# Coding Exercise 4: Single Agent Dynamics[1]

Professor Nathan Miller

In this coding exercise, students estimate a model of single-agent dynamics using methods from Rust (1987), Hotz and Miller (1993), and Aguirregabiria and Mira (2002). I have supplied `R` code comprising dynamics_starter.R, the main file with which the user interacts, and dynamics_functions.R, which defines a number of user-written functions. The first file calls the second.

**Model:**

The model is motivated by Rust (1987). A fleet of buses accumulate mileage over time. Harold Zurcher (the agent) chooses when to replace their engines, which resets accumulated mileage to zero. Thus, in each period $t = 1, \ldots \overline{T}$, with $\overline{T} = \infty$, the agent can choose to replace the engine (choice 1) or not replace the engine (choice 2). The flow utility that the agent receives is $u_{jt} + \epsilon_{jt}$ for subscript $j$ denoting the discrete choice. The first term is given by

$$u_{jt} = \begin{cases} 0 & \text{if replace the engine} \\ \theta_1 + \theta_2 x_t & \text{otherwise} \end{cases} \tag{1}$$

where $\theta_1$ and $\theta_2$ are structural parameters and $x$ is the mileage of the bus. Let $\theta_2 < 0$ so that the relative flow utility of replacing the engine tends to increase with mileage. Each $\epsilon_{jt}$ is an iid draw from a Type I EV distribution.

Let $f_j(x_{t+1}|x_t)$ provide the PDF for mileage in period $t+1$ conditional on choice $j$ and the mileage in period $t$. The idea that replacing the engine resets mileage to zero can be formalized as $f_1(x_1|x_t) = 0$, which incorporates the subtlety that replacement essentially "resets the clock" so that $t+1$ can be replaced with 1. Otherwise, mileage accumulation follows a discrete analog of the exponential distribution up to some $\overline{x}$ where the mileage increment is $\Delta_x$:

$$f_2(x_{t+1}|x_t) = \begin{cases} \exp(-(x_{t+1} - x_t)) - \exp(-(x_{t+1} + \Delta_x - x_t)) & \text{if } x_t \leq x_{t+1} \leq \overline{x} \\ \exp(-(\overline{x} - x_t)) & \text{if } x_{t+1} > \overline{x} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The last line incorporates that mileage cannot decrease, i.e., the probability of realizing and $x_{t+1} < x_t$ is zero if maintenance does not occur. The code

[1] This exercise builds on notes and code provided by Aaron Barkley (Melbourne). Many thanks to Georgetown ECON 631 alumni Ryan Mansley, Tianshi Mu, and Gretchen Sileo for helping me get up to speed with modeling and coding dynamics.

discretizes mileage and operationalizes this accumulation function.

The optimal decision rule maximizes the present value of flow payoffs under the assumption that the optimal decision rule will be employed in the future. The agent's discount factor is $\beta$.

**Code:**

The code in dynamics_starter does the following:

- Discretizes mileage into 301 states with $\Delta_x = 0.05$
- Constructs mileage transition matrices that apply if the agent repairs the engine (F1) and does not repair the engine (F2)
- Parameterizes the model: $\beta = 0.9, \theta_1 = 2, \theta_2 = -0.15$.
- Simulates fake data on $N = 2000$ buses each observed over $T = 30$ periods.
- Plots the data for the first bus
- Estimates the parameters using full information maximum likelihood (FIML). This involves value function iteration and follows Rust (1987).
- Estimates the parameters using estimated conditional choice probabilities, following Hotz and Miller (1993)

Especially to simulate the data and estimate the model, the key code is in dynamics_functions. All of the code is annotated, so may hope is that it should be transparent, but let me know if there are difficult spots.

**Tasks for the Student:**

Please do the following:

1. Familiarize yourself with the code. Confirm that both estimators obtain parameters that are similar to the ones used to create the data.
2. The function valuemap obtains the conditional value functions using the contraction mapping of Rust (1987). Explain how the function does this. Provide an intuition for why this is a contraction mapping.[2]

[2]For example, an intuition for the BLP (1995) contraction mapping might be as follows. If an initial mean valuation is too small, then the observed market share exceeds the predicted market share, and therefore we should adjust the mean valuation upwards.

3. Estimate the model using FIML with (incorrect) discount factors of 0.80, 0.85, and 0.95. Compare the values of the log-likelihood function evaluated at the estimated parameters. What does this imply for the identification of the discount factor?

4. Aguirregabiria and Mira (2002) suggest that the Hotz and Miller (1993) estimation routine can be made more precise using *policy function iteration*. Adjust the f_CCP_EST function to incorporate a user-specified number of policy function iterations. See the lecture notes for how this can be accomplished.