Question 1:

1. At very first after reading the file, I calculated arithmetic return by implementing pct\_change().dropna(). This function exactly matches the equation for arithmetic return we are taught in class: A black text on a white background

   AI-generated content may be incorrect.. Then, I simply remove the mean by subtract the mean value from the dataset. After these processes, just used .std() to calculate the standard deviation. Below is the last five rows of the arithemetic returns by using .tail() and the standard deviation with std SPY = 0.0081, AAPL = 0.01348, and EQIX = 0.01536

A white background with black numbers

AI-generated content may be incorrect.

The total standard deviation is adding all three stocks std value:



1. The process of this question is generally the same with last one except for calculation for log return. Based on the equation showed in class, it will be calculated as: A mathematical equation with black text

   AI-generated content may be incorrect., and that’s why this step is different from last one to be: np.log(data / data.shift(1)).dropna(). Everything else is same for this question to remove mean and calculate standard deviation. The result looks like this with std of SPY = 0.0081, AAPL = 0.01345, and EQIX = 0.01527: A white background with numbers

   AI-generated content may be incorrect.

Total standard deviation is adding all three stocks std value:

A close-up of numbers

AI-generated content may be incorrect.

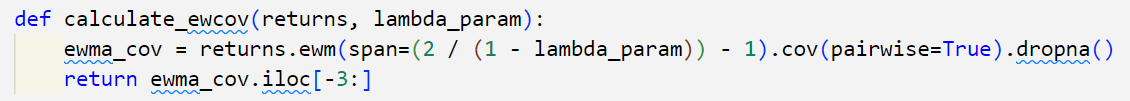
Question 2:

1. Given the date “today”, I simply converted the date from csv file into date form and lock the row with the date “01/03/2025”, then I multiply the value of stock on that date by the number of shares owned and added three of them together.



The final value of the current value of stocks is about $251862.497.

1. 1. In this method, I build the exponentially weighted covariance matrix in python first using the following code:



which stands for this formula

A number of numbers on a white background

AI-generated content may be incorrect.

Then I calculate the portfolio standard deviation by compute arithmetic returns using preprocessed data:

A computer code with text

AI-generated content may be incorrect.

Finally, the VaR is calculated based on the formula shown in the class slide: A black text with a white background

AI-generated content may be incorrect.A close-up of a text

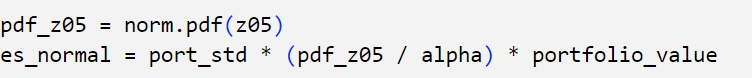
AI-generated content may be incorrect.

where z05 calculates the Z value of 95% confidence level and “\* portfolio\_value” gets the money value of value at risk.

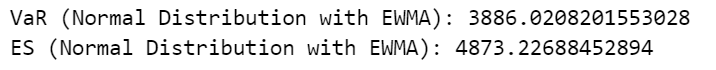
Expected shortfall using this formula:

A black and white image of a mathematical equation

AI-generated content may be incorrect.



The final answer is:



Explanation of result: Based on the Normal Distribution with EWMA covariance, the 5% Value at Risk (VaR) for the portfolio is about $3,886, meaning there's a 5% chance of losing more than this amount in a single day. The Expected Shortfall (ES) is about $4,873, which represents the average loss if the portfolio exceeds the VaR threshold.

* 1. Firstly, Compute arithmetic returns using pct\_changhe().

Following the instructions in class: A white background with black text

AI-generated content may be incorrect.

T-distribution is needed in the first place, so I fit a T-distribution to each stock's returns using

A computer code with black text

AI-generated content may be incorrect.

Then Transform data to uniform distribution using T CDF:



Then convert uniform to normal space using Gaussian copula



By using Z, spearman correlation matrix and Simulation of returns using Gaussian Copula are constructed

A screen shot of a computer code

AI-generated content may be incorrect.

Next step will be converting simulated copula returns back to T-distributed returns

A computer screen shot of text

AI-generated content may be incorrect.

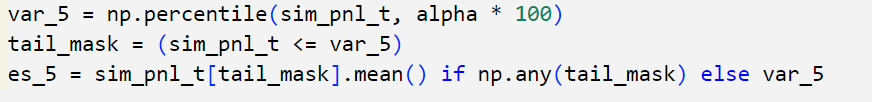
And compute simulated portfolio pnl

A screenshot of a computer code

AI-generated content may be incorrect.

By having the pnl value, I can now calculate VaR by simply computing the 5% percentile of the simulated portfolio profit and loss distribution and ES by formulas we have in class A black and white image of symbols

AI-generated content may be incorrect. which looks like this in codes:

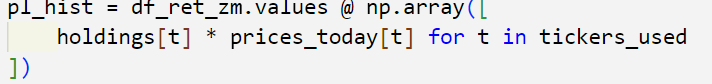


The final result is this:

A number with black text

AI-generated content may be incorrect.

Using the T-Distribution with a Gaussian Copula, we found that the 5% Value at Risk (VaR) is $4,390.50, meaning there’s a 5% chance the portfolio could lose more than this in a single day. The Expected Shortfall (ES) is $6,139.69, which represents the average loss in the worst 5% of cases.

* 1. For the Historic simulation using the full history, I computed historical PnL spread by this code: 

Then compute the historical VaR with 5% quantile 

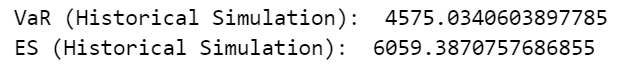
and historical ES using this equation with alpha to be 0.05:



Which is based on the basic expected shortfall equation from class slide: A close up of a number

AI-generated content may be incorrect.

The result is that:



The 5% VaR of $4,575.03 means that, in 5% of the historical scenarios, the portfolio lost *more* than $4,575.03 in one day. The 5% ES of $6,059.39 goes one step further, showing that on those worst 5% of days, the *average* loss was $6,059.39.

The fact that ES is higher than VaR confirms that extreme market drops tend to be worse than the typical risk threshold.

1. The normal distribution method with EWMA assumes that returns follow a normal distribution and estimates risk using an exponentially weighted moving average for volatility. This makes it responsive to recent market changes and computationally simple, but it tends to underestimate extreme losses because normal distributions don’t account for fat tails.

In contrast, the T-distribution with a Gaussian Copula provides a more realistic approach by allowing for heavier tails, meaning it captures extreme market moves better. This makes it useful for modeling financial crises or high-volatility periods, but it requires more complex parameter estimation and is sensitive to how well the T-distribution fits the data.

The historical simulation method is the most straightforward since it doesn’t assume any theoretical distribution. Instead, it directly uses past return data to estimate risk, making it a more empirical approach. However, it assumes that past market conditions will repeat in the future, which may not always be true, and it doesn’t adapt well to changing market conditions. While the normal method is overly optimistic, the T-distribution captures more realistic risk, and historical simulation reflects actual past events but may not be reliable for future uncertainty.

Question 3: (Declaration in advance, all the formulas used under this question are suitable for the situation are because the question states that it is European Call option)

1. Implied volatility is calculated by Black-Scholes formula for sigma located in d2 with this formula: A black text on a white background

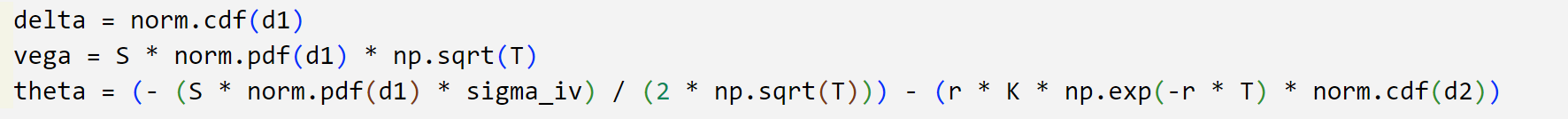
   AI-generated content may be incorrect. using root finder of python just similar to the example shown in Julia by professor. I used brentq package and put all the parameters in like this with sigma to be implied volatilityThe result is this
2. The calculation of Delta, Vega, and Theta are basically following the formula from class slide.
   1. Delta: A black text on a white background

      AI-generated content may be incorrect.
   2. Vega: A black text on a white background

      AI-generated content may be incorrect.
   3. A black text on a white background

      AI-generated content may be incorrect.Theta:

I used these formulas in python and generated the report of these values



The result shows that:

A number with numbers on it

AI-generated content may be incorrect.

The Delta value of 0.6659 means that for every $1 increase in the stock price, the call option price increases by approximately $0.666.

The Vega value of 5.6407 means that for a 1% increase in implied volatility, the call option price increases by approximately $5.64

The Theta value of -5.5446 means that the option loses approximately $5.54 in value per day due to time decay.  
Then, for the price changed when implied volatility increased by 1%, I used Vega method because Vega measures the sensitivity of an option’s price to changes in implied volatility. Therefore I calculated approximate 1% change by multiplying the Vega value by 0.01



For the proof, I add the sigma value by 1%, and used it for B-S model calculations and get the difference.

A screenshot of a computer

AI-generated content may be incorrect.

The result shows that the difference between the estimated change and actual change is very low with the value to be around 0.000091.

A white background with black text

AI-generated content may be incorrect.

1. For the put price using Generalized Black Scholes Merton with the formula below, A black text on a white background

   AI-generated content may be incorrect.

I defined the function in python and plug in the parameters

A close-up of a computer code

AI-generated content may be incorrect.

For whether Put Call Parity holds, I used the formula of Put Call Parity mentioned in class:

A black text with a white background

AI-generated content may be incorrect.

Left-hand side is calculated as follows in python with C\_market for C in formula:



Right-hand side is calculated as follows which Price of the Put Option is directly plugged into the defined function.



After calculating the difference, the result shows that Put Call Parity holds since lhs is equal to rhs.

A number on a white background

AI-generated content may be incorrect.

1. By implementing the functions used before, the general information of the portfolio is this:

A black and white text

AI-generated content may be incorrect.

1. Firstly, I transformed annual volatility into the 20-days volatility using simple math as follows



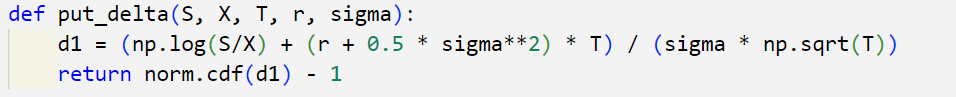
I also recalculated implied volatility as well in order to make the process clear. The reason for calculating this is that implied volatility is not directly observable in the market, so it must be estimated from the option’s market price.

The delta of the portfolio is added by call delta, put delta, and stock delta.

A close-up of letters

AI-generated content may be incorrect.

With Delta Call has been explained before, Delta Put’s formula implemented is A black text on a white background

AI-generated content may be incorrect.which shows in codes as this

By the way, since holding a stock means the value moves 1-for-1 with price changes, stock delta is 1 in the case.

Theta measures the rate at which the option loses value over time, assuming everything else remains constant which includes the option value decay in my calculation. With Theta Call explained in question before, I also want to explain the formula I used for Theta Put which is the formula below

A mathematical equation with black text

AI-generated content may be incorrect.

With function defined in python look like this

A computer code with black and white text

AI-generated content may be incorrect.

All these two functions **ignored the cost-of-carry** term because there is **no dividend** for this European option.

The portfolio theta is then calculated by adding theta call and theta put

A black and white symbol

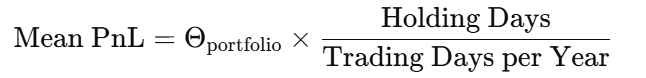
AI-generated content may be incorrect.

A close up of a number of letters

AI-generated content may be incorrect.

Then I calculated the mean PnL since we hold the portfolio for 20 trading days, we approximate the impact of Theta over the period which accounts for the time decay of options A black and white text

AI-generated content may be incorrect. using formula

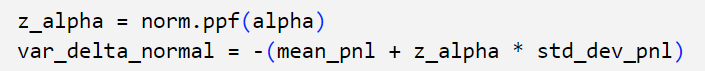
, and calculated standard deviation of PnL for the calculation for VaR by multiplying absolute value of delta by stock price and sigma in 20 trading days using formula A close up of a logo

AI-generated content may be incorrect.



Finally, I calculated the VaR with formula A black text on a white background

AI-generated content may be incorrect. in which z alpha is -1.645. In my code case, negative sign is moved to the front since z\_alpha returned by norm.ppf function is positive.



and ES A black text on a white background

AI-generated content may be incorrect. in which A black and white math equation

AI-generated content may be incorrect.



with the final report to be:

A black text with numbers and a white background

AI-generated content may be incorrect.

Portfolio Delta means that for every $1 increase in the stock price, the portfolio value increases by approximately $1.3319.

Portfolio Theta means that the portfolio loses approximately $8.16 per day due to time decay. As Theta is negative, this portfolio is likely holding long options.

Mean of 20-Day PnL indicates that the expected portfolio loss over the 20-day holding period is about $0.64.

The standard deviation of portfolio returns over 20 days is about $2.89.

Value at Risk at 95% confidence level is $5.3951 which means there is only a 5% probability that the portfolio will lose more than $5.40 over 20 days.

Expected Shortfall is $-5.3225 which measures the expected average loss in the worst 5% of cases.

e.

For the Monte Carlo Simulation, the number of simulated price paths is set to be 100000. Seed (42) ensures reproducibility of random results.

For the parameters below:

A black text on a white background

AI-generated content may be incorrect.

mu = 0 assumes the stock has a mean return of 0% per year which states in the question.

Rands generate normally distributed random returns over 20 days.

S\_20d simulated stock prices after 20 days using the lognormal model which looks like thisA black and white math equation

AI-generated content may be incorrect.

Then, I adjusted the time to expiration to be updated after 20 days which ensures that options are priced at the correct time to expire after 20 days and time decay is accounted for in the Monte Carlo process. 

I used B-S model for computing new call and put option values with formula to be

And

A black text on a white background

AI-generated content may be incorrect.

Which in code, it looks like this



Then I calculated the new portfolio value by adding all three prices





The PnL (profit and loss) is calculated as the change in portfolio value to be





Where portfolio\_0 is calculated by adding C + P + S

A close up of a number

AI-generated content may be incorrect.



Finally, I calculated the VaR using the formula

A black text on a white background

AI-generated content may be incorrect. and ES using the formula A math equations with numbers

AI-generated content may be incorrect.

Which shows in code as

A close up of a text

AI-generated content may be incorrect.

The result shows:

A math equation with numbers and symbols

AI-generated content may be incorrect.

In my report, it shows that the Monte Carlo simulation generated 100,000 different portfolio outcomes.

The average portfolio profit & loss (PnL) over 20 days is -0.2166 which suggests that the portfolio loses $0.2166 over the 20-day period on average.

Value at Risk at 95% confidence is around $4.13 and there is a 5% probability that the portfolio will lose more than $4.13 over the 20-day period.

Expected Shortfall is $4.58, meaning that in the worst 5% of cases, the average loss is $4.58.

1. Firstly, I generated the graph for portfolio value vs stock value and compare the assumptions.

A graph with a line and a red line

AI-generated content may be incorrect.

The graph obviously shows that there is difference between the Monte Carlo Simulation and Delta-Normal Approximation while stock value, portfolio value, and Monte Carlo has nearly overlayed each other with non-linear relationships while Delta Normal Approximation has a linear relationship. My key takeaway of the differences between the 2 methods with the graph is that Monte Carlo captures nonlinear option behavior while Delta-Normal assumes linear relationships, which can be inaccurate for options-heavy portfolios.

To wrap it up, Delta-Normal is the quick and easy method—it assumes that stock price changes follow a normal distribution and that your portfolio moves linearly with Delta. However, it totally ignores Gamma, which means it’s only accurate for small changes in stock price. On the other hand, Monte Carlo Simulation is like running a bunch of what-if scenarios, fully re-pricing the options using Black-Scholes. This method is way more accurate because it captures the nonlinear behavior of options (Gamma, Theta, and even Vega effects), but it takes way more time to compute. So, if we need a fast estimate, Delta-Normal is okay, but if we want realistic risk measurement, Monte Carlo is the way to go.