Replication file for Glynn and Kashin (2018)

Full citation: "Front-Door Versus Back-Door Adjustment With Unmeasured Confounding: Bias Formulas for Front-Door and Hybrid Adjustments With Application to a Job Training Program", Journal of the American Statistical Association, 113:523, 1040-1049, DOI: 10.1080/01621459.2017.1398657.

Last updated: May 10, 2019

Overview

The replication files are in R (run using version 3.5.1) and organized using ProjectTemplate, a project management package for R.

The entire analysis may be run simply by executing the `make.R` script in the main level of the directory. This will carry out data munging and analysis, as well as compile figures and tables seen in the paper. We recommend using RStudio to execute the `make.R` script as RStudio ensures any dependencies for RMarkdown are installed. If you prefer not using RStudio we have included additional instructions in the `make.R` script for execution in `R` which will complete all data munging and analysis but not compile the final report.

Required packages

The versions that were used for this analysis are in parentheses.

- dplyr (0.7.6)
- ggplot2 (3.0.0)
- grid (3.5.1)
- gridExtra (2.3)
- kableExtra (0.9.0)
- knitr (1.20)
- KRLS (0.3-7)
- ProjectTemplate (0.8.2)
- reshape2 (1.4.3)
- readstata13 (0.9.2)
- rmarkdown (1.12)
- R.utils (2.7.0)
- scales (1.0.0)
- stringr (1.3.1)
- xtable (1.8-3)

Additional detail about project structure

Raw data

We include raw data files for the JTPA analysis in the `raw_data` subdirectory. Some of these files were obtained from the Upjohn Institute and others that were obtained in communication with Jeffrey Smith and Petra Todd.

Pre-processing of data

The `src` directory contains files used for the pre-processing of raw data. We provide these files for full transparency of how we assembled the master dataset we work off of for our analyses that are included with this replication.

The cleaned up JTPA data (see `data/jtpa_earn.RData`,`data/jtpa.RData` and `data/data_in.RData`) is constructed using the following scripts:

- `src/construct_jtpa_earnings.R`: script to construct clean earnings data from the JTPA raw data.
- `src/construct_jtpa.R`: script that merges treatment information, compliance information, relevant background characteristics (sex, age, race, site, marriage status) and merges them with income data.
- `src/construct_data_in.R`: script that constructs data_in file from the JTPA files.

The KRLS scripts are also included in the `src` subdirectory so they will not be automatically run by the make.R file; instead, the stored output in the `results` folder will be loaded to run the report.

These files are:

- `run_benchmark_KRLS.R`: benchmark estimates using KRLS method.
- `run_bootstrap_krls.R`: bootstrap estimates using KRLS method.
- `krls`: folder contains all scripts for front-door and back-door estimates across all conditioning sets using KRLS.

Datasets

The cleaned up datasets, are located in the `data` subdirectory:

- `jtpa_earn.RData`: cleaned up earnings data.
- `jtpa.RData`: cleaned up JTPA data.
- `data_in.RData`: cleaned up JTPA data with non-experimental control.

Results

These files are the output of all of the analyses files.

- `benchmark_boot.RData`: KRLS benchmark and bootstraps.
- `bootstrap-arrays.RData`: KRLS front-door and back-door bootstraps.
- `krls-AF` and `krls-AM`: KRLS front-door and back-door estimates. (12 of each)
- `am.results.RData`: all KRLS male combined together
- `af.results.RData`: all KRLS female combined together
- `af.truth.boot.ols`: OLS female benchmark and bootstraps
- `am.truth.boot.ols`: OLS male benchmark and bootstraps
- `af.frontback`: OLS female front-door and back-door estimates
- `am.frontback`: OLS male front-door and back-door estimates

Analysis

The KRLS analysis for the paper was conducted on a remote computing environment. It both takes a long time to run and precise replication is sensitive to the operating system and the version of the KRLS package. For full transparency, we have listed three different options for replication. The first uses our stored results to quickly replicate the exact tables and figures from the paper. The second is the full replication from the cleaned data, which is quite computationally intensive, and relies on exact versions of operating system and the KRLS package in order to obtain the exact numbers in the paper. Finally, in order to demonstrate robustness, provide intuition, and to provide a replication analysis that does not require the use of a remote computing environment, we have included an analysis using OLS. Although this analysis is not in the paper, the results are nearly substantively identical to those presented in the paper.

1. Less computationally intensive (uses stored results)

The analysis of the data is in the `munge` subdirectory and is executed sequentially by ProjectTemplate, which caches the outputs of the analysis in the `cache` subdirectory. This uses bootstrapped estimates that we have have stored in the `results` subdirectory in order to more quickly replicate the findings.

The file run here is:

- `01-combine_krls_results.R`: combines all krls results.

This file utilize various helper files in the `lib` subdirectory to handle estimation.

2. More computationally intensive (requires remote computing environment)

The analysis files to run everything in the paper are in the `src` subdirectory and they are:

- `run_benchmark_KRLS.R`: estimates benchmark using KRLS method.
- `run_bootstrap_krls.R`: estimates bootstraps using KRLS method.
- `krls`: folder contains all scripts for front-door and back-door estimates across all conditioning sets using KRLS.

After this, ProjectTemplate will sequentially execute the analysis in the `munge` subdirectory.

These files utilize various helper files in the `lib` subdirectory to handle estimation. Also, as noted below, the KRLS package is sensitive to the operating system, version of the KRLS package and takes a long time to run. The instructions listed under KRLS replication detail how to set-up an Amazon Web Services (AWS) Elastic Compute Cloud (EC2) to replicate the krls front-door, back-door and bootstraps. KRLS version 0.3-7 is needed to replicate the results.

3. OLS version (not in the paper)

The OLS analysis file is in the `munge` subdirectory and is executed sequentially by ProjectTemplate, which caches the outputs of the analysis in the `cache` subdirectory. The OLS estimates are not in the original paper but are included in the replication report. The OLS analysis is less computationally intensive and can be run quite quickly.

- `02-estimate_OLS.R`: front-door, back-door and benchmark using OLS.

These files utilize various helper files in the `lib` subdirectory to handle estimation.

4. Reports

Reports are automatically compiled from Rmd to pdf files using `knitr` when running the `make.R` file. The Rmd file and the compiled pdf file are available in the `reports` subdirectory.

KRLS Replication

The KRLS front-door, back-door and bootstraps are saved in the `results` folder so that the entire replication file and report may be run quickly. The KRLS package is sensitive to the operating system, version of the KRLS package and takes a long time to run. The following instructions detail how to set-up an Amazon Web Services (AWS) Elastic Compute Cloud (EC2) to replicate the krls front-door, back-door and bootstraps. KRLS version 0.3-7 is needed to replicate the results.

1. Launch Ubuntu Instance

- Type "Ubuntu" in search bar and select the version (18.04)
- Choose an instance type - t2 micro is adequate to run the KRLS scripts
- Configure Instance Details - leave as is
- Add Storage - 20 GB
- Add Tags - Select add another tag. For Key enter "Name" and for Value "Ubuntu-RStudio"
- Configure Security Group
  - Select `Custom TCP` and in Port Range enter "8787" and change `Source` to `Anywhere`. This makes it public to anyone connecting. You will need to create a user with a secure password.
  - If you have a custom IP address select "SSH" and under Source choose `Custom` and enter it here.
- Launch Instance. You will need a Key Pair. If you already have one created select it from the dropdown menu. If not choose `create a new keypair`. Name it and download to your computer. In terminal chmod 400 to change the permissions of the PEM file.
- Select `Launch Instances`, scroll down and go to `View Instances`
- Once the instance is running and a public IP is visible you can continue and ssh into the instance from a terminal application. Selecting the `Connect` button will give instructions on how to ssh.

2. Once you have ssh into the instance run the following commands in

terminal to install R and RStudio:

```
sudo apt-get update
sudo apt-get install r-base
sudo apt-get install build-essential
sudo apt-get install gdebi-core wget <https://download2.rstudio.org/rstudio-server-1.1.456-amd64.deb>
sudo gdebi rstudio-server-1.1.456-amd64.deb
```

- create a username to log-in

```
sudo adduser ruser
```

- Copy the public IP address and paste that into your browser and paste onto the end ":8787"
- An RStudio log-in should appear and use the user and password from above.