# Fintech 545 Project Week05

Queenie Liu
NetID: yl957

**Problem 1:**

For this problem, I used the following libraries:

```
import pandas as pd
import numpy as np
import statsmodels.api as sm #for fit-t
from scipy.stats import norm #for VaR and ES with normal
from scipy.stats import t #for VaR and ES with T
from scipy.optimize import minimize #for fit-t and VaR T
from scipy.integrate import quad #for ES simulation
```

And I organize the functions for calculating covariance and correlation with missing data, and both of them under equally-weighted, near_psd and higham conditions. The normal simulations, PCA simulation. As well as the ways to calculate returns, and how to sort the data into differnet kinds of distributions, the VaR and ES for different kinds of distribution.

To test the correction of my functions, I used all the provided databases and got the results of 'True' for all of them, which means the distances of calculated and expected returns are within 0.001 or even 0.00001 diffrence level. Some of my test outcomes are listed below:

```
Comparison result is:
 [[ True  True  True  True  True]
 [ True  True  True  True  True]
 [ True  True  True  True  True]
 [ True  True  True  True  True]
 [ True  True  True  True  True]]
```

```
Comparison result is:
 [[ True  True  True]]
```

But for the ninth test, the differnece is not that small to be considered, which is:

```
The expecetd outcome is:
    Stock       VaR95        ES95  VaR95_Pct  ES95_Pct
0       A   94.460376  118.289371   0.047230  0.059145
1       B  107.880427  151.218174   0.035960  0.050406
2   Total  152.565684  199.704532   0.030513  0.039941

The calculated result is:
    Stock       VaR95        ES95 VaR95_Pct  ES95_Pct
0       A   93.131154  115.757315  0.046566  0.057879
1       B  108.605932  153.976595  0.036202  0.051326
2   Total   152.25331  202.826173  0.030451  0.040565
```

The difference is not that huge to reject the correction of my models, and with this difference, we can tell that:

For stock A, the calculated VaR & ES are lower than expected. Suggesting that the simulation model predicts a marginally lower risk for stock A compared to the expected model.

For stock B, the calculated VaR & ES are higher than expected, suggests a higher risk profile in the simulation.

For the total portfolio, the calculated VaR is slightly lower than expected, while the ES is higher, indicating that while the overall threshold for extreme losses is similar, the average of the worst losses is higher than expected.

**Problem 2:**

Firstly, the differences between VaR and ES:

VaR is a threshold value such that the probability of loss exceeding this value is at the given confidence level, it's a single point estimate and doesn't show any information about the size of the loss beyond this point.

ES is the expected average loss assuming that the loss is beyond the VaR threshold. It gives an idea of the magnitude of loss in the worst α% of cases.

Thus, ES is always greater than or equal to VaR since it takes into account the average loss in the tail beyond the VaR threshold.

And if the ES is significantly higher than the VaR, it suggests that when losses are bad, they can be much worse than the VaR indicates.

For the data of problem1.csv, I got the results of VaR and ES using these three

| | Method | VaR | ES |
|---|---|---|---|
| 0 | Normal (EW) | 0.091805 | 0.114919 |
| 1 | MLE T-Dist | 0.076476 | 0.113218 |
| 2 | Historical | 0.075981 | 0.116777 |

different methods as below:

The difference between VaR and ES can be explained by the effects of different methods and their underlying distribution assumptions and calculation methods:

1. Normal (EW):

It assumes returns are normally distributed, and accounts for changing volatility by giving more weight to recent observations using exponential weighting.

VaR in this method is relatively higher, which may suggest that the exponential weighting is giving a higher weight to recent volatility.

ES is the average loss in the worst α% of cases, with a normal distribution, so that it should be sensitive to the mean and standard deviation.

Since a normal distribution underestimates the occurrence of extreme events, ES may be smaller compared to the real average of the extreme losses.

2. MLE T-Distribution:

This method fits the data to a T-distribution using MLE. This method typically has heavier tails than the normal distribution, which allows it to better model the occurrence of extreme events.

Both VaR and ES are lower than the normal method, which is because that the model capture tail risk more effectively.

3. Historical:

This method directly uses historical data to estimate VaR and ES.

VaR is based on historical percentiles, while ES is the average of the worst historical outcomes.

Because it is non-parametric, this method can capture the actual distribution of returns, including any skewness or kurtosis present.

Historical simulation tends to be more accurate in capturing tail risk if the historical data is representative of future risk, thus often resulting in a higher ES.

**Problem 3:**

For this problem, I used the library and the functions in the ninth test in problem 1, the results are in the below format:

| Portfolio | VaR95 | ES95 | VaR95_Pct | ES95_Pct |
|---|---|---|---|---|
| A | 8075.455034 | 10630.45588 | 0.026923 | 0.035441 |
| B | 6970.138687 | 9244.929368 | 0.023677 | 0.031404 |
| C | 5963.971886 | 7533.958682 | 0.02208 | 0.027899 |
| Total | 20572.272127 | 26697.552523 | 0.0238 | 0.030886 |

Comparing this week's problem and requirement with week 4 task, it's not hard to find that this week's requirement is more specificly focusing on the use of an exponentially weighted covariance matrix for VaR calculations, as opposed to more general methods that may use simple covariance or other forms of risk estimation.

In this task, I use the following functions to do the calculation:

1. Return Calculation(the 'return_calculate' function):

This function calculates the returns of stocks in different ways based on three passed method: Classical Brownian Motion (CBM), Arithmetic Return System (ARS), and Geometric Brownian Motion (GBM).

2. PCA Simulation (the 'simulatePCA' function):

This function performs Principal Component Analysis (PCA) on the Spearman's correlation matrix of the transformed returns. It's used to simulate potential future scenarios for the portfolio based on the historical data.

It calculates the eigenvectors and eigenvalues of the correlation matrix, uses them to generate multivariate normal distributions, and then scales them according to the square root of the eigenvalues.

3. Copula Simulation (the 'simulate_copula' function):

This function simulates returns for the portfolio using the PCA simulations.

It fits either normal or t distributions to the historical returns based on the 'Distribution' specified in the portfolio DataFrame. It then calculates the portfolio value and pnl (profit and loss) from these simulations.

VaR and ES are computed for each stock in the portfolio and also for the portfolio as a whole.

However, the task in last homework, I used the following steps:

1. Calculate returns: Use the historical returns for each stock in the portfolios.

2. Compute covariance matrix: Calculate the exponentially weighted covariance matrix with lambda value.

3. Apply certain portfolio values: Assess the current value of each portfolio by multiplying the holdings of each stock by its current price.

4. Calculate Portfolio VaR: Apply a portfolio risk model such as the RiskMetrics model to estimate the VaR for each portfolio and the total holdings, involving using the exponentially weighted covariance matrix along with the portfolio weights.

Therefore, the most significant difference is caused by the different levels of detail of the question requirements.