New data cleaning command: `assertlist` improves speed & accuracy of collaborative correction

Dale Rhoda & Mary Kay Trimner

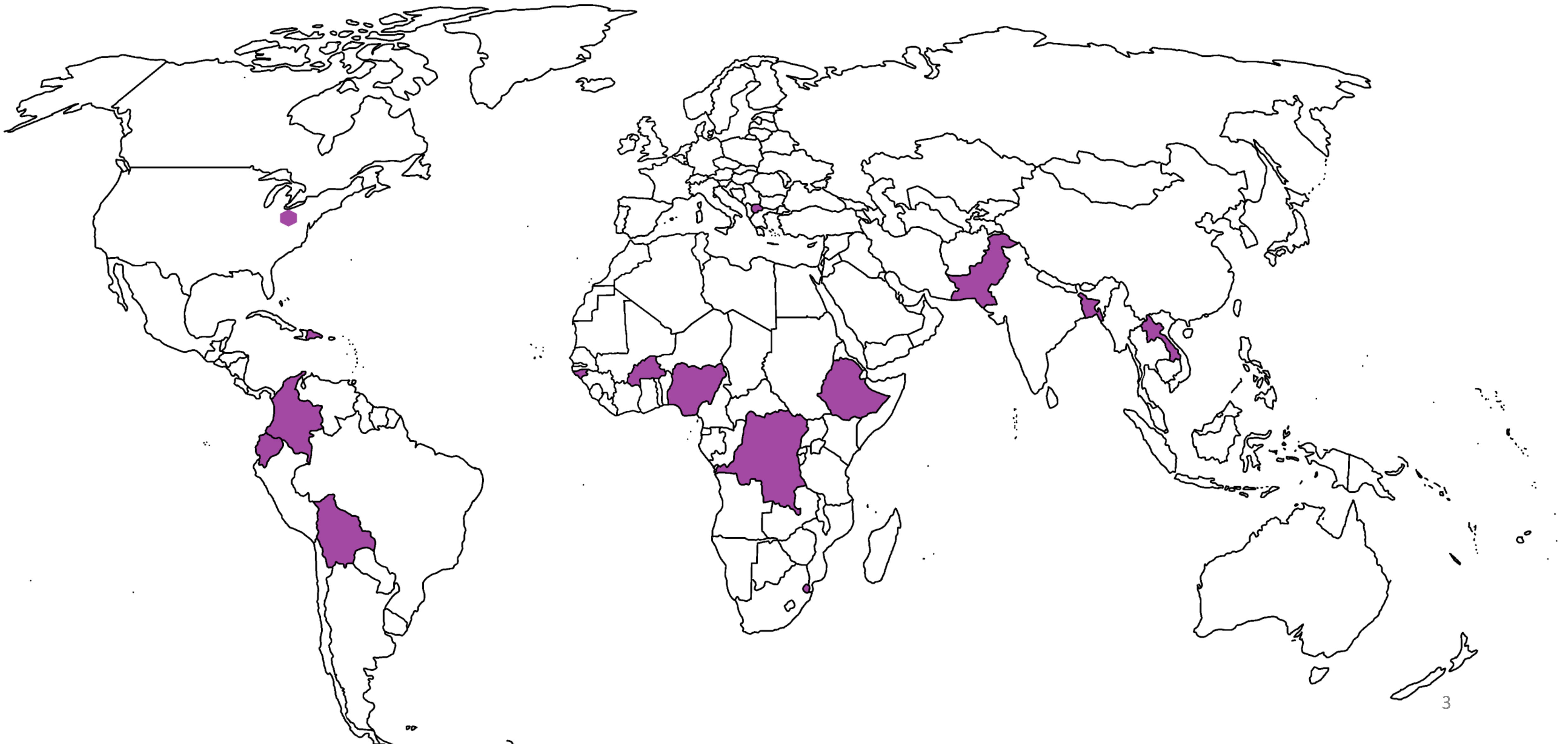
Stata Conference 2018

BIOSTAT
GLOBAL CONSULTING

Context: Data Cleaning

- Primary data collection, entry & cleaning (e.g., household survey)
- There are artifacts (paper, photos, phone numbers) that might be used to confirm correct values
    - Data sometimes stored on paper forms
    - Or, with touchscreen, there is sometimes a photo of important docs (e.g., child's vaccination record)
    - Sometimes collect respondent's mobile phone number to coordinate interview time & place or ask follow-up questions

Where Do We Do It?



Distributed Data Cleaning

Team Constructing the Analysis Dataset

- Check values for correctness, completeness, consistency
- Flag disallowed or inconsistent values for review

Team with Access to Survey Forms

- Check flagged values (via paper form, photo, or phone call)
- Identify corrected data values where possible

- Amend the dataset to include corrections; document as appropriate
- (Sometimes) Change uncorrectable values to special missing

Example: Clean Dataset

stratum	cluster	hhid	respid	iid	idate	name	vxd
1	1	1-1	1	1	01nov2017	Jacob	Yes
1	1	1-1	2	1	01nov2017	Liam	No
1	1	1-1	3	1	01nov2017	Noah	Yes
1	1	1-2	1	1	01nov2017	Mason	Yes
1	1	2-2	1	1	01nov2017	William	No
1	2	3-1	1	2	03nov2017	James	Yes
1	2	3-1	2	2	03nov2017	Ethan	Yes
1	2	3-2	1	2	03nov2017	Alexander	No
1	2	4-1	1	2	03nov2017	Michael	Yes
1	2	4-2	1	2	03nov2017	Benjamin	Yes

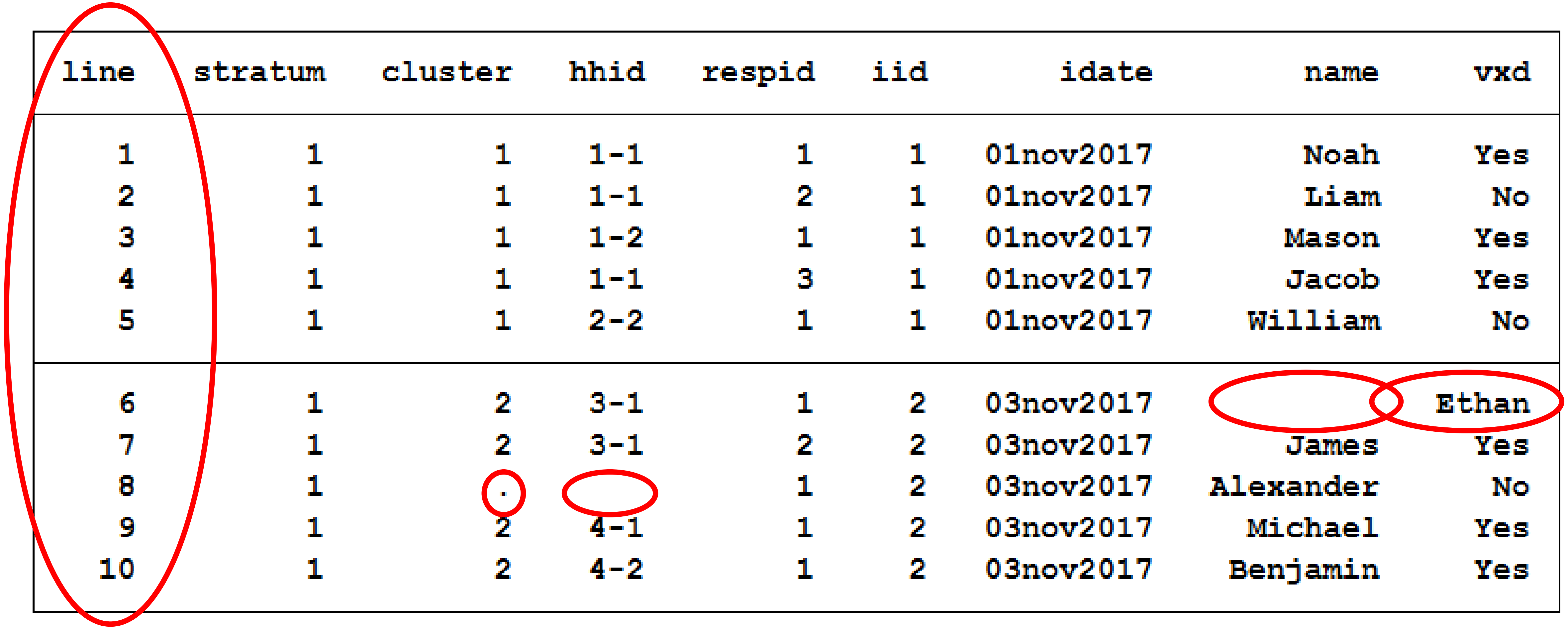
Note: hhid = household ID; respid = respondent ID; iid = interviewer ID; idate = interview date; vxd = vaccinated

List of Expectations

1. No variable should be missing for any respondent
2. No duplicate combos of ID variables: stratum cluster hhid respid
3. The variable vxd holds only values of “Yes” and “No”

Hint: If there appear to be problems in the ID variables, make a variable named *line* that is never missing and always unique and use it for corrections to ID variables

Example: Dataset with Problems



line	stratum	cluster	hhid	respid	iid	idate	name	vxd
1	1	1	1-1	1	1	01nov2017	Noah	Yes
2	1	1	1-1	2	1	01nov2017	Liam	No
3	1	1	1-2	1	1	01nov2017	Mason	Yes
4	1	1	1-1	3	1	01nov2017	Jacob	Yes
5	1	1	2-2	1	1	01nov2017	William	No
6	1	2	3-1	1	2	03nov2017		Ethan
7	1	2	3-1	2	2	03nov2017	James	Yes
8	1	.		1	2	03nov2017	Alexander	No
9	1	2	4-1	1	2	03nov2017	Michael	Yes
10	1	2	4-2	1	2	03nov2017	Benjamin	Yes

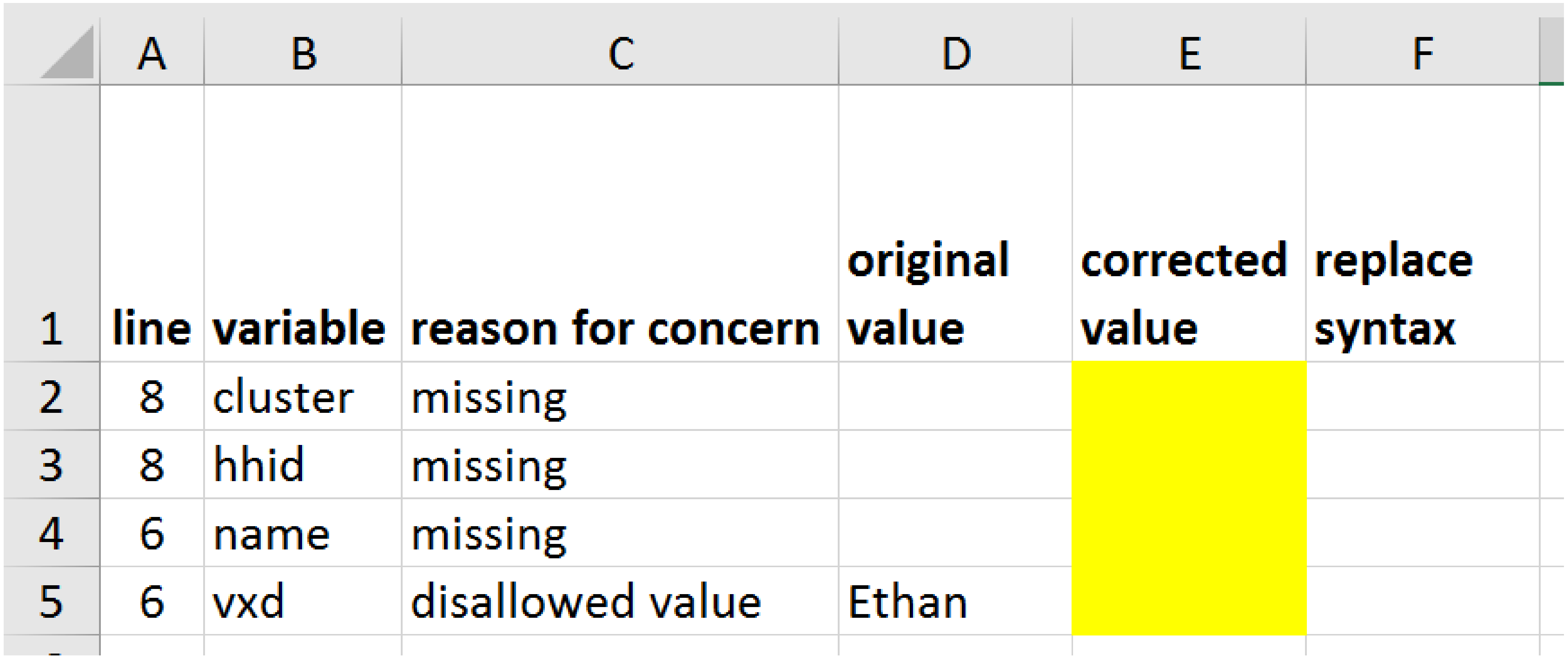
One Approach

```
assert !missing(stratum)
assert !missing(cluster)
assert !missing(iid)
assert !missing(idate)
assert !missing(hhid)
assert !missing(respid)
assert !missing(name)

assert inlist(vxd,"Yes","No")

duplicates list stratum cluster hhid respid
```

Another Approach

```
list line if missing(stratum)
list line if missing(cluster)
list line if missing(iid)
list line if missing(idate)
list line if missing(hhid)
list line if missing(respid)
list line if missing(name)

list line vxd if !inlist(vxd,"Yes","No")

duplicates list stratum cluster hhid respid
```

Conceptually, This Is What We Want



	A	B	C	D	E	F
1	**line**	**variable**	**reason for concern**	**original value**	**corrected value**	**replace syntax**
2	8	cluster	missing			
3	8	hhid	missing			
4	6	name	missing			
5	6	vxd	disallowed value	Ethan		

Stata Command *Assertlist*

- Syntax: assertlist *boolean_assertion* [if] [in]*, options*

- *boolean_assertion* is syntax that resolves to either true (1) or false (0)
  - Can involve the 'and' (&) or the 'or' (|) operators
  - sex must be Male or Female:
    - **`sex == "M" | sex == "F"`**
  - response must be 1 or 2 or 99:
    - **`response == 1 | response == 2 | response == 99`**

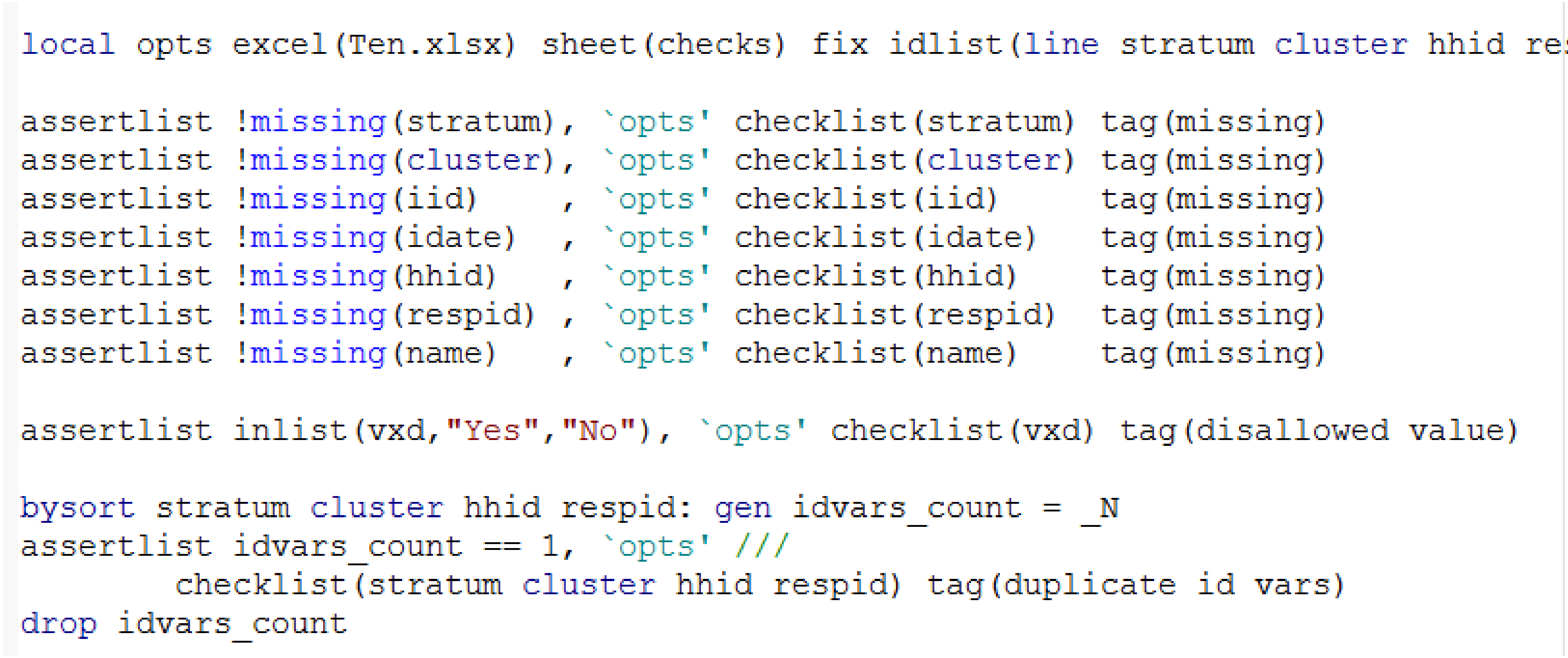
Assertlist Options

- **list(*varlist*)** **variables to list if expression is false**
- **tag(*string*)** **description to list along with output**

- **excel(*filename*)** **name of excel file for output**
  note: excel option requires sheet option
- **sheet(*sheetname*)** **name of worksheet for output**

- **fix** **provide option to correct some values**
  note: fix requires idlist & checklist
- **idlist(*varlist*)** **variables that uniquely identify rows**
- **checklist(*varlist*)** **variables to check & correct**

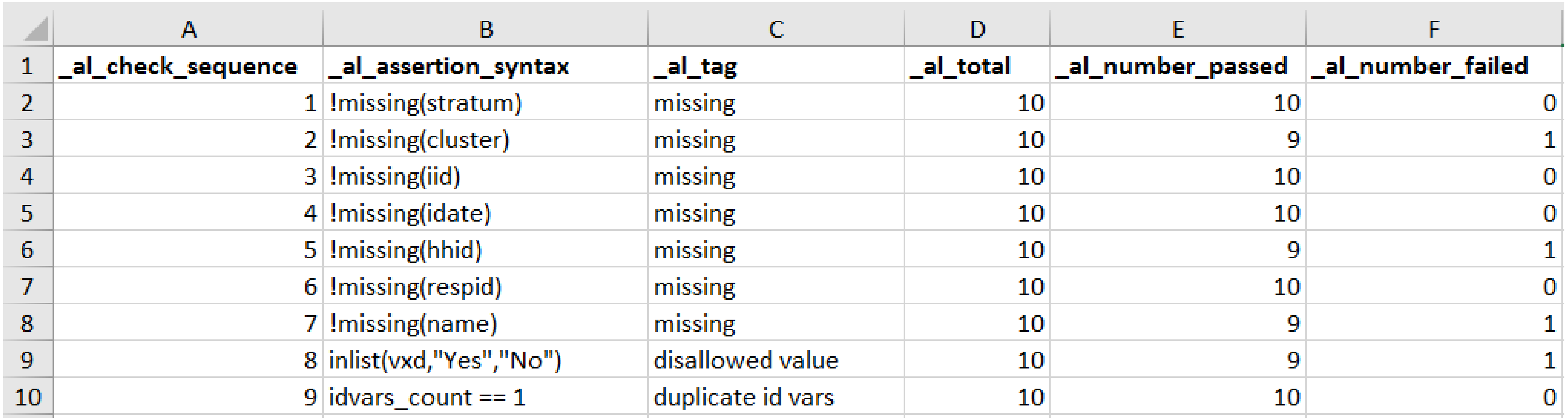
`assertlist` approach



```
local opts excel(Ten.xlsx) sheet(checks) fix idlist(line stratum cluster hhid respid)

assertlist !missing(stratum), `opts' checklist(stratum) tag(missing)
assertlist !missing(cluster), `opts' checklist(cluster) tag(missing)
assertlist !missing(iid)    , `opts' checklist(iid)     tag(missing)
assertlist !missing(idate)  , `opts' checklist(idate)   tag(missing)
assertlist !missing(hhid)   , `opts' checklist(hhid)    tag(missing)
assertlist !missing(respid) , `opts' checklist(respid)  tag(missing)
assertlist !missing(name)   , `opts' checklist(name)    tag(missing)

assertlist inlist(vxd,"Yes","No"), `opts' checklist(vxd) tag(disallowed value)

bysort stratum cluster hhid respid: gen idvars_count = _N
assertlist idvars_count == 1, `opts' ///
        checklist(stratum cluster hhid respid) tag(duplicate id vars)
drop idvars_count
```

Output: Assertlist_Summary Tab

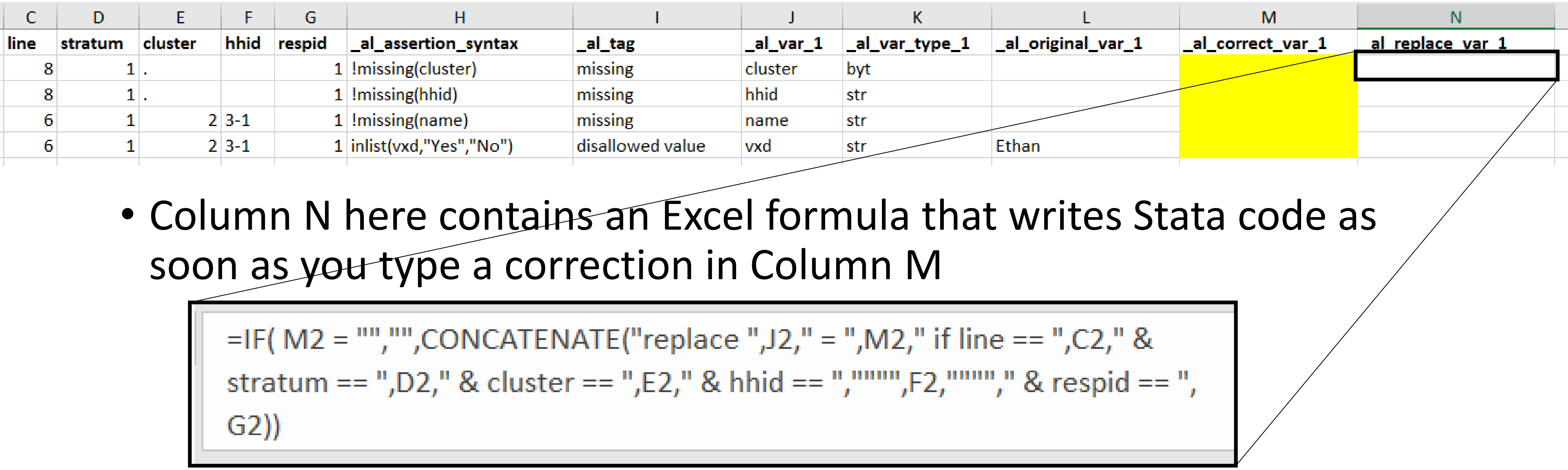
- Includes one row for every check
- Lists the *boolean_assertion & tag*
- Lists # of observations checked, # passed and # failed



	A	B	C	D	E	F
1	_al_check_sequence	_al_assertion_syntax	_al_tag	_al_total	_al_number_passed	_al_number_failed
2	1	!missing(stratum)	missing	10	10	0
3	2	!missing(cluster)	missing	10	9	1
4	3	!missing(iid)	missing	10	10	0
5	4	!missing(idate)	missing	10	10	0
6	5	!missing(hhid)	missing	10	9	1
7	6	!missing(respid)	missing	10	10	0
8	7	!missing(name)	missing	10	9	1
9	8	inlist(vxd,"Yes","No")	disallowed value	10	9	1
10	9	idvars_count == 1	duplicate id vars	10	10	0

Output: Checks Sheet

- The sheet named ‘checks_fix’ provides a space for making corrections



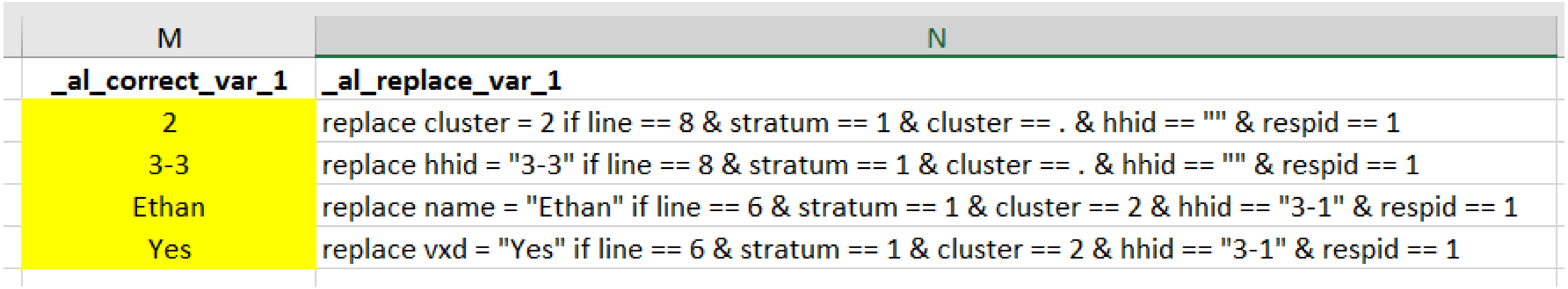
C	D	E	F	G	H	I	J	K	L	M	N
line	stratum	cluster	hhid	respid	_al_assertion_syntax	_al_tag	_al_var_1	_al_var_type_1	_al_original_var_1	_al_correct_var_1	al_replace_var_1
8	1	.		1	!missing(cluster)	missing	cluster	byt		🟨	
8	1	.		1	!missing(hhid)	missing	hhid	str		🟨	
6	1	2	3-1	1	!missing(name)	missing	name	str		🟨	
6	1	2	3-1	1	inlist(vxd,"Yes","No")	disallowed value	vxd	str	Ethan	🟨	

- Column N here contains an Excel formula that writes Stata code as soon as you type a correction in Column M

```
=IF( M2 = "","",CONCATENATE("replace ",J2," = ",M2," if line == ",C2," &
stratum == ",D2," & cluster == ",E2," & hhid == ","""",F2,"""","  & respid == ",
G2))
```

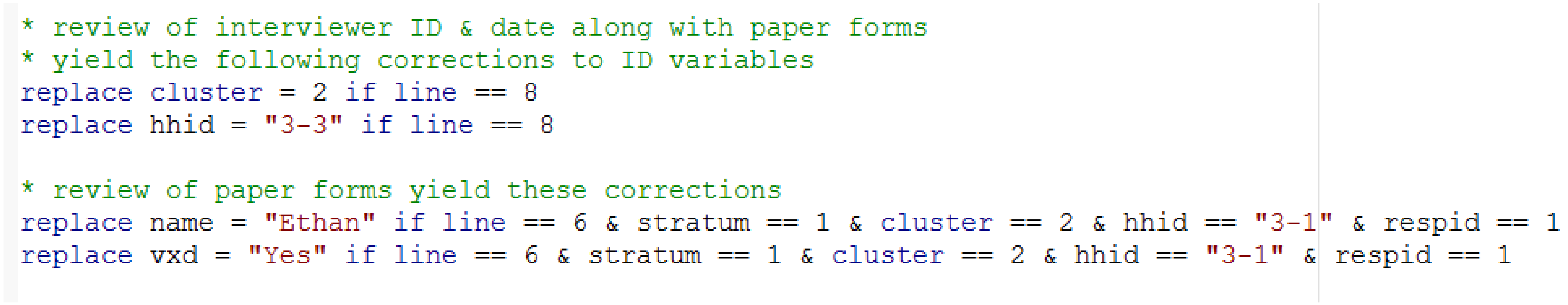
Output: checks_fix sheet

- You can paste the Stata syntax into a downstream .do file (and add some comments)



M	N
_al_correct_var_1	**_al_replace_var_1**
2	replace cluster = 2 if line == 8 & stratum == 1 & cluster == . & hhid == "" & respid == 1
3-3	replace hhid = "3-3" if line == 8 & stratum == 1 & cluster == . & hhid == "" & respid == 1
Ethan	replace name = "Ethan" if line == 6 & stratum == 1 & cluster == 2 & hhid == "3-1" & respid == 1
Yes	replace vxd = "Yes" if line == 6 & stratum == 1 & cluster == 2 & hhid == "3-1" & respid == 1

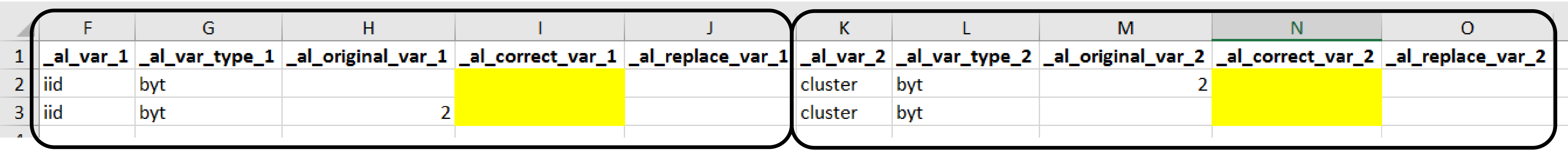
Data Corrections in Stata Program



```
* review of interviewer ID & date along with paper forms
* yield the following corrections to ID variables
replace cluster = 2 if line == 8
replace hhid = "3-3" if line == 8

* review of paper forms yield these corrections
replace name = "Ethan" if line == 6 & stratum == 1 & cluster == 2 & hhid == "3-1" & respid == 1
replace vxd = "Yes" if line == 6 & stratum == 1 & cluster == 2 & hhid == "3-1" & respid == 1
```

Notes

- We’re moving quickly thru this talk, but use a 2-stage approach
  1. Clean up the ID variables first, so they will serve as a unique key: a) to the dataset, and b) to the forms, photos or phone numbers
  2. Then clean up everything else, knowing you have a clean idlist
- The boolean_assertion can be as complex as needs be
- Both idlist and checklist can include numerous variables



	F	G	H	I	J	K	L	M	N	O
1	_al_var_1	_al_var_type_1	_al_original_var_1	_al_correct_var_1	_al_replace_var_1	_al_var_2	_al_var_type_2	_al_original_var_2	_al_correct_var_2	_al_replace_var_2
2	iid	byt				cluster	byt	2		
3	iid	byt	2			cluster	byt			

(Example of correction columns when checklist holds two variables)

Workflow for Collaborative Correction

1. Establish list of expectations as soon as questionnaire is finalized
    a) Valid values
    b) Skip patterns
    c) No duplicates
    d) Responses internally consistent
    (i.e., date of birth ≤ vaccination dates ≤ interview date)
2. Convert to `assertlist` syntax
3. Run the job & send the spreadsheet to data managers
4. Receive back the spreadsheet populated with corrections
5. Copy replace syntax into .do file
6. [Repeat Steps 3-5 as needed]
7. Decide what to do with uncorrected violations of expectations

Companion Program to Clean Up Title Rows

- When you are finished sending output to a particular Excel file, run `assertlist_cleanup` to switch to nicer column titles
- (`assertlist` uses some **nerdy** variable names as column titles because it re-reads & writes its output if you send results from 2+ assertions to the same Excel worksheet)

Can Also Flag Problems in Near-Real-Time

- When data are collected via touchscreen, they are often uploaded to a server on a daily basis
- Assertlist can be wrapped inside a loop that makes a daily list of concerns for each survey team
- E-mail the appropriate output to each supervisor to:
  - Correct errors right away
  - Coach (or replace) error-prone workers

Summary

- Assertlist flags data elements that are illegal or illogical
- Useful to document prevalence of problems
- Useful for replacing those elements with corrected values
- Many benefits:
  - Facilitates clear communication
  - Helps document corrections
  - Saves time
  - Prevents typing errors
- Caution: Requires some care with ID variables

Where Do I Find It?

- Today: Link to our GitHub page here: www.biostatglobal.com
- Soon: Find it on SSC

Questions?

Dale.Rhoda@biostatglobal.com

MaryKay.Trimner@biostatglobal.com

Link to GitHub:

www.biostatglobal.com