FINTECH545; Spring 2024 Project #05
Brandon Kaplan; (03/08)

Problem 1. *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 fixed for correlation matrices*
3. *Simulation Methods*
4. *VaR calculation methods (all discussed)*
5. *ES calculation*

Please check the repo and make sure that all your function can pass test files in the repo. Present your test cases pass results.

Solution. For this problem I conducted every test listed in the course repo. I determined whether I passed or not based on computing the L2 norm between my output for each test and the provided test output files. For every single on of my tests except for Test 9 my L2 norm was less than 1. Many of my results were so small they would be considered zero and the error could be attributed to floating point arithmetic alone. I believed all my L2 norms were small enough to constitute passing. I was curious why some where 'practically' zero and others were still very small but multiple orders of magnitude larger. This was particularly interesting to me within certain test groups. For example, I wonder why my error on Test 8.2 was so much smaller than my error on every other test in Test 8. They were all small enough that I considered them a pass, but it was peculiar to me nonetheless. I assume there are the typical differences due to variance/uncertainty inherent to simulation and perhaps some due to the ways computation occurs in Julia vs. Python. Regardless, I am interested in understanding this more. I thought for my Test 9 that a distance between my output and the test output of ≈ 1.6 was acceptable because in Test 9 you have VaR and ES on two levels from simulations involving a copula. Because there are multiple levels of simulation here, I thought a greater error was acceptable. For each test, I tried to mimic the process in test-setup.jl. The pass results of my tests were as follows:

Differences between my outputs and provided outputs:
Test 1.1: 6.928477745064341e-16
Test 1.2: 4.865149434140654e-16
Test 1.3: 6.651392202765169e-16
Test 1.4: 4.496448759222397e-16
Test 2.1: 3.300171788787641e-16
Test 2.2: 3.8602957068981435e-16
Test 2.3: 4.1319933394177737e-16
Test 3.1: 3.0866478385056124e-15

```
Test 3.2: 1.9480519831873830e-15
Test 3.3: 3.1306813668187338e-15
Test 3.4: 3.5273971410977384e-15
Test 4: 4.364449066124153e-09
Test 5.1: 0.0009686812345492617
Test 5.2: 0.0007414932265371008
Test 5.3: 0.0005428242370464482
Test 5.4: 0.000607111814003481
Test 5.5: 0.0010802885120037437
Test 6.1: 9.261696690692286e-15
Test 6.2: 9.280887109479736e-15
Test 7.1: 0.00023448738921261203
Test 7.2: 5.128581352658074e-06
Test 7.3: 0.0032850469895859638
Test 8.1: 0.0005454585401847807
Test 8.2: 5.122653897487999e-08
Test 8.3: 0.0005267769054423446
Test 8.4: 0.000684027006313027
Test 8.5: 0.0016735032771393855
Test 8.6: 0.0038829589840866877
Test 9: 1.6106203636610492
```

□

Problem 2. *Use the data in problem1.csv. Calculate VaR and ES:*

1. *Using a normal distribution with exponentially weighted variance ($\lambda = 0.97$)*
2. *Using an MLE fitted T distribution*
3. *Using a Historic Simulation*

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

Solution.
For this problem, I tried to take a simpler, more consistent approach to calculating VaR (and then ES) then last week.

For the normal distribution with exponentially weighted variance ($\lambda = 0.97$) I first used my ewCovar() function to compute the exponentially weighted variance of the returns. Because our returns was only a single series (one column) when you calculate covariance, the desired variance is just the entry in the (0,0) position of the resultant covariance matrix (first row, first column). I then took this exponentially weighted variance and took the square root to get the exponentially weighted standard deviation. I calculated a standard normal z-score given our α (probability of VaR break), negated it, and scaled it by the exponentially weighted variance to find VaR. To find ES I used the closed form solution from the Week 05 notes.

For the MLE fitted Student's t-distribution I fit a t-distribution to the returns using scipy's built in stat package. I extracted the parameters—mean (loc), sigma(scale), and degrees of freedom (ν). With theses parameters I used the t-distribution's quantile function along with our α to find VaR. To find ES I simulated $10,000$ random samples from the t-distribution with my MLE fitted parameters. Then I found the the samples that were $\leq$ to VaR (the values whose CDF is $\leq$ to α). Lastly I took the average of these samples to find ES.

For the historic simulation I just used a Python adaptation of the VaR and ES functions from RiskStats.jl for a series of data. In this case the VaR is calculated by sorting the return data in ascending order. Then using α, find the indices corresponding to the lower and upper percentiles of the returns series. Then I compute VaR with the percentile method. Taking the average of the values at the lower and upper percentile indices. ES works similar but is computed by taking the average of the returns that are $\leq$ to the VaR of the historic returns.

For each VaR and ES calculation my results are based on absolute VaR and in terms of return—not dollars. My results were as follows:

Series VaR - Normal EW Variance: 0.09028951366738859
Series ES - Normal EW Variance: 0.11322669274257176

Series VaR - MLE Fitted T-dist: 0.07559610356471451
Series ES - MLE Fitted T-dist: 0.10903067192122626

Series VaR - Historic Simulation: 0.07498168167307463
Series ES - Historic Simulation: 0.114323207368112

We see that the Normal EW Variance VaR is the largest of all three methods. This might be explained by the fact that the Normal EW variance method is placing more emphasis on recent observations than MLE Fitted T-dist and Historic Simulation on account of the exponential smoothing. In addition to this, a visual examination of the data found a few outliers in more recent observations. So not only is the Normal EW Variance method placing more emphasis on recent observations, it is placing more emphasis on a few outliers as well. Plus, the normal distribution is generally less robust to outliers than the T-dist (with its fat tails) and the historic simulation which is non-parametric/makes no distributional assumptions on returns. The ES are all closer together which makes sense given they are averages. However, the MLE Fitted T-dist is clearly lower than the Normal EW Variance and Historic ES. The MLE Fitted T-dist has fatter tails than the Normal EW variance and also also less mass around the mean/peak. Basically we have a situation where the VaR break is pushed further out than in the case of the Normal EW Variance and less returns are falling outside our VaR threshold. The MLE Fitted T-dist seems like it handling the outliers a little better than the other two methods which might be fitting higher likelihood to more extreme losses. $\square$

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 form Problem 3 from Week 4.

Solution. For this problem I first computed arithmetic returns using my $return - calc$ function. Then I implemented the functions for fitting a general t-distribution and a normal distribution. I fight portfolios A and B using $fit - general - t$ and portfolio C using $fit - normal$. Then I constructed the copula. First I access the uniform distribution for each fitted model instance and transform it into a standard normal using the normal quantile function. Next, I calculated the Spearman correlation matrix of my transformed standard normals. Next, I used $simulate - PCA$ to draw from the multivariate normal, transform my standard normals back into uniforms using the standard normal CDF. Lastly transform the uniforms into the fitted distributions using $eval - func$ method for each fitted model instance. After this I do the portfolio valuation. First, I calculated VaR and ES for each stock and then I aggregated the total values as if treating all portfolios as one total and the portfolio values for each portfolio using the stock values and the information on each portfolio's holdings. Finally I calculated the VaR and ES for each individual portfolio as well as the total of all three portfolios. My results were as follows:

Portfolio A: VaR: \$8,099.33; ES: \$10,674.75
Portfolio B: VaR: \$6,784.33; ES: \$8,910.16
Portfolio C: VaR: \$5,844.37; ES: \$7,203.31
Total: Var: \$20,382.65; ES: \$26,375.38

Comparing these results to last week's VaR results in Problem 3 showed that the results were quite different. This is because Professor gave us two different portfolios to do each of these problems with. Since the portfolios were quite different, there is not much to be said about why the VaR's are different other than that they are evaluating risk of two different portfolios.

□