

combining (and reshaping) data

adam okulicz-kozaryn

`adam.okulicz.kozaryn@gmail.com`

this version: Thursday 3rd October, 2019 23:45

outline

intuition

merge

[*] fancy merging

append, reshape, xpose

[*] joinby

outline

intuition

merge

[*] fancy merging

append, reshape, xpose

[*] joinby

overview: merge, append, reshape, xpose, joinby

- ◇ merge, append, joinby combine
 - <https://www.ssc.wisc.edu/sscc/pubs/sfr-combine.htm>
- ◇ reshape, xpose change shape
- ◇ merge is key! perhaps the most important command
- ◇ reshape is useful and difficult
- ◇ append, xpose, joinby are rare
 - but good to know they are there and what they can do

outline

intuition

merge

[*] fancy merging

append, reshape, xpose

[*] joinby

the power of merge

- ◇ merging is one of the most useful things you'll learn here
- ◇ great value comes from simple fact of merging data
- ◇ recall from intro: there's a ton data of (and growing!)
- ◇ but these data are mostly useless unless in one file!
- ◇ somehow organizations (and researchers) are in this persistent habit of having their data chopped up in tiny multiple files
- ◇ they're hungry for knowledge and want to make use of the data and this is where you come in! can make \$ by just merging!
- (and then fun part: visualization/graphics in 2 weeks, but dat man first!)

easy to merge; difficult to do it right

- ◇ it depends on what kind of data (and luck) you have
- ◇ the challenge is to check what happened after the merge
- ◇ sometimes it all merges smoothly without any issues
- ◇ but almost always it doesn't
- ◇ and then the work begins
- ◇ **always investigate carefully non-merges**
- ◇ **make sure that *ALL* nonmerges are as expected**
- ◇ **even matches can be wrong**
 - use a lot of des sta to investigate
 - always be skeptical, ask yourself whether it makes sense

after merge

- ◇ typically some obs did not merge due to diff coding
- ◇ say “Poland” ≠ “Republic of Poland”
- ◇ “CAMDEN” ≠ “Camden” etc
- ◇ then go back and fix it before merge:
- ◇ **replace ctry=“Poland” if ctry==“Republic of Poland”**
- ◇ in many cases it was not supposed to merge, say
 - there was country in A, but not in B
 - data in A was for 1995-2000, in B 1990-1998
 - etc
- ◇ but you have to be 100% sure that nonmerges were correct to happen!

to be honest

- ◇ to confess, what I sometimes do:
 - I simply make a note to myself that I do not care now
 - and I will investigate it later, that is
 - I just put in there a '*LATER:' comment
 - but I only do that if problem is small say around 5% of obs

dirty data

- ◇ the other challenge is to deal with dirty data
- ◇ most data are dirty: weird chars, mistakes, inconsistent names/codes, missing vals
- ◇ weird chars: %, \$, #, etc or non-english letters
- ◇ mistakes: should be 9, but it is 5, etc
- ◇ inconsistent names/codes: 'Camden' \neq 'CAMDEN'

merge (combines variables (same obs))

- ◇ let's generate 2 datasets by splitting one dataset into two
 - and then merge back again
- ◇ use gss.dta, clear
- ◇ gen id=_n
- ◇ keep id region
- ◇ save gss1.dta, replace (**using**) has region
- ◇ use gss.dta, clear
- ◇ gen id=_n
- ◇ keep id inc (**master**) has inc
- ◇ merge 1:1 id using gss1.dta (combine with (**using**))

merge contn'd

- ◇ after merging **always** think about output:
- ◇ `tab _merge`
- ◇ variable `_merge` takes on 3 values:
- ◇ **3** obs in both datasets
- ◇ **1** obs in master only
- ◇ **2** obs in using only
- ◇ `dofile`

merging investigation

- ◇ from my experience, I have found particularly useful:
- ◇ `tab` `_merge` with time and geography
 - say year and state
- ◇ may also want to `list` or `edit` part of datafile
 - especially if it is small
- ◇ can also sort on `_merge` and other key vars
- ◇ it does take time to find out what happened

merge 1:m

- ◇ often you `merge 1:m`
- ◇ very useful command indeed
- ◇ but people often make a mistake of specifying `merge m:m`
- ◇ and I have never seen, cannot even think of situation when this would be applicable

sometimes need to collapse!

- ◇ sometimes may have many (non-unique) obs in one dataset
- ◇ and so the same in the other dataset
- ◇ say multiple animal abuses per zip in one
- ◇ and multiple shelters per zip in the other one
- ◇ cannot merge it!! need to collapse less important one
- ◇ say you're primarily interested in abuse, then collapse shelters: say count them by zip
- ◇ and merge that 1:m with multiple abuses by zip

be clear about merging

- ◇ want to be clear about nonmergers in paper!
 - say how many nonmerges and waht you did about it
 - eg dropped, fixed, etc

merging multiple files

- ◇ can merge at once
 - merge 1:1 id using A B C D
 - avoid at once, too messy
- ◇ better in some steps, eg $A+B$, $C+D$, $AB+CD$
 - or perhaps best $A+B$, $AB+C$, $ABC+D$, etc
- ◇ perhaps best first do easy and clean merges

1:1 merge on 2 vars

- ◇ often need to merge 1:1 on 2 vars
 - when 2 vars uniquely define obs
 - eg country-year, state-county
- ◇ merge 1:1 countryID year using B

what to merge on?

- ◇ geography! usually have some!
- ◇ can always aggregate up! say have city and state, so can merge m:1 on state
- ◇ time! say with weather—usually weather matters!
- ◇ occupation! there are occ codes eg <https://www.onetonline.org/find/descriptor/result/4.A.2.b.2>

outline

intuition

merge

[*] fancy merging

append, reshape, xpose

[*] joinby

merging non-matching ids

- ◇ <http://stats.stackexchange.com/questions/32830/suggestions-on-how-to-merge-multiple-datasets-with-an-imperfect-id>
- ◇
 - (1) The Catcher and the Rye, 7/16/51
 - (2) The Catcher & the Rye, 7/16/51
 - (3) Catcher and the Rye, 1951
 - (4) The Catcher and the Rye (1951), [missing]

merging non-matching ids

- ◇ ssc install strgroup
 - uses Levenshtein distances to do string matching
- ◇ **reclink**
 - probabilistic matching scheme
- ◇ <http://github.com/OpenRefine>

outline

intuition

merge

[*] fancy merging

append, reshape, xpose

[*] joinby

Append

- ◇ Combines Observations (Same Var)
- ◇ Let's generate some data first
- ◇ `use gss.dta, clear`
- ◇ `keep in 1/50`
- ◇ `save gss1.dta, replace` (**using**)
- ◇ `use gss.dta, clear`
- ◇ `keep in 51/100` (**master**)
- ◇ `append using gss1.dta` (combine with (**using**))
- ◇ `dofile`
- ◇ `append` is easy in practice as compared to `merge`

we are about to look at reshape

- ◇ `reshape` is a very peculiar command
- ◇ incredibly powerful, and difficult to understand
- ◇ i thought i have mastered stata
- ◇ but whenever i reshape, i always scratch my head
 - i just always `help reshape`—useful examples to clarify
- ◇ yet reshape is the only way out in many situations
- ◇ we will try to use it often

xpose, reshape

- ◇ `xpose` interchanges Vars and Obs
- ◇ `reshape` converts wide-to-long/long-to-wide
- ◇ `help reshape` (very useful diagram–i always use it!)
- ◇ `reshape long var, i(id) j(year)`
- ◇ `var` is a common part of var that repeats, i.e. prefix,
- ◇ `id` is always unique (eg made by `gen id=_n`)
- ◇ `year` is a new variable that takes unique part from variable that repeats, i.e. suffix

reshape example

- ◇ use gss.dta, clear
- ◇ ren inc inc1
- ◇ gen inc2=2*inc1
- ◇ gen id=_n
- ◇ reshape long inc, i(id) j(period)
- ◇ edit
- ◇ dofile
- ◇ and lets go over output of reshape—it tells you how it changed!

outline

intuition

merge

[*] fancy merging

append, reshape, xpose

[*] joinby

joinby

- ◇ <https://www.stata.com/manuals/u22.pdf>
- ◇ <https://www.stata.com/manuals14/djoinby.pdf>
- ◇ [https://stats.idre.ucla.edu/stata/faq/
how-can-i-create-all-pairs-within-groups](https://stats.idre.ucla.edu/stata/faq/how-can-i-create-all-pairs-within-groups)