Efficient Commands for Data Visualization in Large Datasets

presentation for Oceania Stata Conference

Jan Kabatek
The University of Melbourne, CentER, IZA, LCC, Netspar

February 10, 2022

The environmental imperative of efficiency in popular software packages I.

- **Have you ever wondered about the environmental costs of your work?**
  - We spend most of our working days analyzing large quantities of data, which requires **lots of computing power**
  - Computing power translates into electricity consumption and that translates into **carbon footprint**
  - But how large is this footprint? And what can we do about it?

The environmental imperative of efficiency in popular software packages II.

Carbon footprint of computing

According to Stevens *et al.* (2020), the carbon footprint of computing among Australian astronomers is $\sim$ **22 tCO_2e/yr** per researcher.
This is the **single highest contributing factor** to their net carbon emissions.

- In comparison, the average American household produces $\sim$ **7.5 tCO_2e/yr**
- Carbon emissions of Stata users are bound to differ from those of astronomers, although it is not clear whether they would be much lower.
- Plus, factoring in the much-larger user base of Stata researchers/analysts, we are definitely in the zone of non-trivial emissions.

The environmental imperative of efficiency in popular software packages III.

- **Taking all that into account, we should be putting considerable effort into making our software packages efficient!**

  (but by all means, get that compost bin, too...)

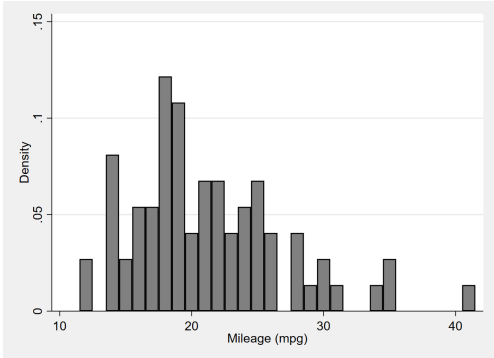
- This presentation is a conversation starter:

  1. It aims to highlight the environmental imperative of efficiency in statistical computing
  2. And, through my "**PLOT**" family of graphing commands, it points out one of the **most straightforward efficiency improvements** that can be achieved in (native) Stata

Illustrative example I.

Stata code I.

```
sysuse auto
hist mpg
```

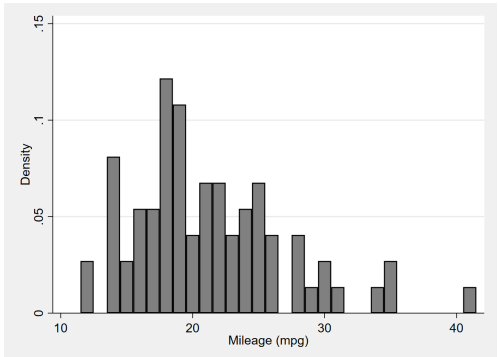


Illustrative example I.

Stata code I.

```
sysuse auto
hist mpg
```

Command 'hist' takes 0.8 sec. to run



Illustrative example I.

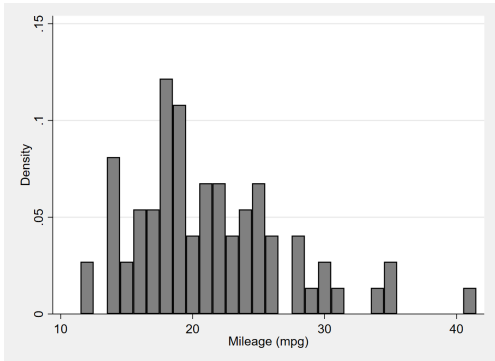
Stata code I.

```
sysuse auto
hist mpg
```

Command 'hist' takes 0.8 sec. to run

Stata code II.

```
sysuse auto
expand 4000000 //4 mil.
hist mpg
```



Illustrative example I.

Stata code I.

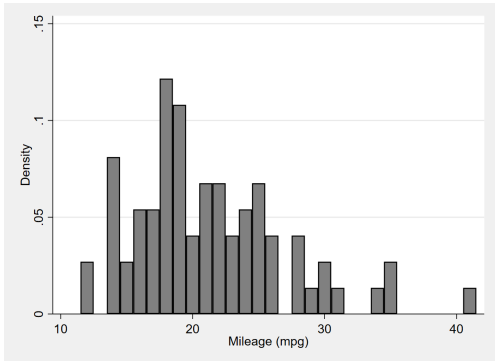
```
sysuse auto
hist mpg
```

Command 'hist' takes 0.8 sec. to run

Stata code II.

```
sysuse auto
expand 4000000 //4 mil.
hist mpg
```

Now, 'hist' takes 31 mins to run!



Illustrative example II.

- Longer execution time of the second exercise is **not** fully attributable to the larger number of observations requiring more time to put into histogram bins!
- Rather, it relates to one of the **most pernicious bottlenecks** of native Stata commands

Illustrative example II.

- Longer execution time of the second exercise is **not** fully attributable to the larger number of observations requiring more time to put into histogram bins!
- Rather, it relates to one of the **most pernicious bottlenecks** of native Stata commands
- Command 'hist' does the following:

Illustrative example II.

- Longer execution time of the second exercise is **not** fully attributable to the larger number of observations requiring more time to put into histogram bins!
- Rather, it relates to one of the **most pernicious bottlenecks** of native Stata commands
- Command 'hist' does the following:
  1. preserves the original dataset

Illustrative example II.

- Longer execution time of the second exercise is **not** fully attributable to the larger number of observations requiring more time to put into histogram bins!
- Rather, it relates to one of the **most pernicious bottlenecks** of native Stata commands
- Command 'hist' does the following:
  1. preserves the original dataset
  2. calculates the histogram bins

Illustrative example II.

- Longer execution time of the second exercise is **not** fully attributable to the larger number of observations requiring more time to put into histogram bins!
- Rather, it relates to one of the **most pernicious bottlenecks** of native Stata commands
- Command 'hist' does the following:
  1. preserves the original dataset
  2. calculates the histogram bins
  3. stores them as a temporary dataset

Illustrative example II.

- Longer execution time of the second exercise is **not** fully attributable to the larger number of observations requiring more time to put into histogram bins!
- Rather, it relates to one of the **most pernicious bottlenecks** of native Stata commands
- Command 'hist' does the following:
  1. preserves the original dataset
  2. calculates the histogram bins
  3. stores them as a temporary dataset
  4. displays the corresponding graph

Illustrative example II.

- Longer execution time of the second exercise is **not** fully attributable to the larger number of observations requiring more time to put into histogram bins!
- Rather, it relates to one of the **most pernicious bottlenecks** of native Stata commands
- Command 'hist' does the following:
  1. preserves the original dataset
  2. calculates the histogram bins
  3. stores them as a temporary dataset
  4. displays the corresponding graph
  5. and restores the original dataset

Legacy bottlenecks

- But: **preserving** and **restoring** big datasets takes ages...
- ...and, as of Stata 16, it appears to be wholly redundant!

- Instead, we can leverage the new environment of **data frames**. We can store the histogram data as a **separate data frame**, thus holding **both** the original and temporary data in memory (no restoring required)
- And this simple adjustment is what makes the "**PLOT**" commands super fast

PLOT family of commands

- Downloadable at github.com/jankabatek/statapack

- Commands: **PLOTTABS, PLOTMEANS, PLOTAREA, & PLOTB**
- Aside from the speed improvements, they also operationalize some features I was missing in the standard visualization commands

    - Critically, the commands store the plotted data in a dedicated frame
    - This data frame can store multiple plots at once, thereby enabling visualizations of complex data systems.

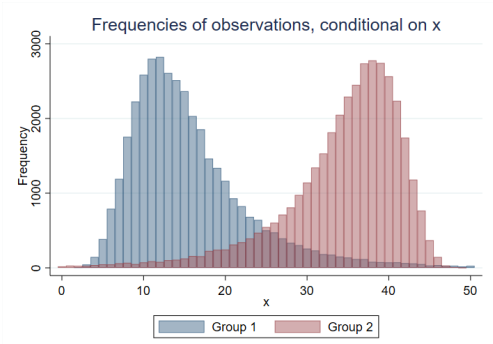
PLOTTABS: one-way and two-way frequency plots

Stata code

```
webuse plotdata, clear
PLOTTABS if gr==1, over(x1) clear
PLOTTABS if gr==2, over(x1) gr(bar)
```

- this example is equivalent to a chart combining two histograms with discrete bins and option '*freq*'

- **PLOTTABS** can also plot conditional rates, similar to the output of `tabulate twoway, row nofreq`



PLOTMEANS: conditional mean plots

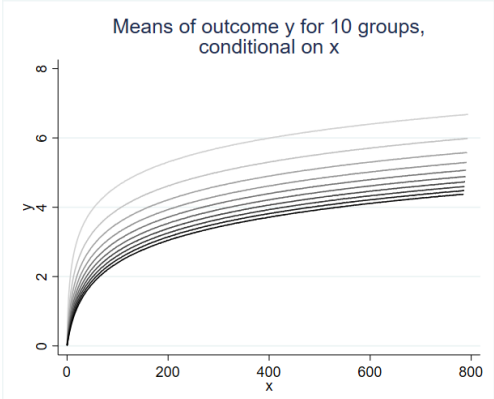
Stata code

```
webuse plotdata, clear
forvalues g = 1/10 {
  PLOTMEANS y if gr10==‘g’, over(x)
}
```

- This example plots means of variable y conditional on a specific value of $x \in [0, 800]$

- Each curve consists of conditional means corresponding to a distinct subgroup of observations g

- Applications: avg wages over time, avg years of education by age, etc.



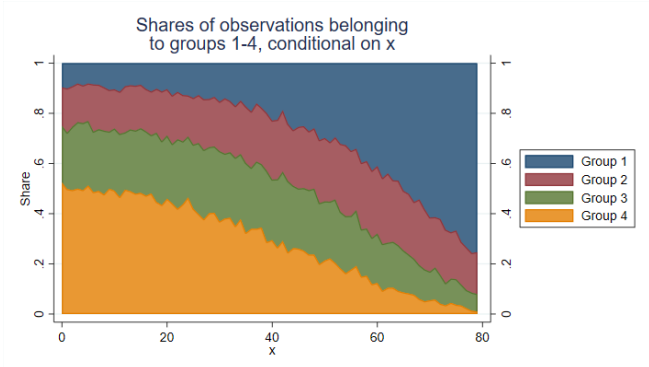
PLOTAREA: conditional share plots

Stata code

```
webuse plotdata, clear
PLOTAREA g, over(x)
```

- This example plots the conditional shares of observations belonging to one of the $g \in [1, 4]$ mutually exclusive categories.

- Applications: highest level of education by age, shares of felonies over time / pop. density, etc.



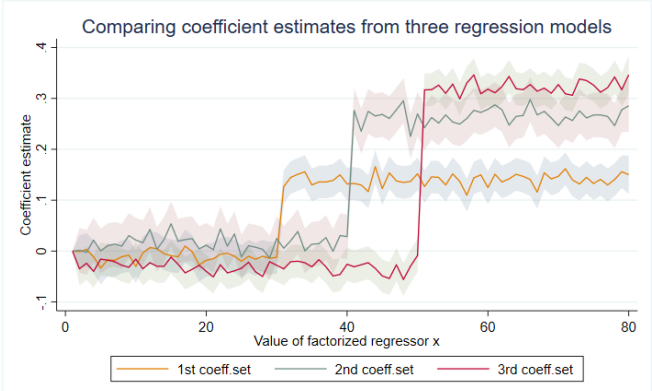
PLOTB: plot sets of coefficient estimates

Stata code

```
webuse plotdata, clear
reg z1 i.x
PLOTB i.x, clear
reg z2 i.x
PLOTB i.x
reg z3 i.x
PLOTB i.x
```

- less verbose alternative to **coefplot**

- Applications: dif-in-dif, RD, heterogeneity analyses, specif. testing & robustness



Benchmarking PLOT against native commands

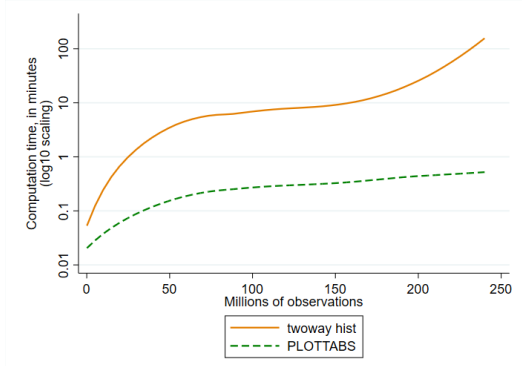
Stata code

```
webuse plotdata, clear
// PLOTTABS
PLOTTABS if gr==1, over(x)
PLOTTABS if gr==2, over(x) gr(bar)

// Twoway native command
twoway (histogram x if gr==1, disc) ///
       (histogram x if gr==2, disc)
```

- the plot comparing the execution times (cond. on sample size) uses the log10 scale

- With large datasets, PLOTTABS finishes **under a minute**, whereas twoway hist takes **almost 2 hrs**



Conclusion

- Efficiency matters! And optimizing popular software packages might just be the most environmentally conscious thing that we (as individuals) will ever do.
- While somewhat quirky, my PLOT commands highlight the efficiency gains that can be reapt by bypassing a legacy bottleneck of native graphing commands in Stata (other commands can be adjusted in the same way too).
- **Disclaimer**: I have neither the skills nor time to turn the PLOT commands into proper .ado commands with correct syntax, help files, and all that...
  So, if you like these commands and would like to help with turning them into something that is more standardized (SSC) please get in touch!

Thank you for your attention!

j.kabatek@unimelb.edu.au