

Replication Package for “To Own or to Rent? The Effects of Transaction Taxes on Housing Markets”

1 Data availability statement

This paper uses data from the following sources:

Greater Toronto Area Multiple Listing Service transaction records (GTA MLS)

The data on residential real-estate sales and leasing transactions come from Multiple Listing Service (MLS) transaction records for the period 2006–2018 in the Greater Toronto Area (GTA). The data are managed and owned by the Toronto Regional Real Estate Board ([TRREB, 2019](#)) and cannot be made publicly available owing to TRREB’s terms of service and privacy policies. Interested researchers may contact TRREB to negotiate access. A detailed description of the data is provided in section 2.1 of the article.

GTA shapefiles Distances to the City of Toronto border are imputed using shapefiles from [Canadian Subdivision Boundaries](#) (source: [Stats Canada](#), public access) and [Scholars GeoPortal](#) (source: [DMTI Inc.](#), restricted access).

Web-scraped rental listings Rental listings were scraped from the websites [realtor.ca](#) ([REALTOR.ca, 2022](#), referred to as ‘MLS’) and [rentals.ca/Toronto](#) ([Toronto Rentals, 2022](#)) between 25th November and 5th December 2022. These data necessarily reflect what was available on the market in real time and hence are no longer publicly available. Moreover, they are governed by the listing platforms’ copyright restrictions and terms of service, so the scraped data cannot be shared in a public repository. The complete archived scraped data are available upon request, subject to approval from the respective platforms. These data are used solely in Appendix A.1.1 of the article and are described in detail there.

2 Software requirements

The replication package code is written in **Stata 18.0**, **Python 3.10** (**Jupyter Notebooks**), and **MATLAB R2024a**. The following software and packages are required to reproduce the results:

- **Stata 18.0 (64-bit)** — Required packages:

- `estout` (as of November 2024): Provides the `esttab` command for creating publication-style regression tables.
- `reghdfe` (as of November 2024): Used for high-dimensional fixed effects regression.
- `ftools` (as of November 2024): Required by `reghdfe` for efficient data handling.

The Stata script `00_setup.do` installs all required user-written packages.

- **Python 3.10.12 (Jupyter Notebook)** — Required packages:

- `geopandas == 1.1.1`
- `matplotlib == 3.8.0`
- `jupyterlab == 4.0.7` (or `notebook`)
- `pandas == 2.1.1`
- `numpy == 1.26.1`
- `scipy == 1.15.1`
- `shapely == 2.0.6`
- Installation command:

```
pip install geopandas==1.1.1 matplotlib==3.8.0 jupyterlab==4.0.7 \
pandas==2.1.1 numpy==1.26.1 scipy==1.15.1 shapely==2.0.6
```

- **MATLAB R2024a (64-bit)** — Required toolboxes:

- Optimization Toolbox (version 24.1)
- Statistics and Machine Learning Toolbox (version 24.1)

The code requires Dynare (version 5.4) for solving the dynamic model. Please ensure Dynare is installed and on the MATLAB path.

- **Operating system:** The Stata and Python code was executed on a university Linstat server for the last time. Exact system specifications are not publicly available, but a standard Linux or Windows server environment is assumed. Stata file `00_init.do` allows changing the file naming system to a different operating system file name.

The MATLAB code was executed on a 10-core Intel-based PC running Windows 11 with 32 GB of RAM and 200 GB of free disk space.

Hardware and runtime requirements

Stata and Python code:

- **Maximum runtime:** Up to 15 hours for full replication.
- **Longest task:** Survival regressions (`0*_table2_*.do`) are the most computationally intensive step and may require several hours on a high-performance server.
- **Other tasks:** All other tables and figures take under 1 hour to generate.
- **Estimated RAM requirement:** Up to 20 GB during peak use.
- **Disk space required:** Approximately 50 GB in total, including installed software and temporary files.

MATLAB code:

- **Maximum runtime:** 10 minutes
- **RAM requirement:** 2 GB
- **Disk space required:** Approximately 5 GB in total to install software

Setup instructions

It is recommended to run the following setup steps before execution:

- `00_setup.do` – installs the Stata packages
- Use the pip command given above to install the Python dependencies
- Ensure Dynare 5.4 is installed and added to the MATLAB path

3 Description of folder and code structure

1. The main folder `HNS_Replication` contains all the programs necessary to complete the replication.
2. The folder contains Stata and Python code for replicating the empirical estimates. These include the following:

- (a) `distance.ipynb` — A Python script that calculates distances to the city border using shapefiles from Canadian Subdivision Boundaries and Scholars GeoPortal.
 - (b) Stata analysis — All the remaining analysis is performed using Stata. The key scripts are:
 - `00_init.do` and `01_run.do` (master) — Run these files to replicate the results.
 - `01_run.do` — This calls all other `.do` files, each of which corresponds to a specific table or figure in the paper.
 - (c) `results` — Contains the output tables and figures.
 - (d) `data_sample` — A sample dataset illustrating the structure of the data, with one unit of observation.
3. The subfolder `replicate_fig1` contains code, sample data, and output files for replicating Figure A.1 in Appendix A.1.1. It includes the following:
- (a) `MLS_TR_data_and_plot.py` — A Python script that verifies the coverage of the MLS data compared to other online rental platforms. It performs data cleaning, spatial analysis, and generates a visualization of the data (Figure A.1).
 - (b) `raw_data` — This subfolder contains:
 - A sample dataset illustrating the structure of the scraped Realtor.ca (MLS) transaction records, with one unit of observation.
 - A sample dataset illustrating the structure of the scraped Toronto Rentals transaction records, with one unit of observation.
 - Toronto shapefiles in the folder `neighbourhoods_140`
 - (c) `output` — Contains output from running the script `MLS_TR_data_and_plot.py`.
 - `TR_vs_MLS_distribution.png` — Visualization of rental listings in Toronto
4. There is Matlab code for replicating the calibration and simulations of the theoretical model and the welfare analysis:
- (a) Calibration — The function `calibration.m` finds parameters matching steady-state targets. This function invokes various subroutines:
 - `calibcritcred.m`, `calibcritown.m`, and `calibcritr.m`

The targets are defined and returned by the function `targets.m`. The script `matchregression.m` matches the empirical response of the moving rate to the tax change.

- (b) Steady-state solution — The function `predict.m` finds the steady-state predictions of the model for given parameters. This function depends on various sub-routines:
 - `checkeqns.m`, `credcostcrit.m`, `findvars.m`, `modelcrit.m`, `ownthrescrit.m`, `popcredcost.m`, `rentalvars.m`, `rentthrescrit.m`, `solveownthres.m`, and `solventthres.m`
- (c) Dynamic solution — The file `rentown.mod` specifies the equations of the model in a format suitable for solving the model using Dynare and computing the perfect-foresight dynamics. The function `rentown_steadystate.m` integrates the steady-state solver function `predict.m` with Dynare.
- (d) Replication — There are scripts to reproduce specific results tables and graphs found in the paper:
 - `creditcostdistrib.m`, `furtherresults.m`, `impulseresponses.m`, `investortax.m`, `parameters.m`, `results.m`, and `welfare.m`

4 Instructions for replication

To replicate the results in the paper, complete the steps described below to prepare the data, run the estimation, and simulate the model. Details about which files produce specific tables and figures from the paper are provided in [Table 1](#) and [Table 2](#).

Estimation

1. `00_init.do` — Sets up the working folder and specifies the path of the input and output files.
2. The following code prepares the data for estimation:
 - `distance.ipynb`
 - `001_cleaning_transactions.do`
 - `001_cleaning_agg_data.do`
3. `01_run.do` (*master*) — Calls all other `.do` files, each of which corresponds to a specific table or figure in the paper. See [Table 1](#) for details.

Calibration and simulation of the quantitative model

1. Include Dynare on the Matlab path.
2. Specific results are generated by running the scripts `parameters.m`, `results.m`, `impulseresponses.m`, `welfare.m`, `creditcostdistrib.m`, `furtherresults.m`, and `investortax.m`. See Table 2 for details. These scripts can be run independently in any order.

Comparison of web-scraped data Running `python MLS_TR_data_and_plot.py` generates the output subfolder in the folder `replicate_fig1`, which contains Figure A.1.

References

- REALTOR.ca. (2022). *MLS rental listings*. Retrieved between 25th November and 5th December 2022, from <http://realtor.ca/>
- Toronto Rentals. (2022). *Rental listings*. Retrieved between 25th November and 5th December 2022, from <http://rentals.ca/Toronto>
- TRREB. (2019). *Toronto Multiple Listing Service data*. Toronto Regional Real Estate Board.

Table 1: List of estimation tables/figures and programs

Figure/Table	Program	Output files
Figure 1	02_figure1.do	results/figure1_*.pdf
Figure A.1	MLS_TR_data_and_plot.py	output/TR_vs_MLS_distribution.png
Table 1	03_table1.do	results/table1_sample3.tex
Table 2	04_table2_hazard.do	results/table2_sample3.tex
Table A.3	05_tableA3_run.do	results/tableA3_*.tex
Table A.4	06_tableA4_x.do	results/tableA4_x.tex
Table A.5	07_tableA5_sales_leases.do	results/tableA5_sales_lease_*.tex
Table A.6	08_tableA6_pdom.do	results/tableA6_pdom_sample3.tex
Table A.7	09_tableA7_crisis.do	results/table*_crisis_*.tex
Table A.8	10_tableA8_placebo.do	results/table1_*_placebo_*.tex
Table A.9	11_tableA9_bto_btr.do	results/tableA9_*.tex
Table A.10	12_tableA10_hazard_period.do	results/table2_sample4.tex results/table2_sample6.tex
Table A.11	13_tableA11_prop_types.do	results/tableA11_prop_sample3.tex
Table A.12	14_tableA12_price_period.do	results/tableA12_*.tex

Notes: The first column lists the figures and tables in the paper. The second column indicates the corresponding program file that generates each output. The third column shows the raw output filename generated by the programs. One program can produce several output files that match the output file naming. The symbol ‘*’ represents a placeholder for relevant variations in naming outcomes and samples. Sample 3 refers to the main sample (2006–2012), Sample 4 covers data up to 2010, and Sample 6 extends to 2018.

Table 2: List of quantitative model results and programs

Place in paper	Program	Output variables
Table 3	targets.m	Targets in tgt
Table 4	parameters.m	Parameters in prm
Table 5	results.m	Results in avresp and ssresp
Figure 2	impulseresponses.m	Results in resps and ssresp
Table 6	welfare.m	Results in C* and l*
Text of Section 5.3	welfare.m	Results in C*
Text of Section 5.4.1	furtherresults.m	Results in avresp
Figure 3	creditcostdistrib.m	Density function in dens
Text of Section 5.4.2	furtherresults.m	Results in avresp
Text of Section 5.4.3	investortax.m	Results in C* and l*
Table A.13	furtherresults.m	Results in avresp

Notes: The first column lists the places in the paper where results from the quantitative model are found. The second column indicates the corresponding program file that generates each output. The programs directly display the results as well as saving them to the variables named in the third column. The symbol ‘*’ represents a placeholder for different variables.