

README

This README file is divided into two main sections. In Part 1, I provide an overview of the Main.do file executed in Stata. This file generates Figures 1-2 and Tables 1-3 in the main text, along with Figures 1-10 and Tables 1-7 in the Appendix. Part 2 discusses the generation of Appendix Figures 11-15, achieved through a combination of Stata and Jupyter Notebook Python scripts.

Part 1

The `Main.do` file is a Stata script serving as the master file for replicating figures and tables from Sheveleva (2014). It constructs Figures 1-2, Tables 1-3 in the main text, and Figures 1-10, Tables 1-7 in the Appendix.

The Main.do and all the files it calls are available in folder Code-Stata.

Necessary Data Inputs

There are two data inputs.

The first data file is `China.dta`, a sample of exports compiled by General Administration of Customs and is available for purchase from <https://www.epschinadata.com/>.

The subset of the variables that we use includes the following variables:

| Variable Name | Description | Data Type |
|-----------------|---|-----------|
| <code>f</code> | Firm identifier | String |
| <code>d</code> | Destination | String |
| <code>hs</code> | Product code according to the 6-digit Harmonized System | String |
| <code>y</code> | Year | String |
| <code>v</code> | Sales value for firm-product-destination | String |

Second data file is `China_country_names.dta`, converts the `d` variable (destination) from the Chinese country classification to English country names and ISO codes (provided with the package). This file is only needed for Chinese exporters.

The data files should be placed in Data-Stata folder.

Users should ensure their dataset contain the variables above, properly renamed, and formatted according to the above specifications.

Stata Version

This script was developed and tested with Stata 17. Users must ensure they have this version or a compatible version of Stata to successfully execute the script.

Execution Instructions

At the beginning of `Main.do`, global variables define the locations of project components:

| Global Variable | Description |
|----------------------|---|
| <code>drive</code> | The root location of the entire project which should contain folders: <ul style="list-style-type: none">• Code-Stata holding the Main.do and all other relevant do files.• Data-Stata containing a country level exports dataset, such as China.dta. |
| <code>country</code> | Identifies the country for which the analysis is carried out. |
| <code>year</code> | The year of the analysis (e.g.,2003) |

Executing the code will create new directories in the `drive` folder:

- `wd`: a directory where intermediate data files are stored
- `Includes`: a directory where outputs are stored. The includes directory assigned the name Includes-country-year (e.g., Includes-China-2003)

To execute the script type `do Main.do` in the Stata command window.

Outputs

The script generates Figures as pdf files and tables as tex files. As discussed above, the outputs are placed in the includes folder (Includes-China-2003). The tex files will

require manual compiling to be converted into PDFs. This can be done with any Tex editor.

Troubleshooting

Should you encounter issues while running `Main.do`, verify the following:

- The dataset is correctly formatted with the appropriate variable names and types.
- Directory paths in the script are correctly set to match your system.
- Note, the individual do files called by the Main file cannot be run individually.
- Synchronizing files in the `$drive/wd` directory using Dropbox or One-Drive software may lead to write errors.

Part 2

There are 3 Jupyter notebooks that replicate Figures 12-15 in the Appendix. They are located in folder Code-python along with a Stata script BinsAndBalls.

The Jupyter notebooks generate the following Figures:

| Jupyter Notebook | Objects Generated |
|------------------|------------------------|
| AppendixFig11 | Appendix Figure 11 |
| AppendixFig12 | Appendix Figure 12 |
| AppendixFig13-15 | Appendix Figures 13-15 |

Executing AppendixFig11 Notebook:

The notebook "AppendixFig11" replicates Appendix Figures 11 A or B.

To replicate the Figures

- Set the `Fig` variable at the top of the script to "A" for Figure 11 A or "B" for Figure 11 B.
- Note, the code was written and run in Jupiter notebook and may not work correctly in other python environments due to how Jupiter notebooks treat global and local variables.
- The program uses 60 threads to run. Please adjust accordingly for your machine by changing 60 to the machine specific number in line:

```
results = Parallel(n_jobs=60)(delayed(process_pair)(i) for i in pairs)
```

Executing AppendixFig11-AppendixFig15 Notebooks:

Each of the files is run independently in a Jupiter notebook. The code may not be suitable to be run in an alternative IDE due to treatment of some local variables as global only allowed in a Jupiter notebook.

To run the code, create a Code-python, Data-python and Includes-python folders in the same directory. Should you name your folders differently be sure to alter their paths throughout python code. Data-python can be generated by running Stata file BinsAndBalls.do in the Code-python directory. This file can be run independently. Make sure to change the main directory identified as the `$drive` variable.

File AppendixFig11 does not require data inputs and the inputs for AppendixFig12 and AppendixFig13-15 are:

| Jupyter Notebook | Data Inputs |
|------------------|--|
| AppendixFig12 | HS_shares.csv, HS_codes.csv, Balls_US.csv, nprod_data.csv |
| AppendixFig13-15 | HS_shares.csv, HS_codes.csv, Balls_US.csv, RankProductData10.csv |

All the data files needed to execute the AppendixFig12 and AppendixFig13-15 should be placed in Data-python directory.

Explanation for the Data Inputs:

- HS_shares.csv is the tab delimited list of sales values that Chinese exporters sell in each HS code.
- HS_codes.csv is the set of HS codes.
- Balls_US.csv is sales per firm for each exporter to the US.
- nprod_data.csv is the number of products each firm export to the US.

These files are generated using a Stata BinsAndBalls.do file (also included) which in turn uses China.dta file explained in Part 1.

Required Libraries:

- `pandas`: For data manipulation and analysis.
- `numpy`: For numerical operations.

- `matplotlib.pyplot`: For creating visualizations in Python.
- `mpl_toolkits.mplot3d.Axes3D`: For 3D plotting capabilities.
- `plotnine`: For creating complex plots using a grammar of graphics approach.
- `statsmodels.formula.api`: For fitting statistical models using formulas.
- `numpy.random.default_rng`: To create a new instance of the default random number generator.
- `joblib.Parallel` and `joblib.delayed`: For parallel computing, allowing for the execution of functions in parallel across multiple processors.
- `multiprocessing.cpu_count`: To determine the number of CPU cores available on the system.

Support

Replicators seeking assistance or wishing to report issues can contact the package author. (yas0306@gmail.com.)