

# Problem Set 7

ECON 6343: Econometrics III

Prof. Tyler Ransom  
University of Oklahoma

Directions: Answer all questions. Each student must turn in their own copy, but you may work in groups. You are encouraged to use any and all Artificial Intelligence resources available to you to complete this problem set. Clearly label all answers. Show all of your code. Turn in jl-file(s), output files and writeup via GitHub. Your writeup may simply consist of comments in jl-file(s). If applicable, put the names of all group members at the top of your writeup or jl-file.

You may need to install and load the following package:

SMM

You will need to load the following previously installed packages:

Optim

HTTP

GLM

LinearAlgebra

Random

Statistics

DataFrames

DataFramesMeta

CSV

In this problem set, we will practice estimating models by Generalized Method of Moments (GMM) and Simulated Method of Moments (SMM).

1. Estimate the linear regression model from Question 2 of Problem Set 2 by GMM. Write down the moment function as in slide #8 of the Lecture 9 slide deck and use `Optim` for estimation. Use the  $N \times N$  Identity matrix as your weighting matrix. Check your answer using the closed-form matrix formula for the OLS estimator.
2. Estimate the multinomial logit model from Question 5 of Problem Set 2 by the following means:
  - (a) Maximum likelihood (i.e. re-run your code [or mine] from Question 5 of Problem Set 2)
  - (b) GMM with the MLE estimates as starting values. Your  $g$  object should be a vector of dimension  $N \times J$  where  $N$  is the number of rows of the  $X$  matrix and  $J$  is the dimension of the choice set. Each element,  $g$  should equal  $d - P$ , where  $d$  and  $P$  are “stacked” vectors of dimension  $N \times J$
  - (c) GMM with random starting values

Compare your estimates from part (b) and (c). Is the objective function globally concave?

3. Simulate a data set from a multinomial logit model, and then estimate its parameter values and verify that the estimates are close to the parameter values you set. That is, for a given sample size  $N$ , choice set dimension  $J$  and parameter vector  $\beta$ , write a function that outputs data  $X$  and  $Y$ . I will let you choose  $N$ ,  $J$ ,  $\beta$  and the number of covariates in  $X$  ( $K$ ), but  $J$  should be larger than 2 and  $K$  should be larger than 1. If you haven’t done this before, you may want to follow these steps:
  - (a) Generate  $X$  using a random number generator—`rand()` or `randn()`.
  - (b) Set values for  $\beta$  such that conformability with  $X$  and  $J$  is satisfied
  - (c) Generate the  $N \times J$  matrix of choice probabilities  $P$
  - (d) Draw the preference shocks  $\varepsilon$  as a  $N \times 1$  vector of  $U[0, 1]$  random numbers
  - (e) Generate  $Y$  as follows:
    - Initialize  $Y$  as an  $N \times 1$  vector of 0s
    - Update  $Y_i = \sum_{j=1}^J 1 \left[ \left\{ \sum_{k=j}^J P_{ik} \right\} > \varepsilon_i \right]$
  - (f) An alternative way to generate choices would be to draw a  $N \times J$  matrix of  $\varepsilon$ ’s from a T1EV distribution. This distribution is already defined in the `Distributions` package. Then  $Y_i = \arg \max_j X_i \beta_j + \varepsilon_{ij}$ . I’ll show you an example of how to do that in the solutions code for this problem set.
4. Use `SMM.jl` to run the example code on slide #21 of the Lecture 9 slide deck.
5. Re-estimate multinomial logit model from Question 2 using SMM. It will be helpful to use the code example from slide #18 of the Lecture 9 slide deck. You will also want to make use of your code from Question 3 to do this. (You can’t do this question without using the code from Question 3.)

6. As with previous problem sets, please wrap all of your code in a function so that you don't evaluate things in the global scope.
7. As with previous problem sets, have an AI write unit tests for each of the functions you've created (or components of each) and run them to verify that they work as expected. Best practice is to provide unit tests in a separate script that first reads in the source code before running the tests. Thus, I would like you to turn in three files:
  - PS7\_LastName\_source.jl
  - PS7\_LastName\_script.jl
  - PS7\_LastName\_tests.jl