Instructions for Computer Exercise (CE) 1

Purpose: The purpose is to practically implement and evaluate some of the various methods available to estimate Value-at-Risk. Feel free to copy and paste code from the course home page. See the file 'Instructions for Assignments' for details about hand-in and grading. All VaR and ES calculations are with alpha=99%.

Data: The data are 4 years of daily Profit and Loss (PL) numbers from the Bank La Caixa from 2005-2008. Numbers are in thousands of Euros, positive numbers are profits.[1] . We call the first 500 observations the estimation period and the rest of the data the test period.

The data are in the Excel file: **DataLab1.xlsx.**

1. Apply basic historical simulation to the losses in order to estimate VaR for each day in the test period. Use a rolling (moving) window of size 500. This means that for the first VaR estimate you use the first 500 loss observations, for the second VaR estimate you drop the very first loss observation but add observation 501 (keeping the window size fixed at 500).

2a. Construct unexpected losses by subtracting the sample mean (mu) estimated as the average of the first 500 loss observations (the unexpected losses are e_t in the EWMA formula.) Subtract this mean from all loss observations. For simplicity, keep the estimate of mu fixed. This means no parameter updating of mu.

b. Use the EWMA conditional variance approximate GARCH model to calculate a variance observation corresponding to each loss observation (both in the estimation and test period):

$\sigma_t^2 = (1-\lambda)e_{t-1}^2 + \lambda\sigma_{t-1}^2$ for $t = 1,2,\dots,1008$

Use the standard values $\lambda = 0.94$, $e_0^2 = 0$ and σ_0^2 = variance of first 500 loss observations.

c. Calculate the EWMA conditional standard deviations (the square root of the EWMA conditional variances).

3. Construct volatility scaled losses by using the standard deviations calculated in 2c. Recall that we need an estimate of σ_{T+1} to construct the scaled losses, i.e. an estimate of volatility for ''the next day''. This means that we have to reconstruct the whole series of scaled losses

[1] Banks typically don't disclose this information so we're using data extracted from the graphs in the annual reports from the paper Perignon, Christophe, Fresard, Laurent and Vilhelmsson, Anders. The Pernicious Effects of Contaminated Data in Risk Management. In *Journal of Banking & Finance*, 2569-2583., 2011.

for each new σ_{T+1}, so we have to create as many new rescaled loss series as we have days in the test period. That is your matrix of scaled losses should by 500x508.

Hint: The code will be something like

```
for j in range(rT,T):
    scaled_losses[:,j-rT] = # your code goes here
```

rt is the size of the rolling sample and T the total sample size.

4. Apply the basic historical simulation to the volatility scaled losses to estimate VaR and ES for each day in the test period.

Plot the Losses (negative PL), the basic historical simulation and the volatility weighted VaR estimate for each day in the test period.

5a. We now move on to parametric approaches. Test if the Loss data is normally distributed. Also compute the first four moments. How do they look compared the SP500 returns? Hint: SP500 (daily) returns are typically, leptokurtic and left skewed with a mean very close to zero.

5 b. Estimate μ, σ and ν (degrees of freedom) for each day in the test period by using a rolling window of the 500 previous observations assuming a student t-distribution. Also estimate μ, σ for each day in the test period by using a rolling window of the 500 previous observations and assume a normal distribution.

5c. Estimate VaR and ES under the normal distribution and under the t-distribution for each day in the test period using the estimates of μ, σ and ν. For those days (if any) you get a value of ν<2, set $\nu = 2.1$.

6. Estimate VaR and ES using the POT method on a rolling sample of 500 observations, use the 5% largest values as the threshold. Plot your VaR estimate over time.

Helpful literature: Video Lectures 1 and 2 as well as Campus lecture 2 with the associated Python code.