Readme file for MATS uncertainty project

Code files to run in order:

1) build_cems_hourly.do:
  -- Louis Preonas' unit-level data didn't have pollution. (only facility level did)
  -- This file pulls the unit-level data again from the downloaded files, keeps
     pollution, and merges it into Louis' cleaned version of the unit level data.

2) merge_CEMS.do:
  -- This code creates our main analysis data by merging together:
    1) EIA923 data
    2) EIA-EPA crosswalk
    3) PRISM weather data
    4) State electricity demand data from PUDL
    5) CPI data
    6) fuel price data, year_state_FERC_AERfuel_fillin.dta,
       created by Gowrisankaran, Langer, and Reguant
    7) electricity price data by state, market_prices_state.dta,
       created by Gowrisankaran, Langer, and Reguant
    8) CEMS unit annual characteristics
    9) hourly CEMS data
  -- This file outputs, in ../data/processed_data:
    state_coal_share_data.dta
    adoption_data.dta
    ownership_data.dta
    CEMS_annual.dta
    CEMS_hourly_merged_data.dta

  -- In addition, it creates a number of intermediate data sources.

3) hourly_regressions.do: generates O&M, technology, and ramping costs.
  -- This code Outputs ramping_0.ster, ... , ramping_700.ster, the estimates of the
     ramping regressions plus estimates_om.ster, the technology and O&M cost estimates.

4) gen_profit_surface.do:
  -- This code generates coefficients of the profit surface

5) gen_dynamic_data.do:
  -- This code generates coefficients and states for C program in CSV format

6) create_halton_sequences.do: generates 3-tuple halton draws

7) reduced_form_evidence.do:
  -- This code generates reduced form tables and graphs from analysis data.

8) Equilibrium estimation:
adoption_estimation: compiled from adoption_estimation.c and other files
  -- First, transfer files to cluster:
     From code/ *.h, *.c, run*ua
     Initially: Makefile, and then probably edit
     From data/processed_data/ : *.csv
  -- Run the estimation on the cluster with the run script (see below)
  -- Create and transfer parameter files back:
     Copy the best parameter vector from the output file and remove everything
       before the coefflabel, so rows should start with "Adoption", "Exit", etc.
     On the first line, put: label, value
     param`paramdate'.csv and param`paramdate'_v.csv

9) Create bootstrapped standard errors:
  adoption_estimation.c:
  -- Generate bootstrap draws using -g option on local computer:
     adoption_estimation -v -g100 -iparamYYMMDD_v.csv
     adoption_estimation -g100 -iparamYYMMDD.csv
  -- Transfer bootstrapped datasets to cluster
     unit_data_*B*.csv and state_year_data_*B*.csv to cluster
  -- Run estimation on bootstrapped data, separately by draw number, on cluster:
  -- Transfer back:
     ../data/bootdata/bootparam*.csv

10) Run equilibrium part of counterfactuals:
  adoption_estimation:
  -- Run on local computer
  -- Counterfactuals can be run with adoption_estimation -v -pparam240503_v.csv

11) results_table.do: creates tables of results

12) counterfactuals.do: creates counterfactual tables

13) Figure 4 is created on the mathcha.io website. The code to create this figure
  (and related figures for the presentation) is in the ra.mathcha file.

To run multiple bootstrapped iterations, with draws 1-100:
Can't run these all on the same account in the same month,
  so need to divide tasks across accounts on UA HPC

```
sbatch -a 1-100 run_boot_ua
In the file run_boot_ua, the command $SLURM_ARRAY_TASK_ID

Using the Arizona cluster:

To transfer files to the server:
sftp alanger@filexfer.hpc.arizona.edu
cd MATS_uncertainty
Upload:
From code/ *.h, *.c, run*ua
From data/bootdata/ : *
Initially:
Makefile, and then probably edit
From data/processed_data/ : *.csv

ssh alanger@hpc.arizona.edu
shell
ocelote [or just don't do anything to stay on puma, which has more
space]
interactive -a alanger [wait to connect to a computer]
cd MATS_uncertainty/code
make adoption_estimation
* If we switch to xdisk, we would need:
cp adoption_estimation /xdisk/alanger/MATS_uncertainty/code
cd /xdisk/alanger/MATS_uncertainty/code

To run the code:
sbatch run_est_ua

To run multiple bootstrapped iterations, with draws 1-5:
sbatch -a 1-5 run_boot_ua
sbatch -a 1-5 run_boot_samefedstate_ua
In the file run_boot_ua, the command $SLURM_ARRAY_TASK_ID

Then transfer files back to local computer
```