Coding Exercise 1: Bertrand with Logit Demand

The purpose of this coding exercise is to build experience converting an economic model into practical code. You should complete it individually or with at most two other students. But it is simple enough that really you should probably do it on your own. Build skills!

I have supplied R code that *almost* generates the merger simulation results shown in class. The code consists of two files: simcode.R is the main file which which the user interacts. It calls the second file, simcode_functions.R, which defines a number of user-written functions. This is the way practical R code is structured: write simple functions, check them for accuracy, and keep them in the background. To run the code, you need to (1) download R, (2) download a GUI like RStudio, (3) install the nleqslv package, and (4) edit the path on line 9 of simcode2.R.

You have the option of translating my R code into Python or Julia, and proceeding that way. If you do so, then please send me your code so I can share it with the class.

Please do the following:

1. The code only *almost* generates the results from class because I have introduced a small number of bugs into simcode_functions.R. Fix the bugs and replicate the results in class. Report the line number at which each bug is located, and briefly describe its nature.

2. Compute equilibrium under the new assumption that the merger decreases the marginal costs of the merging firms by ten percent. Report the new post-merger prices.

3. Compute equilibrium under the new assumption that firms no longer maximize their own profit but instead collude to maximize joint profit. Report the new post-merger prices.

4. With a fully identified structural model in hand, you are *The Master of the Universe*. Change something and see how market outcomes shift.

The deliverable is a short PDF writeup that you can turn in via Canvas.

Notes on Calibrating the Bertrand/Logit

Suppose you have data on:

- Prices $p = (p_1, p_2, \ldots)$
- Market shares $s = (s_1, s_2, \ldots)$
- The first products' marginal cost, c_1

The objective is to calibrate a Bertrand/logit model with single-product firms. As specified in the model, demand is given by:

$$s_j = \frac{\exp(\alpha p_j + \xi_j)}{1 + \sum_k exp(\alpha p_k + \xi_k)} \tag{1}$$

and the supply-side given by:

$$p_j = mc_j - \frac{1}{\frac{\partial s_j}{\partial p_j}} s_j \tag{2}$$

which given our specification of logit demand simplifies to:

$$p_j = mc_j - \frac{1}{\alpha} \frac{1}{1 - s_j} \tag{3}$$

for price parameter $\alpha < 0$ (but you should verify). The structural parameters to be recovered are:

- Qualities $\xi = (\xi_1, \xi_2, \ldots)$
- Marginal costs $mc = (mc_1, mc_2, \ldots)$ of which the first element is known
- The price parameter α

How can this be done? Rearranging equation (3) for product 1 obtains:

$$\alpha = -\frac{1}{1 - s_1} \frac{1}{p_1 - mc_1}$$

and the RHS of this is observed. This obtains α. Next, write for each product

$$\ln(s_j) - \ln(s_0) - \alpha p_j = \xi_j$$

and as we know α, the LHS is known, so this obtains ξ_j for all products. Third, returning to equation (3), marginal costs can be inferred for all products because α is known and prices/shares are data.