Lecture 4: Importing and cleaning data

BIOSTAT 212, Summer 2016

Kristen Aiemjoy
Department of Epidemiology and Biostatistics
University of California San Francisco

Review/ questions from last week?

- If you want to keep notes on the commands we use in class you can try using a .do file

- Reminder to look for data for your final project

Outline for today

- Importing data
- Label variables
- Missing data
- Checking data/ranges
- Generating variables
- Generating categorical variables from continuous data
- Dropping data
- Merging/appending

Clean data

- The last 2 data sets we worked with (coronarycalcium.dta and acupuncture_baseline.dta) were clean
  - The data set came as a .dta file
  - The variables were ready to analyze
  - Most variables were labeled

In the real world:

- Your data will arrive:
  - On paper forms
  - In a text file (comma or tab delimited)
  - In Excel
  - In Access
  - In another data format (SAS, etc)

Importing data: .CSV files

- Delimited text files are the best file type to import into STATA

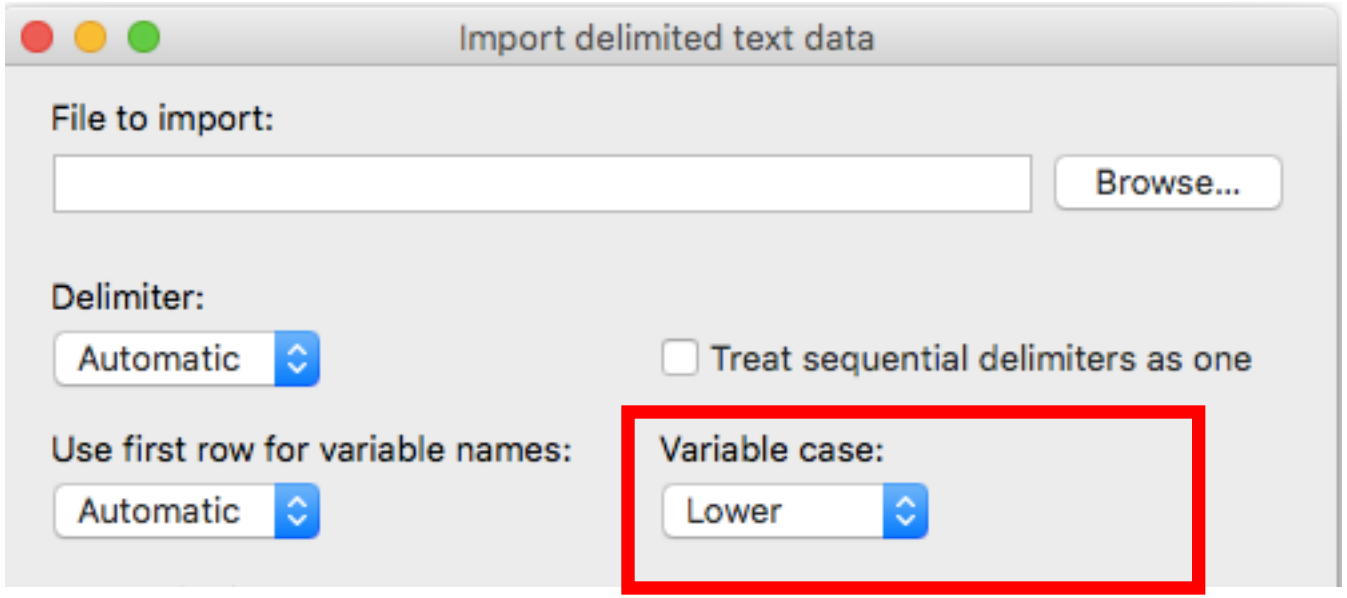
```
id,age,htn,dm,smoke,chol,cac,dead,obstime,anycac,caccat,agecat,male,ageover60,sex
1,25,0,0,0,0,0,0,1125,0,"0","<30",1,0,"male"
2,25,0,0,0,1,0,0,1125,0,"0","<30",0,0,"female"
3,25,1,0,1,0,0,0,1064,0,"0","<30",0,0,"female"
4,25,1,0,1,0,0,0,1064,0,"0","<30",0,0,"female"
5,25,0,0,0,0,0,0,913,0,"0","<30",1,0,"male"
6,25,0,0,1,0,0,0,852,0,"0","<30",0,0,"female"
7,25,0,0,0,1,3,0,699,1,"1-99","<30",1,0,"male”
```

- .CSV files use commas to separate (delimit) data
- Export data from other programs (like excel) as a .csv file
- Note: .CSV files will not preserve variable labels

Importing data

1. Use the command line
   - `import delimited "yourfile.csv"`
2. File < Import < Text Data (delimited, *.csv, …)
3. Copy paste into data editor – dangerous
4. Enter directly into data editor

Importing data: File<import



- First row = variable names
- Variable case = lower
- Use the dropdown menu and then copy command into your .do file

Importing data

- Whatever you do, make sure it worked!

	id	age	htn	dm	smoke	chol	cac	dead
1	1	25	0	0	0	0	0	0
2	2	25	0	0	0	1	0	0
3	3	25	1	0	1	0	0	0
4	4	25	1	0	1	0	0	0
5	5	25	0	0	0	0	0	0
6	6	25	0	0	1	0	0	0
7	7	25	0	0	0	1	3	0
8	8	26	0	0	0	1	0	0
9	9	26	1	0	1	0	0	0

Practice

- Download the accupunture_d.csv dataset from the course website

- Import the data set into STATA

```
import delimited "acupunture_d.csv"
```

-or-

File<Import<Text data (*.csv)

Cleaning Data

- STEP 1: Check variable names and labels
- STEP 2: Missing data
- STEP 3: Manipulate/generate variables
- Other optional steps:
  - Subset data
  - Append/merge data
  - Reshape data

STEP 1: Check names and labels

- Do you need to rename the variable?
- Do you need to label the variable?
- Do you need to label the values of categorical variables?

STEP 1: Variable names

- Short enough so easy to type, long enough to remember
- All lower case (or remember upper case)

STEP 1: Variable names: `rename`

```
rename old_varname new_varname
```

-changes the name of an existing variable
-the content of the variable is unchanged

Practice

- Variable f1 = headache frequency at baseline
- Rename f1 → hfreq_bl

```
rename f1 hfreq_bl
```

STEP 1: Variable labels: `label`

Label variables

```
label variable varname "labelname"
```

Label values of variables

```
label define labelname # "label" # "label" …
label values varname labelname
```

example

```
label define yesno 0 "no" 1 "yes"
label values exercise yesno
```

Practice

- Variable f1 = headache frequency at baseline
- label hfreq_bl → “headache frequency at baseline”

```
label variable hfreq_bl "headache frequency baseline"
```

STEP 2: Missing data

- Is there missing data?
- How are missing values coded?
- Standardize all missing data to (.)

STEP 2: Missing data

- Stata designates missing data with a period (.)
- Important to distinguish between ‘missing’ and ‘no’
- Why is it missing?
  - Loss to follow up
  - Death
  - Skip pattern in a survey
    - Certain questions can only be answered if respondent has children
  - Data collection errors
  - Respondent refusal/non-response

STEP 2: Missing data

- Is there missing data?

```
tab varname, missing
misstable sum
```

- How are missing values coded?
  - Range checks (`sum, codebook`)
  - `tabulate varname, missing`

Practice

- How many missing observations are there for ‘posths’?

```
misstable sum
codebook posths
inspect posths
```

- How many missing observations are there for ‘completed’?

```
tab completed, missing
```

STEP 2: Missing data: `mvdecode`

Standardize all missing data to (.)

`mvdecode _all, mv(99)` – recode all ‘99’ as ‘.’

`mvdecode varname, mv(99)` – for `varname` recode ‘99’ as ‘.’

Practice

- Change ‘99’ to (.)

```
mvdecode completed, mv(99)
```

- How many missing observations are there for ‘completed’?

```
tab completed, missing
```

STEP 3: Manipulate/generate variables

- Generate new variables

<u>Categorical variables</u>

- Do the categories make sense?
- Are dichotomous variables coded 0 and 1?

<u>Continuous variables</u>

- Check the range for outliers
- Change into categorical variable?

STEP 3: Create new variables: `generate`

`generate new_varname = exp` -generate a new variable

`exp = 0,#, ".", _n, addition(+), subtraction(-)`

example: `gen calsum = cal1 +cal2`

`generate new_varname=(condition)` -generate a new dichotomous variable where 1= condition_True & 0=condition_False

Options

`before(varname)/after(varname)` –place new variable before/after specified variable

Practice

- Generate a new running id variable

```
gen newid = _n

browse newid
```

STEP 3: Create new variables: `replace`

`replace varname = exp` – replace contents of an existing variable

`exp` = 0, #, “.”

Options

`if` –use logic operators

example `replace diarrhea=1 if freq>=3`

`in` –replace only in a subset of observations

Practice

- Generate a new variable for ‘severe migraine’ for a migraine + a headache score >50

```
gen severemigraine = (migraine==1 & basehs>50)
```

-or-

```
gen severemigraine = 0
replace severemigraine = 1 if migraine==1 & basehs>50
```

- How many participants have severe migraine at baseline?

```
tab severemigraine
```

STEP 3: Create new variables: `xtile`

`xtile new_varname = varname` -create a new variable containing percentiles

Options

`if` –use logic operators

`nq(#)` –set number of percentiles (default is 2)

Practice

- Create a new categorical variable that is a quartile of baseline headache square.

```
xtile hs_quartile = basehs, nq(4)
```

- What is the mean baseline headache score in the first quartile?

```
bysort hs_quartile: sum basehs
```

STEP 3: Create new variables: `egen`

```
egen new_varname = function(varname)
```

-create a new variable that is a function of a different variable(s)

Functions

mean	`egen new_varname = mean(varname)`
row mean	`egen new_varname = rowmean(varlist)`
row total	`egen new_varname = rowtotal(varlist)`
concatenate	`egen new_varname = concat(varlist)`

Options: by, if

*many more functions → look in help file

Practice

- Create a variable for average headache score (basehs, posths, oneyrhs)

```
egen meanhs = rowmean(basehs posths oneyrhs)
```

- What is the average headache score for the 5th participant?

```
gsort +id
list meanhs in 5
```

STEP 3: Create new variables: `encode/decode`

```
encode varname, generate(new_varname)
```

-Encode string into numeric variables with labels

```
decode varname, generate(new_varname)
```

-Decode numeric variables into string variables

Practice

- Change withdrawal_reason from string to numeric/categorical:

```
browse withdrawal_reason
encode withdrawal_reason, gen(dropout)
browse withdrawal_reason
```

- What percent of individuals who dropped out withdrew consent?

```
tab dropout
```

STEP 3: Manipulate data: `recode`

`recode varlist (rule)` – changes the values of numeric variables according to the rules specified.

rule	Example	Meaning
# = #	3 = 1	3 recoded to 1
# # = #	2 . = 9	2 and . recoded to 9
#/# = #	1/5 = 4	1 through 5 recoded to 4
nonmissing = #	nonmiss = 8	all other nonmissing to 8
missing = #	miss = 9	all other missings to 9

Options

`, generate` –generate a new variable for transformed data

STEP 3: Recode 0/1

- 1/0 convention for dichotomous variables
- 1=yes
- 0=no

```
tab varname, nolabel

recode 1=0 2=1
```

Practice

- Encode sex from a string variable to a numeric/categorical variable

```
encode sex, gen(sex2)
tab sex2, nolabel
```

- Recode 1/2 to 0/1

```
recode sex2 1=0 2=1
label define sex2_label 0 "female" 1 "male”
label values sex2 sex2_label
tab sex2
tab sex2, nolabel
```

STEP 3: Create categorical variable from continuous

1. Generate a new variable that is a copy of the variable you want to categorize

```
gen new_varname = old_varname
```

2. Recode new variable into new categories

```
recode new_varname min/25 = 1 26/50 = 2 51/max = 3
```

3. Create and assign a new label for your variable

```
label define label_name 1 "<25" 2 "26-50" 3 ">50"
label values new_varname label_name
```

-OR-

```
recode new_varname min/25 = 1 26/50 = 2 51/max = 3, gen(new_varname) label(label_name)
```

Practice

- Create a dichotomous variable for age (>50)

```
gen age2 = age
recode age2 min/49 = 1 50/max = 2
```

- Label new variable

```
label define age_label 1"<49" 2">=50”
label values age2 age2label
```

- How many study participants are 50 years old or older?

Subset data: `drop`

`drop if` -drops subset of data given condition

example `drop if age>100`

`drop in` -drops subset of observations

example `drop in 1/4`

`drop varlist` -drop one or more variables

**careful! you cant undo

Subset data: `keep`

`keep if` -keep observations meeting condition, all others are dropped!

example `keep if age<100`

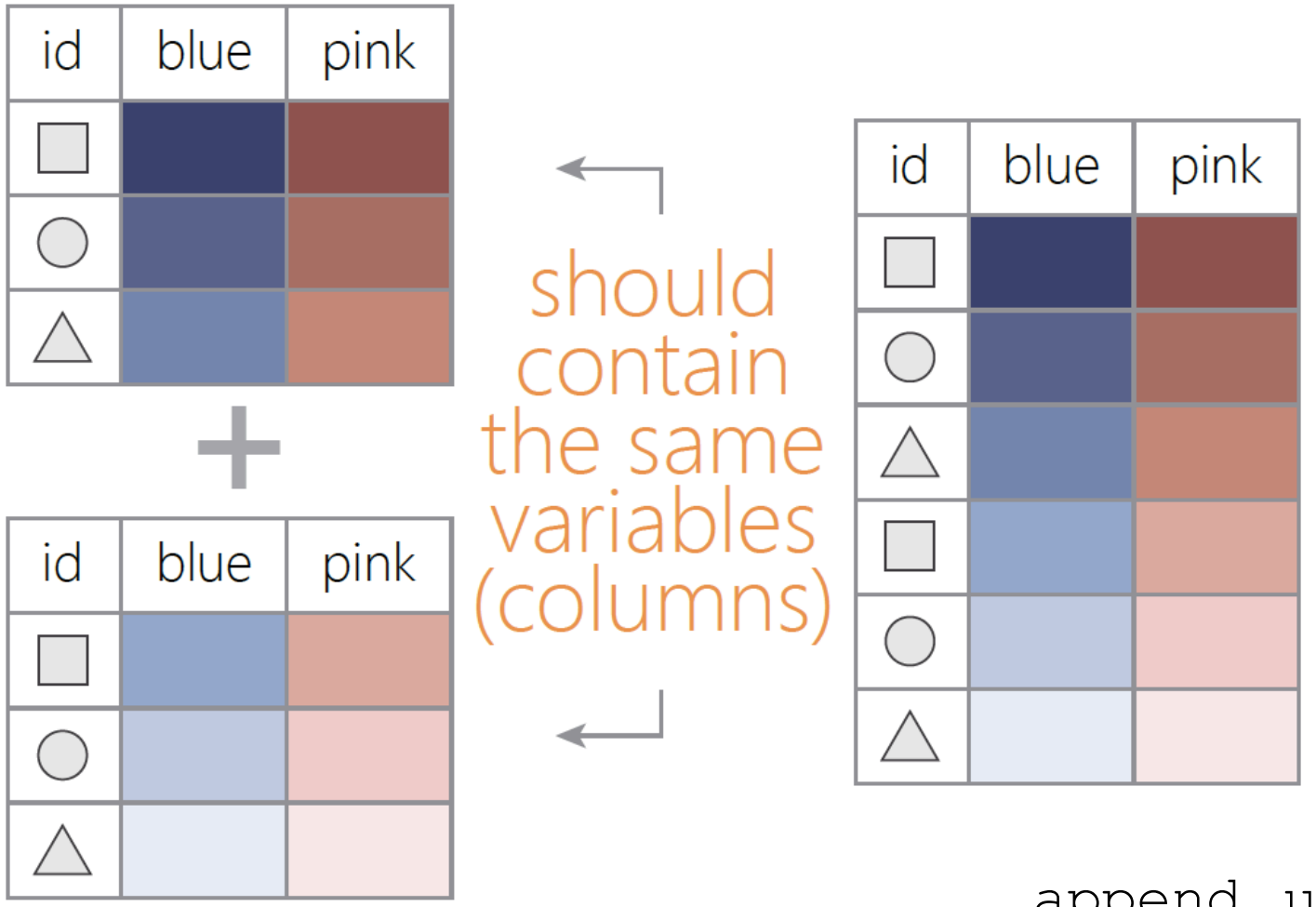
`keep in` -keep subset of observations

example `keep in 1/30`

`keep varlist` -keep one or more variables, all others dropped!

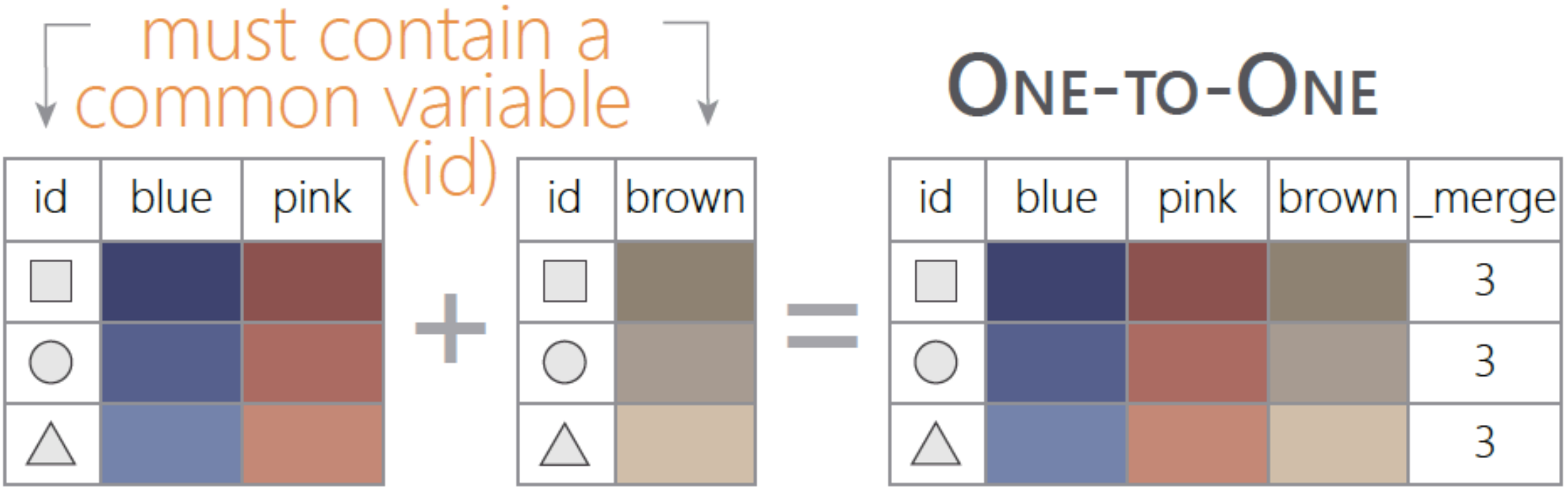
**careful! you cant undo

Adding (appending) new data: `append`




```
append using "newdata.dta"
```

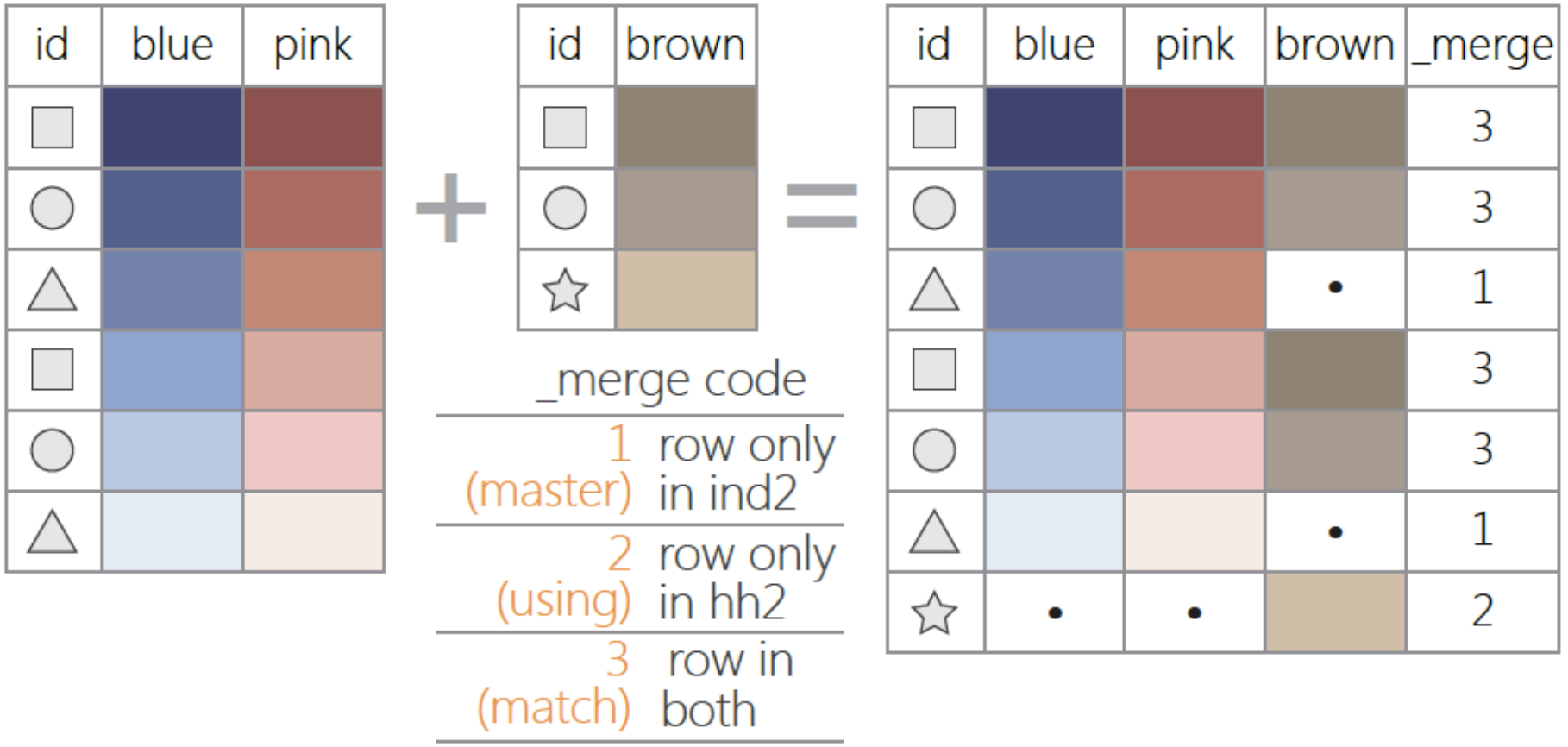
Merge data: One-to-One




```
merge 1:1 id_variable using "new_dataset.dta"
```

-merge into loaded dataset

Merge data: Many-to-One




```
Merge m:1 id_variable using "new_dataset.dta"
```

-merge into loaded dataset

Reshape data

- Handout

Insider tips for data cleaning

- We live in a dirty world
- Your data will be dirty also (no matter how well run the study is)
- Data cleaning can take up a huge amount of your Stata code and analysis time
- Beware missing values in Stata (assigned largest size)
- Think long and hard about outliers (removing vs sensitivity analyses vs other)

- Keep a raw unedited version of the original dataset and do not ever touch it. Save revised data in a new file.
- All your data cleaning decisions must be fully documented in your do-file
- You will not remember what you did in 6 months time

```
/*Document everything and trust no
one
...especially yourself */
```

Slide credit: Simon Pollet