*README Replication Files*

# The End of the American Dream? Inequality and Segregation in US Cities

Alessandra Fogli [*] Veronica Guerrieri [†] Mark Ponder [‡] Marta Prato [§]

April 20, 2025

## Contents

**A Overview** **1**

**B Quantitative Analysis** **2**

B.1 Calibration. . . . . 2

B.2 Main exercises . . . . . 3

B.3 MTO exercises . . . . . 5

## A Overview

This document describes the instructions to replicate the results in 'The End of the American Dream? Inequality and Segregation in US Cities" by Alessandra Fogli, Veronica Guerrieri, Mark Ponder, and Marta Prato, for the *Journal of Political Economy*.

[*]Federal Reserve Bank of Minneapolis. afogli00@gmail.com
[†]University of Chicago, Booth School of Business. vguerrie@chicagobooth.edu
[‡]NERA. ponder.mark@gmail.com
[§]Bocconi University, IGIER, and CEPR. marta.prato@unibocconi.it

# B Quantitative Analysis

This section describes the programs to replicate the quantitative analysis and produce the figures and tables described in the paper. We split the programs into three groups: calibration, main exercises, and MTO exercises.

## B.1 Calibration.

- "calibration.m" - This program runs the calibration and delivers the calibrated parameters. It calls several functions for the calibration exercise, described below.
- "err_obj_func_ls.m" - This function catches and handles parameter values where the objective function is not well defined without stopping the program execution.
- "obj_func.m" - The main objective function to minimize. Takes a vector of input parameters and returns the weighted least squares distance between the data moments and the model moments. It calls three functions: "tauchen.m", "steady_state.m" and "generate_summary_statistics_new", described further below.

  We implement the following strategies to improve the efficiency of the calibration algorithm. First, some parameters parameters can be backed out by directly inverting model equations given the moment targets. In this case, so we solve for such parameters within the model solution routine instead of searching over parameter values. We implement this strategy for the following parameters:

  - Given the running guesses for parameters $\phi_2$ and $\phi_3$, $\phi_1$ is calculated as residual given the target moment for the average city elasticity to be equal to 1.75.
  - The parameters $\lambda_A$ and $\lambda_B$ are calculated by inverting the housing supply equation, given the running solution for equilibrium rental rates and housing demand. We also note that the value of $\lambda_C$ is pinned down by the normalization of the city size to 1 in 1980.

  Second, the calibration algorithm also searches over the value of two objects that can have arbitrary normalizations but aide the calibration:

  - $\tau$. In particular, we can make the transformation $\tilde{e} = \tau^{\frac{1}{\gamma}}$ and scale $\beta_0$ and $\beta_1$ by $\tau^{\frac{1}{\gamma\zeta}}$. This leaves the optimization problem (where we now optimize over $n$ and $\tilde{e}$ instead of $n$ and $e$) and the wage equation unchanged.
  - $r_C$. As described above, instead of searching over the parameters for $\lambda_A$ and $\lambda_B$, we search over the value of one of the rental prices (we pick $r_C$, but we could pick any of the other two). Then we solve for the equilibrium value of the other two rental rates relative to $r_C$, and we invert the housing supply equations to back out $\lambda_A$ and $\lambda_B$.
- "tauchen.m" - Tauchen's method to discretize the AR(1) process governing ability.
- "steady_state.m" - Iteratively updates the wage, ability distrubtion until it converges to the steady-state. It calls two additional functions "find_spillovers_ss.m" and "update_dist_ss.m".

- "find_spillovers_ss.m" - Given a (wage, ability) distribution, solves for a fixed point in the rental rate and the spillover such that demand for housing equals the sizes of A, B, and C in 1980. There are two nested loops. The inner loop uses a spectral algorithm to solve for the market clearing prices given a vector of spillovers. The outer loop iterates on the spillovers to find a fixed point. The parameters ($b$, $\beta_0$, $\beta_1$, $\tau$, and $r_C$) are updated at each step so that the mean wage is normalized to 1 in the steady-state. Note that $r_C$ is a parameter we search over in place of $\lambda_C$. This is to ease calibration.

  It calls two functions: "find_rents_ss.m" and "choiceProb.m".

- "find_rents_ss.m" - Given vectors for the rental rate and the spillovers, calculate the excess housing demand. This calls the function

- "choiceProb.m" - Given a vector of rental rates and spillovers, calcualte the optimal level of education for each neighborhood, the probability of choosing each neighborhood, and the expected future wage. It calls "f_w.m".

- "f_w.m" - Determines how the parent's wage directly impacts their children's wage.

- "update_dist_ss.m" - Takes the current distribution of (wage, ability) and updates it by one period. Note that the wage grid is updated in each step so that it stays centered around the mean wage.

- "generate_summary_statistics.m" - This program generates the relevant moments from the model. Several programs are called that are analogous to the ones above with slight modification.

- "inv_hs.m" - Inverts the housing supply equation to generate a vector of lambdas given the steady-state price vector and the 1980 sizes of each neighborhood.

- "find_spillovers_dyn.m" - Similar to "find_spillovers_ss.m" but allows $r_C$ to adjust flexibly. It also calls the function "find_rents_dyn.m".

- "find_rents_dyn.m" - Similar to "find_rents_ss.m" but uses the housing supply equation to determine neighborhood sizes, rather than setting sizes equal to the 1980 levels. Calculates demand and supply and returns the residual. Note that population is allowed to growth at a rate of 1.0893 per decade. Calls "housing_supply.m".

- "housing_supply.m" - Given a vector of rental rates, determines the supply of houses in each neighborhood.

- "update_dist_dyn.m" - Similar to "update_dist_ss.m" with one modification. Allows the future wage to be updated for the skill shock without impacting the optimal decisions. This is to allow the parent's wage to adjust in 1980 after the skill premium shock without impacting the education or location choices of the grandparents.

## B.2 Main exercises

- "main_exercises.m" - This program runs the steady state, the counter-factual exercises, and produces the figures and tables in the paper. It calls two main functions to conduct exercises: "counter_factual.m" and "decomposition.m", described below.

- "counter_factual.m" - Calculates the various counterfactuals. Re-solves for the steady-state and then calculates one of the following counterfactual exercises:
  - Random mixing - Neighborhood choice probabilites are uniform
  - Fixed location 1 - Keeps individuals in their steady-state neighborhood. Prices don't adjust.
  - Fixed location 2 - Keeps individuals in their steady-state neighborhood. Prices adjust according to the SS housing supply.
  - Fixed location and education - Keeps individuals in their steady-state neighborhood and fixes education decisions to the steady state.
- "generate_counterfactual.m" - Does the actual counterfactual calculations. The distribution updating function is adjusted in one of the following ways:
  - "update_dist_int_edu_rm.m" - Updates the (wage, ability) distribution under random mixing case.
  - "update_dist_int_edu_ss.m" - Updates the (wage, ability) distribution fixing the location choice as well.
  - "update_dist_noedu.m" - Updates the (wage, ability) distribution fixing the location and education choices.

  It also uses the additional function:
  - "choiceProb_no_ed.m" - Similar to "choiceProb.m" but does not update the education decision.
- "decomposition.m" - Calculates additional exercises. Re-solves for the steady-state and then calculates one of the following exercises:
  - Fixed spillover - Keep spillovers fixed at their initial steady state levels
  - Mechanical effect - Fixing the rental rates and the optimal choices of the parents, but letting the spillovers adjust
  - Partial equilibrium - Fixing the rental rates, but letting the optimal choices of the parents and the spillovers adjust
  - Fixed Education - Keep the education policy function fixed at the steady state

  The distribution updating function is adjusted in one of the following ways:
  - "update_dist_int_edu_fix_p.m" - Updates the (wage, ability) distribution for given neighborhood choices.
  - "update_dist_int_edu_dn_Ss.m" - Updates the (wage, ability) distribution in the mechanical effect case.
  - "update_dist_int_noedu.m" - Updates the (wage, ability) distributionwith fixed education.

  These programs use the following additional functions:
  - "find_spillovers_mech.m" - solves for equilibrium fixed point while keeping spillovers fixed
  - "find_spillovers_noedu.m" - solves for equilibrium fixed point while keeping education fixed

### B.3 MTO exercises

- "experiments.m" - This program runs the MTO counterfactual experiments and produces the figures and tables in Section 6 of the paper. It calls one main function to conduct exercises: "counter_fact_mto.m", described below.
- "counter_fact_mto.m" - Re-solves for the steady-state and then calculates the MTO counterfactual exercises.
- "generate_counterfactual_mto.m" - Implements the MTO experiment at a given scale.
- "choiceProb_ct_par.m" - Similar to "choiceProb.m", but incorporates the MTO into the education and neighborhoods decisions.
- "choiceProb_ct_par_eu.m" - Similar to "choiceProb_ct_par.m", but it additionally calculates the expected utility before the neighborhood preference shock realizes.
- "update_dist_int_edu_mto.m" - Updates the (wage, ability) distribution before the MTO experiment is implemented.
- "update_dist_int_edu_mto_2.m" - Updates the (wage, ability) distribution after the MTO experiment is implemented.
- "find_spillovers_mto.m" - Similar to "find_spillovers_dyn.m" but includes the MTO implementation in the spillover fixed point algorithm.
- "find_rents_mto.m" - Similar to "find_rents_dyn.m" but includes an additional fixed to calculate the tax rate to finance the MTO experiment.