Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

My favorite overlooked life savers in Stata

Jan Ditzen

Free University of Bozen-Bolzano, Bozen, Italy

www.jan.ditzen.net, jan.ditzen@unibz.it

https://janditzen.github.io/

September 12, 2024

Motivation I

Not so much econometrics...

- Empirical work often relies on small programs which are overlooked.
- This presentation puts the spotlight on my personal favourites:
  1. `adotools`
  2. `psimulate2`
  3. `xttools`

adotools[1]

- Community contributed or custom made ado files can be located in folders and added using `adopath + ``path''`.
- Many users are often not aware which folders are active.
- Adding those paths by hand is often time consuming.
- `adotools` helps with 3 functions:
  - Remove all user specified ado paths
  - Create list of custom ado paths
  - Add and remove ado paths using keys

[1] Work in progress, will be released soon.

adodefine

- adodefine creates a list of paths of ado folders and associated keys.
- List is saved as a dta file in the folder adotools is located.
- Syntax:

```
adodefine key , [path() remove list clear ]
```

- adodefine *key* , path() adds key with path to list
- remove removes key-path from list
- list displays all key folder
- clear deletes list

Example adodefine

```
. adodefine, list
ado list

. adodefine simulate2, path("D:\Other computers\Laptop (Main)\StataCode\simulate2\ado")
Add entry with name: simulate2 and path D:\Other computers\Laptop (Main)\StataCode\simulate2\ado

. adodefine xtdcce2, path("D:\Other computers\Laptop (Main)\StataCode\xtdcce2\working\")
Add entry with name: xtdcce2 and path D:\Other computers\Laptop (Main)\StataCode\xtdcce2\working\

. adodefine, list
ado list

       id        name                                                          path
  1.    1   simulate2      D:\Other computers\Laptop (Main)\StataCode\simulate2\ado
  2.    2     xtdcce2   D:\Other computers\Laptop (Main)\StataCode\xtdcce2\working\

. adodefine xtdcce2, remove

. adodefine, list
ado list

       id        name                                                       path
  1.    1   simulate2   D:\Other computers\Laptop (Main)\StataCode\simulate2\ado
```

adotools

adoadd and adoclear

- After defining the path list, paths can be added and removed using the key.
- Adding and removing path to ado path:

addoadd *key*

adoclear [*key*] , [clear all]

- Option clear implies clear all.
- adoclear also removes any ado programs in the folder.

adotools

Example adoadd and adoclear

```
. adopath
  [1]  (BASE)      "C:\Program Files\Stata18\ado\base/"
  [2]  (SITE)      "C:\Program Files\Stata18\ado\site/"
  [3]              "."
  [4]  (PERSONAL)  "C:\Users\JDitzen\ado\personal/"
  [5]  (PLUS)      "C:\Users\JDitzen\ado\plus/"
  [6]  (OLDPLACE)  "c:\ado/"

. adodefine, list
ado list

       id       name                                                          path
  1.    1  simulate2      D:\Other computers\Laptop (Main)\StataCode\simulate2\ado
  2.    2    xtdcce2   D:\Other computers\Laptop (Main)\StataCode\xtdcce2\working\

. adoadd xtdcce2
Adding adopath D:\Other computers\Laptop (Main)\StataCode\xtdcce2\working\

. adopath
  [1]  (BASE)      "C:\Program Files\Stata18\ado\base/"
  [2]  (SITE)      "C:\Program Files\Stata18\ado\site/"
  [3]              "."
  [4]  (PERSONAL)  "C:\Users\JDitzen\ado\personal/"
  [5]  (PLUS)      "C:\Users\JDitzen\ado\plus/"
  [6]  (OLDPLACE)  "c:\ado/"
  [7]              "D:\Other computers\Laptop (Main)\StataCode\xtdcce2\working\"

. adoclear xtdcce2
Path D:\Other computers\Laptop (Main)\StataCode\xtdcce2\working\ removed.
```

psimulate I

parallel version of simulate

- (Monte Carlo) simulations are popular in econometrics, statistics, medical sciences, ...
- Often they require many repetitions over many parametrisations.
- Stata offers the `simulate` command, which allows for easy simulations.
- However, simulations are not in parallel.
- Two ways of paralleising tasks: parallelise part of command or entire tasks.

psimulate II

parallel version of simulate

- psimulate2 was inspired by a discussion at the 2019 UK Stata conference and multishell.
- psimulate2 splits the number of repetitions into chunks and runs the simulations via shell.
- Parent instance runs in current Stata session and displays progress, child instances are doing the simulation.
- Advantages:
  - Supports Windows, macOS, Unix.
  - Save and load repetition specific seeds.
  - Can be combined with Stata MP.
  - Works with frames.
  - Strings can be returned via simulate2

psimulate III

parallel version of simulate

- Simple syntax:

```
psimulate2 [exp_list] , reps(#1)  parallel(#2,  options1)
                 [options2] :  command
```

- #1 controls the number of repetitions and #2 number of parallel instances.
- options1 sets path to exe file, CPUs for Stata MP
- options2 controls seed, output and further behaviour

psimulate

Example

- Example: show unbiasedness of the OLS estimator using a simulation with 10,000 repetitions and 1000 observations.
- Data is generated as

$$y_i = \beta_0 + x_i\beta_1 + e_i, \;\; x_i \sim U(0,1), e_i \sim N(0,1)$$
$$\beta_0 = 1, \;\; \beta_1 = 2$$

- We would program:

```
. clear all

. program define mc_test, rclass
  1.         syntax anything, nobs(integer)
  2.         tokenize `anything´
  3.         local b0 `1´
  4.         local b1 `2´
  5.         clear
  6.         set obs `nobs´
  7.         gen x = runiform()
  8.         gen e = rnormal()
  9.         gen y = `b0´ + `b1´ * x + e
 10.         reg y x
 11.         return scalar b1 = _b[x]
 12. end
```

And then use:

`simulate b1_sim = r(b1), reps(10000) nodots: mc_test 1 2, nobs(1000)`

to run the simulation

psimulate

Example - parallel

- To run the simulation in parallel:

  ```
  psimulate2 b1_psim = r(b1), reps(10000) p(4) :  mc_test 1 2,
  nobs(1000)
  ```

- *p*(4) sets 4 parallel instances.
- We do not need to set a seed or a seed stream. `psimulate` uses the current seed. Seedstreams are set for each parallel instance automatically.
- In the background parent instance creates do files and starts child instances.
- Results and progress information saved in temporary folder, can also be specified.
- If more than one parent instance is running, use option `globalid()`.

psimulate

Example - simulations in parallel - Output

Output

```
psimulate2 - parallelise simulate2
command: mc_test 1 2, nobs(1000)
Timings (hour, minute, sec):          Estimated:
  Average Run:         00:00:00.004     Time left (min):   00:00:46
  Time Elapsed:        00:00:46         finishing time:    10:28:36
Instance 1:
  Done     100.00%  (2500/2500)
Instance 2:
  Done     100.00%  (2500/2500)
Instance 3:
  Done     100.00%  (2500/2500)
Instance 4:
  Done     100.00%  (2500/2500)
Total
  Done     100.00%  (10000/10000)
Current Time: 10:27:51 - next refresh in 00.45 sec.
No seed set. If psimulate is used in a loop,
all iterations of the loop will have the Stata default seed.
Click on link to open log file:
Log files
    Instance 1:     Log File
    Instance 2:     Log File
    Instance 3:     Log File
    Instance 4:     Log File
. timer off 2
```

Timings

```
. timer list
   1:      86.73 /         1 =       86.7310
   2:      47.96 /         1 =       47.9610
```

xttools[2]

- Stata comes with some very handy tools to anaylse and process panel data.
- However, some functions from `ts` are missing.
- `xttools` consists of three programs:
  - `xtgetpca` obtain principal components from panel data.
  - `xtcorri` calculate unit specific correlations or covariances.
  - `xtplot2` identify panel structure for variables

[2] Work in progress

xttools

xtgetpca - PCs in panel data

- pca extracts principal components (PC) from a variable list.
- Not possible with repeated samples (panel data).
- xtgetpca internally reshapes the data into wide format, extracts the PCs and then adds them to the using dataset.
- Allows standardisation with respect to overall, unit and time dimension.
- Can predict score, fit, residual and q from pca.
- Wide dataset can be copied into frame.
- Requires balanced panel.

xttools

xtgetpca - PCs in panel data

Syntax:

```
xtgetpca [varlist] , num(real) [name(string) standardize(string)
frame(string) predict(string) covariance correlations output]
```

- `num()` defines number of PCs
- `name()` prefix for new variable, default PCA_
- `stand()` standardizes data
- `frame()` copies wide data to frame
- `predict()` adds prediction to data, default is score
- `output` display output from pca

Example

xttools I

xtcorri - Unit specific correlations

- When working with panel data, often correlation between variables on unit level are of interest, i.e. $corr(X_i, Y_i)$
- `by id, sort: corr variables` does the job, but contains too much information and output not clear.
- `xtcorri var1 varlist` calculates the correlation on a unit level between *var1* and variables specified in *varlist*.
- Also calculates overall correlation and correlation between cross-section averages.

xttools II

xtcorri - Unit specific correlations

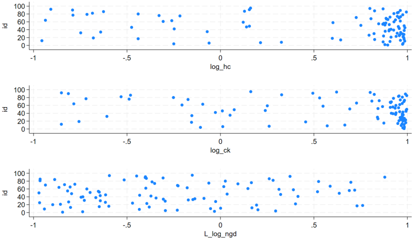
- Syntax:

```
xtcorri var1 [varlist] [if], [ivar(varname) cov fmt(string)
plotoptions(string) nocsa ycsa ]
```

- ivar() specifies unit name
- cov calculate covariance rather than correlation
- fmt set format
- plot plot scatterplot
- nocsa do not calculate cross-section averages
- ycsa convert *var1* to cross-section averages

Example - xtcorri

```
. xtcorri log_rgdpo log_hc log_ck L.log_ngd, ivar(country) plot combine(rows(3))
Correlations with log_rgdpo on unit level
             |   log_hc     log_ck  L.log_ngd
   Country_1 | .8948912   .9824009  -.8446966
   Country_2 | .8798358    .979756  -.7372838
   Country_3 | .9743094   .9754843  -.0320244
  ...
  Country_93 | .8699171   .8010562  -.4484579
       Total | .7993354   .6211605  -.5942171
         CSA | .1608569   .1620696    -.15088
```



xttools

xtplot2

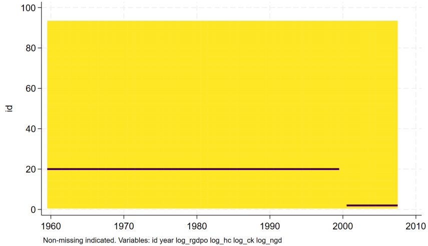
- First stage of any empirical project should be data investigation.
- Identifying outlier and missing values in large datasets difficult.
- `xtplot2` is inspired by `panelview`, but quicker to handle.
- It is essentially a wrapper for Ben Jann's `heatplot`.
- Syntax:

```
xtplot2 [varlist] [if], [values combineopt(string)
ivar(varlist) title seperate ]
```

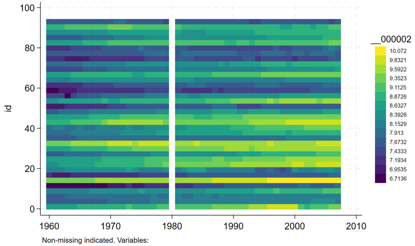
- `values` plots values rather than indicating missing values.
- `combine()` passes options to graph combine.
- `ivar()` specifies unit name.
- `title` uses variable label as title
- `seperate` plot each variable separate.

Example - `xtplot2`

`xtplot2`



`xtplot2 log_rgdpo, values`



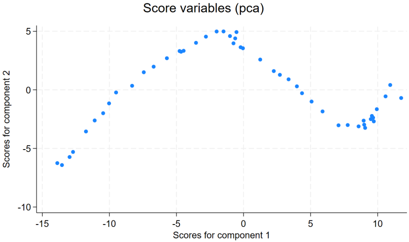
Conclusion

- I presented my personal life savers, small programs which are often overlooked.
- They helped me to
  1. control the ado directories
  2. run simulations fast
  3. investigate panel data
- `psimulate2` is already available on SSC and GitHub.
- `adotools` and `xttools` will be available soon.

Example - xtgetpca

back

```
. use "xtdcce2_sample_dataset.dta" , clear

. xtgetpca log_rgdpo, frame(test) num(2)
Dataset used for cacluation of PCs copied to frame test
Variables PCA_1 PCA_2 containing score(s) added to dataset.

. return list

scalars:
                r(rho) =  .7894428507822338

macros:
          r(predict) : "score"
            r(frame) : "test"

. frame test: scoreplot
```


Score variables (pca)
Scores for component 2
Scores for component 1