Lecture 4: Data Cleaning Part 2

BIOSTAT 212

Amanda Irish | Amanda.Irish@ucsf.edu
Department of Epidemiology and Biostatistics
University of California San Francisco

Midterm Data Cleaning Project

- Due next week (no assignment due this week)
  - Worth the same # pts as the final
- Three parts:
  - Baseline data cleaning
  - Follow-up data cleaning
  - Combining datasets
- Will cover the remaining commands you need this week
- You may work together, but turn in your own assignments

Today

Data Cleaning Part 2:

- Working with dates
- Missing data
- Duplicates
- Combining datasets
  - Appending
  - Merging
- Reshaping data

Download files for today's lecture

- Download:
  - pt1.dta
  - pt2.dta
  - doctors.dta
  - reshape_example.dta
  - xfill.do
- Save .dta files to your "data" folder
- Save .do file to your "do" folder

Dates

Dates

- Dates can be recognized in many formats:
  - 8/8/17 or 8/8/2017
  - 8-8-2017
  - August 8, 2017
- Dates are usually imported as string (character) variables and need to be converted into dates that Stata recognizes

Credit: Biostat 212 TAs Sankari Ayyaluru & Yiwey Shieh, MD

Dates

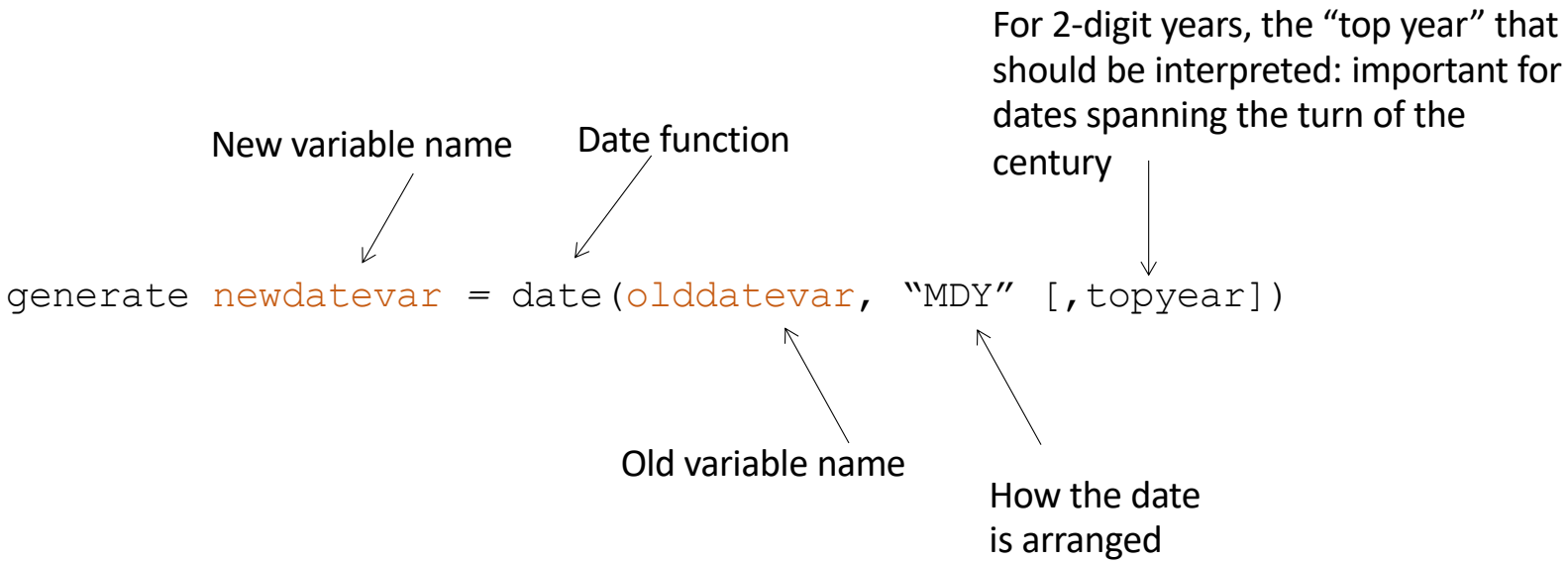
- Stata recodes dates as # of days elapsed from a reference date
  - January 1, 1960 = 0
  - January 2, 1960 = 1
  - January 3, 1960 = 2
  - December 31, 1960 = 364

- Makes it easy for Stata to manipulate dates mathematically

Credit: Biostat 212 TAs Sankari Ayyaluru & Yiwey Shieh, MD

Dates

Generate new date variable



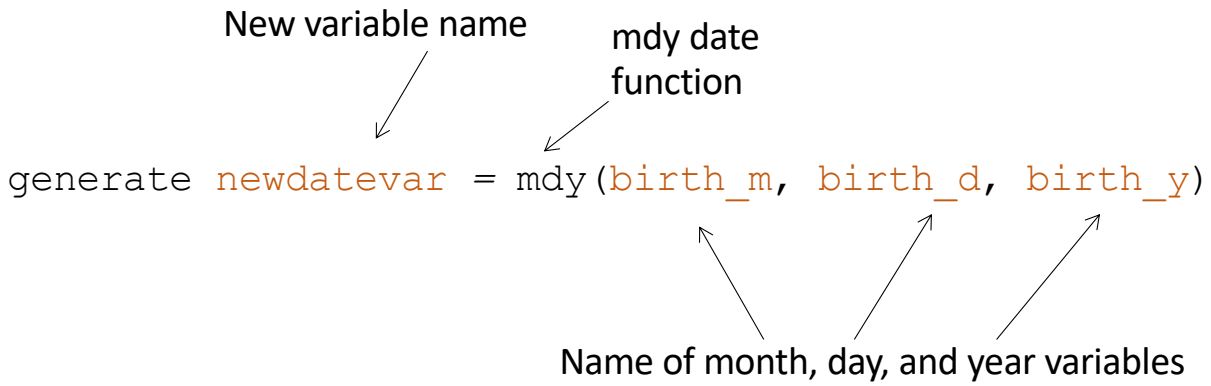
Credit: Biostat 212 TAs Sankari Ayyaluru & Yiwey Shieh, MD

Dates — MDY command

- If date components housed in separate variables:

birth_m	birth_d	birth_y
7	4	1953
1	18	1948
6	28	1949
5	30	1960

- Use `mdy` command to concatenate:



Dates — 2-digit years

- For 2-digit years, you need to specify first two digits of the year

```
generate newdatevar = date(olddatevar, "MD20Y")
```

8/03/17 = August 03, 1917→ will be miscoded
9/10/10 = September 10, 2010
03/22/16 = March 22, 1916→ will be miscoded
9/10/09 = September 10, 2009

Dates — 2-digit years

- Alternatively, (if you have dates that are in the 20th and 21st century), you can specify the top (most recent) year
- Anything after topyear = previous century

```
generate newdatevar = date(olddatevar, "MDY", 2010)
```

8/03/17 = August 03, 1917
9/10/10 = September 10, 2010 ← top year
03/22/16 = March 22, 1916
9/10/09 = September 10, 2009

Dates — Reformat date

- The numbers that emerge in Stata after generating date variable are meaningless to most humans
- The solution is to apply a format:

```
format newdatevar %td
```

%td = regular date with format DDmonCCYY
(other options & formats available - see Stata help file)

Dates — Mathematical operations

- We can add and subtract numbers to dates and find time (in days) elapsed between dates.

```
gen delta = visit1 - visit0
```

  - If you want this as # of hours:

```
gen delta_hrs = (visit1 - visit0) * 24
```

Dates

Logic operators

- Use the `td` prefix
- Variable must already be formatted as a date

- Suppose you wanted to categorize patients by a date
  - Patients starting ARV < 1996 = pre-HAART
  - `artstart` is the date of ARV initiation, already formatted as a date

```
gen prehaart = 0
replace prehaart = 1 if artstart<=td(01jan2006)
replace prehaart =. if artstart==.
```

Dates — Extract components

- Day

```
gen nameofdayvariable = day(datevariable)
```

- Day of week

```
gen nameofdayvariable = dow(datevariable)
```

*where 0(Sunday) - 6(Saturday)

- Month

```
gen nameofdayvariable = month(datevariable)
```

- Year

```
gen nameofdayvariable = month(datevariable)
```

Practice

- Open “pt1.dta”
- Create a new date variable (admit_date) in Stata’s date format from the string variable date_str
- Create a new variable indicated the date of discharge (discharge_date)
  - Hint: remember that Stata stores dates as the **number of days** since January 1, 1960 in order to make mathematical operations like addition much easier

Missing Data

Missing data

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

Missing data

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

Missing data

- Is there missing data?

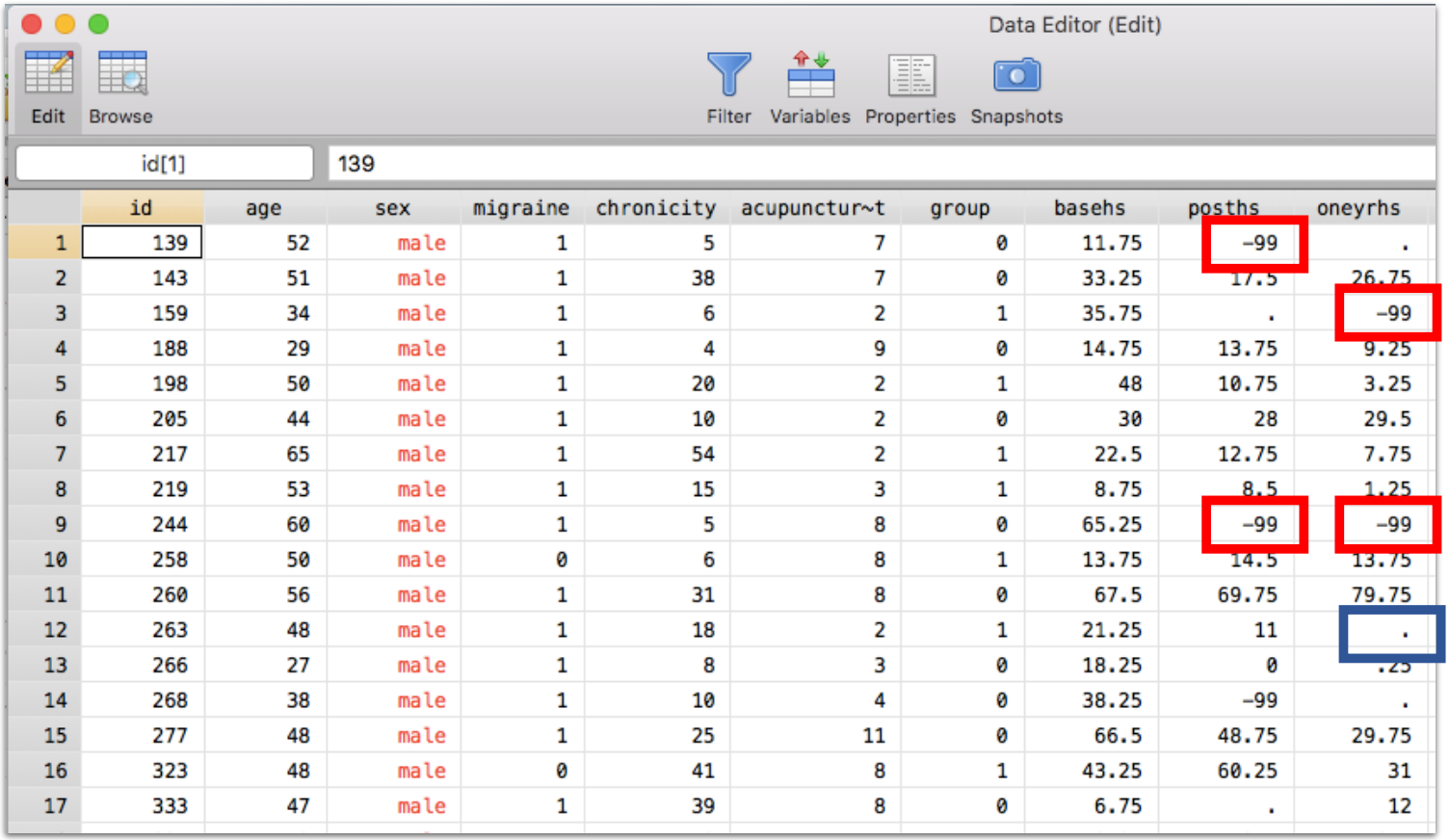
```
tab varname, missing
misstable sum
```

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

Stata codes for missing data

- "." is the default & most commonly used
- Can specify extended missing values (.a, .b … .z)
  - May designate reasons for missing data
  - Need to label as such
- Relationship to arithmetic operators
  - Non-missing observation < **.** < .a < .b < … < .z
  - Implications for how you specify non-missing values in your dataset
    - E.g. <. vs !=.

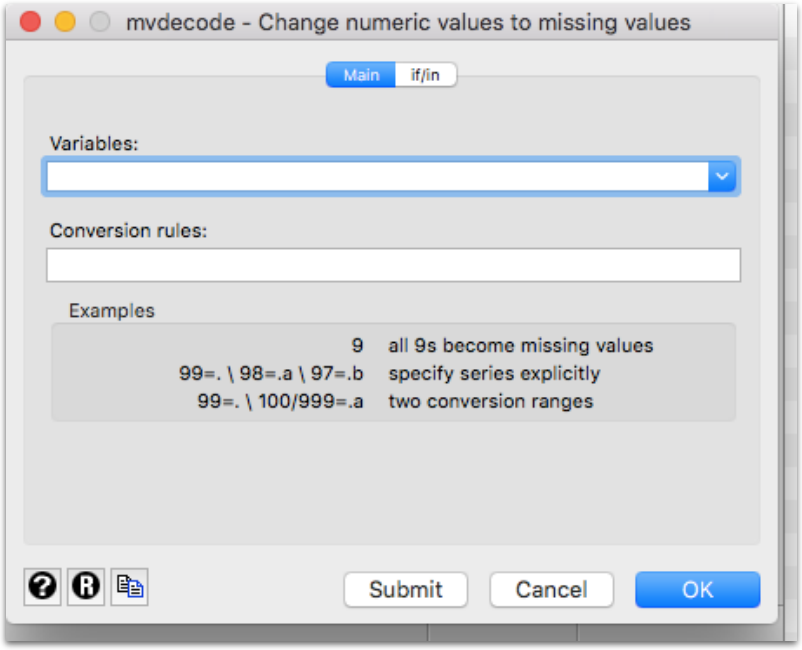
Convert numerical indicators to missing values



	id	age	sex	migraine	chronicity	acupunctur~t	group	basehs	posths	oneyrhs
1	139	52	male	1	5	7	0	11.75	-99	.
2	143	51	male	1	38	7	0	33.25	17.5	26.75
3	159	34	male	1	6	2	1	35.75	.	-99
4	188	29	male	1	4	9	0	14.75	13.75	9.25
5	198	50	male	1	20	2	1	48	10.75	3.25
6	205	44	male	1	10	2	0	30	28	29.5
7	217	65	male	1	54	2	1	22.5	12.75	7.75
8	219	53	male	1	15	3	1	8.75	8.5	1.25
9	244	60	male	1	5	8	0	65.25	-99	-99
10	258	50	male	0	6	8	1	13.75	14.5	13.75
11	260	56	male	1	31	8	0	67.5	69.75	79.75
12	263	48	male	1	18	2	1	21.25	11	.
13	266	27	male	1	8	3	0	18.25	0	.25
14	268	38	male	1	10	4	0	38.25	-99	.
15	277	48	male	1	25	11	0	66.5	48.75	29.75
16	323	48	male	0	41	8	1	43.25	60.25	31
17	333	47	male	1	39	8	0	6.75	.	12

-99 → .

Convert -99 to missing values



Data > Create or change data > Other variable-transformation commands > Change numeric values to missing

Convert -99 to missing values

```
mvdecode _all, mv(-99)
```
– recode all '-99' as '.'

```
mvdecode varname, mv(-99)
```
– for specific `varname` recode '-99' as '.'

Convert strings to missing values

- Blank string = empty quotes (“”)
- Number coded as a string, e.g. “-99”
- Need to process string variables during data cleaning anyway
  - Stata will generate missing values during cleaning for “”

- Recall from last week: `encode` for categorical variables coded as strings
  - Need to `recode` & modify the value label after `encode`

Recode missing values

- May want to recode missing values to non-missing
- For longitudinal datasets, can have missing data for visits after baseline for variables that do not change over time

`xfill varname, i(uniqueid)`

- `xfill` is a user-written STATA command that fills in missing values if the values are constant over a unique identifier

	ptid	visit	yearofbirth
1	1	1	1986
2	1	2	.
3	1	3	.
4	2	1	1988
5	2	2	.
6	2	3	.
7	3	1	1995
8	3	2	.
9	3	3	.
10	4	1	1975
11	4	2	.
12	4	3	.

ptid	visit	yearofbirth
1	1	1986
1	2	1986
1	3	1986
2	1	1988
2	2	1988
2	3	1988
3	1	1995
3	2	1995
3	3	1995
4	1	1975
4	2	1975
4	3	1975

Practice

- Find which variables have missing values
- Recode numeric variables with missing values
- Recode categorical variables with missing values
- Use a special missing value to indicate “refused to answer” for “lungcapacity” and “co2”
- Set any out-of-range age values to a special missing value to indicate “data error”
- Fill in missing values of “hospital” to match the value present for “hospid”

Duplicates

Look for duplicates in all or specific variables

Count duplicates across all variables:

```
duplicates report
```

Count duplicates in a particular variable or variables:

```
duplicates report varlist
```

Check to see if variable or combo of variables is unique identifier (you'll get an error if they are NOT unique)

```
isid varlist
```

Tag Duplicates

Tag duplicates so you can look at them

```
duplicates tag varlist, gen(newvar)
```

Drop all but first observation **CAUTION**

```
duplicates drop
```

Practice

- How many duplicates are there in pt1.dta across all variables?
- How many duplicate are there in pt1.dta on ptid?
- Drop duplicates across all variables
- Check to see if ptid and hospid are unique identifiers (together)

Linking do-files

Linking do-files

Often want to use the same do-file on multiple datasets

Can call one do-file from within another

Keeps your do-files succinct and tidy

Short, well-documented do-files are best

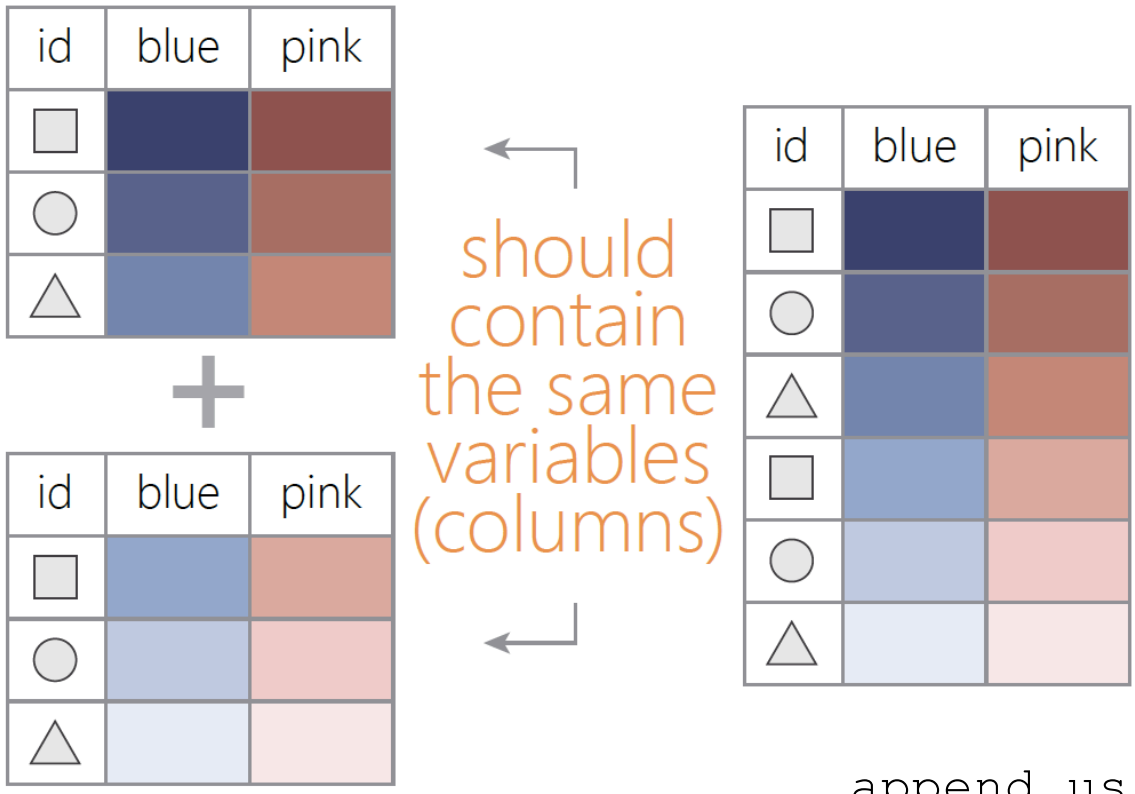
`quietly run "dofile.do"`

- `run` – will run a do file from within the command line or another do file

- Using the `quietly` option suppresses the output of the do-file that's running in the background – speeds things up

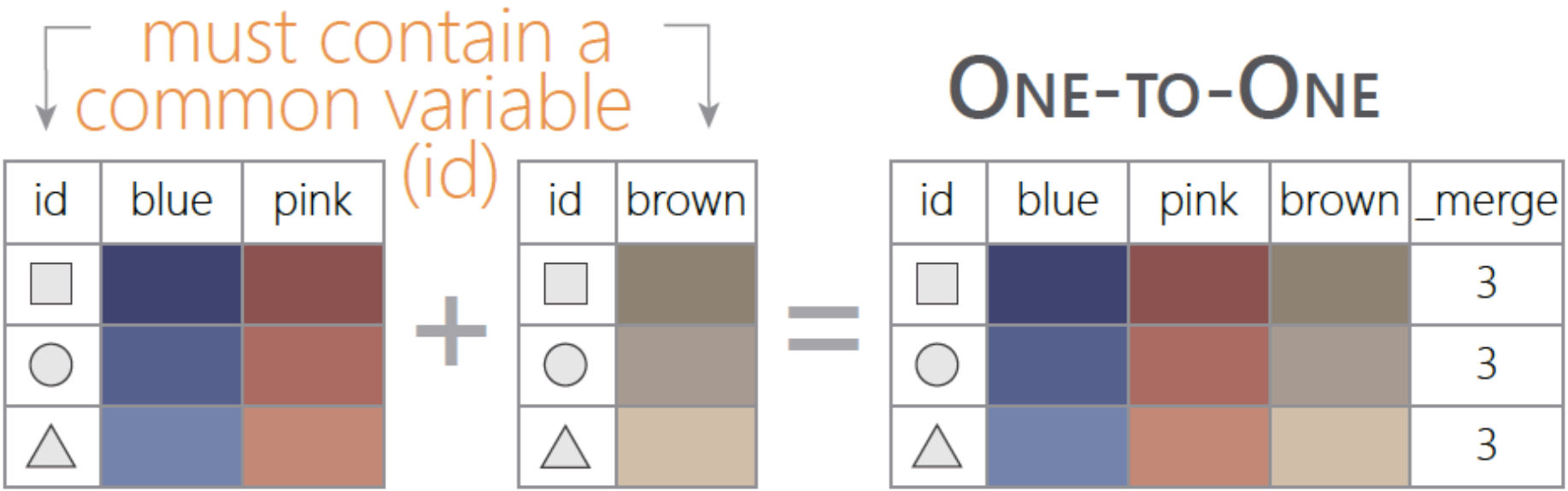
Combining Datasets

Adding (appending) new data: `append`




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

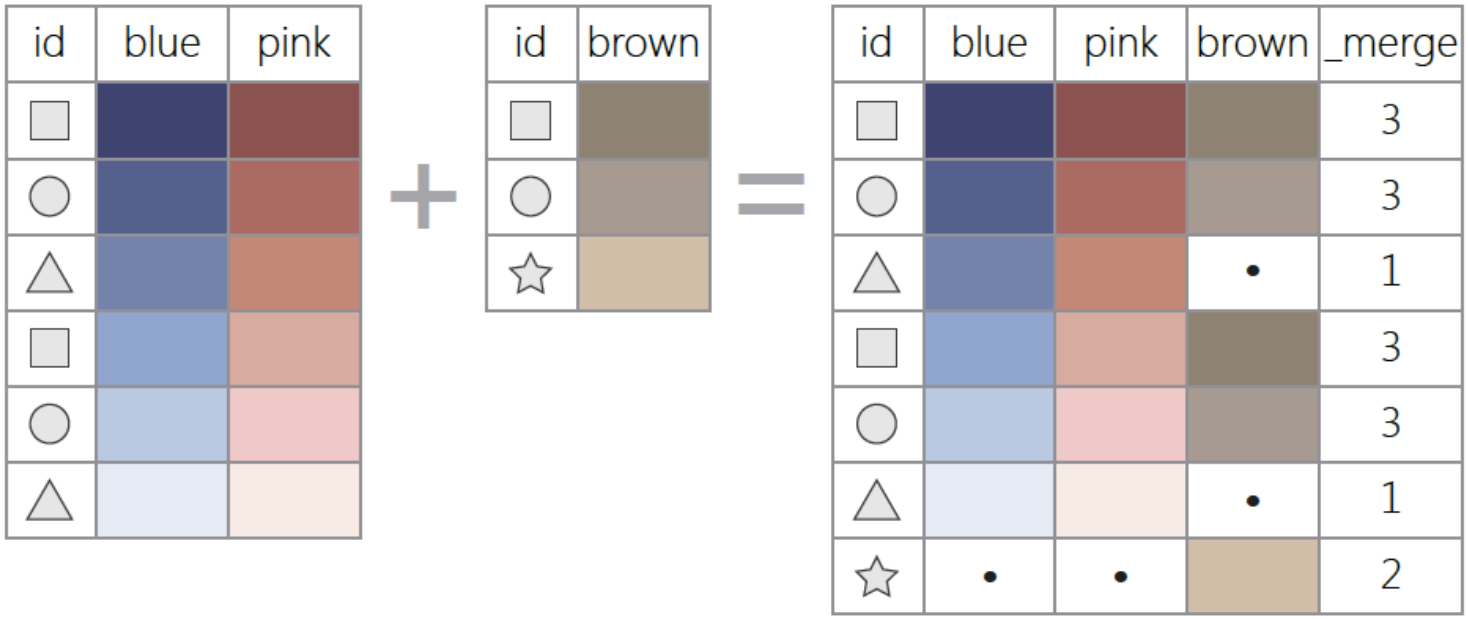
Merge data: One-to-One




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

-merge "using" dataset into loaded dataset

Merge data: Many-to-One




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

-merge "using" dataset into loaded dataset

Merge data: other options

- One-to-many
- Many-to-many

- Same principles/syntax structure as the previous examples

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

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

`_merge` variable

- Generated automatically with the `merge` command
- Tells you how the merge went
- 3 possible values:
  - 1 = observation found only in “master” dataset
  - 2 = observation found only in “using” dataset
  - 3 = observation found in both (i.e. the merge worked)
- Use `tab _merge` after a merge
  - Could use `browse _merge` for small datasets
  - Helpful to `sort` data first on matching variable prior to using `browse`

Practice

- Clean pt1.dta and pt2.dta and save them as clean datasets within a new, separate do-file
- Append pt2_clean.dta to pt1_clean.dta
- Save the combined dataset
- Merge the combined dataset with the doctors.dta dataset
- Drop any observations that were only in the doctors.dta dataset
- Save the final dataset

Reshaping Data

Reshape

LONG

id	male	visit	weight
1	0	1	158
1	0	2	155
1	0	3	155
1	0	4	156
2	1	1	289
2	1	2	293
2	1	3	290
2	1	4	287
3	1	1	318
3	1	2	315
3	1	3	315
3	1	4	312

WIDE

id	weight1	weight2	weight3	weight4	male
1	158	155	155	156	0
2	289	293	290	287	1
3	318	315	315	312	1

```
reshape wide weight, i(id) j(visit)
```

Reshape

LONG

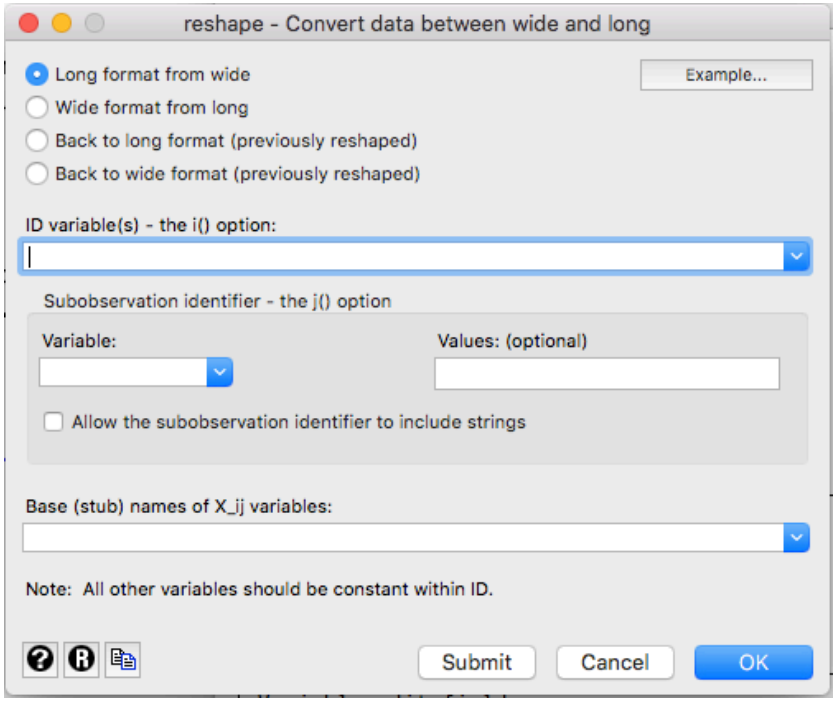
id	male	visit	weight
1	0	1	158
1	0	2	155
1	0	3	155
1	0	4	156
2	1	1	289
2	1	2	293
2	1	3	290
2	1	4	287
3	1	1	318
3	1	2	315
3	1	3	315
3	1	4	312

WIDE

id	weight1	weight2	weight3	weight4	male
1	158	155	155	156	0
2	289	293	290	287	1
3	318	315	315	312	1

```
reshape long weight, i(id) j(visit)
reshape long
```

Reshape



Data > Create or change data > Other variable-transformation commands > Convert data between wide and long

Practice

- Open “reshape_example.dta”
- Reshape the data from long to wide format

Next Week

Data Visualization