

Semester Project - Week 3 Report

Alexia Paratte

March 27, 2024

Goals

Last week, I did not get the expected results, as there was an error in the code translation from Apollo to Biogeme. I did not add the fact that we had to use *Panel data*: it is multi-dimensional data involving measurements over time. We make multiple observations for the same individual over time. We also could not estimate $\mu_{\log(\beta_{ch})}$, but this was due to a typing error. This week, we thus want to:

- Correct the `MMNL_Preference_Model_1` in Biogeme and re run it to get the correct results (ie the same results that we got in Apollo)
- Finalize the CPU/RAM `monitor_sytstem()`: I could not make it work properly
- Try to see how the parameters `second_derivatives` and `optimization_algorithm` change the estimation time.

Model

The choice model is a Multinomial Logit Model (MMNL). We are trying two slightly different models, for reasons we will see later on. See the [week 2 report](#) to see data description and introduction.

Recall of the MMNL Preference Model

Model parameters: For $i \in \{tt, tc, hw, ch\}$:

- $\mu_{\log(\beta_i)}$ is estimated mean for the log of β_i , set to -3 before estimation
- $\sigma_{\log(\beta_i)}$ is standard deviation of β_i , set to -0.01 before estimation

Utility functions:

$$V_{alt1} = \beta_{tt} \cdot tt_1 + \beta_{tc} \cdot tc_1 + \beta_{hw} \cdot hw_1 + \beta_{ch} \cdot ch_1$$

$$V_{alt2} = \beta_{tt} \cdot tt_2 + \beta_{tc} \cdot tc_2 + \beta_{hw} \cdot hw_2 + \beta_{ch} \cdot ch_2$$

Estimation:

After modifying the code in Biogeme in order to have panel data, we get a new model which is `MMNL_Preference_Model_Final`. The estimated parameters are given by Table 1. We observe now that the values between Apollo and Biogeme are closer than the one we had last time (see [week 2 report](#)). We also need to look at the statistics results we get in Table 2. We verify that the number of estimated parameters/threads/individuals sampled are the same in both cases. We see however that the initial and final log likelihoods are not quite the same. This might be due to translation errors (from Apollo to Biogeme), optimization differences between the two libraries or that there is a random number generation, which are thus not the same in both codes.

Parameter	Value in Apollo	Value in Biogeme
$\mu_{\log(\beta_{tt})}$	-1.984	-2.541
$\sigma_{\log(\beta_{tt})}$	-0.442	-0.577
$\mu_{\log(\beta_{tc})}$	-1.016	-2.62
$\sigma_{\log(\beta_{tc})}$	-0.991	-1.89
$\mu_{\log(\beta_{hw})}$	-2.938	-3.07
$\sigma_{\log(\beta_{hw})}$	-0.834	0.268
$\mu_{\log(\beta_{ch})}$	0.631	0.251
$\sigma_{\log(\beta_{ch})}$	0.858	0.801

Table 1: Parameter values for the `MMNL_Preference_Model_final`

Statistic	Value in Apollo	Value in Biogeme
Number of estimated parameters	8	8
Number of threads	8	8
Number of individuals	388	388
Initial log likelihood	-2253.78	-1703.926
Final log likelihood	-1444.35	-1534.219
AIC	2904.7	3084.437
BIC	2953.97	3116.125
Time of estimation (hh:mm:ss)	00:03:17	00:32:57

Table 2: Model Estimation Results `MMNL_Preference_Model_final`

Optimization

First of all, we clearly see that there is a big estimation time difference between the two library. Apollo takes only **00:03:17** to estimate the model, whereas Biogeme takes nothing less than **00:32:57**. There is almost a 30 minutes gap between the two. As for CPU/RAM monitoring, I could not correctly implement the `monitor_system()` function as I wished to do. I wanted to be able to run the monitoring while the code is running (in both Python and RStudio), however after some research and test and trial I could not make it work. I have however measured CPU and RAM using the `monitor_system` as it was and plotted the usage for both Apollo and Biogeme. Because the time of estimation was different, I monitored the CPU/RAM usage during 4 minutes in Apollo and 50 minutes in Biogeme. I have created the `CPU_RAM_plots` function to plot the percentage of CPU and RAM. The plots can be seen in Figures [1](#) and [2](#). We observe that the RAM usage is slightly different between Apollo and Biogeme, however it stays approximately within the same range. As for the CPU, it is clearly more solicited by the Biogeme code. There are big peaks as well, which I am not sure to what it corresponds.

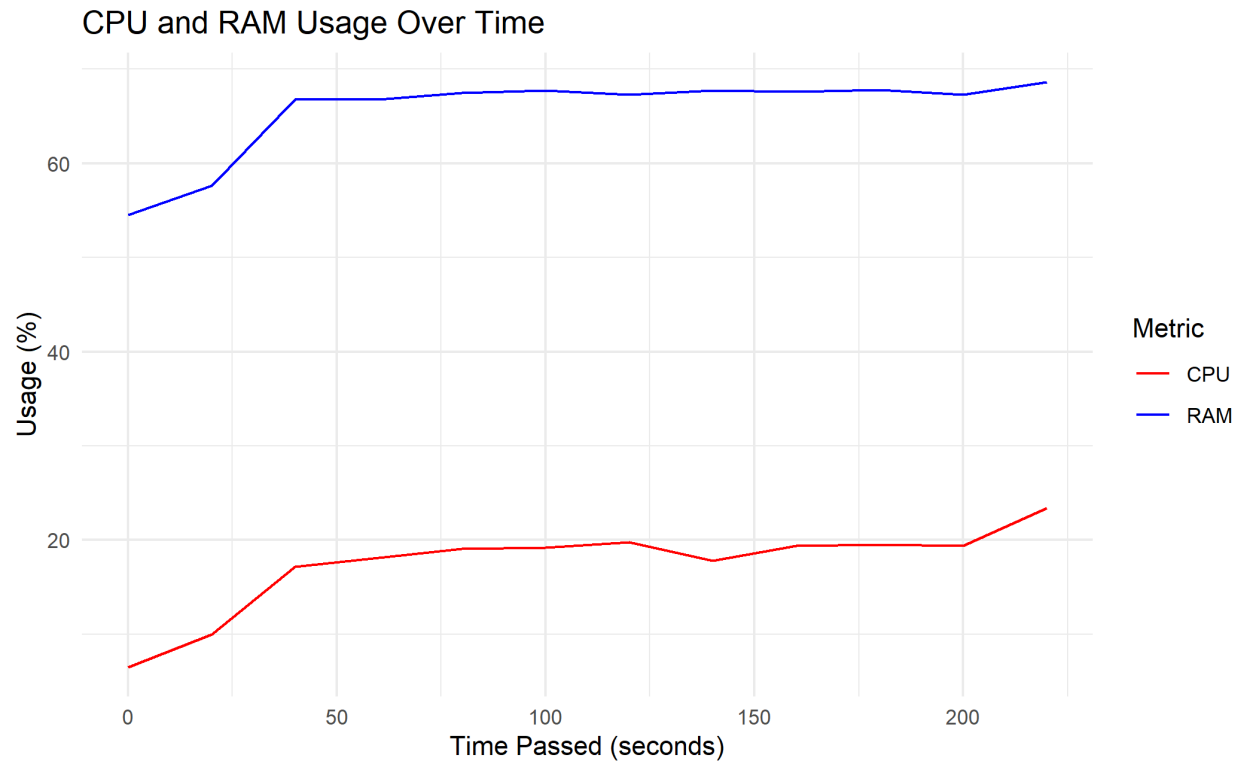


Figure 1: CPU/RAM usage in Apollo

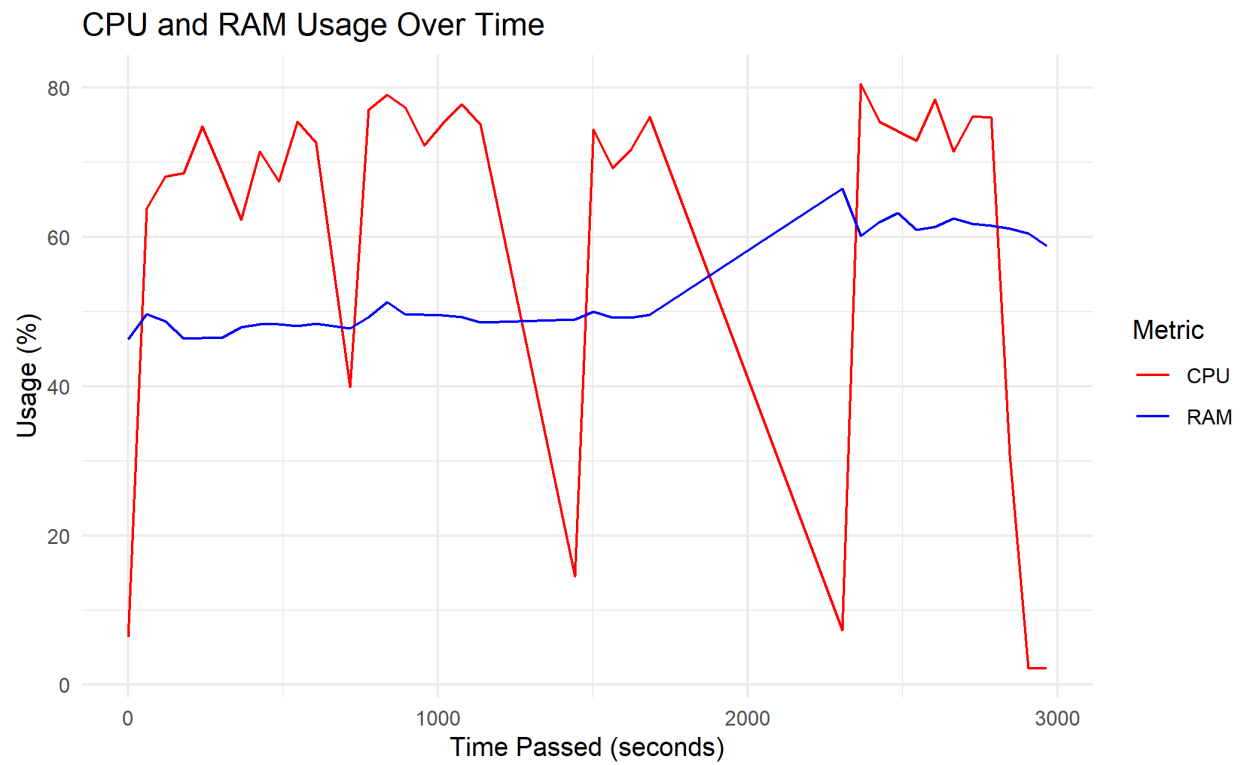


Figure 2: CPU/RAM usage in Biogeme