Financial Econometrics 2023/2022
Exercise set 3

1 Notes

- Due date: December 13^{th}, 2023, 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_E1`, where `X` is the group number (e.g., `G4_E1.txt` is the name for group 4).
- Fill up the `Groups_and_grades` Excel file in teams to make up the groups.
- If you work in a group, please designate *one* person to send the solutions and open only *one* chat for submitting the assignment. Please *avoid* changing the groups in the following exercise sets, and please *reuse* the same chat for the following submissions.
- 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. This means that if I copy-paste the *whole* content from your .txt file in the R console, *the code should run all the way with no errors*, see the file *Example.txt* and copy-paste its content in R. Makes sure your code runs.
- Additionally, write your student names, surnames and number(s) on the top of the document as a comment. You can work in groups of 2-5 students or individually.
- If you would like to discuss the exercises and get some feedback before the due date you can come to office 2.23 during regular office hours (please book), or ask to arrange a different time.
- Avoid sending lines and lines of code through Teams: rely on office hours to get help.
  - As the deadline approaches, I get more and more requests for help: then I end up being fully booked and don't have time for all. Plan the work well ahead!
  - Be considerate, do not expect replies on the weekend/late at night. Note that the deadline is on Monday: you don't wanna work on the assignment the day before, as I won't be around to help.
- In the tasks, name the variables exactly as indicated! so that when, e.g., I see `r` I have a reference and understand what you are trying to do. Write your code in a way that is understandable: e.g., you would name a variable that contains prices (returns) as `p` (`r`) and not `h` (`u`).
- Last updated on 2023-11-20 at 16:34:08 (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 population coefficient $\beta = (2, 0.5)^\top$ 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?
   (v) Now add an intercept to the *estimated* model, comment on the standard error & p-value of the estimated intercept Hint: (i) create the first independent variable x_1 (which has n elements), then $x_2 = \log x_1$, then generate n errors from the above normal distribution, and 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: 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 $\mathbf{X} = [x_1, x_2]$.
   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 $\varepsilon \sim N(\mu = 0, \sigma = 1)$, starting value $u_1 = 0.1$ and $i = 2, 3, ..., n = 200$.
   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 (check `?matrix`)

except at the first element, `u[1]` $= u_1$) 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: 0.2.

5. Variance of the OLS estimates.
This exercise is about generating N 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 y's from the model $y = 2x_1 + 0.5x_2 + \varepsilon$, with $x_1 = X \times 10$, $x_2 = \log(x_1)$, $\varepsilon \sim \mathcal{N}(\mu = 0, \sigma = 2)$ and X are 500 draws from a uniform distribution on the interval $[0, 1]$. (i) Estimate the above model $N = 5000$ times by,

- keeping the independent variables always fixed
- re-drawing the ε's each time
- storing in a $N \times 2$ matrix `B` the estimated coefficients
- storing in a $N \times 1$ vector `s` the estimates of residuals' *variance* $\hat{\sigma}^2$

(ii) Compute the sample covariance of the N elements in B, and 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/iteration i the data y is re-drawn (by re-drawing the ε's), the model `y=x1+x2` fitted, the relevant quantities extracted (`fittedmodel$...`) 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.

Define X
Compute x_1, x_2

loop $i = 1, \dots, N$
- generate y_i ($y_i = 2x_1 + \frac{1}{2}x_2 + \varepsilon_i$)
- fit the model
- extract $\hat{\beta}_i$ and $\hat{\sigma}_i^2$
- Assign $\hat{\beta}_i$ to the i-th row of B
- Assign $\hat{\sigma}_i^2$ to the i-th row of s

Compute the cov matrix of B
Compare this with its theoretical counterpart

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` $= \sigma^2 = 3$. Check on the slides what do the theoretical results on the MLR say about the finite sample distribution of `Ratio`? Make two separate plots:

1 Plot the histogram of `Ratio` and overprint on it its theoretical distribution.

2 Plot the histogram of `s` and overprint on it the pdf of its theoretical distribution.

Hint: (i) Both for point 1 and 2, first generate the histogram (`?hist`)! (ii) The theoretical pdf and the histogram need to be one the same "scale": what does the argument `prob` of `hist` do? In hist, you need to set the argument `prob=TRUE`.

- Task 1.
  (i) what is the theoretical pdf of `Ratio`? Check the slides on the finite-sample distribution of the OLS estimator.
  (ii) To plot the theoretical distribution, you need to initialize a sequence of x-points (use: `xp=seq(350,750,1)`) and evaluate the density (d'pdf') of those points, i.e. `yp` = dpdf(`xp`), and then go with `lines(xp,yp)`. `lines` prints on the current plot (the histogram).
- Task 2.
  (i) `Ratio` $= \hat{\sigma}^2 c \sim \chi^2_{n-2}$, therefore $\hat{\sigma}^2 \sim \frac{1}{c}\chi^2_{n-2}$. Here $c = 3/498$ and $n - 2 = 498$. .
  (ii) Note that $a\chi^2_\nu \sim \Gamma(\kappa = \nu/2, \theta = 2k)$, where $k > 0$ is a constant Γ denotes the pdf of a Gamma distribution with shape parameter κ and scale parameter θ, and ν is the degrees of freedom parameter of the chi-squared distribution. This is a general result: a chi-squared distribution times a constant is a gamma distribution.
  (iii) Based on point (ii), the distribution of $\hat{\sigma}^2$ is a Gamma... you have to reason across points (i) and (ii) and understand what are the numbers for κ and θ.
  (iv) To plot this Gamma, first use the command `yp = dgamma(xp,scale=kappa, shape=theta)` and again `lines(xp,yp)`, here `xp=seq(1,6,0.01)`).

Points: 0.2.