SEMESTER PROJECT – FINAL PRESENTATION

**RECAP :**

* Goal and process
  + Widely used in transportation, for environmental issues, economics, marketing, etc. For now, will mainly focus on choices of transportation modes
  + Many tools on the market to do modelling. Will focus on two packages: *Biogeme* (in RStudio, developed by Bierlaire), and *Apollo* (in Python, developed by Hess & Palma).
  + The main goal is to compare the two packages
    - will look at the performances (computational speed, resources consumption),
    - potential differences in the results and statistics (for same model/dataset),
    - features (how the packages differs or have the same features)
  + The process will always be as follows:
    - select a model/dataset from Apollo or Biogeme and translate it to the other language
    - make sure the model equivalent: need to look at the specifications in both scripts before estimation, as well as after estimation; Initial and Final LL, estimated parameters
* Recap midterm pres
  + We developed two models during the first part of the project
    - Multinomial Logit Model with only revealed preferences (observed behavior): not much result (estimation time very small, around 1min each), decide to go with more complex model
    - Mixed Multinomial Logit Model: Stated preference, introduction to 1 random coefficient with Halton draws: add heterogeneity to the model. Not truly equivalent but already got some results
  + Tried to measure CPU and RAM usage for the MMNL model, will come back to this later on
  + After the midterm pres, decided to set second derivative to 0 in Biogeme for better performances

**SWISSMETRO:**

* Description:
  + We think the not equivalence in the last MMNL model might be due to data issues. We thus decide to choose a script and dataset from the Biogeme website instead
  + Again, use randomness with the Halton draws, and will vary the number of draws later on to see how the estimation time varies
  + Change of algorithm:
    - Biogeme: Newton to BFGS (because of second derivative set to 0)
    - Apollo: BGW to BFGS as well (will justify this result later on)
* Utilities
  + As usual, set one of the ASC to 0 (in this case, the swissmetro one)
  + Decide add one random coefficient: for time -> different individuals have different sensitivities or preferences regarding travel time.
* Estimation and statistics
  + First look at the equivalence for 100 draws, to make sure we can start varying the number of draws
  + Same parameters, same Initial and Final logelikelihood -> ok to go
* Estimation time / draws plots
  + Try for 100 draws to 10’000 draws
  + Gap up to 44 minutes for 5’000 draws
  + 10’000 draws in Apollo: seems to be an outlier. We tried to rerun but my computer kept crashing

**SWISSMETRO 2:**

* Utilities
  + Add three random coefficients: one for cost, two directly in the ASC
  + Observe that we changed the types of draws (in Biogeme only) from Normal Halton to just normal, will tell about it later
* Estimation and statistics
  + Again, try again for 100 draws to make sure the models are equivalent
  + Same parameters, same Initial and Final logelikelihood -> ok to go
* Estimation time / draws plots
  + Try for 100 draws to 10’000 draws
  + Gap up to **4h and 13min** (for 10’000 draws)
* Problem encountered
  + **Apollo:** False convergence from 500 draws: change the algorithm from BGW to BFGS; apparently, BFGS has mechanisms to escape local minimums more effectively than BGW, which might have been the case here
  + **Apollo:** Crashing issue for > 7000 draws -> might be due to the use of RAM in Apollo, will talk about it later
  + **Biogeme:** equivalence error >1 random coeff. Indeed, first try to add randomness into cost (so we have time and cost which are random) -> models are not equivalent anymore. We observe then that by changing from Normal Halton to normal, no problem (estimated parameters are not exactly equal but final LL is)

**CPU/RAM USAGE:**

* MMNL Preference space:
  + We had already observed that RAM is a bit more solicited for Apollo than Biogeme, however CPU much more used by Biogeme
  + However, second derivative turned to 1, which might be the cause of these big peaks we can see
* Swissmetro 2 (1000 draws)
  + Decide to be more precise in the measurement time
  + Estimation time are around 10 minutes in Apollo, 50 in Biogeme
  + Again, we observe this big peaks in Biogeme. However, RAM seems way more solicited in Apollo than Biogeme
  + Try to delve deeper into the investigation:
    - CPU: much more used in Biogeme, see mean and max
    - RAM: much more used in Apollo, see mean and max
  + Conclusion: This suggests that Biogeme’s implementation is computation-intensive but not memory-demanding, while Apollo appears to hold a substantial amount of data in memory without performing extensive calculations
  + This might explain why Apollo was crashing for over 6’000 draws. We tried on a different computer and got a RAM error after a while for the 10’000 draws scripts

**FEATURES COMPARAISON:**

* + Just look at the functions in Apollo that does not seem to exist in Biogeme, or that need to be scripted by hand
  + apollo\_basTest(): likelihood ratio test need for one model to be a restrictive version of the other. In this case, can compare any models we want
  + apollo\_choiceAnalysis() and apollo\_sharesTest(): go a step further from just computing marketshare. Will either compute marketshare across subsamples in dataset, or compared shares predicted by the models with the shares observed in the dataset
  + apollo\_fitsTest(): compares how well the estimated parameters fit with any subsets of the data

**CONCLUSION:**

* Indeed, Biogeme is slower than Apollo:
  + This might be due to the solicitation of the CPU, meaning that they are more calculation in the biogeme case
  + However, advantage: RAM is less solicited than for Apollo, no crashing issues at all, can use as much number of draws as desired
  + User friendly: not anyone has powerful computer (as mine) but can still get results
  + Main error: the equivalence error when adding more than one random coefficient, should work with Halton draws
  + It was a pleasure to work on the project !!