**Code and Input Notes**

1. The code to create the inputs is **prep\_size\_term\_data\_estimation.ipynb**
2. The working directory for this work is: **R:\activitysim\estimation\2017\_2019\_data\size\_terms**
3. The code to run the estimation of each trip type is: **estimate\_size\_terms.Rmd**

It sources the code to do the maximum likelihood estimation is called **size\_estimation\_dest\_choice.R** We can’t check size\_estimation\_dest\_choice in currently because we don’t own it.

1. The input files are designated by the model selctor: workplace, school, non\_mandatory, atwork, or trip, and within the files are the segments such as work\_low, work\_med, work\_high, work\_very\_high, university, gradeschool, highschool, shopping, eatout, atwork, etc.

It has MAZ level household, employment, and park data, as well as the number of trips by type destined to each MAZ.

1. Validate\_size\_terms.Rmd is a notebook that helps you find trips going to MAZs with nothing in them in terms of the size variables select in your model. The function validate\_inputs will list out the MAZs that have trips going to them but nothing in them. You can also make a map of the region with these MAZ to visualize them with the function: map\_trips\_no\_stuff
2. I’m writing out the individual size variables sets by purpose to an outputs directory, with the file name being the trippurpose\_size.csv. Each trip purpose has the number of columns equal to all potential land use predictors, with zeros if the land use predictor is not used.
3. Then I’m collating them manually into outputs/destination\_choice\_size\_terms\_psrc.csv. we should script that, sorry. I’m just copy-pasting the individual csvs I wrote out into the overall csv.
4. Code should be rewritten in python and be made open source to be consistent with the rest of ActivitySim

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Estimation notes**

* **Interesting to see the different coefficients between non-mandatory tour destination and intermediate stops on the same purpose**
* **People are really attracted to food jobs**
* There is too much typing in this thing; error prone; fix the typiness
* Just trying to get a quick reasonablish estimation through
* Sometimes adding in variables with a little less oomph to have some dispersion across destinations because sometimes people do weird things, go weird places
* You could do everything programmatically and find the best coefficients and variables for the modeling given some constraints if you wanted to; like you could say optimize across all variables, with some down weighting for adding an additional variable. Then you wouldn’t have to go through and estimate each purpose separately. Does the human touch improve the model or bias it?