Reproducible research in Stata: Managing dependencies and project files

Sergio Correia[1] Matthew P. Seay[1]

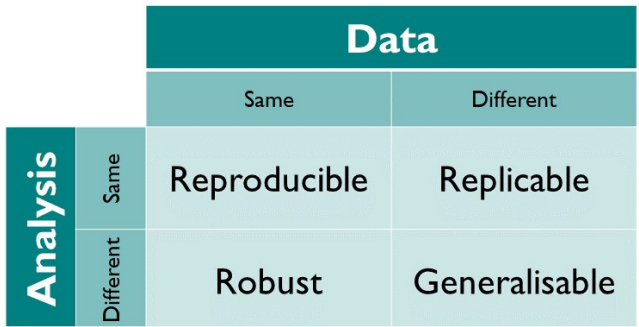
2023 Stata Conference
21 July 2023

[1]Board of Governors of the Federal Reserve System. Views are our own.

Background: what is reproducibility?

- Multiple definitions, but at its core:

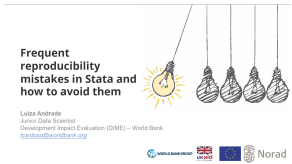
"Given the necessary data and code, can research results be recreated?"



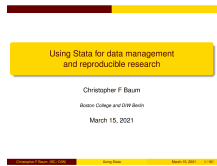
Reproducibility matrix from The Touring Way

Background: why reproducible research?

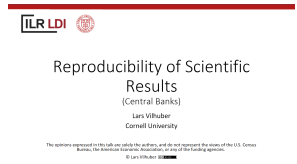
- Can I recreate results on a new computer?
- Can my coauthors recreate them?
- Can the journal's data editor, in a year?
- What about other researchers N years in the future?
- Clearly an important topic for others as well!



(a) *Andrade (2021)*



(b) *Baum (2021)*



(c) *Vilhuber (2021)*

This talk...

- Two aspects of reproducibility (within Stata):
  - Manage dependencies on external user-contributed packages: `require.ado`
  - Access and save files: `setroot.ado`

`require.ado`: motivation

- Stata projects often depend on user-contributed packages
- How can we ensure users are not running outdated/incompatible versions?
- Personal experience:
  1. *Three* different `rdrobust` estimates on Windows, Linux, and coauthor's laptop
  2. Internal policy tool relies on internal packages with frequent releases; users forget to "`ado update`"
  3. How to meet journal reproducibility requirements?

- Is including all the dependencies the only way?

require.ado: motivation

- Missing or incompatible package dependencies behind many reproducibility errors in Sebastian Kranz's "Repbox"

Search Results | Help | About | Replications

200 newest economic articles in data base (total 8475)

1. Evidence and Lessons on the Health Impacts of Public Health Funding from the Fight against HIV/AIDS (5300 MB aer, 2023 Jul)
   fixed effect (33), RDD (13), DID (8), bootstrap (3), nonparametric (2)
   Data: 152 MB, Code: ado 1650 KB, do 180 KB (Data & Code) (README) (Stata reproduction 3%)
2. Confidence, Self-Selection, and Bias in the Aggregate (16 MB aer, 2023 Jul)
   equilibrium (6), Bayesian (3), lab experiment (3)
   Data: 15 MB, Code: r 38.7 KB (Data & Code) (README)
3. The Value of Working Conditions in the United States and the Implications for the Structure of Wages (85 MB aer, 2023 Jul)
   logit (38), bootstrap (5), panel data (4), equilibrium (3), nonparametric (2), probit (2)
   Data: 84.2 MB, Code: do 193 KB (Data & Code) (README) (Stata reproduction 19%)
4. Dividend Taxes and the Allocation of Capital: Comment (0.072 MB aer, 2023 Jul)
   DID (7)
   Data: 0.0092 MB, Code: do 39.3 KB, sas 16.8 KB (Data & Code) (README) (Stata reproduction 6%)
5. Estimating the Optimal Inflation Target from Trends in Relative Prices (7800 MB aejmac, 2023 Jul)
   equilibrium (7), fixed effect (5), time series (2)
   Data: 7.82 GB, Code: m 115 KB (Data & Code) (README)

***Repbox* screenshot**

require.ado: solution

- Most packages have version numbers in their first comment line!

```
. which ivreg2
*! ivreg2 4.1.11 22Nov2019

. which estout
*! version 3.30 25mar2022  Ben Jann

. which reghdfe
*! version 6.12.2 02Nov2021
```

require.ado: solution

- Read package code; extract version numbers and dates
- Users can require a minimum or exact version/date; optionally install it

```
. require ivreg2 >= 4.1

. sreturn list
macros:
            s(package) : "ivreg2"
            s(version) : "4.1.11"
      s(version_major) : "4"
      s(version_minor) : "1"
      s(version_patch) : "11"
       s(version_date) : "22nov2019"
```

require.ado: syntax

```
require <package> == <version> , [options]
require <package> >= <version> , [options]
```

Examples:

```
require reghdfe
require reghdfe >= 6
require reghdfe == 6.0.3
require reghdfe >= 6, install
```

require.ado: advanced usage

Inspired on Python's requirement.txt:

mydofile.do

```
clear all
require using requirements.txt
...
```

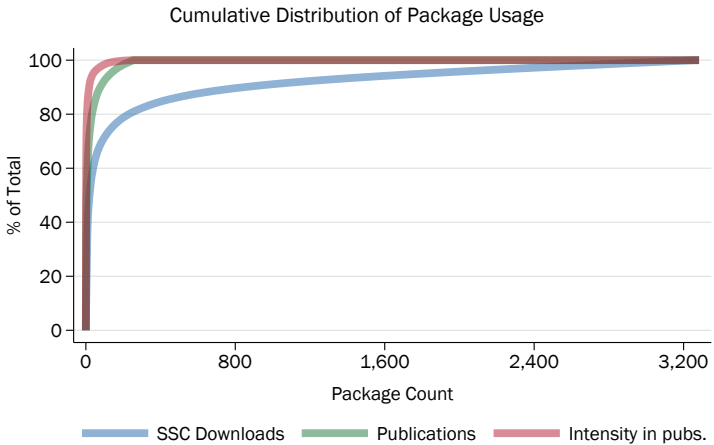
requirements.txt

```
require   >= 0.9.4
winsor2   >= 1.1
estout    >= 3.23
```

`require.ado`: usefulness

- To be useful it needs to support all packages than a researcher might use
  $\rightarrow$ It needs to deal with the long tail

The long tail: user-contributed packages seem to follow a power law

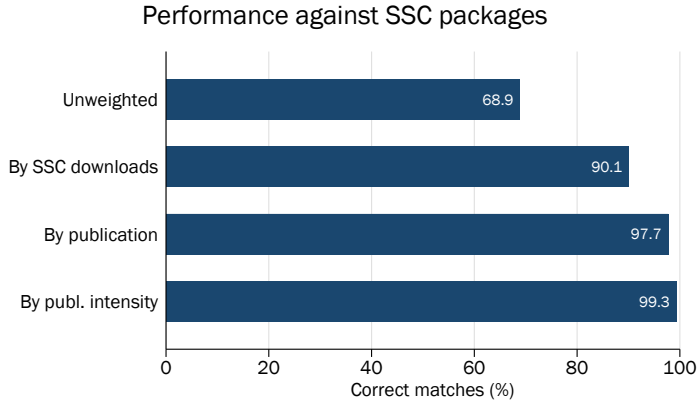


`require.ado`: solution strategy

- Test-driven development
  - Download the universe of SSC packages (plus Github, etc.)
  - Construct ground truth of version and dates
  - Validate against ground truth!
- Inside require
  - Lots of regular expressions (to deal with all version and date variants)
  - Ad-hoc exceptions (Mata, graphic schemes, etc.)

require.ado: performance

- How to measure performance?



Publication data based on analysis of journal replication files by Kranz (2023)

`require.ado`: missing pieces

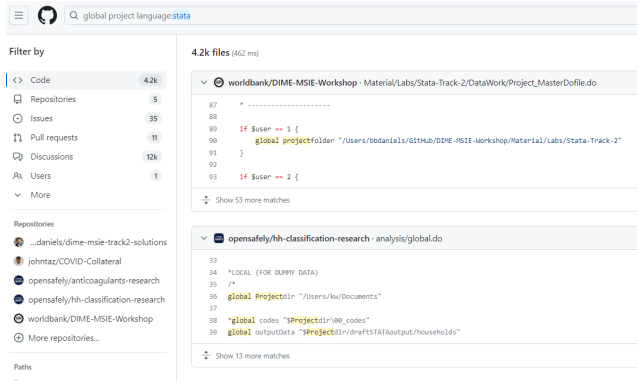
- To install an older version we need to store it somewhere
  - Feasible on Github through "releases" and commit history
  - For SSC, see SSC-Mirror by Lars Vilhuber
  - Also an issue in other software tools (`GRAN` and `groundhog` in R)
- Q: should we encourage minimum or exact versions? (`==1.0` vs. `>=1.0`)
  - Exact version ensures maximum reproducibility
  - But might be missing bugfixes, speedups, etc.
- Q: How to bootstrap the package? What if `require` is not installed?

setroot.ado: motivation

- Deals with accessing files (data, do-files, output) within a project
- Alternative to:

```
global data "C:/Dropbox/Sergio/mypaper/data"
use "$data/responses.dta"
```

- This is a **very** common problem



setroot.ado: solution

- Start on working directory
- Navigate upwards to detect the root folder of a project
  - Detects root folder based on `.git`, `README.md`, etc.
- Store the root path in global variable `$root`

```
. setroot // simplest, store in $root
. setroot, local // store in `root' instead
```

- Inspired on the R and Stata **`here`** packages

Putting it all together...

```
* Header
  version 18
  clear all
  ...
  setroot, more
  require gtools >= 1.7.5
  require rdrobust >= 3.2.1

* Analysis
  use "$root/data/..."
  rdrobust ...
```

Thank you!