Advanced Risk Management

Computer Exercise Week 3

In this exercise, we compare risk measurement based on a fitted Student's $t(d)$ distribution to EVT-based risk measurement. We consider a simulated data-set $\{x_i\}_{i=1}^T$ of i.i.d. random variables, drawn from a standardized student's $t(d)$ distribution with $d = 4.5$. The x_i's may be interpreted as minus the standardized shocks; we will focus on the right tail of the distribution.

The Python script `ComputerExercise3.py` loads the relevant packages and creates a simulated, sorted Pandas series `x`. The data have been sorted in descending order, so from large to small. We take a large sample $T = 10000$, such that we still have a fairly large number of observations in the tail, i.e., $T_u = 0.05T = 500$.

After completing the code, you can run it repeatedly, and see whether your answers to the questions varies with every time you run it, or remains fairly constant. (A full Monte Carlo simulation exercise would involve extending the code to include a loop over different replications, and a method to asses how well different methods work on average.)

1. The script already has a simple implementation of the maximum likelihood estimator of the degrees-of-freedom parameter d. Extend the script to calculate the simple estimate of d based on the sample kurtosis, see Slide 10 of Week 3. Compare the outcomes; which estimate is closer to the true value of $d = 4.5$ that we have used to simulate the data?

2. We know that the t-distribution in this case should fit the data very well, because that is how we have simulated it. Let us check this by making a QQ plot of the data versus the $t(d)$ quantiles, with d equal to the ML estimate `d_ML`. You can do this using the `sm.qqplot` function, see the file `CodeWeek3.py` on Canvas. Alternatively, you could make the QQ plot yourself, by using the array of p_i values provided in the script, together with the `stats.t.ppf` function and `plt.scatter`. Run the program repeatedly; you should see that the majority of observations lie on the diagonal line as expected, but sometimes there are a few extreme observations quite far from this line.

3. Calculate the *VaR*^{p} based on the estimate t distribution for $p = 0.02$, $p = 0.01$ and $p = 0.005$ (making use of `stats.t.ppf`). Check how well these measures work for this data set, by calculating the percentage of exceedances.

4. It has been discussed in the theory exercises of this week that EVT implies that we expect a log-log relationship between `x[0:Tu-1]` and `p[0:Tu-1]`, where `p` is the array of p_i values. Investigate this, using the `plt.loglog` function. The slope should be comparable to $-\xi = -1/d$.

5. Apply the Hill estimator to `x[0:Tu-1]`, and compare the estimate $\hat{\xi}$ to $1/d$ and to $1/\hat{d}_{ML}$. Are they close to each other?

6. Recalculate the *VaR*^{p} based on the EVT estimate $\hat{\xi}$ for $p = 0.02$, $p = 0.01$ and $p = 0.005$, using the formula given on Slide 21. Again, calculate the percentage of exceedances, and compare with the theoretical probabilities.