we assume $r_t \sim \mathcal{N}(0, 1)$, it is unlikely to have $1 + r_t < 0$, thus we got a positive simulated results of its expectation. $E[P_t]$ of Arithmetic Return System and Classical Brownian Motion are the same, that is P_{t-1}, but the standard deviation of Arithmetic Return System will be influenced by the previous price, which sounds more reasonable in the practical world.

Geometric Brownian Motion is commonly used to represent stock price, especially when we price their derivatives. This will ensure that $P_t > 0$, for positive previous stock prices. The expectation of Geometric Brownian Motion will be larger than the former two, because $e^{\sigma^2} >= 1$, which will reflect the steadily increase of stock price in the practical world I think, and the standard deviation will be larger than the former two in most situations. Log returns can simplify many calculations used in financial world.

Problem 2

Problem

Implement a function similar to the "return_calculate()" in this week's code. Allow the user to specify the method of return calculation. Use DailyPrices.csv. Calculate the arithmetic returns for all prices. Remove the mean from the series so that the mean(META)=0.

Calculate VaR

1. Using a normal distribution.
2. Using a normal distribution with an Exponentially Weighted variance (λ = 0. 94)
3. Using a MLE fitted T distribution.
4. Using a fitted AR(1) model.
5. Using a Historic Simulation.

   Compare the 5 values.

Answer

1.Implement `return_calculate()` function and calculate the arithmetic returns.

I implemented the `return_calculate()` function which can use the method that user inputs. And I calculated the arithmetic returns of each stock in the `DailyPrices.csv` file. Here are the results.

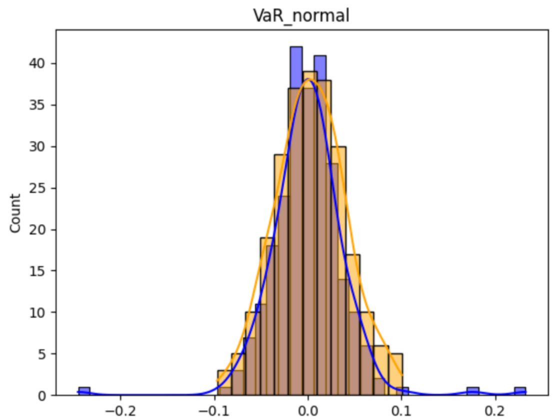
```python
def return_calculate(data, method):

    rt = pd.DataFrame(index=data.index, columns=data.columns)

    # Classical Brownian Motion
    if method == "BM":
        rt = data - data.shift(1).fillna(0)
    # Arithmetic Return System
    elif method == "ARITH_RT":
        rt = data / data.shift(1).fillna(1) - 1
    # Geometric Brownian Motions
    else:
        rt = np.log(data / data.shift(1).fillna(1))

    # The first row of data has no return
    rt.iloc[0,:] = "NaN"
    return rt
```

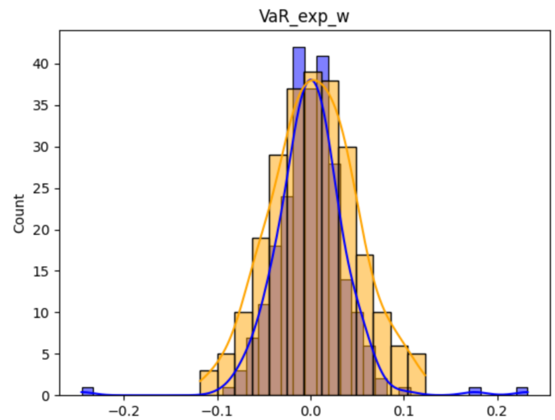
```
Arithmetric Return for DailyPrices
                      SPY      AAPL      MSFT      AMZN      TSLA  ...       LMT       SYK        GM       TFC       TJX
Date                                                               ...
2/14/2022 0:00        NaN       NaN       NaN       NaN       NaN  ...       NaN       NaN       NaN       NaN       NaN
2/15/2022 0:00   0.016127  0.023152  0.018542  0.008658  0.053291  ... -0.012275  0.033021   0.02624  0.028572  0.013237
2/16/2022 0:00   0.001121 -0.001389 -0.001167  0.010159  0.001041  ...  0.012244  0.003363  0.015301 -0.001389 -0.025984
2/17/2022 0:00  -0.021361 -0.021269 -0.029282 -0.021809 -0.050943  ...  0.004833 -0.030857 -0.031925  -0.03338 -0.028763
2/18/2022 0:00  -0.006475 -0.009356 -0.009631 -0.013262 -0.022103  ... -0.005942 -0.013674 -0.004506 -0.003677  0.015038
...                   ...       ...       ...       ...       ...  ...       ...       ...       ...       ...       ...
2/3/2023 0:00   -0.010629    0.0244 -0.023621 -0.084315  0.009083  ...  0.004134  0.002336 -0.008916 -0.005954  0.001617
2/6/2023 0:00   -0.006111 -0.017929 -0.006116 -0.011703  0.025161  ...  0.021826 -0.041181  0.005106 -0.009782 -0.004595
2/7/2023 0:00    0.013079  0.019245  0.042022 -0.000685  0.010526  ... -0.001641  0.003573  0.001451  0.008669 -0.003618
2/8/2023 0:00   -0.010935 -0.017653 -0.003102 -0.020174  0.022763  ...  0.002819 -0.015526  0.004106 -0.015391  0.009363
2/9/2023 0:00   -0.008669 -0.006912  -0.01166 -0.018091  0.029957  ...  0.000937 -0.014391  0.001443 -0.016619  0.005603

[249 rows x 100 columns]
```

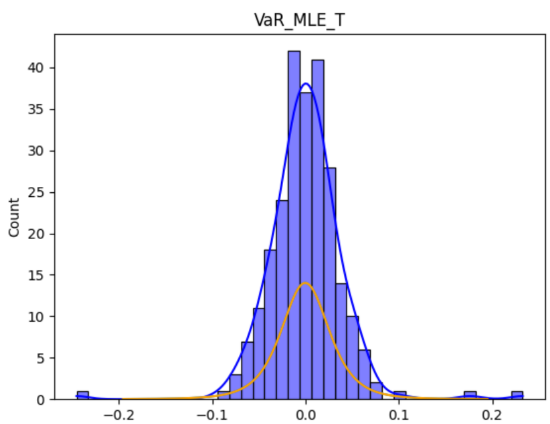
2.Remove mean from the series and calculate VaR.

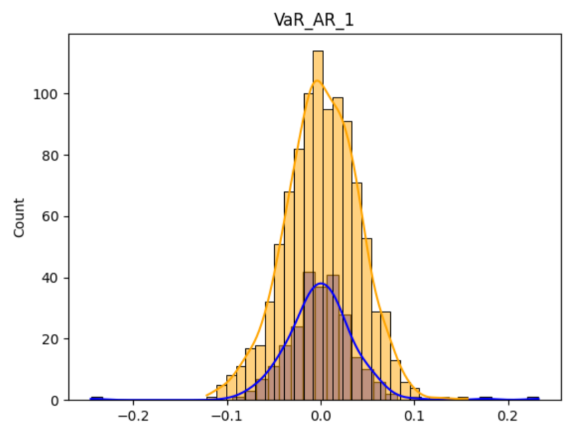
I removed the mean from the series, mean(META)=0, and used the return series of META to calculate VaR, `alpha=0.05`. Here are the results.

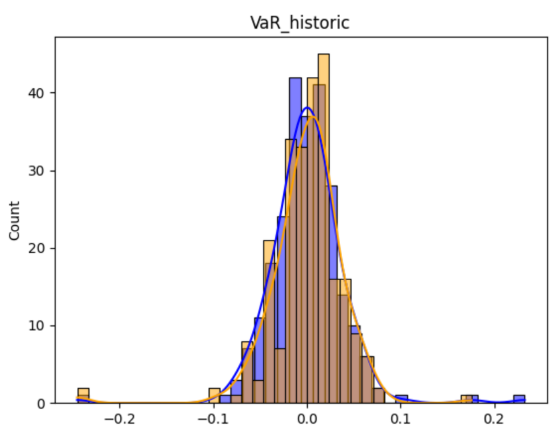
	Normal	Exponentially Normal	MLE_T	AR(1)	Historic
VaR	6.5469%	9.1385%	5.7581%	6.4337%	5.5907%
$	11.65	16	10.24	11.45	10.52











Normal Distribution is a good fit for this. The blue and orange lines are close, but it cannot simulate the outliers in the return data. Exponentially Weighted variance is theoretically a better fit than normal distribution, but when $\lambda = 0.94$, it does not perform very well, it is less centered than normal distribution, which gives evidently a larger VaR. If $\lambda = 0.97$, it will be better, but it still cannot simulate outliers very well, and still gives a larger VaR. (Maybe a larger VaR is what the risk management people need, just a joke:)

MLE fitted T distribution is better than normal distribution, it fits outliers better, and the trend looks closer to the blue line, although the count seems not to be true in the graph. I guess that maybe this is because when I maximized the log likelihood, I constrained sigma to be larger than 0, which decrease the count. But the trend and simulation is very good, to fit the outliers.

AR(1) model is the best, I think. I simulated 1000 numbers for the return after fitting AR(1) model, instead of only 248 times, to better fit the "PDF". That's why the count is far larger than the blue one. We can see that the trend is the closest to the blue line, and has a good reflect on outliers. Compared with Exponentially Weighted variance method, although the Exponentially Weighted variance gives different weights to returns to reflect the influence of time, it is not so ideal as the time series model to fit the returns.

Historic Simulation is, to some extent, more casual, because it randomly sample returns from the historic values. It will be influenced by the values it picks from the historic data, it will perform better if the historic data and the number of samples are large enough.

Problem 3

Problem

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

This file contains the stock holdings of 3 portfolios. You own each of these portfolios. Using an exponentially weighted covariance with lambda = 0.94, calculate the VaR of each portfolio as well as your total VaR (VaR of the total holdings). Express VaR as a $.

Discuss your methods and your results. Choose a different model for returns and calculate VaR again. Why did you choose that model? How did the model change affect the results?

Answer

1.Use arithmetic return and exponentially weighted covariance with lambda = 0.94

	Portfolio A	Portfolio B	Portfolio C	Portfolio A+B+C
VaR	1.868241%	1.485023%	1.388189%	1.543658%
$	5603.79	4371.69	3748.70	13343.05

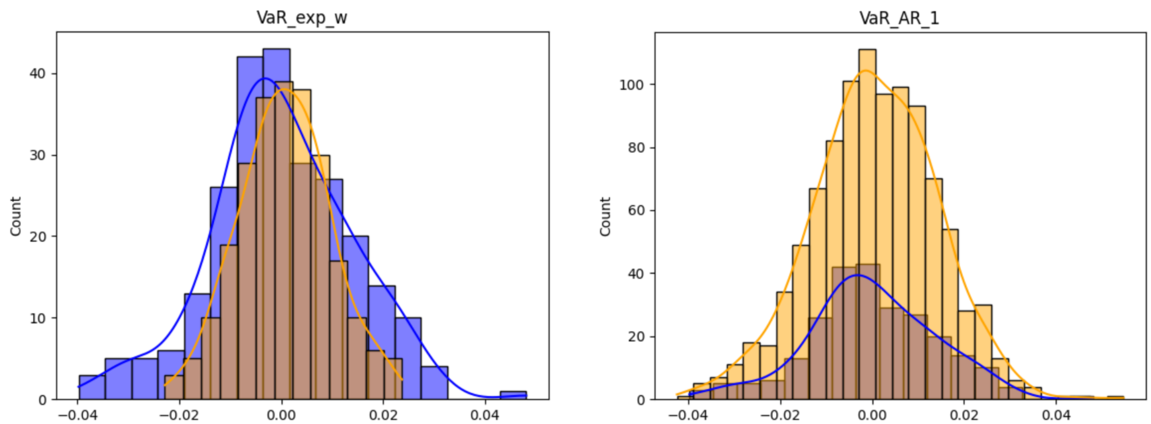
As the calculation above, the 5% VaR of the portfolios are between 1%-2%, which is significantly smaller than one single asset in problem2. This shows the advantage of allocating assets as portfolios. In this allocation, $VaR(A) > VaR(A+B+C)$, while $VaR(B), VaR(c) < VaR(A+B+C)$. We can also find that $VaR(A) + VaR(B) + VaR(C) > VaR(A+B+C)$. This “sub additive" property shows the value of diversification of assets.

2.Use log return and fitted AR(1) model

	Portfolio A	Portfolio B	Portfolio C	Portfolio A+B+C
VaR	2.618736%	2.169598%	2.028810%	2.246988%
$	7854.90	6386.98	5478.65	19422.48

This method get similar results for $VaR(A) + VaR(B) + VaR(C) > VaR(A + B + C)$, but all the VaR are larger than the results in method 1. Method 1 used exponentially weighted covariance with lambda = 0.94, which does not have very good fit for outliers.

In method two, I tried Log returns, because this will ensure the prices to be positive and is also commonly used. And I used a fitted AR(1) model, which fits the outliers better. I made similar graphs as problem 2. For the fitted AR(1) model, I still used a large size of fitted data to get a better simulation, which makes the count larger than the original ones.



From the graphs, the difference between Arithmetic Return and Log Return is evident. The Arithmetic Return is more like a normal distribution, with some outliers, and a slightly negative skewness, while the Log Return has lower peak, thicker tail, with negative skewness. The log return distributed more smooth from the center to the two ends. In this way, there are more values distributed near the two ends than the Arithmetic Return. Therefore, the VaR will be larger.

For AR(1) model, the trend line shows a good fit, which is a better choice when the returns have outliers, especially when using log returns, which are less likely to be normal distributed and has more outliers.