Big Data in Stata

Paulo Guimaraes[1,2]

[1]Banco de Portugal
[2]Universidade do Porto

Portuguese Stata UGM - Sept 18, 2015

What do I mean by "big data"?

- "big data" has several meanings
- the Vs of big data
- "large data set" may be more appropriate
- many observations, many variables
- typical examples: large administrative data sets, panel data

Why does it matter?

- your computer may not be able to load the data
  - Stata stores data in RAM
  - memory is allocated dynamically
  - Stata imposes a limit of 2.1 billion observations (except Stata/MP)
- time becomes relevant - usual procedures may take hours, even days
- usual procedures may not be feasible at all

Basic advice

- use a powerful computer (many MhZ) with lots of RAM
- invest in your code
- test your code in a small data set
- take advantage of many user-programmed tools
- use the latest version of Stata
- use Stata/MP

Stata MP

- Stata/MP takes advantage of computers with multiple cores and multiple processors
- runs 1.6 times faster on 2 cores, 2.1 times faster on 4 cores, and 2.7 times faster on 8 cores (Statacorp)
- All timings are on a 1 million observation dataset. The two regressions included 50 covariates.

Timing (seconds)		
Analysis	24 cores	1 core
generate a new variable	0.03	0.33
summarize 50 variables	0.88	19.55
twoway tabulation	0.45	0.45
linear regression	0.65	11.48
logistic regression	7.19	59.27

Source: Statablog

- for details see Stata/MP Performance Report

Reading data

- Stata reads faster from its native format
- Stata reads all data to RAM and there are limits on the number of observations and number of variables. These limits depend on your version of Stata
- if you have trouble importing a large Excel file try using `set excelxlsxlargefile on`
- you can approximate the size of your data set with

$$M = \frac{N * V * W + 4 * N}{1024^2}$$

M - size in megabytes
N - number of observations
V - number of variables
W - average width in bytes of a variable
Source: Statacorp

Read only the variables that you need

- you can read only a select number of observations or variables
  `use [varlist] [if] [in] using filename [, clear nolabel]`
- not all I/O commands allow a variable list and the `[if] [in]` qualifiers. Some that do are:
  `infile`, `infix`, `fdause`
- you can also use odbc to extract just the needed variables
- use third-party software such as DBMS or Stattransfer to select a subset of the variables

Simple coding tips

- make sure to specify the correct type for the variables
  - it saves space
  - it avoids problems
- compress your data
- avoid strings if you can (use value labels) **
- take advantage of Stata's factor-variable operators *
  - use only one variable per category
  - do not store squared variables, interactions, or lagged values
- use built-in commands if possible (see `which`)

More coding tips

- sort *
  - use `sort` instead of `gsort` for "decreasing sorts" (Feenberg)
  - if you need to sort on several variables (byte, int, or long) consider using the user-written utility `hash` (Maurer)
- collapse *
  - may be faster to write your code for collapse
  - use the user-written `fastcollapse` (Maurer)
- recoding *
  - it makes a big difference how you (re)code
  - `recode` is typically slow
  - for additional examples see Canner and Schneider

And a few more ...

- reshape *
  - the `reshape` command is very slow
  - it is usually faster to break the data into several files and reassemble it on the desired format
- egen *
  - the `egen` command can also be very slow
  - it may pay to code alternatives to egen

Making Stata run faster

- learn Mata
  - Mata is a fast matrix language built into Stata
- write a Stata plugin
  - plugins are compiled code that you can attach to Stata
- if you have a desktop with multiple cores use the package `parallel` (Vega)
  - `parallel` runs multiple Stata instances on the same computer
- Lokshin and Radyakin (2014) showed that it is possible to join the power of multiple computers in a network
  - they built a set of tools to implement distributed computations (HPCCMD)

Keep the data simple

- use a "clean" dataset
  - data should have just the variables needed for the analysis
  - cases with missing observations should be removed
  - store the variables efficiently
- will a sample do?
  - for many procedures the results will be similar
  - it is fairly easy to sample observations or clusters (see Stata FAQs)

Understand your data

- do you have duplicate observations?
  - create a variable with frequency of unique cases
  - do the analysis with the "weights" option
- are observations repeated on the X variables?
  - instead of `logit` use `binreg` or `glm` on grouped data
  - instead of `clogit` use `multin` on grouped data
  - instead of `poisson` use `poisson` with exposure on grouped data
  - instead of `regress` use `regress` with weights on grouped data

Regressions ...

- What if you want to estimate a regression with thousands of regressors?
  - It is possible using the iterative procedure of Guimaraes and Portugal (2010)
  - Torres et al (2015) use Stata to estimate a linear regression function with 28 million observations and $33,491$ covariates, 18 year dummies and a fixed effect
  - the procedure may be adapted to other problems
- What if you want to estimate a regression with a single fixed-effect?
  - consider using `areg` or `_regress` instead of `xtreg`
  - but pay attention to clustered standard errors

More regressions ...

- What if you to estimate a linear regression with two or more fixed effects?
  - there are many user-written commands (`a2reg`, `gpreg`, `felsdvreg`, `reg2hdfe`)
  - but the gold standard nowadays is `reghdfe` by Sergio Correia
    - absorbs any number of fixed effects and their interactions
    - implements IV estimation
    - much faster and takes advantage of multiple cores
    - excellent support (github)
- What if you want to estimate a Poisson regression with two fixed effects?
  - use the package `poi2hdfe`

Advice on estimation of high-dimensional models

- be patient! this is not OLS regression!
- you can probably use a lower convergence criterion
- be careful about using the estimated fixed effects for secondary analysis
- remember that fes are only identified by imposing restrictions
- if you use clustered ses make sure you have a high enough number of clusters

An example

- with large data sets we can use more flexible parametrizations
- consider the typical wage regression

$$log(wage) = \beta_1 age + \beta_2 tenure + firm_f e + ind_f e + year_f e$$

- employee-employer panel data set with 28 million observations (1986-2013)
- age and tenure were introduced as discretized variables

References

- Canner, J. and Schneider, E. "Optimizing Stata for Analysis of Large Data Sets" Stata Conference New Orleans, LA (July, 2013)
- Feenberg, D., "Stata for very large datasets" available at http://www.nber.org/stata/efficient/ (July 2012)
- Guimarães, P., "Understanding the Multinomial-Poisson transformation" Stata Journal, vol4, 3, pp.265-273 (2004)
- Guimarães, P. and Portugal P., "A simple feasible procedure to fit models with high-dimensional fixed effects", Stata Journal, vol 10, 4, pp.628-649 (2010)

References

- Lokshin, M. and Radyakin, S. "Distributed computations in Stata" Stata Conference Boston, MA (August, 2014)
- StataCorp "Stata/MP Performance Report" available at http://www.stata.com/statamp/statamp.pdf
- Torres, S., Portugal, P. and Addison, J., and Guimarães, P. "The Sources of Wage Variation: A Three-Way High-Dimensional Fixed Effects Regression Model," unpublished mnuscript (2015)
- Vega, G. "Just tired of endless loops! or parallel: Stata module for parallel computing" Stata Conference New Orleans, LA (July, 2013)