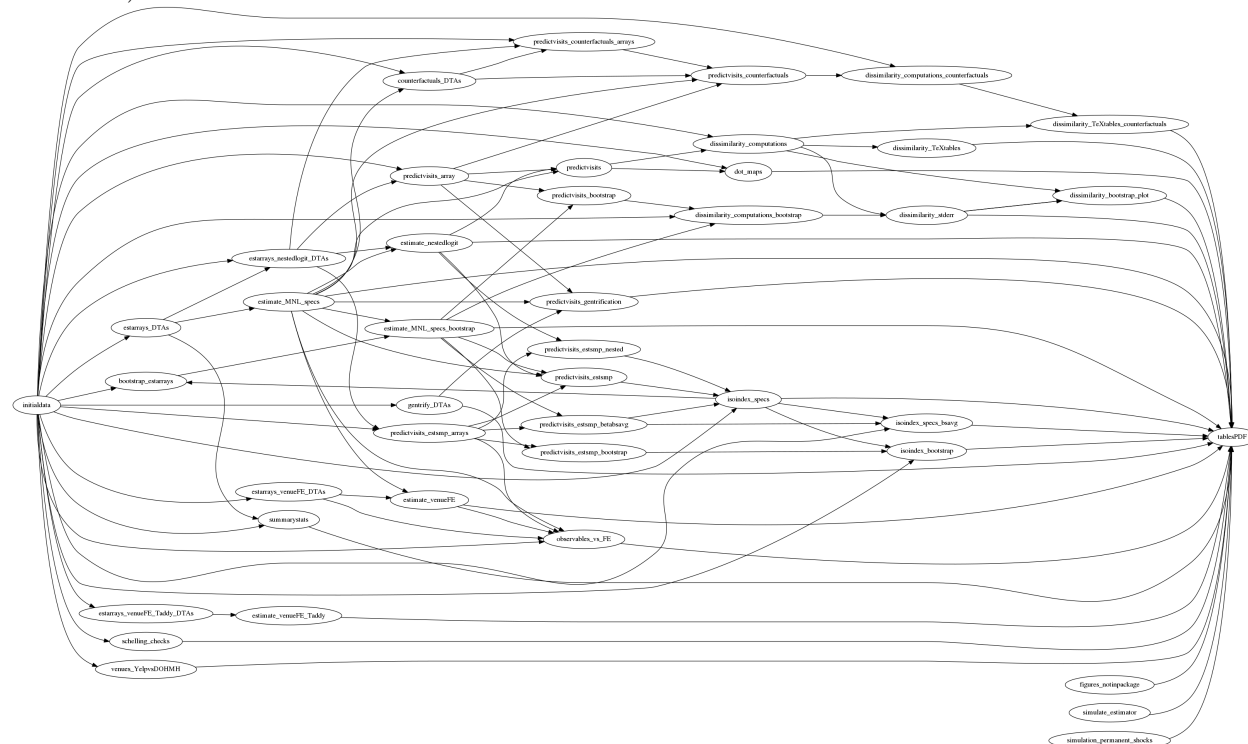


This repository contains the data and code underlying the paper “How Segregated is Urban Consumption?” in the *Journal of Political Economy* by Don Davis, Jonathan Dingel, Joan Monras, and Eduardo Morales.

Kevin Dano provided outstanding research assistance and contributed very substantially to this code.

## Code organization

Our project is organized as a series of tasks. The main project directory contains 48 folders that represent 48 tasks. Each task folder contains three folders: **input**, **code**, **output**. A task’s output is used as an input by one or more downstream tasks. This large graph depicts the input-output relationships between tasks (zoom in to view):



We use Unix’s [make](#) utility to automate this workflow. After downloading this replication package (and installing the relevant software), you can reproduce the figures and tables appearing in the published paper and the online appendix simply by typing **make** at the command line.

## Software requirements

The project’s tasks are implemented via [Julia](#) code, [Stata](#) code, and Unix shell scripts. In particular, we used Julia 0.6.2, Stata 15, and GNU bash version 4.2.46(2). To run the code, you must have installed [Julia 0.6.2](#), Stata, and Bash. The taskflow structure employs [symbolic links](#).

*Note to Mac OS X users:* The code presumes that Julia and Stata scripts can be run from Terminal via the commands **julia** and **stata-se**, respectively. Please follow the instructions for [Running Julia from the Terminal](#) and [Running Stata from the Terminal](#).

## Replication instructions

### Download and configure

1. Download (or clone) this repository by clicking the green **Clone** or **download** button above. Uncompress the ZIP file into a working directory on your cluster or local machine.
2. From the Unix/Linux/MacOSX command line, navigate to the working directory and configure the project based on whether you will be running the code using the [Slurm workload manager](#) in a computing cluster environment or locally:
  - If you are using Slurm, type `bash slurm_configuration.sh` and enter the name of the computing partition to which the batch jobs should be submitted.
  - If you want to run code locally, type `bash local_configuration.sh` to impose the local configuration. The script will ask you to specify the number of CPUs available.

*Warning:* We strongly recommend using a computing cluster if possible. This is a large project (in terms of both disk space and computation time) that heavily exploits parallel computation.

### Run code

After setting up your configuration, typing `make` in the working directory will execute all the project code.

*Warning:* A few of the tasks are computationally intensive and take days to run. The slow tasks are: `dissimilarity_computations_bootstrap`, `estimate_MNL_specs_bootstrap`, `estimate_nestedlogit`, `estimate_venueFE`, `estimate_venueFE_Taddy`, `predictvisits_bootstrap`. `predictvisits_bootstrap` alone needs more than 1TB of disk space.

- To replicate everything, at the command line type `make` or `make full_version`. It may take several days to produce everything. You need at least 4TB of disk space.
- To replicate the results in the main text that can be computed in less than a day on typical hardware (you need at least 10GB of RAM and 70GB of disk space), type `make quick_version`.

### Notes

- It is best to replicate the project using the `make` approach described above. Nonetheless, it is also possible to produce the results task-by-task in the order depicted in the [flow chart](#). If all upstream tasks have been completed, you can complete a task by navigating to the task's `code` directory and typing `make`.
- Given Julia's ongoing development (i.e., evolving syntax), it is important to use [Julia version 0.6.2](#) to run this code.
- An internet connection is required so that scripts can install Julia packages and Stata programs.
- The `slurm_configuration.sh` script asks the user to name a single computing partition on which the batch jobs are run. To vary this at the folder level, type `make edit_sbatch` within the task folder and select that task's partition.