Overview

The code in this replication package for the paper "Bunching in Real-Estate Markets: Regulated Building Heights in New York City" by Jan Brueckner, David Leather, and Miguel Zerecero for publication in the Journal of Urban Economics constructs the analysis file from 15 releases (02a, 03c, 04c, 05d, 06c, 07c, 09v1, 10v1, 11v1, 12v1, 13v1, 14v1, 15v1, 16v1, & 17v11) of NYC's PLUTO database published by the NYC Department of City Planning using Julia and R. One main file runs all of the code to generate the data for the tables and figures (with the exception of Figure 1, which is illustrative) in the paper.

Data Availability and Provenance Statements

The paper uses data obtained from the New York City Department of City Planning (NYCDCP). Fifteen releases of the Public Land Use Tax Lot Output (PLUTO) dataset (02a, 03c, 04c, 05d, 06c, 07c, 09v1, 10v1, 11v1, 12v1, 13v1, 14v1, 15v1, 16v1, & 17v11), as well as release 17v11 of the MAPPluto dataset, are used for producing Figure 8, the map of sample properties. All raw data (including codebooks) are included in the archive under the directories `~/raw_data/PLUTO/` and `~/raw_data/MAPPLUTO/`, and can be downloaded from the "BYTES of the BIG APPLE - Archive" on the NYCDCP's website. All data are licensed under the NYC Open Data Law.

- [x] I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.

License for Analysis Data

The data resulting from the replication code are licensed under an MIT license. See `LICENSE.txt` for details.

Summary of Availability

- [x] All data **are** publicly available.
- [ ] Some data **cannot be made** publicly available.
- [ ] **No data can be made** publicly available.

Computational Requirements

Software Requirements

- [x] The replication package contains one or more programs to install all dependencies and set up the necessary directory structure.

- Julia 1.10.4 (download)

    - The main script - `~/run.jl` - installs and precompiles all dependencies. Uncomment and change the directory path in `line 48` to ensure you are in the root directory of the repository before running.
    - `AlgebraOfGraphics` v0.6.19
    - `Bootstrap` v2.4.0
    - `CSV` v0.10.14
    - `CategoricalArrays` v0.10.8
    - `DataFrames` v1.6.1
    - `DataFramesMeta` v0.15.2

    - `JLD2` v0.4.48
    - `LaTeXStrings` v1.3.1
    - `Latexify` v0.16.3
    - `Plots` v1.40.4
    - `RCall` v0.14.1
    - `Dates`
    - `Printf`
    - `Random`
    - `Statistics` v1.10.0

- R 4.3.1 (download)

    - The file `~/code/scripts/create_sample_map.R` will automatically install all packages. Change `line 2` to ensure the working directory is set to the root of the repository.
    - `data.table` 1.15.4
    - `ggplot2` 3.5.1
    - `ggmap` 4.0.0
    - `sf` 1.0-16
    - `RColorBrewer` 1.1.3
    - `svglite` 2.1.3
    - `dplyr` 1.1.4

Controlled Randomness

- [x] Random seed is set at `line 49` of the program `~/run.jl`. It is set to `12431413412`.
- [ ] No pseudo-random generator is used in the analysis described here.

Memory, Runtime, Storage Requirements

Summary

Approximate time needed to reproduce the analyses on a standard 2023 desktop machine:

- [ ] <10 minutes
- [ ] 10-60 minutes
- [x] 1-2 hours
- [ ] 2-8 hours
- [ ] 8-24 hours
- [ ] 1-3 days
- [ ] 3-14 days
- [ ] > 14 days

Approximate storage space needed:

- 10GB if `save_intermediate_files = false` at `line 54` of `./run.jl`
- 25GB if `save_intermediate_files = true`

Approximate memory needed:

- 16GB if `produce_raw_data = false` at `line 51` of `./run.jl`

- 32GB if `produce_raw_data = true`

Details

The code was last run on a 6-core AMD-based desktop with Windows 11 Pro, 64GB of RAM, and over 200GB of free space.

Description of Programs/Code

The codebase is written into the Julia module `Bunching`, whose contents can be found in the file `~/code/Bunching/Bunching.jl`. Five functions are called in the script `~/run.jl` to go from the raw data to results. Two additional functions are included to estimate and bootstrap θ.

- `clean_raw_pluto_data(save_results::Bool)`
  - Description: This function cleans and combines all releases of NYCPLUTO into a single `DataFrame`.
  - Input: `save_intermediate_files` - Boolean - `true` or `false`.
    - If `true`, the resulting `DataFrame` is saved to the file `~/processed_data/df_cleaned_pluto.jld2`.
  - Output: `df_pluto` - A `DataFrame` object.
- `transform_cleaned_pluto_data(df::DataFrame, save_results::Bool)`
  - Description: This function accepts the `DataFrame` outputted by `clean_raw_pluto_data()` and applies several transformations and filters to the data before compiling the analysis data.
  - Inputs:
    - `df` - `DataFrame` - This is the output `df_pluto` from `transform_cleaned_pluto_data()`.
    - `save_results` - Boolean.
      - If `true`, the resulting output `DataFrame`, `df`, will be saved to the file `~/processed_data/df_transformed_cleaned_pluto.jld2`.
  - Output: `df` - A `DataFrame` object.
- `generate_analysis_subsamples(df::DataFrame)`
  - Description: This function uses the cleaned PLUTO data outputted from `transform_cleaned_pluto_data()` to save the analysis data to five different files corresponding to each of the values of the FAR limits, $\bar{F} \in [2.0, 1.25, 0.9, 0.6, 0.5]$. The files are written in `CSV` format to `~/processed_data/df_2p0.csv`, `~/processed_data/df_1p25.csv`, `~/processed_data/df_0p9.csv`, `~/processed_data/df_0p6`, and `~/processed_data/df_0p5.csv`. They are also saved in `JLD2` format using the extension `.jld2` for convenience.
  - Input: `df` - A `DataFrame` object.
    - The inputted `DataFrame` should be the output from `transform_cleaned_pluto_data()`.
- `generate_tables_figures(seed::Int)`
  - Description: This function creates Figures 2 - 7 and 9, as well as Tables 1 - 4 in the paper. The figures are saved in the `~/figures` directory as `svg`, `pdf`, and `png` formats. All filenames are formatted as `figure_X_*desc*` where `X` is the figure number and `*desc*` is the description of the figure. Tables 1 - 4 are saved in the `~/tables` directory with the file name format `tableX.csv` where `X` is the table number.
  - Input: `seed` - `Integer` - Controls the randomness of the bootstrap of θ.
- `create_sample_map(use_rcall::Bool)`

    - Description: This function writes a `CSV` file, `~/processed_data/df_map.csv`, that is used by the provided `R` script, `~/code/scripts/create_sample_map.R`, to create Figure 9. If `use_rcall = true`, then the script will attempt to call the R script using the `RCall` package. Otherwise, one must manually run the R script.
    - Input: `use_rcall` - Boolean - If `true`, Julia attempts to run `create_sample_map.R`; otherwise, one must run the script manually.
- `est_θ(F::Vector{Float64}, δ::Float64, F_bar::Float64)`
    - Description: This function computes the point estimate of θ as described in the paper. It first computes the bunching area, B, and then computes θ using the quadratic formula.
    - Inputs:
        - `F` - `Vector{Float64}` - A vector of the FAR values. Must be of type `Float64`.
        - `δ` - `Float64` - The size of the bins.
        - `F_bar` - `Float64` - The regulatory FAR limit.
    - Output: `θ` - `Float64` - Point estimate.
- `bootstrap_θ(F::Vector{Float64}, δ::Float64, F_bar::Float64, N_boot::Int64, ci_lvl::Float64)`
    - Description: This function first computes the point estimate of θ and then performs a bootstrap using random sampling with replacement.
    - Inputs:
        - `F` - `Vector{Float64}` - A vector of the FAR values. Must be of type `Float64`.
        - `δ` - `Float64` - The size of the bins.
        - `F_bar` - `Float64` - The regulatory FAR limit.
        - `N_boot` - `Int64` - The number of bootstraps to perform.
        - `ci_lvl` - `Float64` - Must be in $(0,1)$ - The confidence level of the bootstrap.
    - Outputs: A `tuple` of length 5 whose elements are described in order below:
        - `θ` - `Float64` - Point estimate.
        - `θ_σ` - `Float64` - Point estimate of the bootstrapped standard error.
        - `θ_bias` - `Float64` - The estimated bias of the point estimate.
        - `θ_bci` - `Vector{Float64}` - The bootstrapped confidence intervals per input `ci_lvl`.
        - `θ_bs` - `Vector{Float64}` - The bootstrapped sample of θ.

There is also a single script written in `R` to produce Figure 8.

- `~/code/scripts/create_sample_map.R`
    - Description: This script generates Figure 8 in the paper. It depends on the file `~/processed_data/df_map.csv` produced by the Julia function `create_sample_map(use_rcall::Bool)`.
    - Instructions: One must set the working directory in `line 2` to match the root of the repository.

License for Code

The code is licensed under an MIT license. See `~/LICENSE.txt` for details.

Instructions to Replicators

- Install Julia 1.10.4. For instructions on setting up your Julia environment, see the QuantEcon lecture.
- Install R 4.3.1. For installation instructions for Windows, see here.
- Edit `~/code/scripts/create_sample_map.R` to adjust the working directory to the repository root.

- Optional: Edit `~/run.jl` to set `save_intermediate_files` to `true` if you wish to write the intermediate data files.
- Open Julia in your preferred method (I recommend opening VS Code from the root of the repository using the terminal command `code .`). Make sure the working directory is set to the root of the repository.
- Run `~/run.jl` either from the Julia IDE or a command prompt.
- If `RCall` fails to run `~/code/scripts/create_sample_map.R`, open R Studio and run the script manually.

List of Tables and Programs

The provided code reproduces:

- ☐ All numbers provided in the text of the paper.
- ☐ All tables and figures in the paper.
- ☑ Selected tables and figures in the paper, as explained and justified below.

Figure/Table #	Program -> Function	Line Number	Output File
Table 1	Bunching.jl -> generate_tables_figures()	249	~/tables/table1.csv
Table 2	Bunching.jl -> generate_tables_figures()	273	~/tables/table2.csv
Table 3	Bunching.jl -> generate_tables_figures()	322	~/tables/table3.csv
Table 4	Bunching.jl -> generate_tables_figures()	349	~/tables/table4.csv
Figure 1	n.a. (no data)		
Figure 2	Bunching.jl -> generate_tables_figures()	194	~/figures/figure_2_bunching_areas_0p6.png
Figure 3	Bunching.jl -> generate_tables_figures()	166	~/figures/figure_3_FAR_distribution_2p0_4xbins.png
Figure 4	Bunching.jl -> generate_tables_figures()	138	~/figures/figure_4_FAR_distribution_1p25_4xbins.png
Figure 5	Bunching.jl -> generate_tables_figures()	110	~/figures/figure_5_FAR_distribution_0p9_4xbins.png
Figure 6	Bunching.jl -> generate_tables_figures()	83	~/figures/figure_6_FAR_distribution_0p6_4xbins.png
Figure 7	Bunching.jl -> generate_tables_figures()	56	~/figures/figure_7_FAR_distribution_0p5_4xbins.png
Figure 8	create_sample_map.R		~/figures/figure_8_FAR_distribution_2p0_4xbins.png

Figure/Table #	Program -> Function	Line Number	Output File
Figure 9	Bunching.jl -> generate_tables_figures()	427	~/figures/figure_9_bootstrap_hist_2p0.png

References

NYC Department of City Planning. 2002. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 02a [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2003. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 03c [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2004. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 04c [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2005. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 05d [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2006. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 06c [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2007. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 07c [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2008. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 09v1 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2010. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 10v1 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2011. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 11v1 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2012. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 12v1 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2013. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 13v1 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2014. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 14v1 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2015. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 15v1 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2016. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 16v1 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2017. "Primary Land Use Tax Lot Output (PLUTO) Data, Version 17v11 [dataset]." NYC Department of City Planning.

NYC Department of City Planning. 2017. "MAPPLUTO Data, Release 17v11 [dataset]." NYC Department of City Planning.