

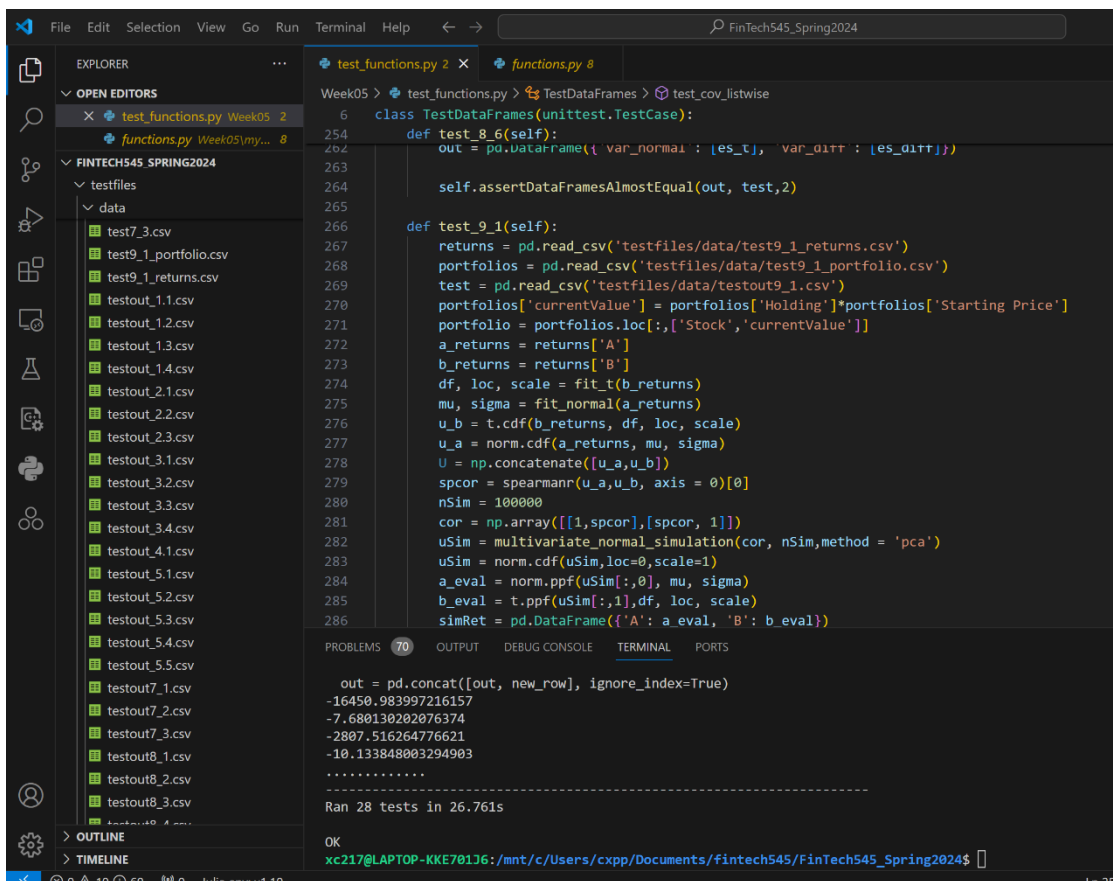
Q1

In your main repository, create a Library for risk management. Create modules, classes, packages, etc as you see fit. Include all the functionality we have discussed so far in class. Make sure it includes:

1. Covariance estimation techniques.
2. Non-PSD fixes for correlation matrices
3. Simulation Methods
4. VaR calculation methods (all discussed)
5. ES calculation

Please check the repo https://github.com/dompazz/FinTech545_Spring2024/tree/main/testfiles and make sure that all your functions can pass test files in the repo. Present your test cases pass results.

The Library and Test are provided in the file. The screen shot is as follows:



The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar displays the file explorer with a folder named 'FINTECH545_SPRING2024' containing a 'testfiles' subfolder and a 'data' subfolder. The 'testfiles' folder contains several CSV files, including 'test7_3.csv', 'test9_1_portfolio.csv', 'test9_1_returns.csv', and various 'testout' files. The main area shows a Jupyter Notebook with two tabs: 'test_functions.py 2' and 'functions.py 8'. The 'test_functions.py 2' tab is active, displaying a Python class 'TestDataFrames' that inherits from 'unittest.TestCase'. The class contains two test methods: 'test_8_6' and 'test_9_1'. The 'test_9_1' method performs a series of data manipulations, including reading CSV files, calculating returns, fitting a normal distribution, and simulating a multivariate normal distribution. The bottom of the notebook shows the output of the tests, indicating that 28 tests passed in 26.761 seconds. The terminal output shows the results of the tests, including the path to the test files and the execution time.

```
Week05 > test_functions.py > TestDataFrames > test_cov_listwise
6 class TestDataFrames(unittest.TestCase):
254     def test_8_6(self):
262         out = pd.DataFrame({'var_normal': [es_t], 'var_dirt': [es_dirt]})
263
264         self.assertDataFramesAlmostEqual(out, test, 2)
265
266     def test_9_1(self):
267         returns = pd.read_csv('testfiles/data/test9_1_returns.csv')
268         portfolios = pd.read_csv('testfiles/data/test9_1_portfolio.csv')
269         test = pd.read_csv('testfiles/data/testout9_1.csv')
270         portfolios['currentValue'] = portfolios['Holding']*portfolios['Starting Price']
271         portfolio = portfolios.loc[:, ['Stock', 'currentValue']]
272         a_returns = returns['A']
273         b_returns = returns['B']
274         df, loc, scale = fit_t(b_returns)
275         mu, sigma = fit_normal(a_returns)
276         u_b = t.cdf(b_returns, df, loc, scale)
277         u_a = norm.cdf(a_returns, mu, sigma)
278         u = np.concatenate([u_a, u_b])
279         spcor = spearmanr(u_a, u_b, axis = 0)[0]
280         nSim = 100000
281         cor = np.array([[1, spcor], [spcor, 1]])
282         uSim = multivariate_normal_simulation(cor, nSim, method = 'pca')
283         uSim = norm.cdf(uSim, loc=0, scale=1)
284         a_eval = norm.ppf(uSim[:,0], mu, sigma)
285         b_eval = t.ppf(uSim[:,1], df, loc, scale)
286         simRet = pd.DataFrame({'A': a_eval, 'B': b_eval})

PROBLEMS (70) OUTPUT DEBUG CONSOLE TERMINAL PORTS

out = pd.concat([out, new_row], ignore_index=True)
-16450.983997216157
-7.680130202076374
-2807.516264776621
-10.133848003294903
.....
Ran 28 tests in 26.761s

OK
xc217@LAPTOP-KKE701J6: /mnt/c/Users/cxpp/Documents/fintech545/FinTech545_Spring2024$
```

Q2

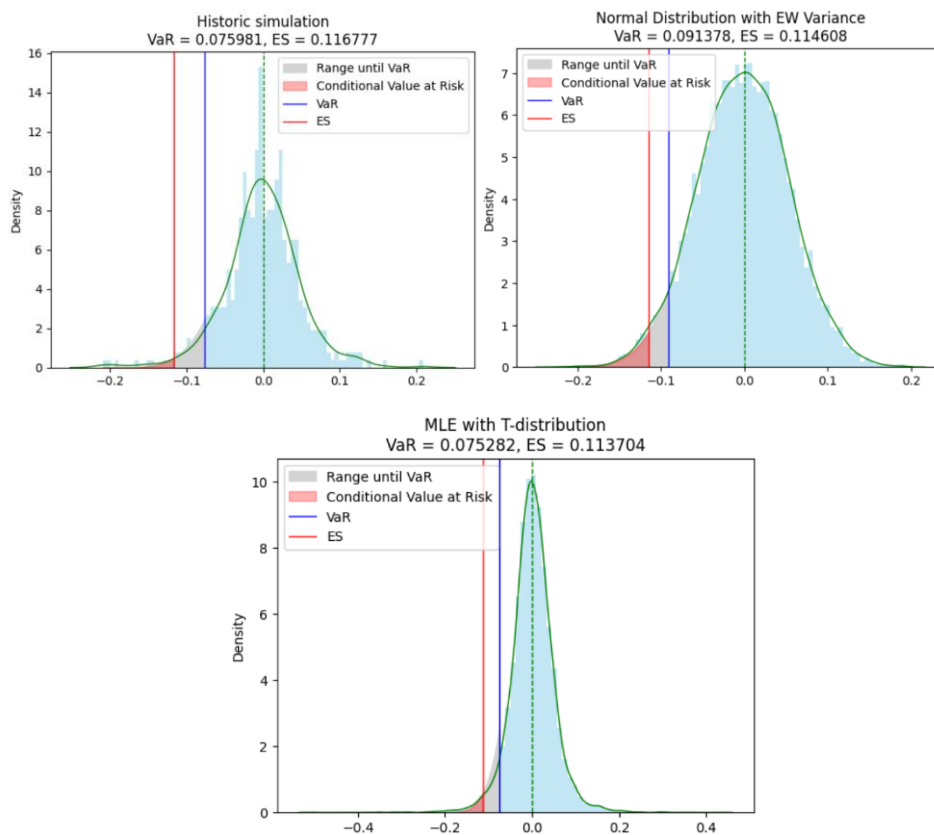
Problem 2:

Use the data in problem1.csv.

Calculate VaR and ES:

- Using a normal distribution with an exponentially weighted variance ($\lambda=0.97$);
- Using a MLE fitted T distribution
- Using a Historic Simulation

Compare difference between VaR and ES under different probabilistic distributions. Explain the differences.



Methods	VaR	ES
Normal + EW	0.092366	0.115236
MLE + t	0.077007	0.115599
Historical	0.075981	0.116777

Explain the differences:

For the VaR, the normal +EW method can get the biggest one, since EW method gives the latest data more weight in calculating, so maybe the recent data's volatility is larger. The MLE + t method uses maximum likelihood estimation to fit the data and assumes that returns follow a t-distribution, a distribution with thicker tails than the normal distribution, so is better able to capture extreme changes in financial asset returns, so this approach is generally considered more accurate than the normal distribution model. Using the t-distribution may result in higher risk estimates because it takes into account extreme events in the return distribution. The historical method is totally decided by the given data, and my result is that the historical data has the least VaR, and it is very similar to MLE + t method. This condition maybe because the data is more likely to be T distribution.

For ES, the results calculated by the three methods are very similar, which means that although EW method's VaR is larger, its left tail is shorter(thinner) than the other two method's tails. Based on the graph, we can see that the historical method's tail is the longest one.

Q3

Problem 3:

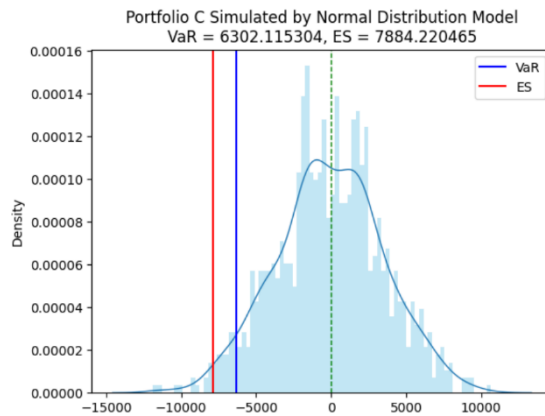
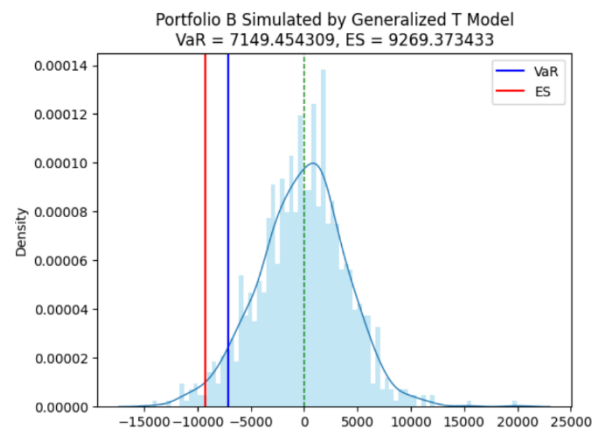
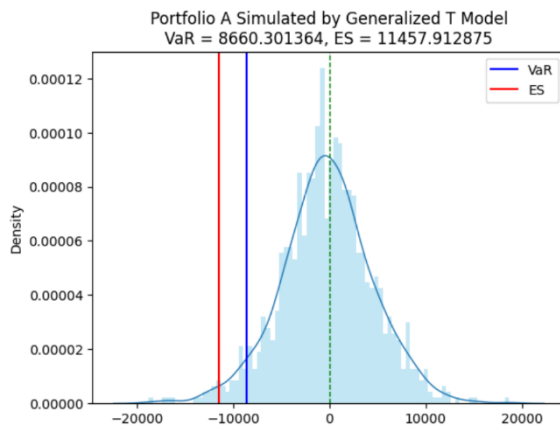
Use your repository from #1.

Using Portfolio.csv and DailyPrices.csv. Calculate arithmetic returns. Assume the expected return on all stocks is 0. This file contains the stock holdings of 3 portfolios. You own each of these portfolios.

Fit Generalized T models to stocks in portfolios A and B, and fit a normal distributions to stocks in portfolio C. Calculate the VaR and ES of each portfolio as well as your total VaR and ES. You will need to use a copula. Compare the results from this to your VaR from Problem 3 from Week 4.

My results are as follows:

	VaR	ES
A	8660.30	11457.91
B	7149.45	9269.37
C	6302.11	7884.22
Total	21889.52	29088.54



Compare VaR from problem 3 in Week4:

	VaR	Week4 Normal+ EW	Week4 Monte Carlo
A (Fit T)	8660.30	15426.97	14014.13
B (Fit T)	7149.45	8082.57	7474.11
C (Fit Normal)	6302.11	18163.29	16285.41

Total	21889.52	38941.38	35642.05
-------	----------	----------	----------

For A and C, the VaR calculated this week is smaller than the VaR in week 4, this is because the EW in week 4 includes more recent data, thus making the VaR larger since the recent data's volatility is larger. For B, I think the data is more like a distribution between t distribution and normal distribution, so the three method's results are similar.