Financial Econometrics 2022/2023
Exercise set 3

1 Notes

- Due date: December 19^{th}, 2022, 23:59 (CET).
- Send your solutions through Microsoft Teams. Start a one-to-one private chat with the instructor and attach a *single* .txt file, name it as `GX_E3_ECXXX_ECXXX_ECXXX`, where `GX` is the group number, `ECXXXX` is the student number (there are up to four `ECXXX` strings depending on how many students are working together). Use only the student numbers of those who, within your group, worked on the exercises.
- The document is supposed to contain, answers, codes, and possible comments, for each question. Make a .txt file with the codes and comments that runs smoothly in the R console (if I copy-paste the *whole* content in R there are no issues and errors), see the file *Example.txt* and copy-paste its content in R.
- Additionally write your student number(s) on the top of the document. You can work in groups of 2-3 students or individually.
- If you would like to discuss the exercises and get some feedback before the due date contact me for scheduling a meeting over Teams.
- In financial econometrics by "returns" we mean "log-returns", unless differently specified.
- Last updated on 2022-11-26 at 08:04:56 (UT).

2 Exercises

Total points: 1.3.

1. Simulate data and fit a MLR model
   (i) Generate some data under a multiple linear regression model where (a) the independent variables are $(1, 1.1, 1.2, \ldots, 9.9, 10)$ and its logarithm, (b) errors are normally distributed with mean zero and variance $\sigma^2 = 0.5^2$, (c) the coefficient vector is $(2, 0.5)^\top$ and there is no intercept. (ii) Fit the MLR model. (iv) From the model's summary extract the residuals and manually compute their standard error. How does it compare with σ and why?
   Hint: (i) create the fist independent variable x_1 (which has n elements), then $x_2 = \log x_1$, then generate n errors from the above normal distribution, lastly combine x_1, x_2, the errors and the coefficients to obtain the corresponding n values of the dependent variable. (ii) In `lm()` you need to specify that the model does not have an intercept.
   Points: 0.1.

2. OLS estimates.
   (i) Simulate some data under the model $y = 1 + 2x_1 + 0.5x_2 + \varepsilon$, with $x_1 = (1, 1.1, 1.2, \ldots, 9.9, 10)$, $x_2 = \log(x_1)$ and $\varepsilon \sim N(\mu = 0, \sigma = 0.5)$. Have a look at the OLS estimates $\hat{\beta}$ that `summary(lm())` displays and extract the variance-covariance matrix of the estimated parameters with the function `vcov()`. (ii) Manually compute the OLS estimates by implementing $\hat{\beta} = (X'X)^{-1}X'y$. (iii) Estimate as well β's variance-covariance matrix $\hat{V} = \hat{\sigma}^2(X'X)^{-1}$.
   Hints: (i) This is all about specifying the correct design matrix X and implementing the definition of $\hat{\beta}$ correctly. Search on google how to do matrix multiplication, transposition and inversion. (ii) The exercise is solved correctly if your manual estimates match those that `R`'s functions return.
   Points: 0.2.

3. A problematic model.
   Generate the dependent variable under model $y = x_1 + x_2 + \varepsilon$, with $x_1 = X/10$ where X are 1000 draws from a uniform distribution between 0 and 1. Let x_2 be $\log(x_1 + 1)$ and $\varepsilon \sim N(\mu = 0, \sigma = 1)$. (ii) Fit the linear model, display its summary and the estimated variance covariance matrix $\hat{V}$. (iii) The values in $\hat{V}$ are not reasonable: considering how $\hat{V}$ is computed where should the problem be? Explain why and how the problem arises. Does this issue have a name?. (iv) Do you expect the estimates to be biased or not?
   Hints: (i) To identify the problem try running the code with $x_1 = X/100$, $x_1 = X/1000$, $x_1 = X/10000$..., and look at the respective design matrices $\mathbf{X} = [x_1, x_2]$. (i) For an explanation, have a look at the slides in Chapter 1 dealing with the relationship between returns and logreturns.
   Points: 0.2.

4. For loop.
   One of the most important and useful operations that you might use in R as well as in any programming language are the so called "for loops". Read some online documentation, tutorial, examples on how for loops work in R. It may seem hard at the beginning but you will quickly see how these turn useful many circumstances.

   By using a for loop, construct the values $u_i = -0.9u_{i-1} + \varepsilon_i$ with $\varepsilon \sim N(\mu = 0, \sigma = 1)$ and starting value $u_1 = 0.1$ and $i = 2, 3, \ldots., n = 200$.
   Hint: (i) the loop goes from 2 to n and implements the code `u[i] <- u[i-1]*-0.9+rnorm(1)`. (ii)

You need to pre-initialize an empty matrix `u` (e.g. a matrix filled with missing values except at the first element) to be filled up as the loop iterates across i. The matrix need to be of appropriate size (e.g. $n \times 1$). (ii) Plot the autocorrelation function of u and provide an *explanation* for what you see (not a description).
Points: 0.3.

5. Variance of the OLS estimates.
   This exercise is about generating Ns values of $\hat{\beta}$ (and $\hat{\sigma}$) and checking whether the variance of its elements is actually aligned with the one that the theory predicts, i.e. with $\sigma^2(X'X)^{-1}_{k,k}$, where $k = 1, 2$.

   Generate n draws of the ys from the model $y = 2x_1 + 0.5x_2 + \varepsilon$, with $x_1 = X \times 10$, $x_2 = \log(x_1)$, $\varepsilon \sim N(\mu = 0, \sigma = 2)$ and X are $n = 500$ draws from a uniform distribution on the interval $[0, 1]$. (i) Estimate the above model $Ns = 5000$ times by keeping the independent variables always fixed, re-drawing the εs each time, storing in a $Ns \times 2$ matrix **b** the estimated coefficients and in a $Ns \times 1$ matrix `s` the estimate of residuals' variance ($\hat{\sigma}^2$ aka s^2) at each draw. (ii) Compute the sample covariance of the Ns elements in `b`, compare it with the *theoretical* variance-covariance matrix of the estimates under the above MLR.

   Hint: (i) This is done with a for loop that envelopes a code where at each step i the data y is generated, the model `y=x1+x2` fitted, the relevant quantities `fit$coefficients[k,i]` extracted and stored in `b[i,1]` `b[i,2]` and `s[i]`. (ii) Before the loop generate the dependent variables to be reused at every iteration, so they remain unchanged.
   Points: 0.4.

6. Plot about the finite sample distribution of $\hat{\sigma}^2$.
   By using the vector `s` from the above exercise, create a new vector `Ratio = s/sig`^{2}`*(n-2)`, `sig` $= \sigma = 2$ as above. (i) What do the theoretical results on the MLR say about the finite sample distribution R of `Ratio`? Make a plot of the histogram of `Ratio` and overprint on it the pdf of R.
   Hint: (i) the pdf of R and the histogram need to be one the same "scale": what does the argument `prob` of `hist` do?
   Points: 0.1.