README File for the Replication Package of "Do Parties and Voters Disagree? An Equilibrium Analysis of Candidate Selection"

by Arvind Magesan, Andrea Szabó, and Gergely Ujhelyi

Contents

1 Overview **2**

2 Package Contents **2**

3 Data **4**

3.1 Data sources 4

3.2 List of included data files 8

4 Computing requirements and instructions **11**

4.1 Stata 11

4.2 Matlab 11

4.3 Python 11

4.4 ArcGIS 12

5 Description of Codes **13**

5.1 Part A - Preliminaries 14

5.2 Part B - Setup data for demand estimation and demand estimation 16

5.2.1 Setup for demand estimation 16

5.2.2 Demand estimation 17

5.3 Part C - Setup for supply estimation and supply estimation 18

5.3.1 Setup for supply estimation 19

5.3.2 Supply estimation 21
5.4 Part D - Counterfactuals 21

6 List of Tables and Figures **22**

1 Overview

The code in this replication package constructs the analysis datasets and then performs the analysis using Stata, MATLAB, Python, and ArcGIS Pro. Stata and Python are used to build the analysis data. ArcGIS Pro is used to construct the crosswalk between old and new parliamentary constituency boundaries. MATLAB is used for estimating the demand side of the model. Python is used for the supply-side estimation and counterfactual analysis.

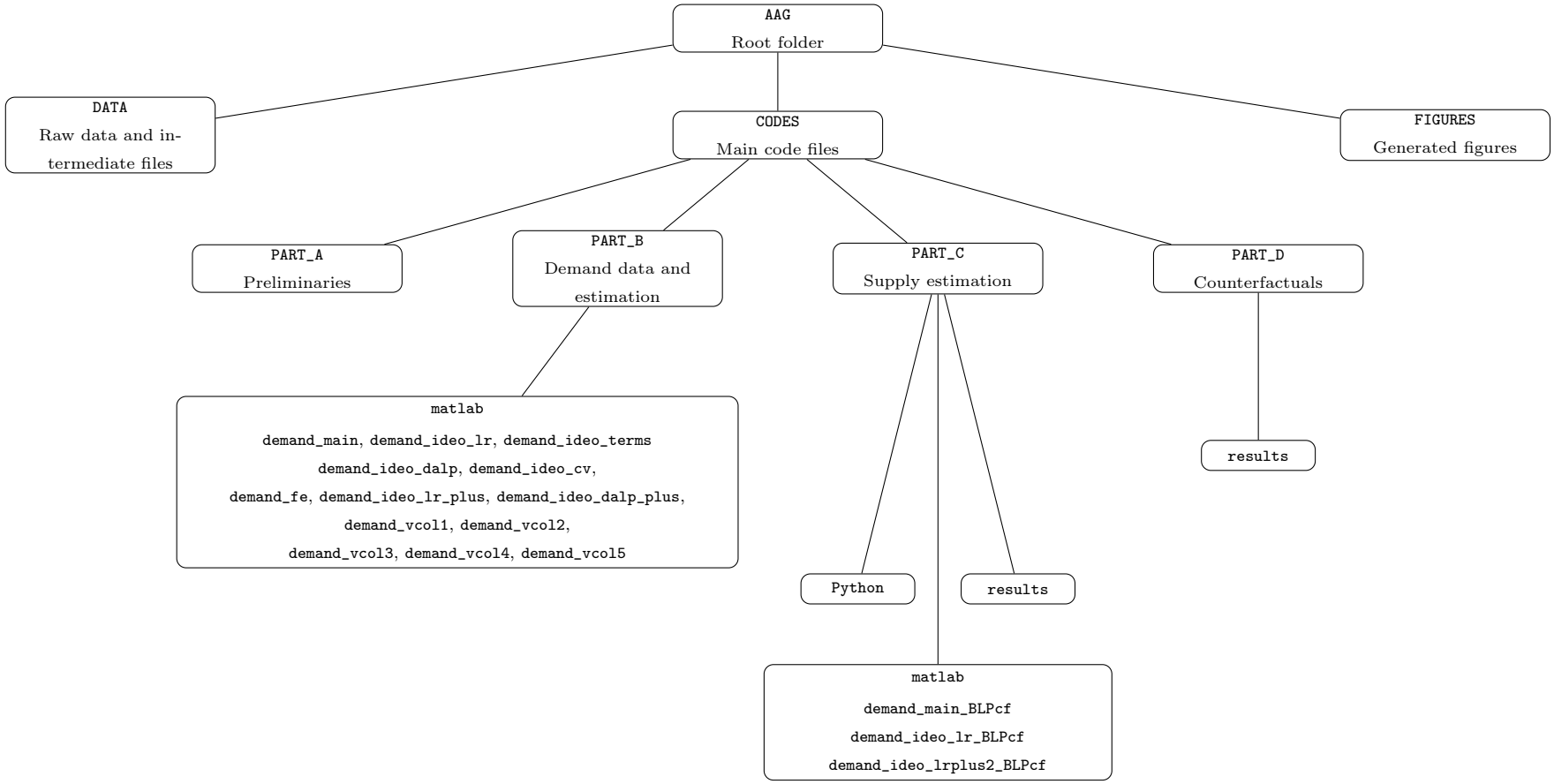
The replication consists of four parts, numbered Part A - Part D. Some parts of the package consist of master files that call a sequence of other scripts, while other parts require running separate files manually. The replication should be run in the order described in each section under **Run order**. Figure 1 illustrates the structure of the replication package.

2 Package Contents

The root folder of the replication package is `AAG`. This can be downloaded to the user's desired location. The `AAG` folder contains three subfolders: `DATA`, `CODES`, and `FIGURES`. All raw data files used in the replication package are located in the `DATA` folder. The `CODES` folder has four subfolders:

- `PART_A`: Preliminaries.
- `PART_B`: Setup data for demand estimation and demand estimation.
- `PART_C`: Setup for supply estimation and supply estimation.
- `PART_D`: Counterfactuals.

Figure 1: Replication package structure



The files that users need to run are located in the `PART_A`, `PART_B`, `PART_C`, and `PART_D` folders. Some of these files call other scripts or functions stored elsewhere in the replication package. For example, `Master1.do` is located in the `PART_A` folder, but it calls several other do-files. Users should therefore run only the files specified for `PART_A`, `PART_B`, `PART_C`, and `PART_D`, in the order described in this README.

The `FIGURES` folder is initially empty. Figures from the paper are created by the replication code and will be saved in this folder.

3 Data

All data used in this paper are publicly available and included in the replication package. When the original files are in a proprietary format, the package includes copies in .csv in a subfolders named "`* csv versions`".

3.1 Data sources

- Ashoka. The electoral dataset compiled by the Trivedi Center for Political Data at Ashoka University (Bhogale et al., 2019) is publicly available at `https://lokdhaba.ashoka.edu.in/browse-data`. The user needs to specify the Election Type, either General Elections (National) or Assembly Elections (State), choose the election year and then download the associated .csv file. We provide the version used in our paper, downloaded on November 23, 2020.

- Election Commission of India. A number of datasets were obtained from the Election Commission's website at `https://www.eci.gov.in/`. The website changes often, and not every file is always available. We have provided the data files necessary to replicate our analysis. This includes the following types of files:

  - Election returns for the 2004, 2009 and 2014 national elections, including candidate characteristics.
  - The list of assembly constituencies (AC) belonging to each parliamentary constituency (PC)
  - Broadcast allowance of parties in the 2009 and 2014 elections
  - Geo-coded boundaries of the pre-2008 parliamentary constituencies. State codes are shown in Table 1 below.

- Myneta. Candidate affidavit information is publicly available at `myneta.info` (Association for Democratic Reforms, 2019). Because the information may be updated, we have provided the

Table 1: State boundary folders

Original folder name	State
S01	Andhra Pradesh
S03	Assam
S04	Bihar
S06	Gujarat
S07	Haryana
S27	Jharkhand
S10	Karnataka
S11	Kerala
S12	Madhya Pradesh
S13	Maharashtra
S18	Odisha
S20	Rajasthan
S22	Tamil Nadu
S24	Uttar Pradesh
S25	West Bengal

original data files.

- SHRUG. The SHRUG (Asher et al., 2021) contains information from the Indian Census merged to electoral information and other datasets. It is publicly available at `https://www.devdatalab.org/shrug`. We used version 1.5 and provide all the original files necessary to replicate our analysis. These include village demographics from the census, election and affidavit data from assembly elections, and keys to assign this information to the corresponding parliamentary constituencies.

- Indian Census. Village level tables from the Population Census are publicly available at `https://censusindia.gov.in/census.website/data/census-tables`. We provide the downloaded data file used in our analysis.

- DALP. The DALP I data (Kitschelt and Smidt, 2013) and codebook can be downloaded (once the user has registered) from this site: `https://sites.duke.edu/democracylinkage/data/`. Download the file `DALP_datasets_20131108.zip`. We use the file `partylevel_20130907.dta`.

- Post-delimitation constituency boundaries. Geo-coded post-delimitation (post-2008) constituency boundaries can be downloaded from: https://github.com/datameet/maps/tree/master/parliamentary-constituencies. The data used in this paper are the files starting with `india_pc_2019` and we included all necessary files in this replication package.

- Deflators. In our codes, monetary values for 2008-2017 are deflated using the Consumer Price Index available at https://alfred.stlouisfed.org (All Items for India Vintage: 2020-11-10).

- Datasets tabulated by the authors

  - Wikipedia-based ideology measures. We start from https://en.wikipedia.org/wiki/List_of_political_parties_in_India and the individual party websites linked from this page, and we collect information on the ideology of the parties that enter our dataset. We also collect Wikipedia's list of up to six specific ideological keywords for each party.

  - Muslim and Non-Muslim name libraries. We hired a Hindi native speaker research assistant (RA) with a working knowledge of multiple Indian languages to help us generate these libraries. Specifically, the RA identified 470 Muslim and 1210 non-Muslim name and name substrings/fragments from the set of all names of candidates who contested Indian elections between 2008-2019 as they appear in the affidavit data (Association for Democratic Reforms, 2019) described above.

  - Alliances. The full list of all candidates who ran as members of the NDA and UPA alliances in the 2009 and 2014 elections, including their parties and constituencies is available at the Wikipedia pages https://en.wikipedia.org/wiki/List_of_National_Democratic_Alliance_candidates_in_the_2009_Indian_general_election, https://en.wikipedia.org/wiki/List_of_United_Progressive_Alliance_candidates_in_the_2009_Indian_general_election, and similarly for 2014.

  - Keys and identifiers. The following files contain keys and identifiers assigned by the authors in order to merge different datasets together: `relpopcum_order.dta`, `pc_villages2_order`, `v1.dta`, `ordering_for_matlab.dta`

  - Intermediate files. We provide several intermediate files that can either be used to proceed with the analysis faster, or that are used for manually correcting data. `group checks.xls` is used to manually check and correct candidate ids in the Ashoka data. `missing_premuslim.csv` is a separate tabulation for creating Muslim indicators for candidates missing this information. Using the boundary data with ArcGIS Pro 3.2, we created 15 files containing the union of the old and new parliamentary constituency boundaries for the states used in the paper.

For convenience, we include the resulting 15 files, called `india_pc_2019_Union1*.xlsx` where * stands for a state.

3.2 List of included data files

Table 2 lists all the data files provided in the replication package.

Table 2: Data files and sources

File location	File name	Data source
\DATA\ECI data GE 1999_2019	GE2009.xls	ECI
\DATA\ECI data GE 1999_2019	Constituency wise detailed result 2014.xls	ECI
\DATA\ECI data GE 1999_2019	Constituency wise detailed result 2014.pdf	ECI
\DATA\ECI data GE 1999_2019	GE2004.xls	ECI
\DATA\ECI data GE 1999_2019	33.Constituency_Wise_Detailed_Result 2019.xls	ECI
\DATA\ECI data GE 1999_2019	parties99.xlsx	ECI
\DATA\ECI data GE 1999_2019	parties04.xlsx	ECI
\DATA\ECI data GE 1999_2019	parties09.xlsx	ECI
\DATA\ECI data GE 1999_2019	List of Political Parties Participated 2014.xls	ECI
\DATA\ECI data GE 1999_2019	List of Political Parties Participated 2019.xls	ECI
\DATA\ECI data GE 1999_2019	missing_age.dta	ECI
\DATA\ECI data GE 1999_2019	missing_preage_precaste.xls	ECI
\DATA\ECI data GE 1999_2019	missing_premuslim.csv	ECI
\DATA\AffidavitData\2004	35 files, each starting with an Indian state name and ending with 2004	Myneta
\DATA\AffidavitData\2009	35 files, each starting with an Indian state name and ending with 2009	Myneta
\DATA\AffidavitData\2014	36 files, each starting with an Indian state name and ending with 2014	Myneta
\DATA\AffidavitData\2019	36 files, each starting with an Indian state name and ending with 2019	Myneta
\DATA\shrug	assembly_elections_clean.dta	SHRUG
\DATA\shrug	actopc.xlsx	ECI

Continued on next page

Table 2 continued

File location	File name	Data source
\DATA\shrug	relpopcum_order.dta	keys and identifiers
\DATA\shrug	assembly_candidates_clean.dta	SHRUG
\DATA\shrug	pc_villages2_order.dta	keys and identifiers
\DATA\shrug\census	shrug_pc11r_key.dta	SHRUG
\DATA\shrug\census	shrug_pc11u_key.dta	SHRUG
\DATA\shrug\census	shrug_pc11.dta	SHRUG
\DATA\shrug\census	shrug_names.dta	SHRUG
\DATA\shrug\Census Downloads	Population_Census_AllStates.csv	Census
\DATA\shrug\con_shrug	con_shrug_2008.dta	SHRUG
\DATA\shrug\secc	shrug_secc.dta	SHRUG
\DATA\shrug\extrakeys	shrug_con_key_2008.dta	SHRUG
\DATA\shrug\affidavits	affidavits_clean.dta	SHRUG
\DATA\GISoldPCtonewPC	15 files india_pc_2019_Union1*.xlsx, where * is an Indian state name	intermediate file
\DATA\GISoldPCtonewPC\NEW PC boundaries	parliamentary-constituencies: ESRI shapefile dataset	DataMeet
\DATA\GISoldPCtonewPC\OLD PC boundaries	15 ESRI shapefile datasets, one per state	ECI
\DATA\MuslimNames	MuslimNames.csv	tabulated by the authors
\DATA\MuslimNames	NonMuslimNames.csv	tabulated by the authors
\DATA\MuslimNames	MuslimSubstrings.csv	tabulated by the authors

Continued on next page

Table 2 continued

File location	File name	Data source
\DATA\MuslimNames	NonMuslimSubstrings.csv	tabulated by the authors
\Data\Ashoka	TCPD_GE_all.xls	Ashoka
\Data\Ashoka	group checks.xls	intermediate file
\DATA\new_data	broadcast.dta	ECI
\DATA\new_data	candidates_national_alliances.csv	Wikipedia
\DATA\new_data	v1.dta	keys and identifiers
\DATA\ideology	ideology_wiki.dta	Wikipedia
\DATA\ideology	ideology_pca.csv	intermediate file
\DATA\ideology	party_names.csv	keys and identifiers
DATA\ideology	partylevel_20130907.dta	DALP
\CODES\PART B\matlab	ordering_for_matlab.dta	keys and identifiers

Note: broadcast.dta includes information entered manually from the included national_2009.pdf and national_2014.pdf. missing_age.dta includes information entered manually from the included Constituency wise detailed result 2014.pdf.

4 Computing requirements and instructions

Downloading and running the entire replication package requires approximately 4 GB of hard disk space. We recommend that the user fully download the entire package on their hard drive and turn off any automatic cloud syncing (e.g., Dropbox syncing) because delays in syncing may prevent some scripts from running correctly. Note that the `CODES/PART_C/matlab` subfolder is 890 Mb in size and contains over 15,000 files. The user should make sure that it is fully downloaded before proceeding with the replication.

4.1 Stata

The Stata code was run using Stata 18.0 SE. The code requires the following user-written packages: `rangestat`, `unique`, `ivreg2`, `avar`, `weakivtest`, `ranktest`, `outreg`. All Stata scripts run under 2 minutes, except `Master1.do` in Part A (8 minutes).

Instructions. Before running Stata files, the user needs to set the location of the `AAG` folder in `AAG/set_path.do`, and run this file. This do file must also be run anytime Stata is restarted during the replication process.

4.2 Matlab

The MATLAB files were run using MATLAB R2022b. The required toolboxes are the Global Optimization Toolbox and the Statistics and Machine Learning Toolbox. All MATLAB code was run on Sabine, the University of Houston's shared high-performance computing cluster. The code can also be run on a standard laptop with a 12-core processor and 16 GB of RAM. Most MATLAB codes run under an hour, except a few specifications, the longest of which runs under 5 hours.

Instructions. MATLAB files should be run from the folder where they are located (i.e., the "Current Folder" should be set to that location). All necessary files are included in each folder.

4.3 Python

All Python codes were run through Jupyter Notebooks on a 2024 Mac Mini computer with 10 cores, consisting of 4 performance cores and 6 efficiency cores, an Apple M4 chip, and the Sequoia 15.7.2 operating system. Each code runs in under 30 minutes, with the longest runtime being for `Supply_Estimation.ipynb`. All codes should be run in Python 3.13.9, preferably packaged by Anaconda. For the replication the user will need the following Python packages and versions: `pandas`

(2.3.3), `numpy` (2.3.5), `scikit-learn` (1.7.2), `statsmodels` (0.14.5), and `Levenshtein` (0.14.5). The `ipykernel package` (version 6.31.0) is required to register the replication environment as a Jupyter kernel. For convenience, we have included the yaml file `environment.yml` in the main folder, which the user can use to recreate the environment. We describe how to use environment.yml in the instructions below.

Instructions. The Python codes provided were written in Jupyter Notebooks, and, though not strictly necessary, it is most convenient for the user to run the files from Jupyter. The notebook form allows the user to run code one cell at a time, if desired, and to view the code together with results (tables) as they appear. To recreate the Python environment for replication, follow these steps:

1. Install Anaconda if it is not already installed on the machine.
2. Open the terminal and change the directory to the location of the replication package so that the file `environment.yml` can be found.
3. Create and activate the environment by entering:
   `conda env create -f environment.yml`
   and then
   `conda activate replication_python`
4. Register the Jupyter kernel:
   `python -m ipykernel install -user -name replication_python -display-name "Replication Python"`
5. Launch Jupyter
6. Once in the Jupyter interface with notebook open, go to: Kernel $--> $ Change Kernel $--> $ Replication Python.

The notebook should now be running using the correct environment for replication. NOTE: the user only needs to do this once - there is no need to repeat these steps for every notebook in the package. The file `config.py` does not need to be run by the user. All the Python notebooks import this file, as it captures the base file path of the package and then generates file paths that are used in all the Python codes for calling and saving data and results.

4.4 ArcGIS

The ArcGIS step uses ArcGIS Pro 3.2. The user needs to have access to the `Union` tool form the Analysis toolbox and the `Table To Excel` tool from the Conversion toolbox. The provided code

requires access to ArcGIS Pro's Python library, `arcpy`. The code runs in under 5 minutes.

Instructions. ArcGIS Pro includes its own Python environment. This Python environment is different from the standard, freely available Python installation used for other parts of the replication package. `arcpy` is available only through an installed and licensed copy of ArcGIS Pro. To run the ArcGIS Python code:

1. Open ArcGIS Pro 3.2.
2. Create a new project.
3. In the top ribbon, click **Analysis**.
4. Click **Python Window**. This opens the Python window inside ArcGIS Pro.
5. To check that the ArcGIS Python environment is active, type the following into the Python window and press Enter:
   ```
   import arcpy
   print(arcpy.GetInstallInfo()["Version"])
   ```

This should print the ArcGIS Pro version. If the command `import arcpy` fails, the code is being run in the wrong Python environment. For this replication package, the ArcGIS step was run using ArcGIS Pro 3.2. The user needs to edit the location of the `AAG` folder at the top of the ArcGIS Python code. Then the code should be pasted into the ArcGIS Pro Python window and run there. When running the ArcGIS code, users should ensure that the output folders are not being actively synced or locked by cloud-storage software such as Dropbox. For convenience, a `.txt` version of the ArcGIS script is also provided, as well as the 15 output files `india_pc_2019_Union1*.xlsx`, where `*` is an Indian state name.

5 Description of Codes

Files should be run in the order described in each section in the yellow boxes under **Run order**. These files are located in the `PART_A`, `PART_B`, `PART_C`, and `PART_D` folders under `CODES`.

5.1 Part A - Preliminaries

Run order.

0. `create_boundary_unions.py` (only if the user wants to recreate the ArcGIS boundary-union files).
1. `ideology_merge.ipynb`
2. `Master1.do`
3. `MuslimType_State.ipynb`
4. `Master2.do`
5. `MuslimType_National.ipynb`
6. `Master3.do`
7. `kmeans_hca.ipynb`
8. `clustering_figure.do`

File descriptions.

`create_boundary_unions.py`: ArcGIS Pro Python script that creates the 15 union Excel files matching old and new parliamentary constituency boundaries. This script must be run from the ArcGIS Pro Python environment because it uses `arcpy`.

`Ideology_merge.ipynb`: This file sets up the ideology measures from DALP.

`Master1.do`: Runs a series of do-files that merge and clean the raw data. `Master1.do` calls the following files:

- `setup parties.do`: Sets up election data from Ashoka. Creates `parties.dta`.
- `insheet AffidavitData.do`: Sets up affidavits data. Creates `affidavits.dta` and `affidavits2.dta`.
- `pctoac_merge.do`: Assigns AC information to PCs. Creates `acid.dta`, `censusid.dta`, `kids_work.dta`, `pc_villages.dta`,

  `formerge.dta`, and `acformerge.dta`.

- `draw demographics.do`: Draws simulated voters for the demand estimation. Creates `demogr.dta` and a series of files `demogr200_`f'.dta`.
- `boundaries_merge.do`: Sets up the data on old-new constituency correspondence. Creates `boundaries_formerge.dta`.
- `assembly_affidavits_merge_p1.do`: Merges the state election and affidavits data. Creates `assembly_withaffidavits_allmainpc.dta`.
- `ideology_setup.do`: Sets up the party ideology data.

`MuslimType_State.ipynb`: This code uses libraries of Muslim and non-Muslim names and substrings to classify state-level candidates.

`Master2.do`:

- `assembly_affidavits_merge_p2.do`: Updates `assembly_withaffidavits_allmainpc.dta` with the Muslim indicator.
- `kmean.do`: Does the k-means clustering. Creates `state.dta`.
- `merge age.do`: Merges election data from ECI with Ashoka data. Creates `withage.dta` and `ashoka_updated.dta`.
- `withgroups.do`: Updates candidate identifiers. Creates `tempRA.dta` and `ashoka_updated_withgroups.dta`.
- `merge affidavits.do`: Merges in the affidavit data. Creates `candidates.dta`.
- `merge candidates_shrug3p1.do`: Merges in old constituency information and Shrug data. Creates `winners_temp.dta`, `temp2014inc.dta`, `candidates_mainpc.dta`, and `z_mychars.dta`.

`MuslimType_National.ipynb`: This code uses libraries of Muslim and non-Muslim names and substrings to classify national candidates.

`Master3.do`:

- `merge candidates_shrug3p2.do`: Merges in Muslim indicator and state-candidate averages. Creates `alliance.dta`, `alliance_red.dta`, `state_all.dta`, `ass_type.dta`, `xtemp.dta`, `winners.dta`, and `maindata.dta`.
- `setup2.do`: Setup for `mattemp0.dta` (created in Master3.do), the starting point for the demand estimation. Creates Table A.1 Panels A and B, and Table A.3.

- `can_incumbent.do`: Setup for `mattemp0_canincumbent.dta` (created in Master3.do) which includes incumbent information used in supply estimation. Creates `missing_preage_formerge.dta`, `temp_national.dta`, `winners_formerge.dta`, `temp_missing_inc.dta`.

`kmeans_hca.ipynb`: Sets up data for Figure A.3.

`clustering_figure.do`: Produces Figure A.3.

Notes. The DALP principal component measures called in by `Ideology_merge.ipynb` can only be reproduced up to a numerical precision of $1e-5$. This is due to changes in Python versions and updates to the libraries used to do Principal Components analysis in Python since the initial code was written. We therefore provide these as data. The code that was used to produce them is provided in `Ideology_pca.ipynb`.
In the HCA clustering in `kmeans_hca.ipynb`, we have found that earlier versions of the main Python libraries result in slightly different values, and thus small differences on Figure A.3. The figure in the paper was produced using `numpy` 2.3.5, `pandas` 2.3.3, and `sklearn` 1.7.2.

5.2 Part B - Setup data for demand estimation and demand estimation

Run order.

1. `demand.do`
2. `est.m` file(s) corresponding to the desired specification(s).
3. `elasticity.do`
4. `sensitivity_figure.do`

5.2.1 Setup for demand estimation

`demand.do:` Runs a series of do files to set up the different demand specifications.

- `setup_mattemp.do`: Sets up the differentiation IVs. The file also creates `mattemp.dta` and Table A.1 Panel C.
- `setup for demandest.do`: Sets up the files that will be imported by Matlab to run the demand estimation. Files go into separate folders (see below). Creates `formatlab2.dta`, `formatlab2.csv`,

varnames.csv, formatlab.dta, formatlab.csv, formatlab dummies.dta, formatlab dummies.csv, demogr200_1.csv and demogr200_2.csv in each folder.

- patterns.do: Produces Table 3.
- ivcheck_regressions.do: Produces Tables A.4-A.6.

5.2.2 Demand estimation

Each demand specification has its own estimation file est.m, located in one of the folders below. Each folder is self-contained, i.e., once demand.do has been run, each folder will contain all the files necessary to run the estimation.

- demand_main, demand_maincol2, demand_maincol3, demand_maincol4, demand_maincol5: Contain the files necessary to produce Table 1, columns 1-5, respectively.
- demand_ideo_lr: Contains the files necessary to produce Table 6 column 1.
- demand_ideo_terms: Contains the files necessary to produce Table 6 column 2.
- demand_ideo_dalp: Contains the files necessary to produce Table 6 column 3.
- demand_ideo_cv: Contains the files necessary to produce Table 6 column 4.
- demand_vcol1, demand_vcol2, demand_vcol3, demand_vcol4, demand_vcol5: Contain the files necessary to produce Table A.7 columns 1-5, respectively.
- demand_fe: Contains the files necessary to produce Table A.9.
- demand_ideo_lr_pluscol1, demand_ideo_lr_pluscol2, demand_ideo_lr_pluscol3: Contain the files necessary to produce Table A.10 columns 1-3, respectively.
- demand_ideo_dalp_pluscol4, demand_ideo_dalp_pluscol5, demand_ideo_dalp_pluscol6: Contain the files necessary to produce Table A.10 columns 4-6, respectively.

Each of the above folders contains the following files:

- est.m: Main MATLAB script for the demand estimation. Users should run this file to replicate the demand estimation results. All other .m function files and data files are called by this code.
- gmmobj_weighted.m: Computes the GMM objective function.
- meanval.m: Computes the mean utility values.

- `mktsh.m`: Computes model-predicted market shares.
- `mufunc.m`: Computes the individual-specific utility component.
- `jacob.m`: Computes the Jacobian.
- `ind_sh.m`: Computes individual-level choice probabilities or shares.
- `v.mat`: MATLAB data file containing simulation draws for the Normally distributed random coefficients v_i. These are necessary for the code to run even in specifications that do not include v_i.
- `formatlab.csv`: Input data file.
- `formatlab2.csv`: Input data file.
- `formatlab dummies.csv`: Input data file.
- `varnames.csv`: File containing variable names used by the MATLAB code.

After the main specification has been estimated, `elasticity.do` produces Figure A.1 and `sensitivity_figure.do` produces Figure A.2.

Notes. The v_i draws (used in Tables A.7 (`demand_vcol*`), A.9 (`demand_fe`), and A.10 (`demand_*_plus`) are provided as data and cannot be replicated. The code that was used to produce the draws is provided in each `est.m` file (usually line 42) and can be used to create new draws.

5.3 Part C - Setup for supply estimation and supply estimation

Run order.

1. `get estimates.do`
2. `predicted_shares.m`
3. `put together cases.do`
4. `Supply_Estimation.ipynb`
5. `coefficient magnitudes.do`
6. `strategic complementarity.do`

5.3.1 Setup for supply estimation

In order to set up the supply estimation, "BLP counterfactuals" need to be run, which compute predicted market shares and win probabilities from the demand model for all combination of type choices. The files necessary to do this based on a specific demand specification are located in separate folders in the `matlab` subfolder:

- `demand_main_BLPcf` corresponds to the main specification produced by `PART_B/matlab/demand_main/est.m`.
- `demand_ideo_lr_BLPcf` corresponds to the specification produced by `PART_B/matlab/demand_ideo_lr/est.m`.
- `demand_ideo_lrplus2_BLPcf` corresponds to the specification produced by `PART_B/matlab/demand_ideo_lr_pluscol2/est.m`.

Before proceeding to the supply estimation, all three of these folders must be set up - i.e., the corresponding demand specifications must be set up and estimated in Part B, and the files `get estimates.do`, `predicted_shares.m` and `put together cases.do` in the corresponding folders must be run.

`get estimates.do`: Once the corresponding demand estimation has been run, this copies the necessary files to the correct folder.

`predicted_shares.m`: Computes the predicted market shares for all type choice combinations, creating a series of files `case_1.dat-case_25.dat`. The type choices that these cases correspond to are listed in Table 3 below.

Table 3: Counterfactual cases

	UPA					NDA				
Case	**TYPE 1**	**TYPE 2**	**TYPE 3**	**TYPE 4**	**Inc**	**TYPE 1**	**TYPE 2**	**TYPE 3**	**TYPE 4**	**Inc**
1	x					x				
2	x						x			
3	x							x		
4	x								x	
5	x									x
6		x				x				
7		x					x			
8		x						x		
9		x							x	
10		x								x
11			x			x				
12			x				x			
13			x					x		
14			x						x	
15			x							x
16				x		x				
17				x			x			
18				x				x		
19				x					x	
20				x						x
21					x	x				
22					x		x			
23					x			x		
24					x				x	
25					x					x

`put together cases.do`: Sets up the file `supply_BLP_column1_v2.dta` used for the supply estimation.

Note. `predicted_shares.m` uses 1000 characteristic draws conditional on types. These draws are provided as data in the `type_characteristics_alt/sim1-sim10` subfolders. Each of these contains 100 files for each of the 5 types (Types 1-4 plus Incumbent), where all observations in the data are assigned a simulated characteristic vector conditional on the given type. These 5000 files cannot be replicated. The code that was used to produce them is provided: `draw_type_characteristics.do` creates the $4 \times 100 \times 10$ files for Types 1-4 and `draw_incumbent_chars.do` creates the 100×10 files for Incumbent. These files can be used to create new draws.

5.3.2 Supply estimation

`Supply_Estimation.ipynb`: Carries out the estimation of the various supply specifications and performs the model validations. The user can "run all cells" in the notebook to produce Tables IV, V, A.11, A.12, A.13, and A.14. The program also produces the following files: `baseline_estimates.csv` (parameter Estimates from column 3 of Table IV), `baseline_eqm.csv` (dataset that includes payoff variables and choice variables in the equilibrium associated with column 3 of Table IV), and `blpdata.csv` (dataset with win probabilities and vote shares). These data files are saved to `/CODES/PART_C/results` as they are created.

`coefficient magnitudes.do`: produces Figure 1 After the main supply specification has been estimated.

`strategic complementarity.do`: computes the derivatives necessary to assess strategic complementarity of criminals, discussed in Section 6.2.2 of the paper.

5.4 Part D - Counterfactuals

Run order.

1. `Supply_Counterfactuals.ipynb`
2. `counterfactuals_figure.do`

`Supply_Counterfactuals.ipynb`: After the main supply specification has been estimated, this code takes as inputs `baseline_estimates.csv, baseline_eqm.csv` and `blpdata.csv` and carries out the counterfactual exercises. The user should "run all cells" in the notebook to produce Tables VII and VIII

as well as a dataset with counterfactual equilibrium probabilities called `crime_CF_baseline_eqm.csv` which is saved to `/CODES/PART_D/results` when it is created.

`counterfactuals_figure.do`: produces Figure A.4. after the criminal ban counterfactual has been run, using the dataset `crime_CF_baseline_eqm.csv`.

6 List of Tables and Figures

In Table 4 we list the Tables, Figures and in-text results generated in the replication and the code file that produces them.

Table 4: Location of tables, figures, and text outputs in the replication package

Figure/Table	Code File Name	Line / Cell	Folder	Output
Table I – Voter preference parameter estimates, column 1	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_main`	Displayed as code is run. Order of nonlinear parameter estimates may be different.
Table I – Voter preference parameter estimates, column 2	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_maincol2`	Displayed as code is run. Order of nonlinear parameter estimates may be different.
Table I – Voter preference parameter estimates, column 3	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_maincol3`	Displayed as code is run. Order of nonlinear parameter estimates may be different.
Table I – Voter preference parameter estimates, column 4	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_maincol4`	Displayed as code is run. Order of nonlinear parameter estimates may be different.
Table I – Voter preference parameter estimates, column 5	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_maincol5`	Displayed as code is run. Order of nonlinear parameter estimates may be different.
Table II – Centroids of candidate types	`kmean.do`	Line 67	`\DATA`	Displayed as code is run.
Table III – Electoral performance and frequency of different candidates in the raw data	`patterns.do` called in by `demand.do`	Lines 41 and 56	`\CODES\PART_B`	Displayed as code is run.
Table IV – Party objective function estimates	`Supply_Estimation.ipynb`	Cell 13	`\CODES\PART_C\Python`	Displayed as code is run.
Table V – Model fit	`Supply_Estimation.ipynb`	Cell 17	`\CODES\PART_C\Python`	Displayed as code is run.

Continued on next page

Table 4 – continued from previous page

Figure/Table	Code File Name	Line / Cell	Folder	Output
Table VI – Voter preference parameter estimates with different ideology measures, Left-Right	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_ideo_lr`	Displayed as code is run.
Table VI – Voter preference parameter estimates with different ideology measures, Terms	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_ideo_terms`	Displayed as code is run.
Table VI – Voter preference parameter estimates with different ideology measures, DALP	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_ideo_dalp`	Displayed as code is run.
Table VI – Voter preference parameter estimates with different ideology measures, CV	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_ideo_cv`	Displayed as code is run.
Table VII – Party choices with criminal ban	`Supply_Counterfactuals.ipynb`	Cell 10	`\CODES\PART_D`	Displayed as code is run.
Table VIII – Choice probabilities with and without term limits	`Supply_Counterfactuals.ipynb`	Cell 17	`\CODES\PART_D`	Displayed as code is run.
Table A.1, Panel A – Summary statistics	`setup2.do` called in by `CODES/PART_A/Master3.do`	Line 4	`\DATA`	Displayed as code is run.
Table A.1, Panel B – Summary statistics	`setup2.do` called in by `CODES/PART_A/Master3.do`	Line 76	`\DATA`	Displayed as code is run.
Table A.1, Panel C – Summary statistics	`setup_mattemp.do` called in by `demand.do`	Line 13	`\CODES\PART_B`	Displayed as code is run.
Table A.2 – Coding of Left–Right ideology positions from Wikipedia	N/A; presents definition of coding			

Continued on next page

Table 4 – continued from previous page

Figure/Table	Code File Name	Line / Cell	Folder	Output
Table A.3 – Ideological keyword measure	`setup2.do` called in by `CODES/PART_A/Master3.do`	Line 230	`\DATA`	Displayed as code is run.
Table A.4 – Characteristics regressed on instruments	`ivcheck_regressions.do` called in by `demand.do`	Lines 38–41	`\CODES\PART_B`	Displayed as code is run.
Table A.5 – Specification choice: differentiation IVs	`ivcheck_regressions.do` called in by `demand.do`	Line 74	`\CODES\PART_B`	Displayed as code is run.
Table A.6 – Specification choice: differentiation IVs and voter demographics	`ivcheck_regressions.do` called in by `demand.do`	Line 104	`\CODES\PART_B`	Displayed as code is run.
Table A.7 – Additional demand specifications, column 1	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_vcol1`	Displayed as code is run.
Table A.7 – Additional demand specifications, column 2	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_vcol2`	Displayed as code is run.
Table A.7 – Additional demand specifications, column 3	`est.m`	Line 212	`\CODES\PART_B\matlab\demand_vcol3`	Displayed as code is run.
Table A.7 – Additional demand specifications, column 4	`est.m`	Line 213	`\CODES\PART_B\matlab\demand_vcol4`	Displayed as code is run.
Table A.7 – Additional demand specifications, column 5	`est.m`	Line 213	`\CODES\PART_B\matlab\demand_vcol5`	Displayed as code is run.
Table A.8 – Distribution of candidate types	`setup2.do` called in by `CODES/PART_A/Master3.do`	Lines 8–9 and 93–96	`\DATA`	Displayed as code is run.
Table A.9 – Voter preference parameter estimates with heterogeneity on parties, column 1	`est_col1.m`	Line 218	`\CODES\PART_B\matlab\demand_fe`	Displayed as code is run.

Continued on next page

Table 4 – continued from previous page

Figure/Table	Code File Name	Line / Cell	Folder	Output
Table A.9 – Voter preference parameter estimates with heterogeneity on parties, column 2	`est_col2.m`	Line 218	`\CODES\PART_B\matlab\demand_fe`	Displayed as code is run.
Table A.9 – Voter preference parameter estimates with heterogeneity on parties, column 3	`est_col3.m`	Line 217	`\CODES\PART_B\matlab\demand_fe`	Displayed as code is run.
Table A.10 – Additional voter preference parameter estimates with alternative ideology measures, column 1	`est.m`	Line 215	`\CODES\PART_B\matlab \demand_ideo_lr_pluscol1`	Displayed as code is run.
Table A.10 – Additional voter preference parameter estimates with alternative ideology measures, column 2	`est.m`	Line 215	`\CODES\PART_B\matlab \demand_ideo_lr_pluscol2`	Displayed as code is run.
Table A.10 – Additional voter preference parameter estimates with alternative ideology measures, column 3	`est.m`	Line 215	`\CODES\PART_B\matlab \demand_ideo_lr_pluscol3`	Displayed as code is run.
Table A.10 – Additional voter preference parameter estimates with alternative ideology measures, column 4	`est.m`	Line 215	`\CODES\PART_B\matlab \demand_ideo_dalp_pluscol4`	Displayed as code is run.
Table A.10 – Additional voter preference parameter estimates with alternative ideology measures, column 5	`est.m`	Line 215	`\CODES\PART_B\matlab \demand_ideo_dalp_pluscol5`	Displayed as code is run.

Continued on next page

Table 4 – continued from previous page

Figure/Table	Code File Name	Line / Cell	Folder	Output
Table A.10 – Additional voter preference parameter estimates with alternative ideology measures, column 6	`est.m`	Line 215	`\CODES\PART_B\matlab \demand_ideo_dalp_pluscol6`	Displayed as code is run.
Table A.11 – Party objective function estimates with ideology measure	`Supply_Estimation.ipynb`	Cell 21	`\CODES\PART_C\Python`	Displayed as code is run.
Table A.12 – Alternative payoff specifications	`Supply_Estimation.ipynb`	Cell 25	`\CODES\PART_C\Python`	Displayed as code is run.
Table A.13 – Party objective functions, with interaction between win probability and vote share	`Supply_Estimation.ipynb`	Cell 29	`\CODES\PART_C\Python`	Displayed as code is run.
Table A.14 – Party objective function estimates: constituency-level heterogeneity	`Supply_Estimation.ipynb`	Cell 33	`\CODES\PART_C\Python`	Displayed as code is run.
Figure 1 – Distribution of each party's costs of different types across constituencies	`coefficient magnitudes.do`	Line 45	`\CODES\PART_C`	Saved as `FIGURES/costs.pdf`.
Figure A.1 – Magnitude of voter preference for different candidate characteristics	`elasticity.do`	Line 30	`\CODES\PART_B`	Saved as `FIGURES/elasticity.pdf`.
Figure A.2 – Sensitivity of demand estimates to failures of the moment conditions	`sensitivity_figure.do`	Line 114	`\CODES\PART_B`	Saved as `FIGURES/sensitivity.pdf`.
Figure A.3 – WCSS and Silhouette	`clustering_figure.do`	Line 21	`\CODES\PART_A`	Saved as `FIGURES/cluster_fig.pdf`.

Continued on next page

Table 4 – continued from previous page

Figure/Table	Code File Name	Line / Cell	Folder	Output
Figure A.4 – Distribution of the change in parties' payoffs with a criminal ban	`counterfactuals_figure.do`	Line 85	`\CODES\PART_D`	Saved as `FIGURES/payoffs2.pdf`.
"... we find captures almost 80% of the variation in the data ..."	`ideology_pca.ipynb`	Cell 3	`\CODES\PART_A`	Displayed as code is run.
"... we find that criminals are strategic complements in 84% of the cases."	`strategic complementarity.do`	Line 108	`\CODES\PART_C`	Displayed as code is run.
"... 97% of incumbents are retained by the parties ..."	`Supply_Counterfactuals.ipynb`	Cell 19	`\CODES\PART_D`	Displayed as code is run.

References

Asher, S., T. Lunt, R. Matsuura, and P. Novosad (2021): "Development research at high geographic resolution: an analysis of night-lights, firms, and poverty in India using the SHRUG open data platform," *The World Bank Economic Review*, 35, 845–871.

Association for Democratic Reforms (2019): "MyNeta Candidate Affidavit Database," https://myneta.info.

Bhogale, S., S. Hangal, F. R. Jensenius, M. Kumar, C. Narayan, B. U. Nissa, and G. Verniers (2019): *TCPD Indian Elections Data v1*, Trivedi Centre for Political Data, Ashoka University.

Kitschelt, H. and D. M. Smidt (2013): "Democratic Accountability and Linkages Project (DALP)," https://sites.duke.edu/democracylinkage/.