MAX-PLANCK-INSTITUT FÜR DEMOGRAFISCHE FORSCHUNG

MAX PLANCK INSTITUTE FOR DEMOGRAPHIC RESEARCH

Data Structures in Stata

Daniel C. Schneider

German Stata Conference, Mannheim, June 7, 2024

Introduction

Stata Forum user:
"I have a lot of programming experience with C and R,
so Stata in that regard seems a bit clumsy."

My own take:
"Compared to Python, Stata in some respects can be a bit clumsy
(and in some other respects it shines)."

Introduction

- Stata commands can be great (achieve much with little code), but custom operations can be very tedious
- Those things oftentimes can be done well in Mata
- But Mata can't do many Stata things
  so there is often a lot of back-and-forth
- In general, Stata has many different parts to be learned
- This is also reflected in its data structures

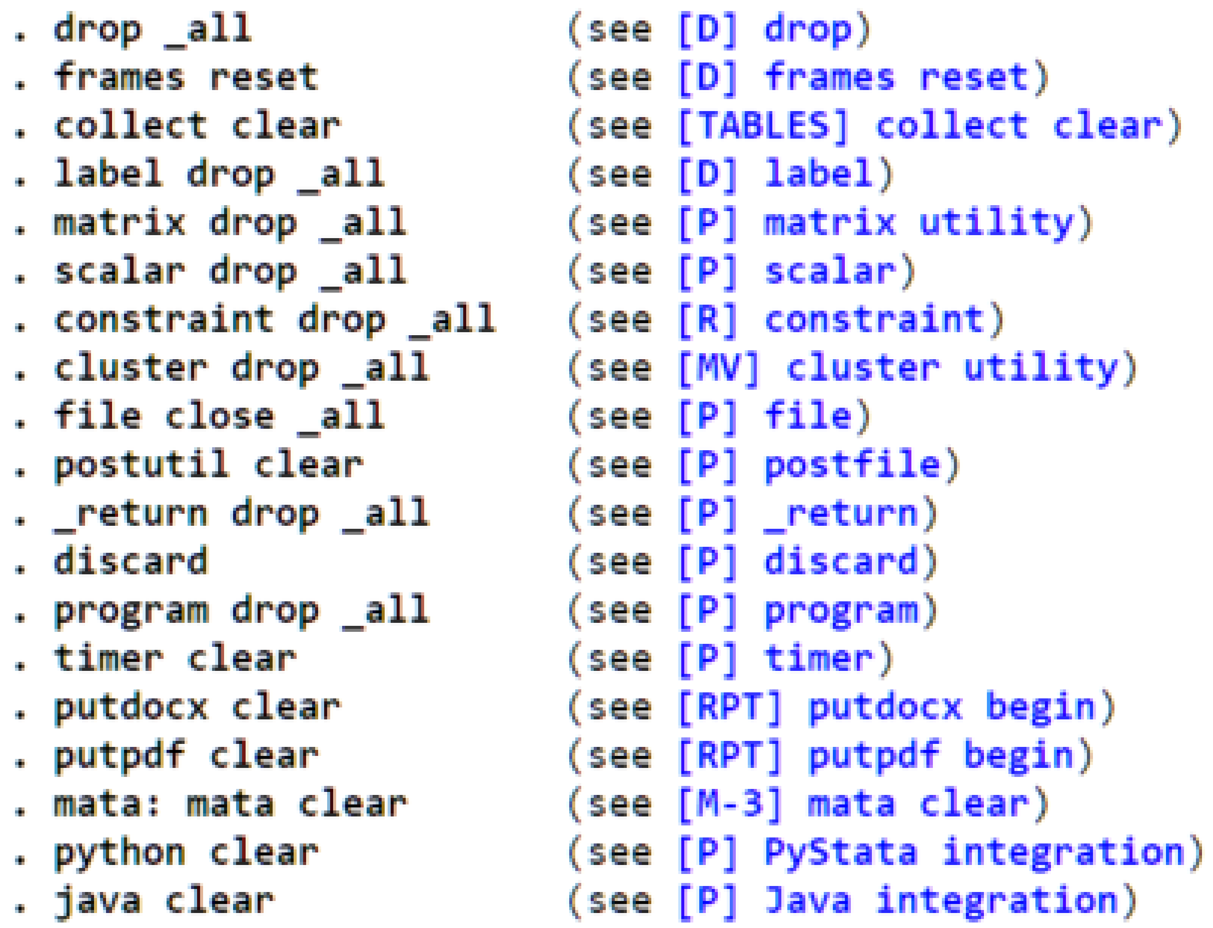
Introduction

- Can one change/extend/combine Stata's data structures in a better way?
- Real long-term planning would address the Stata-Mata divide and where Mata is going fundamentally
- This presentation: medium-term suggestions
- Surprisingly little discussion about these issues in the Stata community

Stata's data structures

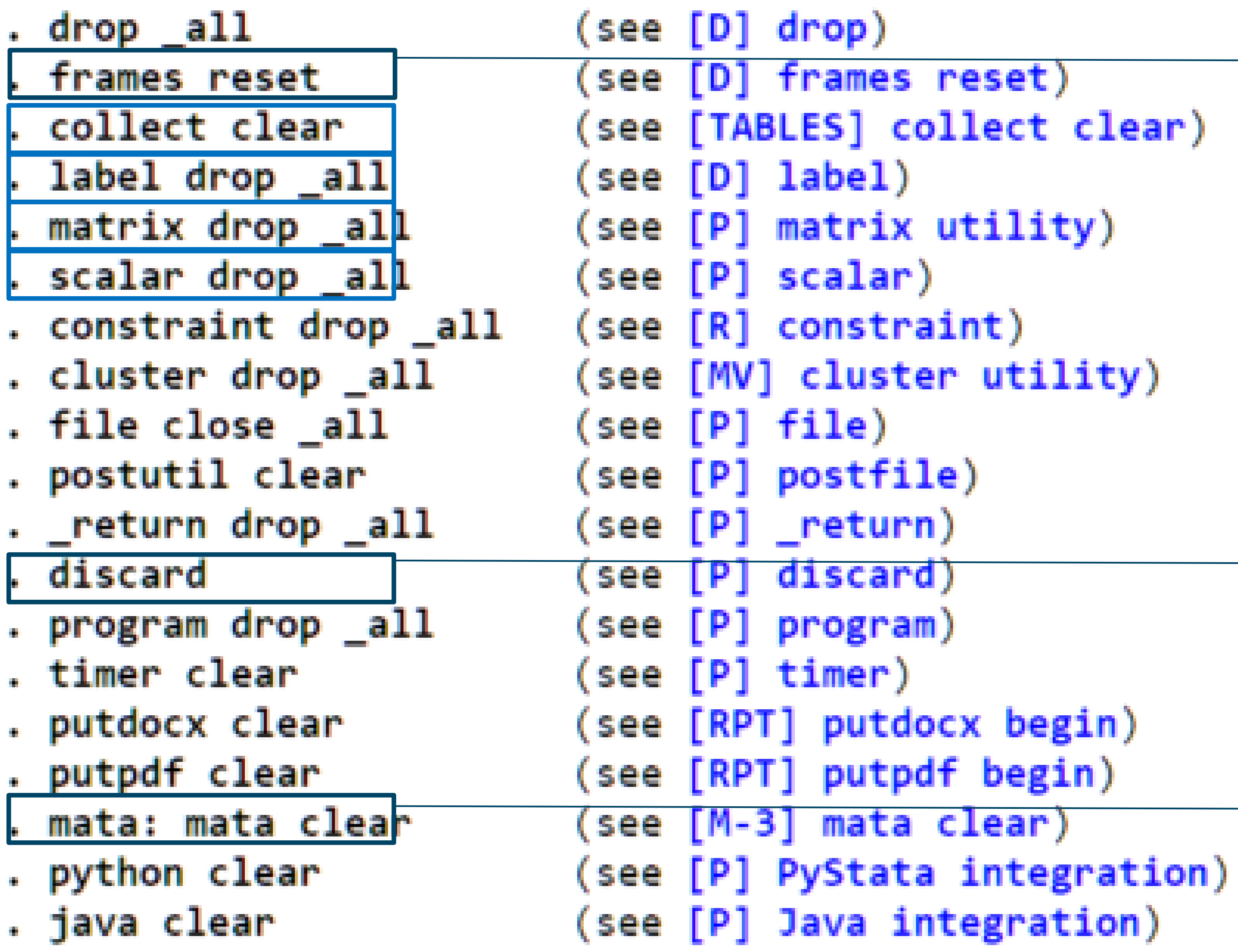
See `help clear`:

`. clear all` is equivalent to:



```
. drop _all                 (see [D] drop)
. frames reset              (see [D] frames reset)
. collect clear             (see [TABLES] collect clear)
. label drop _all           (see [D] label)
. matrix drop _all          (see [P] matrix utility)
. scalar drop _all          (see [P] scalar)
. constraint drop _all      (see [R] constraint)
. cluster drop _all         (see [MV] cluster utility)
. file close _all           (see [P] file)
. postutil clear            (see [P] postfile)
. _return drop _all         (see [P] _return)
. discard                   (see [P] discard)
. program drop _all         (see [P] program)
. timer clear               (see [P] timer)
. putdocx clear             (see [RPT] putdocx begin)
. putpdf clear              (see [RPT] putpdf begin)
. mata: mata clear          (see [M-3] mata clear)
. python clear              (see [P] PyStata integration)
. java clear                (see [P] Java integration)
```

Stata's data structures



```
. drop _all                (see [D] drop)
. frames reset             (see [D] frames reset)
. collect clear            (see [TABLES] collect clear)
. label drop _all          (see [D] label)
. matrix drop _all         (see [P] matrix utility)
. scalar drop _all         (see [P] scalar)
. constraint drop _all     (see [R] constraint)
. cluster drop _all        (see [MV] cluster utility)
. file close _all          (see [P] file)
. postutil clear           (see [P] postfile)
. _return drop _all        (see [P] _return)
. discard                  (see [P] discard)
. program drop _all        (see [P] program)
. timer clear              (see [P] timer)
. putdocx clear            (see [RPT] putdocx begin)
. putpdf clear             (see [RPT] putpdf begin)
. mata: mata clear         (see [M-3] mata clear)
. python clear             (see [P] PyStata integration)
. java clear               (see [P] Java integration)
```

frames / datasets contain:
variable and data set labels
variable and data set characteristics
display format information

linked frames and alias variables

drop ops include:
e(), r(), s()
graphs
classes (can contain arrays)

numeric/string/pointer matrices
objects (official Stata or user-defined)

not in the list:
macros!

Preview of Suggestions

- Improve Stata matrices
- New table-like object (here called "tabular") to complement collections
- Improve conversions:
  Dataset/frame <> collection <> tabular <> Stata matrices
- Introduce a new list data structure
  - Many use cases, see below

Suggestion: Stata Matrices and Mata

- New command/prefix : Mata environment but with immediate access to Stata matrices

```
. matmata A[4::8,.] = inv(B)[7::rows(B)] * r(C)' * mata(D)
```

- Introduce Stata string matrices ("smatrix") to improve string processing?

```
. smatmata A = colshape(tokens("`mylocal'"), 3)'
```

- Problem of cumulating matrices
  not possible with `collapse`, `statsby`

Before `collect`

- How I envisioned tables in Stata before `collect`:
  - Create new `table` object that can be populated and that can format and export information
  - **All tabular Stata output** should be generated by populating such an object and then calling a table.display() method
  - **Every output table** should be returned in new r()/e() return type `table`

Along Came Collections...

- Ingenious implementation, game changer for tabular output production
- `collect` wish list:
  - Make faster
  - Show contents in tag form as data set
    (=> user-written `collect_to_frame`, Daniel Alves Fernandes, GitHub)
  - Generally allow string cells
  - Make work with tempnames
  - Make data accessible, e.g., conversion ("export") to Stata data set
  - Make formatting/layout easier, but how?
    I propose conversion to a new object `tabular` instead

Suggestion: New `tabular` Object

- Primary purpose: "freeze" a collection layout, apply final touches, then save or export
- Why needed:
  - Final formatting / polishing
  - Much easier to use than collections
  - Set values
- Data type by cell
- Conversion from collection: `. collect convert tabular ...`
- Conversion from dataset: `. mktabular` *`varlist`* `...`
- new r()/e() return types `tabular` and `collection`

Suggestion: Easy Conversions

Conversions among datasets <> collections <> tabular <> matrices

- Datasets <> matrices
  - existing `mkmat`, `svmat`; problem: matrix labels
  - Some benefits: save matrices as dta; view in data browser
- Datasets <> collections
  - Allows math ops on or plain setting of collection items:
    first convert collection=>dataset, calculate and set items, then convert back)
  - Conversely, collections can be used to do data set calculations:
    create collection (e.g., using `table`), then convert back to dataset

Suggestion: Easy Conversions

Conversions among datasets <> collections <> tabular <> matrices

- Collections <> tabular
  - (was discussed already)
- Matrices <> collections
  - A lot of old code use matrices for tabular display
  - Conversion matrix=>collection done via `collect`
    Would be nice to be able to collect matrices directly, without storing in r()
  - Conversely: New command `collect convert matrix` ?

Implementing Conversions: Changes to Dataset

- One or more levels of supercolumns (supervariables) and superrows)
- Implement as labels only?

```
. label rows 1-10 "2001-2010" 11-20 "2011-2020"
. label columns mpg-rep78 "Covariate set A" ///
      headroom-length "Covariate set B"
. label columns mpg-length "All covariates" , level(3)
```

- If supercols/rows have more meaning, this would be more difficult:

```
. supercol generate set_A = mpg-rep78
. supercol label set_A "Covariate set A"
. regress price [set_A]
```

- Get inspiration from axis indexes in Python pandas dataframes?

Implementing Conversions: Changes to Dataset

- New mixed (num/str) variable type
  `syntax varlist(mixed)`
- Maybe not allow mixed data types in expressions? Just in `generate/replace` with numeric / string literals?

```
. tabular export frame , name(myframe, replace)
. frame myframe            // all variables are potentially of mixed type
. gen double tmp = mycol // generates missings for string values of mycol
. replace tmp = tmp^2     // some transform
. replace mycol = tmp if !missing(tmp)
. drop tmp
. frame export tabular , name(mytab, replace)
. tabular mytab
. tabular export excel using [...]
```

Suggestion: New `list` Object in Stata

- list: like an array, but can contain anything
- Here: matrix, dataset / frame, scalar, local, collection?, graph?, ...
  What about Mata matrices/objects?
- Implementation suggestions:
  - Allow nesting: list of lists
  - Entries are named, index by name or numerically
  - Can be saved in a .stlist file

Suggestion: New `list` Object in Stata

- Use cases:
  - Saving the "workspace" (ex Mata)
    Saving a set of results in one place (estimates+graphs+tables+...)
  - Accumulating matrices
  - Storing / saving sets of estimation results
    A single set of e()-results can be stored in a list
    Then a list of lists can be built up
  - New r()/e() return type `list`
  - … and probably many more

Suggestion: New `list` Object in Stata

Fantasy syntax:

```
. stlist [define] L = []
. stlist [define] L[1] = matrix(A)
. matrix Ainv = invsym(L[1])
. stlist [define] L[2]      = frame(auto)
. stlist [define] L[3..10] = frame(_all)
. stlist [define] L[8..10] = estim(_all), replace
. stlist K = L[3..7]  // 5 different frames
. stlist [define] L[9] = stlist(K)   // nesting
. frame copy L[K[1]] newframe
. stlist restoreelems L[K[2]] , as(newframe, replace)
```

```
. stlist describe
. stlist save L using myfile
    // saves myfile.stlist
. clear all
. stlist use myfile.stlist
. stlist restoreelems L , replace
. stlist M = [] , holds(estimates)
    // list restricted to
    // certain elems
```

Summary of Suggestions

- Improve Stata matrices
- New table-like object (here called "tabular") to complement collections
- Improve conversions:
  Dataset/frame <> collection <> tabular <> Stata matrices
- Introduce a new list data structure

Thank you

schneider@demogr.mpg.de

Suggestion: Stata Matrices and Mata

- New command/prefix : Mata environment but with immediate access to Stata matrices

```
. matmata A[4::8,.] = inv(B)[7::rows(B)] * r(C)' * mata(D)
```

- Currently would be something like

```
. mata : st_replacematrix(A , (st_matrix(A)[1::3,.] /
    inv(st_matrix(B))[7::rows(st_matrix(B))] *
    st_matrix("r(C)")' * D))
```

- What about row/col labels?
- Merge indexing features of both types of matrices?

Suggestion: Stata Matrices and Mata

- Introduce Stata string matrices ("smatrix") to improve string processing?

```
. smatrix A = J(3,3,"abc")
. smatrix A = J(20000, 1, "") // preallocate
```

- Command analogous to -matmata- for Mata-like syntax for string matrices

```
. smatmata A = colshape(tokens("`mylocal'"), 3)'
```

- Note: problem of cumulating matrices: not possible with `collapse`, `statsby`