Lecture 3: Data cleaning

BIOSTAT 212, Summer 2018

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

Final Project

- Due 9/18 @ 9 PM
- 100 pts
- Create a publication quality table and graph
- Instructions and example on the course website
- Health-related data set (ideally your own data)

Doctoral Program in Epidemiology and Translational Sciences

Missing Data and Multiple Imputation with Stata

Chuck Huber, PhD
Senior Statistician, Stata Corp LP
Adjunct Associate Professor of Biostatistics, Texas A&M School of Public Health

In this talk, Chuck will introduce different kinds of mechanisms that lead to missing data such as missing at random (MAR), missing completely at random (MCAR), and missing not at random (MNAR). He will then demonstrate how to use Stata's -mi- commands to impute missing data and fit models using the imputed datasets.

August 20
12:00 to 1:00 pm
Mission Hall-1407

Pizza lunch.

We would appreciate RSVPs so we can ensure the appropriate room capacity

RSVP: For Aug. 20th
https://bit.ly/2uHlV4u

Sponsored by the UCSF PhD program in Epidemiology and Translational Science

UCSF Department of Epidemiology & Biostatistics

Doctoral Program in Epidemiology and Translational Sciences

Multilevel/Longitudinal Modeling with Stata

Chuck Huber, PhD
Senior Statistician, Stata Corp LP
Adjunct Associate Professor of Biostatistics, Texas A&M School of Public Health

In this talk, Chuck will introduce the concepts and jargon of multilevel modeling for nested and longitudinal data. Chuck will demonstrate how to fit multilevel/longitudinal models using Stata's -mixed- command, and how to visualize the results using Stata's -predict-, -twoway-, -margins-, and -marginsplot- commands.

August 21
12:00 to 1:00 pm
Mission Hall-1407

Pizza lunch.

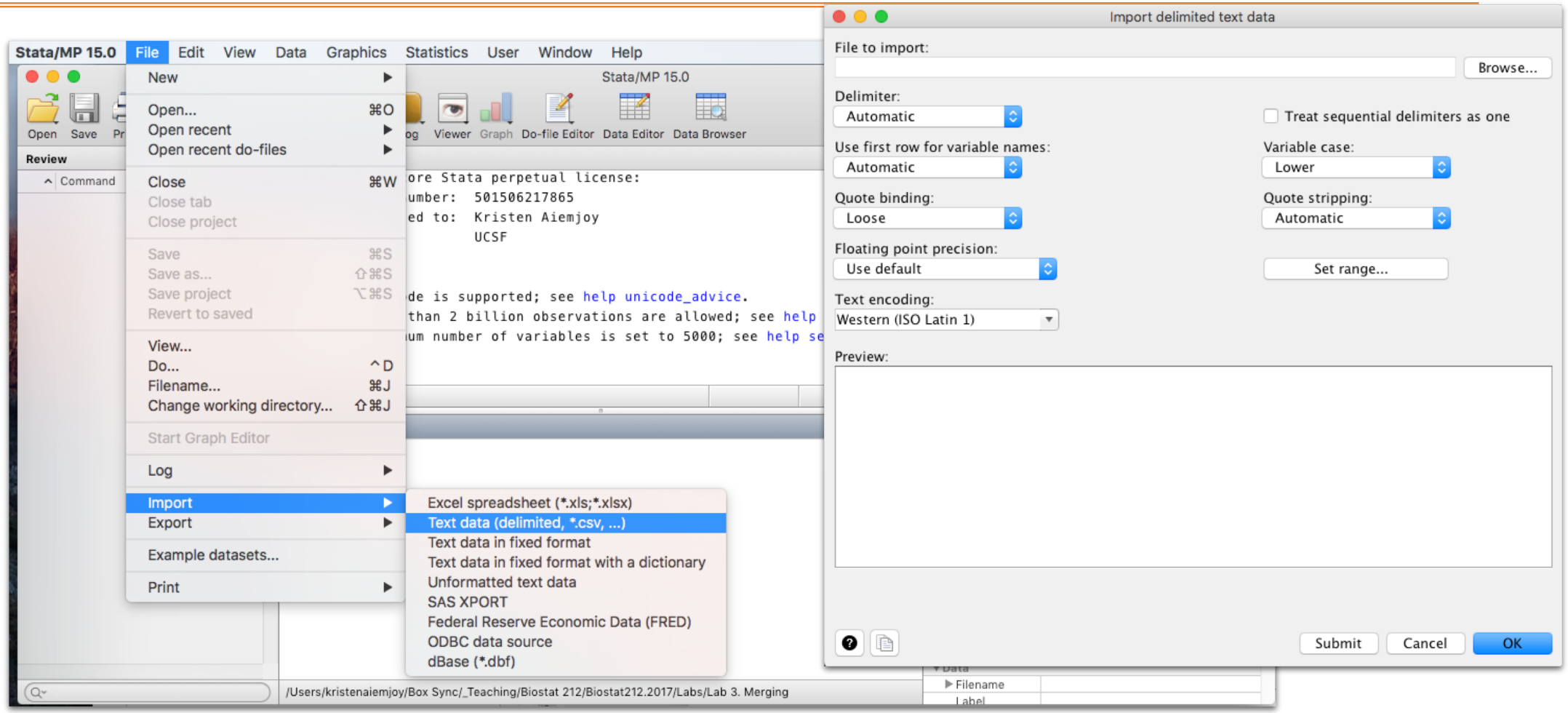
We would appreciate RSVPs so we can ensure the appropriate room capacity

RSVP: https://bit.ly/2NRWkyB

Sponsored by the UCSF PhD program in Epidemiology and Translational Science

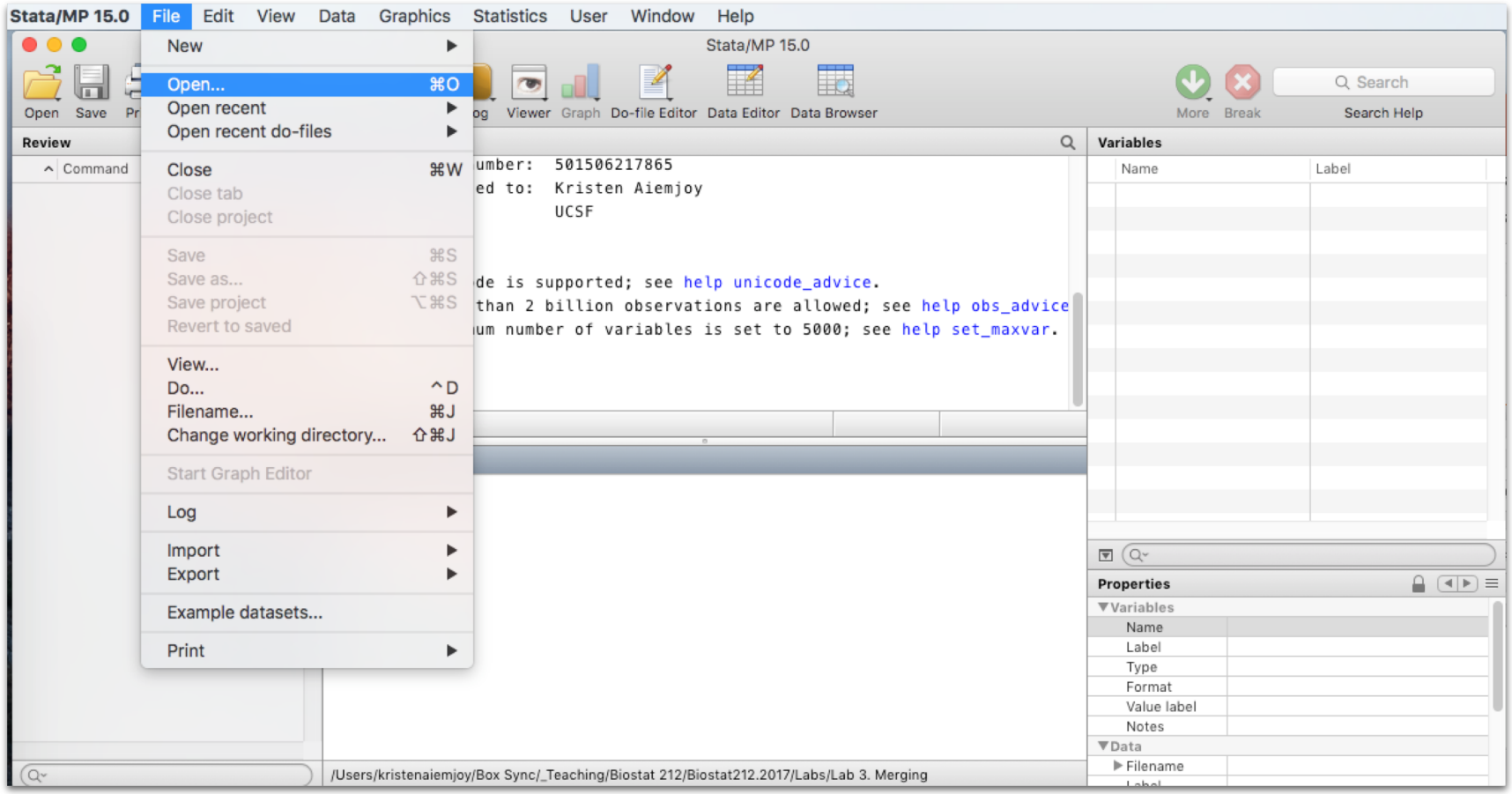
UCSF Department of Epidemiology & Biostatistics

Review: Import .csv file




```
import delimited acupuncture_raw.csv, clear
```

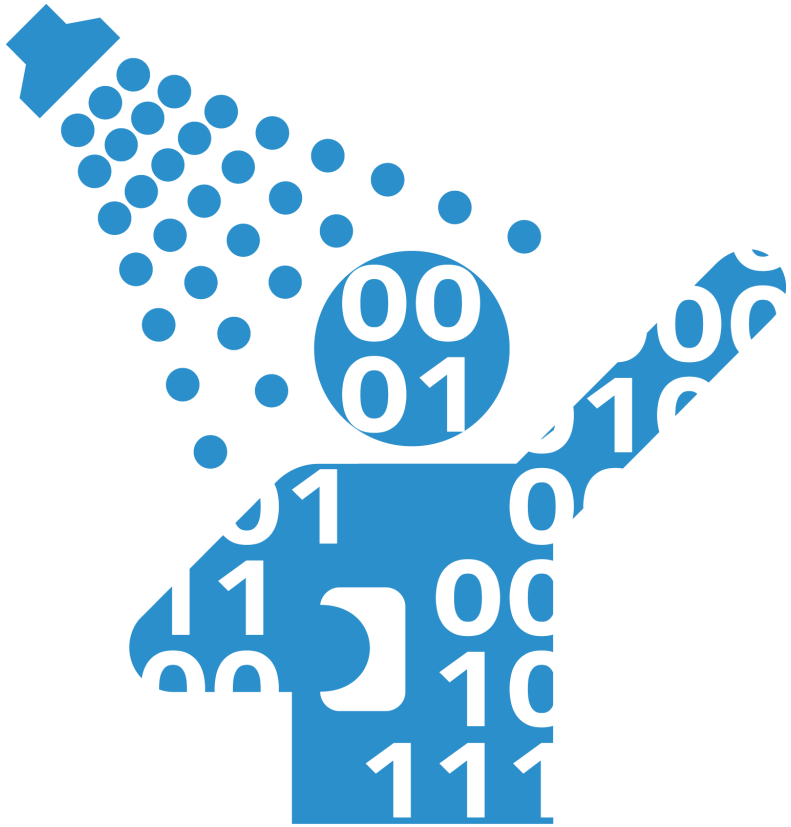
Review: Open Stata data set




```
use acupuncture.dta, clear
```

Clean data

- The data sets we have worked with (nhanes2.dta & acupuncture_baseline.dta) were clean
  - The data came as a .dta file
  - The variables were ready to analyze
  - Most variables were labeled



In the real world:

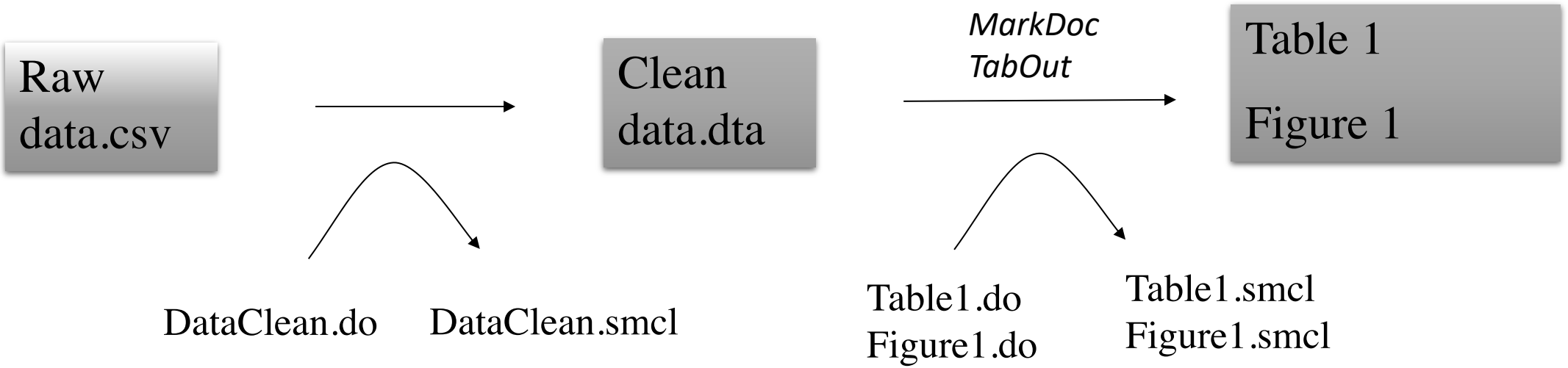
- Your data will arrive:
  - On paper forms
  - In a text file (comma or tab delimited)
  - In Excel
  - In Access
  - In another data format (SAS, etc)

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
- 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 should 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 */
```

Example workflow



Outline for today

- Name and label variables
- Checking ranges and removing outliers
- Missing data
- Generating variables
- Convert string variables to numeric
- Subset data
- Adding new data (append & merge)
- Reshape
- Dates

Load dataset for lecture today

- Download the “acupuncture_raw.csv” dataset from the course website

- Import the data set into Stata

```
import delimited acupuncture_raw.csv, clear
```

-or-

File<Import<Text data (*.csv)

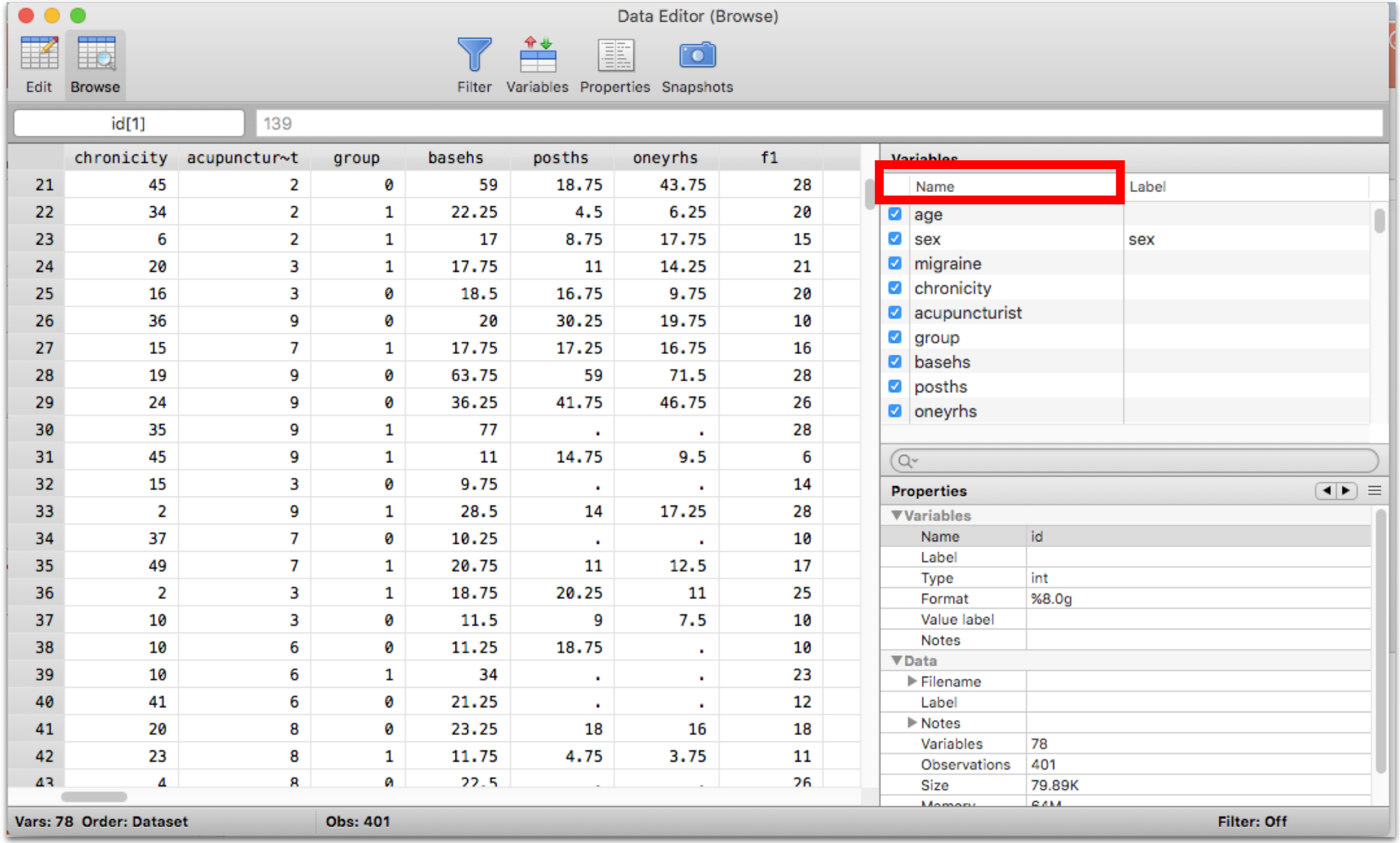
Outline for today

- **Name and label variables**
- Checking ranges and removing outliers
- Missing data
- Generating variables
- Convert string variables to numeric
- Subset data
- Adding new data (append & merge)
- Reshape
- Dates

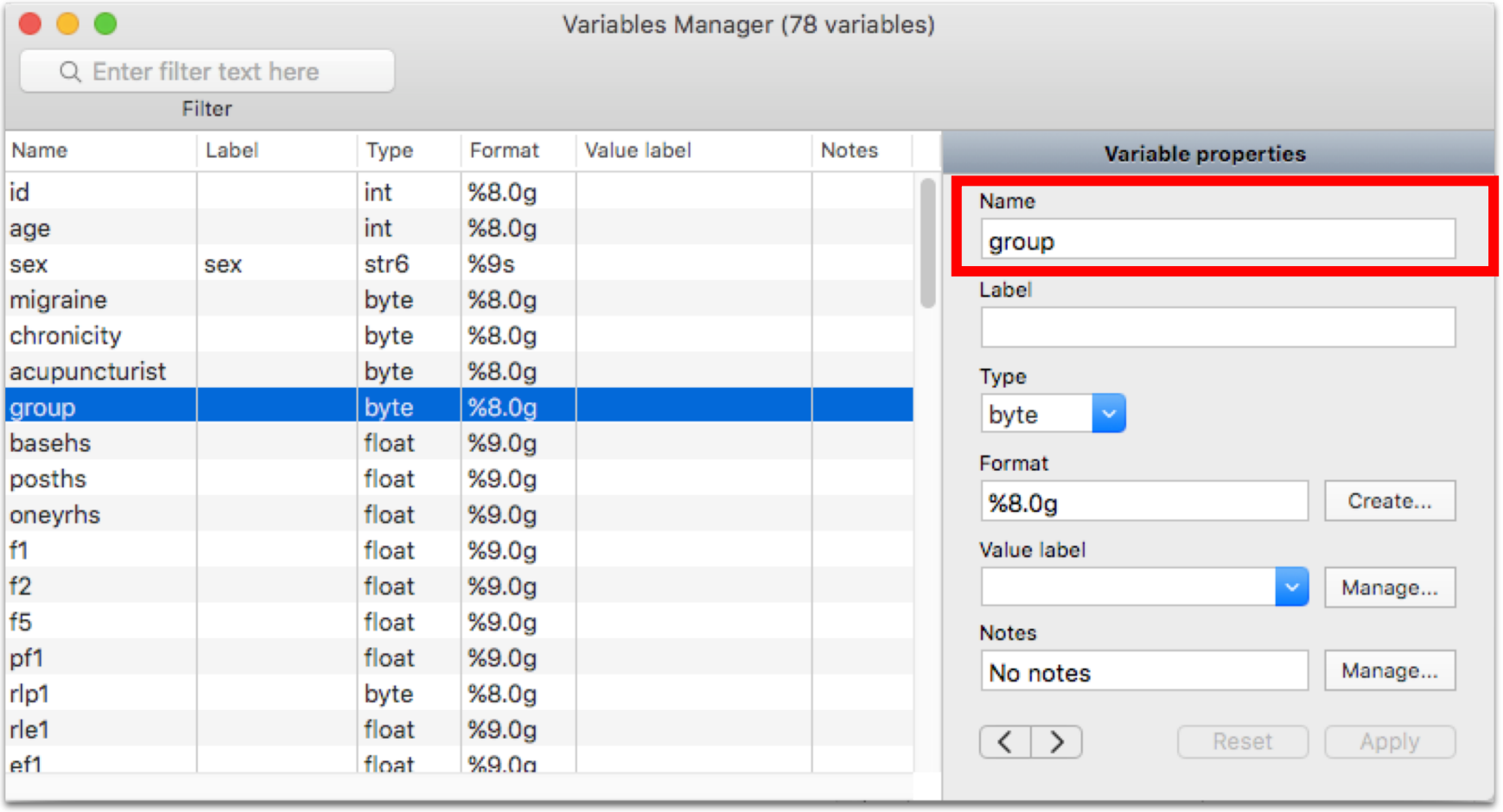
Name and label variables

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

Variable names: keep short and sweet!



Change variable names



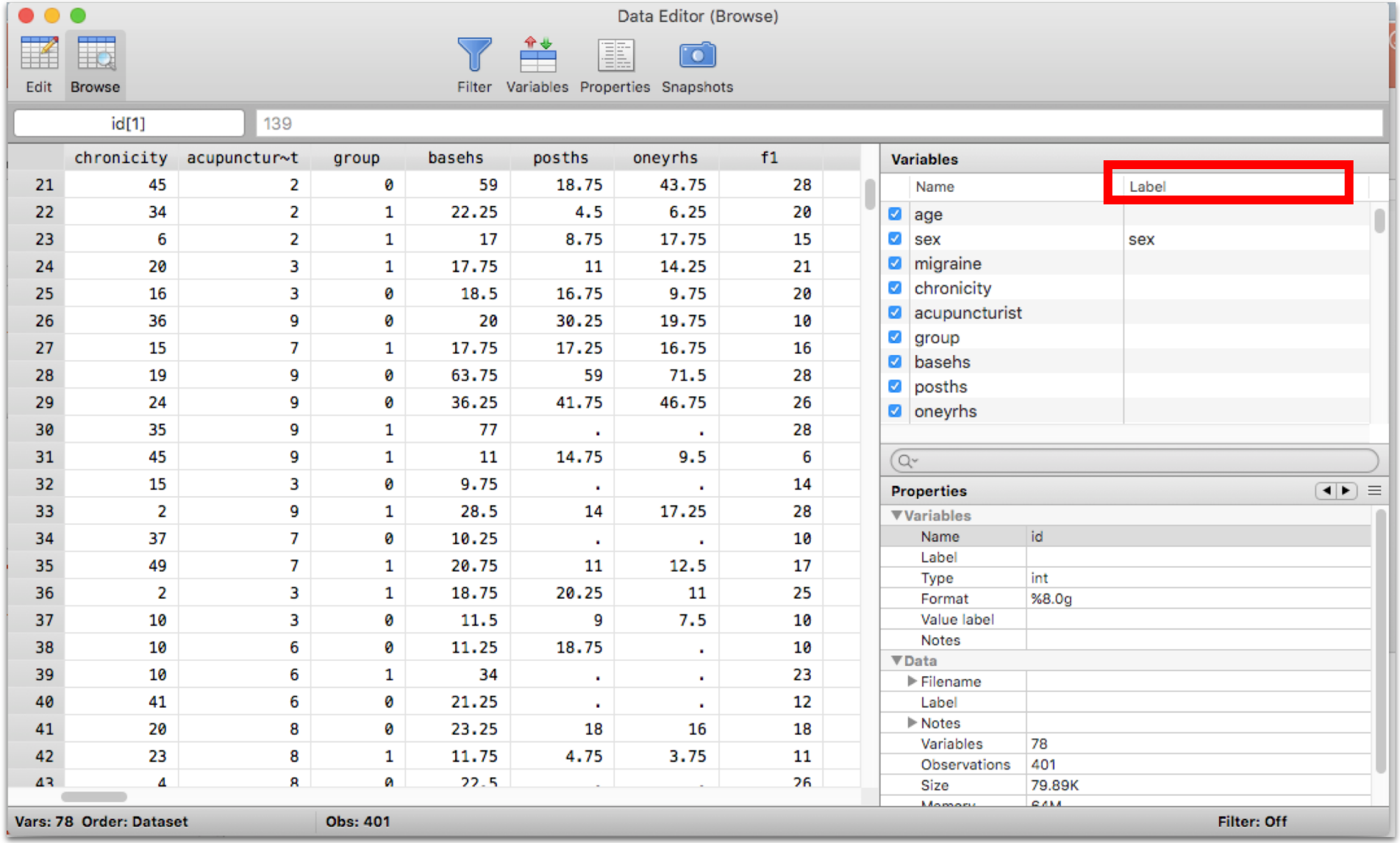
Data >Variable manager

Change variable names

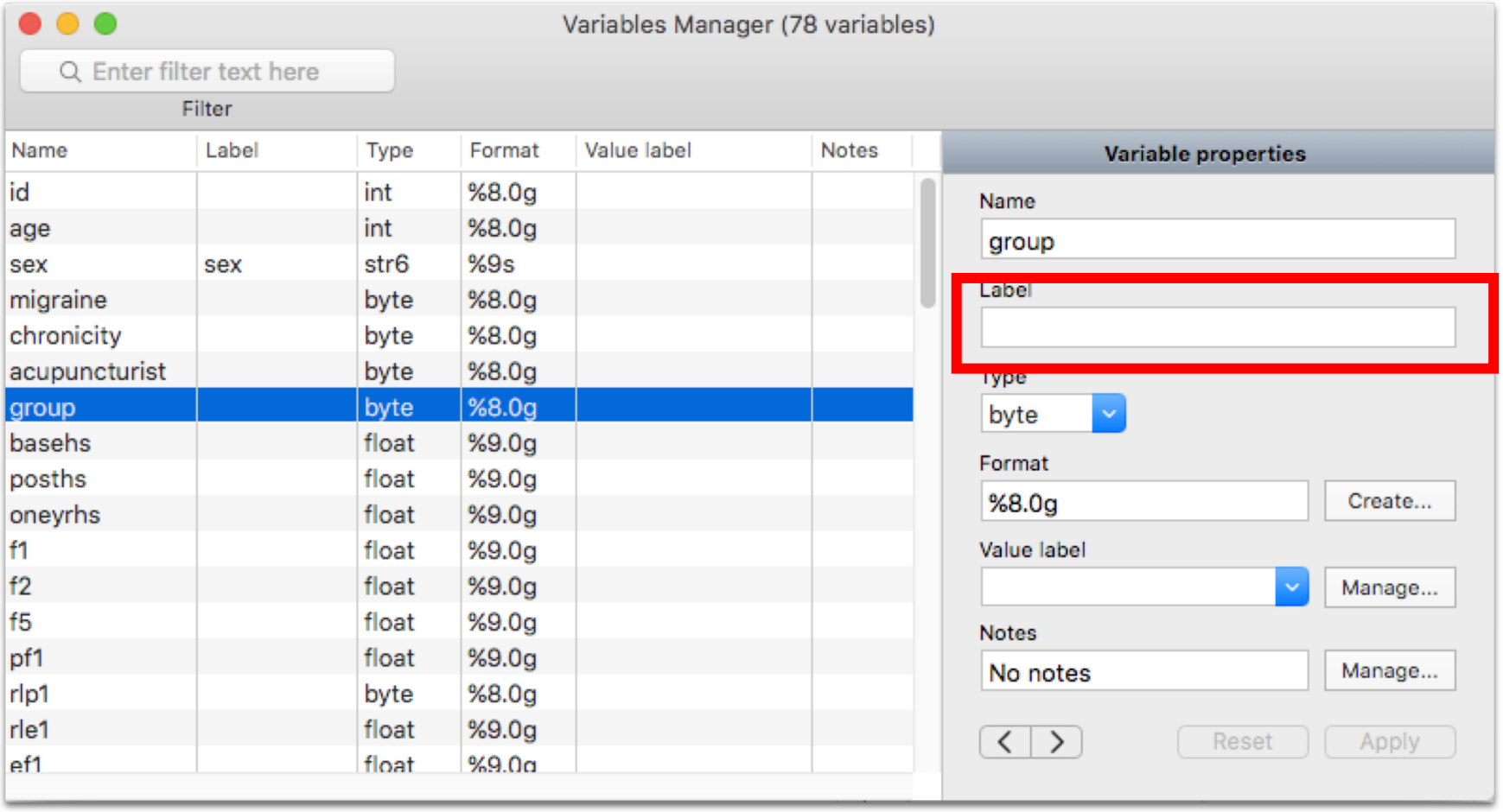
```
rename old_varname new_varname
```

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

Variable labels: help you remember details



Change variable label



Data >Variable manager

Change variable label

```
label variable varname "label name"
```

*label name can have spaces, but must be in quotes

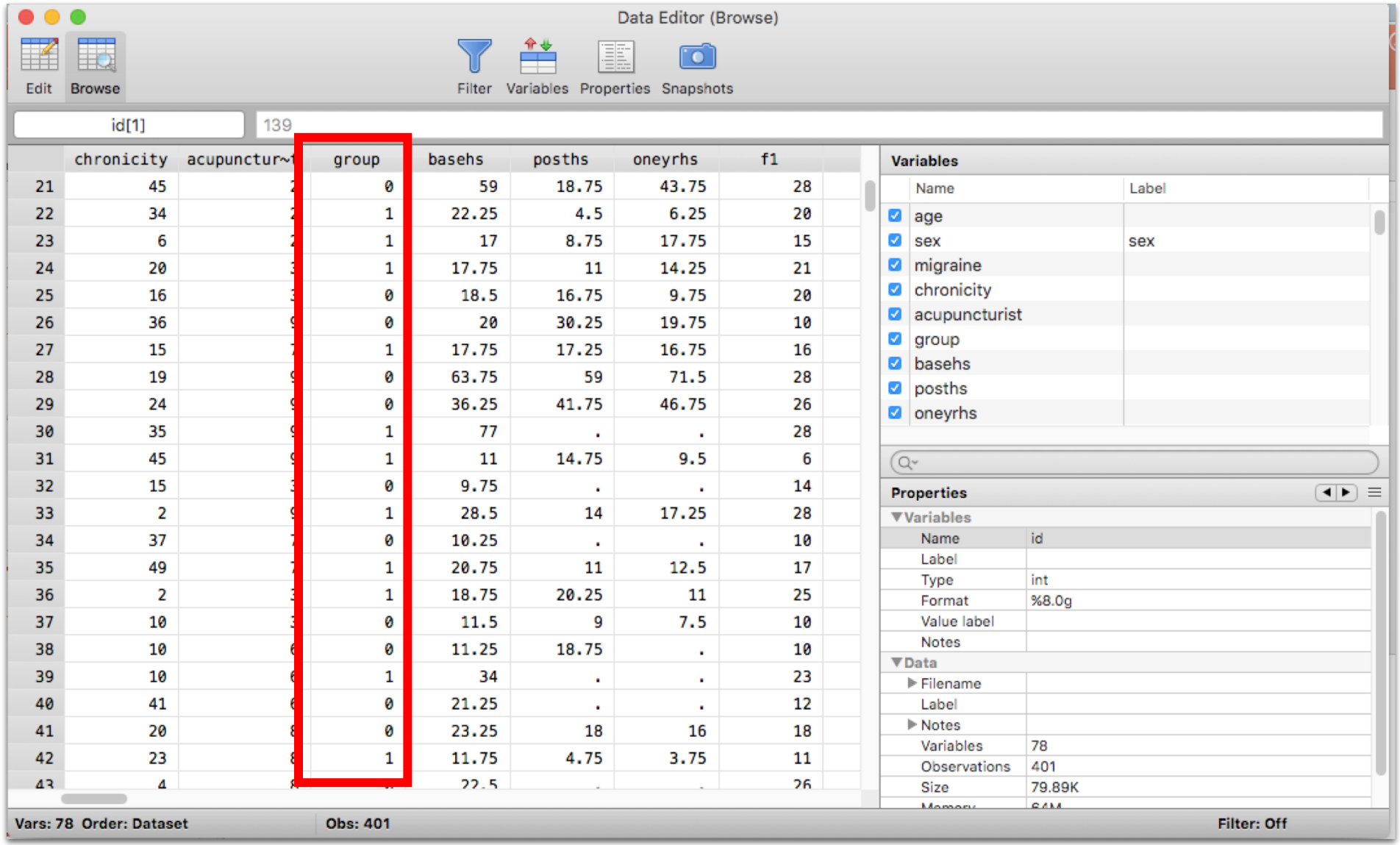
Practice

Variable f1 = headache frequency at baseline

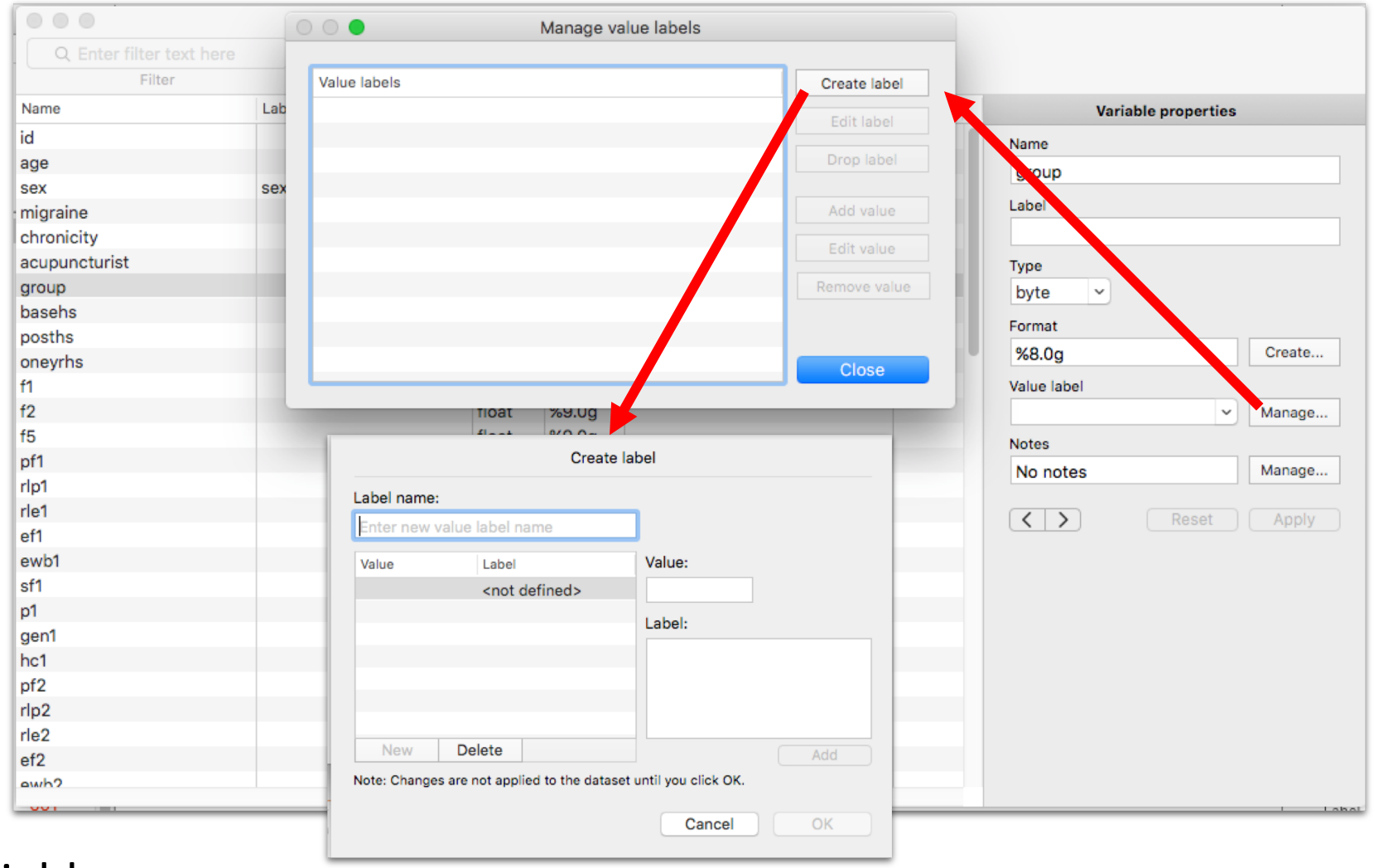
1. Rename f1 → hfreq_bl
2. label f1 = headache frequency at baseline

```
rename f1 hfreq_bl
label variable hfreq_bl "headache frequency baseline"
```

Value labels



Add value labels



Data >Variable manager

Add value labels

Label values of categorical variables

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

example

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

Practice

- Label group (1=intervention) (2=control)

```
tab group
label define grouplabel 1 “intervention” 0 “control”
label values group grouplabel
tab group
```

Displaying numeric values for labels

`numlabel, add`

```
. tab sex

  1=male,
 2=female |      Freq.     Percent        Cum.
----------+-----------------------------------
   1. Male |      4,915       47.48       47.48
 2. Female |      5,436       52.52      100.00
----------+-----------------------------------
     Total |     10,351      100.00
```

`labelbook` – gets you a list of all numeric labels and their values

Outline for today

- Name and label variables
- **Checking ranges and removing outliers**
- Missing data
- Generating variables
- Convert string variables to numeric
- Subset data
- Adding new data (append & merge)
- Reshape
- Dates

Range checks

- Are there any outliers? (unrealistic values?)
  - `codebook varname`
  - `inspect varname`
  - `histogram varname`
  - `summarize varname`
  - `spikeplot varname`

Removing outliers: `replace`

```
replace varname = # if condition
```

– replace the value of a variable meeting a certain condition

Example

```
replace weight = 230 if id == "13des4"
replace age = . if age>110
```

Practice

- Are there any outliers for age?

```
spikeplot age
sum age
list if age >120
```

- Replace outlier

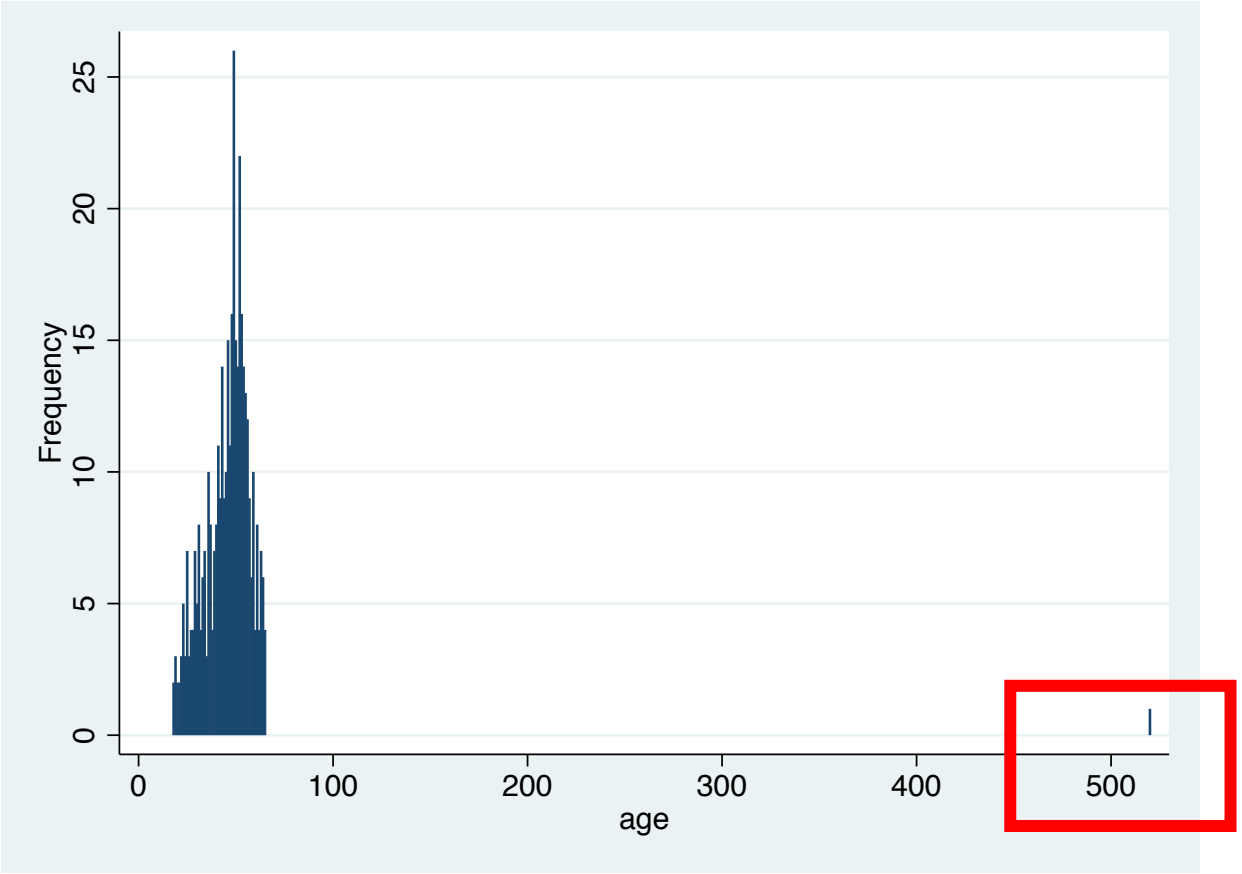
```
replace age = 52 if id==363
```

-or-

```
replace age = 52 if age>115
spikeplot age
```

Range checks

```
spikeplot age
```



Outline for today

- Name and label variables
- Checking ranges and removing outliers
- **Missing data**
- Generating variables
- Convert string variables to numeric
- Subset data
- Adding new data (append & merge)
- Reshape
- Dates

Missing data

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

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

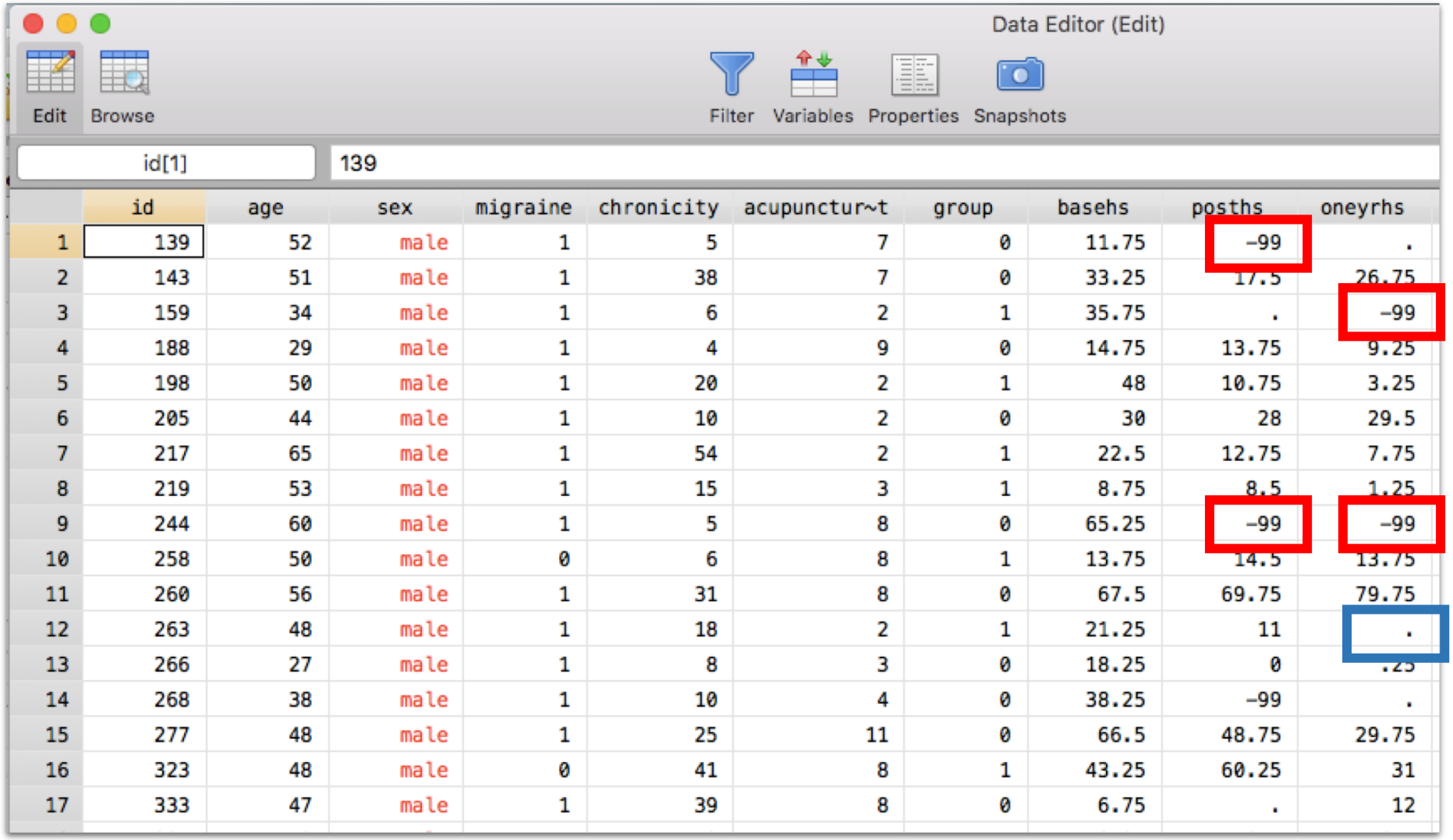
Missing data

- Is there missing data?

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

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

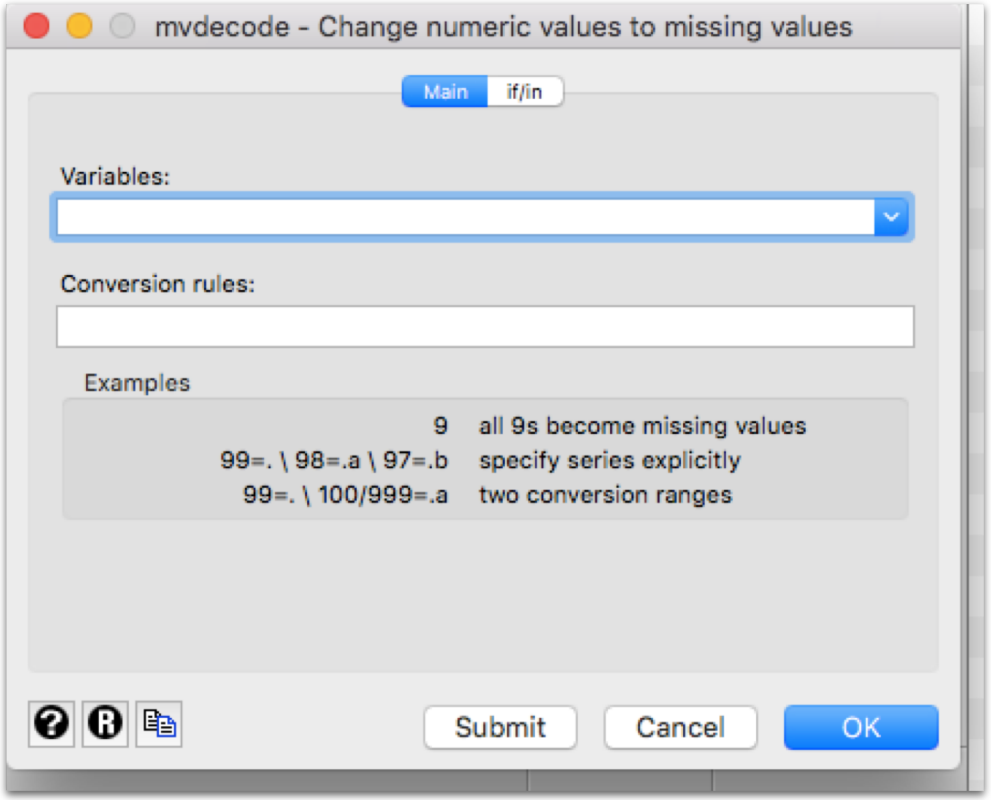
Convert -99 to missing values



	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 ‘.’

Practice

- Look for missing data

```
misstable sum
```

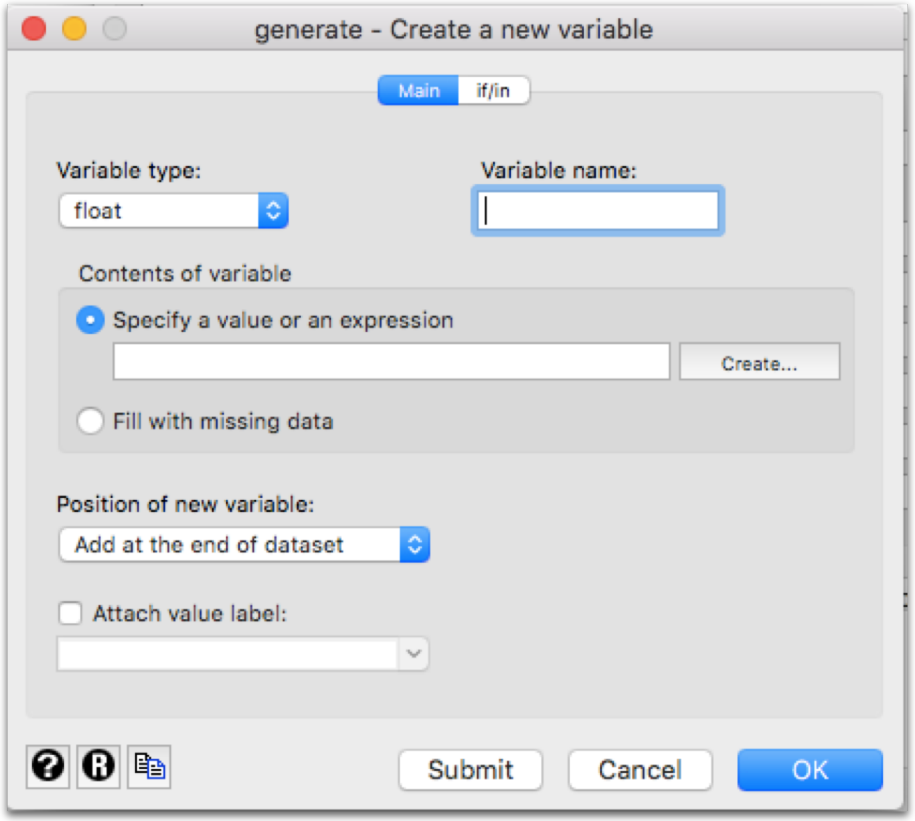
- Change all ‘-99’ to (.)

```
mvdecode _all, mv(-99)
misstable sum
```

Outline for today

- Name and label variables
- Checking ranges and removing outliers
- Missing data
- **Generating variables**
- Convert string variables to numeric
- Subset data
- Adding new data (append & merge)
- Reshape
- Dates

Create a new variable



Data > Create or change data > Create a new variable

Create a new variable

`generate new_varname = exp` -generate a new variable

`exp = #, ".", _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

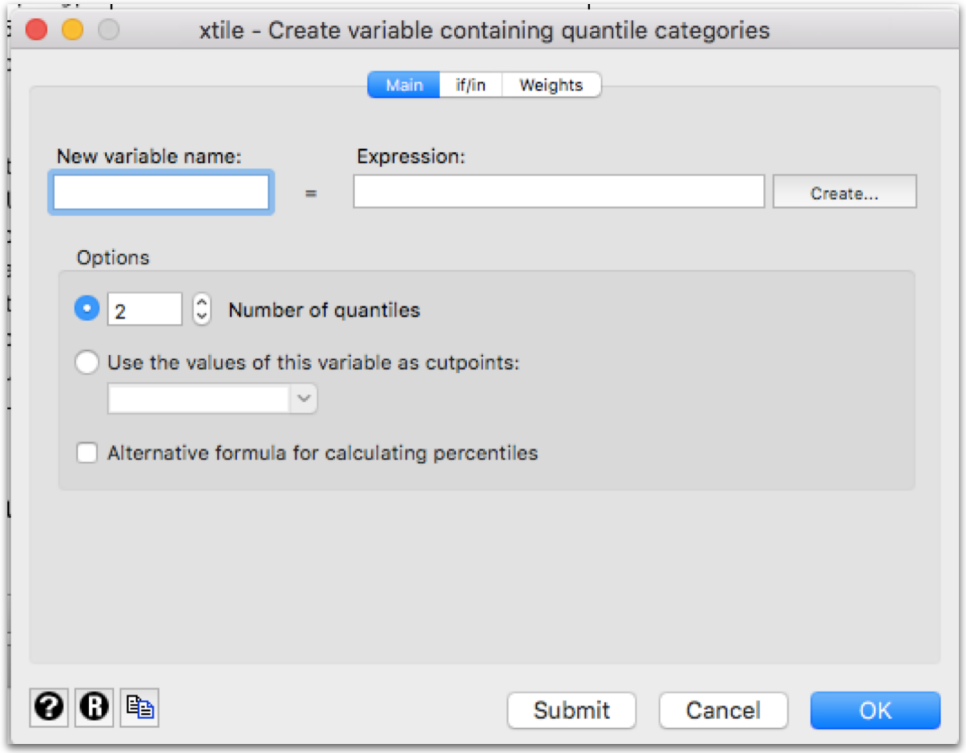
- Generate a new running id variable and a blank variable

```
gen newid = _n
gen blank = .

browse newid blank
```

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

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

tab severemigraine
```

Create variable containing quantile categories



Statistics > Summaries, tables, and tests > Summary and descriptive statistics > Create variable of quantiles

Create variable containing quantile categories

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

Options

`if` –use logic operators

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

Practice

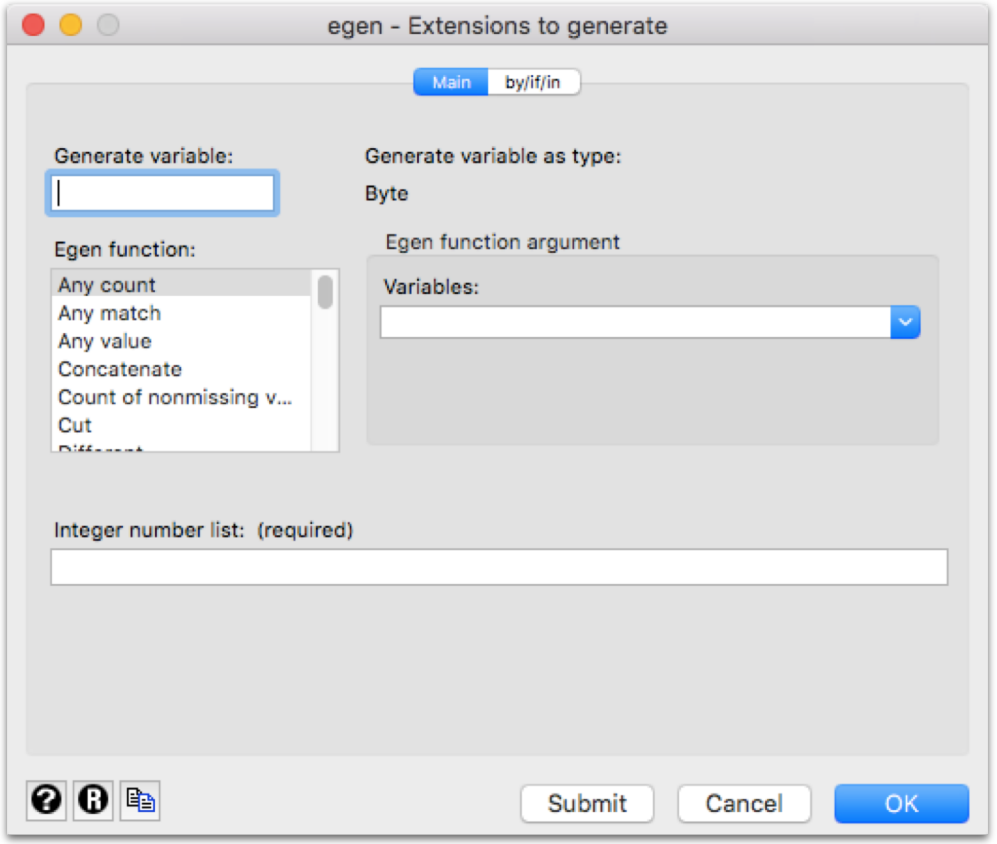
- Create a new categorical variable that is a quartile of baseline headache square.

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

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

```
bysort hs_quartile: sum basehs
```

Create new variable (extended)



Data > Create or change data > Create new variable (extended)

Create new variable (extended)

```
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

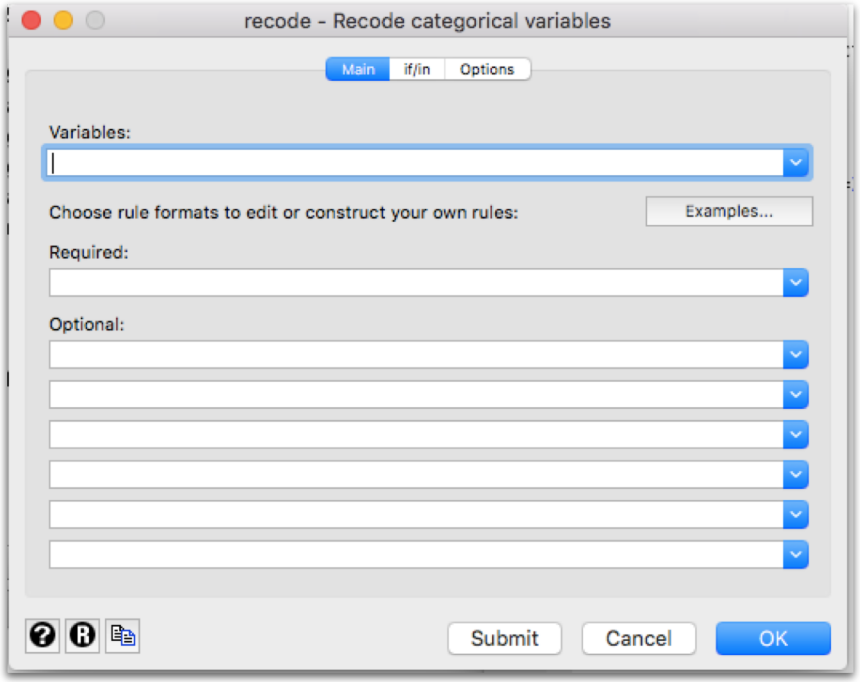
*many more functions → look in help file

Practice

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

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

Create categorical variable from continuous variable



Data > Create or change data > Other variable-transformation commands > Recode categorical variable

Create categorical variable from continuous variable

```
recode old_varname (min/# = 0 “value label”)(#/# = 1 “value
label”)(#/max = 2 “value label”), gen(new_varname)
```

Practice

- Create a dichotomous variable for age (>50)

```
recode age (min/49 = 1 "less than 50") (50/max = 2 "50
and older"), gen(age2)
```

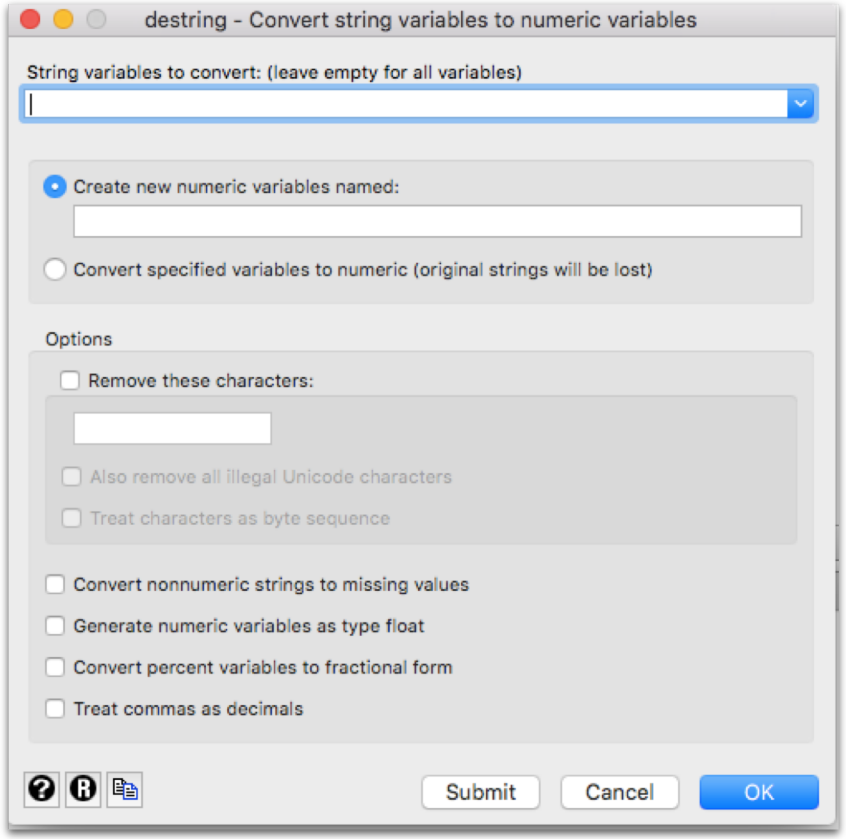
- How many study participants are 50 years old or older?

```
tab age2
```

Outline for today

- Name and label variables
- Checking ranges and removing outliers
- Missing data
- Generating variables
- **Convert string variables to numeric**
- Subset data
- Adding new data (append & merge)
- Reshape
- Dates

Convert string variables to numeric



Data > Create or change data > Other variable-transformation commands > Convert variables from string to numeric

Convert string variables to numeric

```
destring varname, generate(new_varname)
```

-convert string into numeric variables

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

-Encode string into numeric variable with labels

`destring` or `encode`?

`encode` and `destring` both convert string variables to numeric variables. What's the difference?

- Use `destring` to:
  - convert intrinsically numeric variables such as **age** that do not have labels but happen to be stored as strings.
- Use `encode` to:
  - convert categorical variables that will be included in statistical models but have been stored as strings. **encode** can convert the categories to value labels for the new numeric variable.

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
```

Outline for today

- Name and label variables
- Checking ranges and removing outliers
- Missing data
- Generating variables
- Convert string variables to numeric
- **Subset data**
- Adding new data (append & merge)
- Reshape
- Dates

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 can't 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`

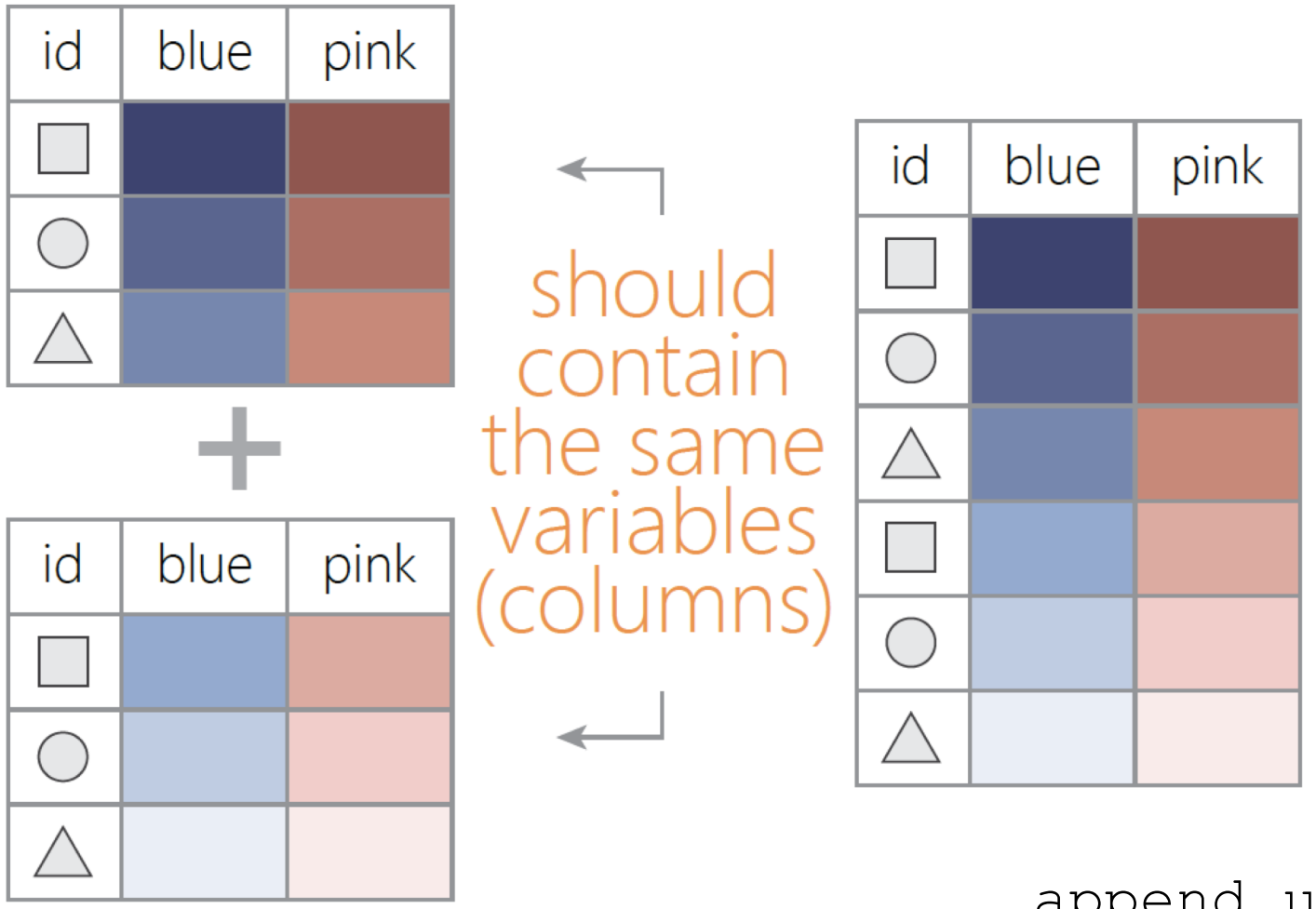
`keep varlist` -keep one or more variables, all others dropped!

**careful! you can't undo

Outline for today

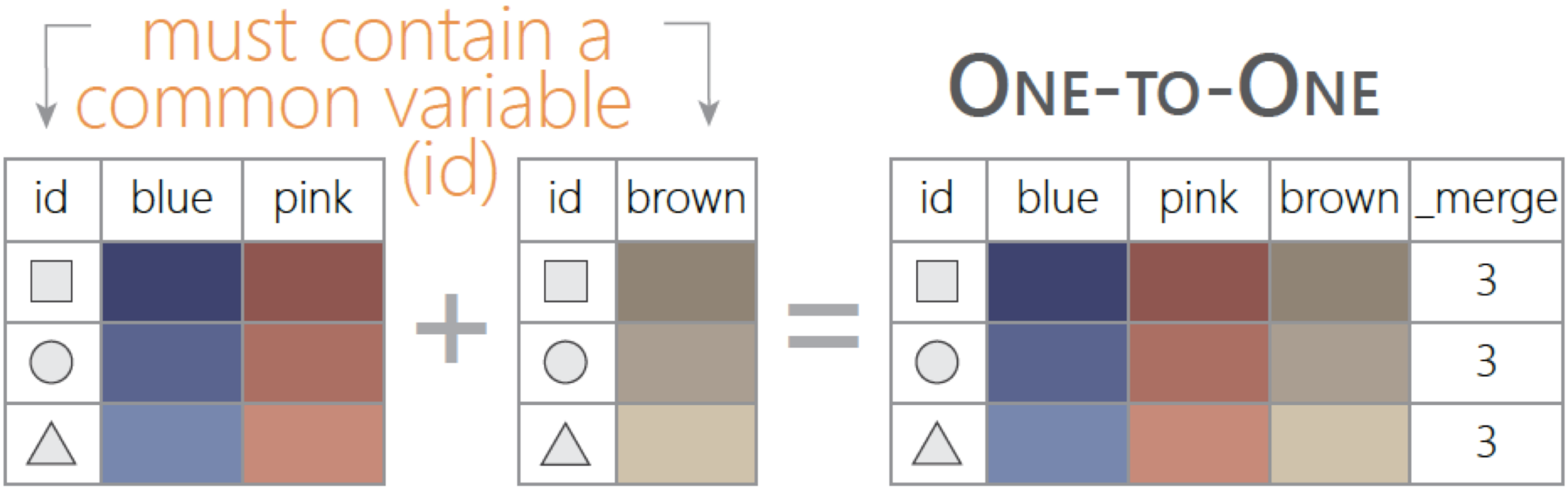
- Name and label variables
- Checking ranges and removing outliers
- Missing data
- Generating variables
- Convert string variables to numeric
- Subset data
- **Adding new data (append & merge)**
- Reshape
- Dates

Adding (appending) new data: `append`




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

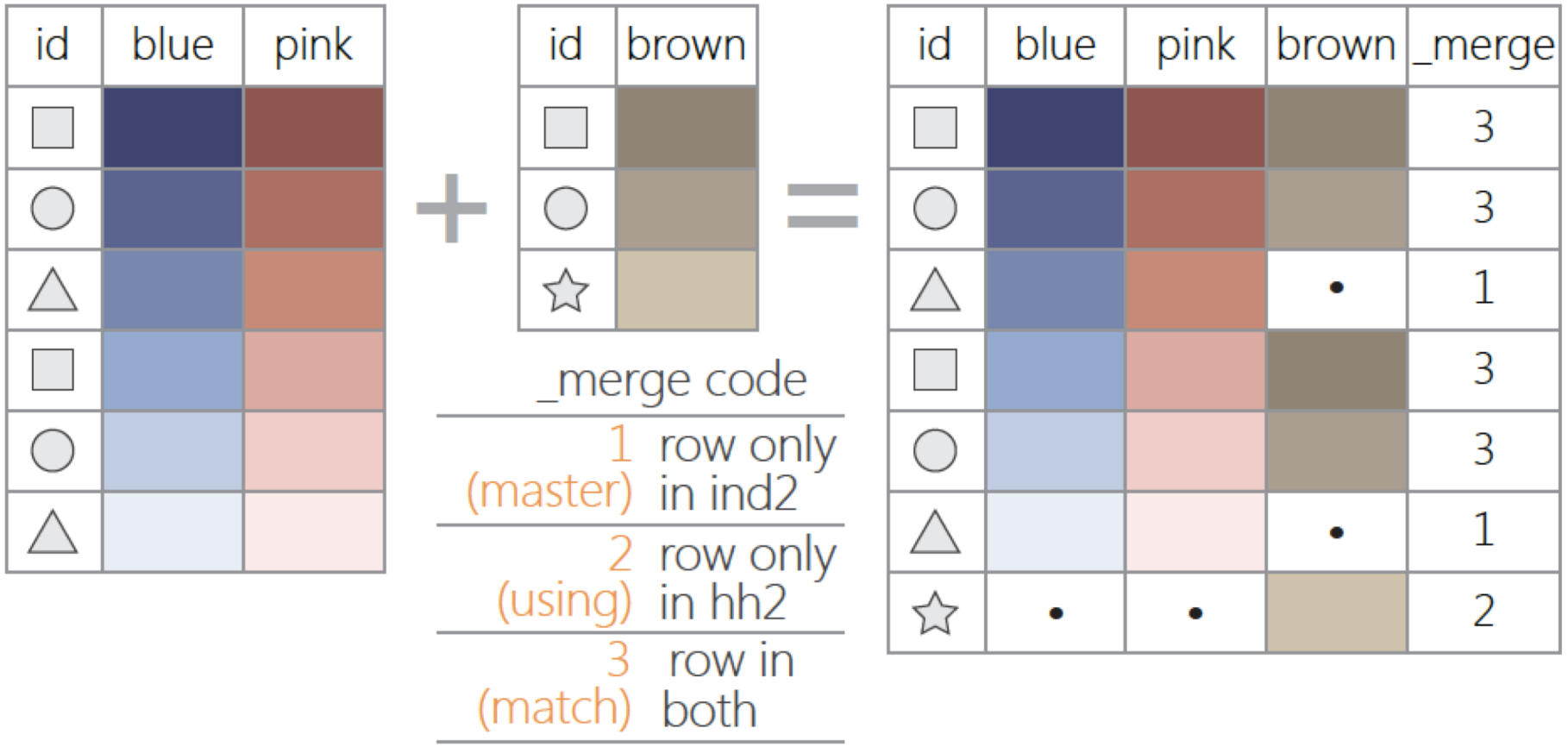
Merge data: One-to-One




```
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

Outline for today

- Name and label variables
- Checking ranges and removing outliers
- Missing data
- Generating variables
- Convert string variables to numeric
- Subset data
- Adding new data (append & merge)
- **Reshape**
- Dates

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

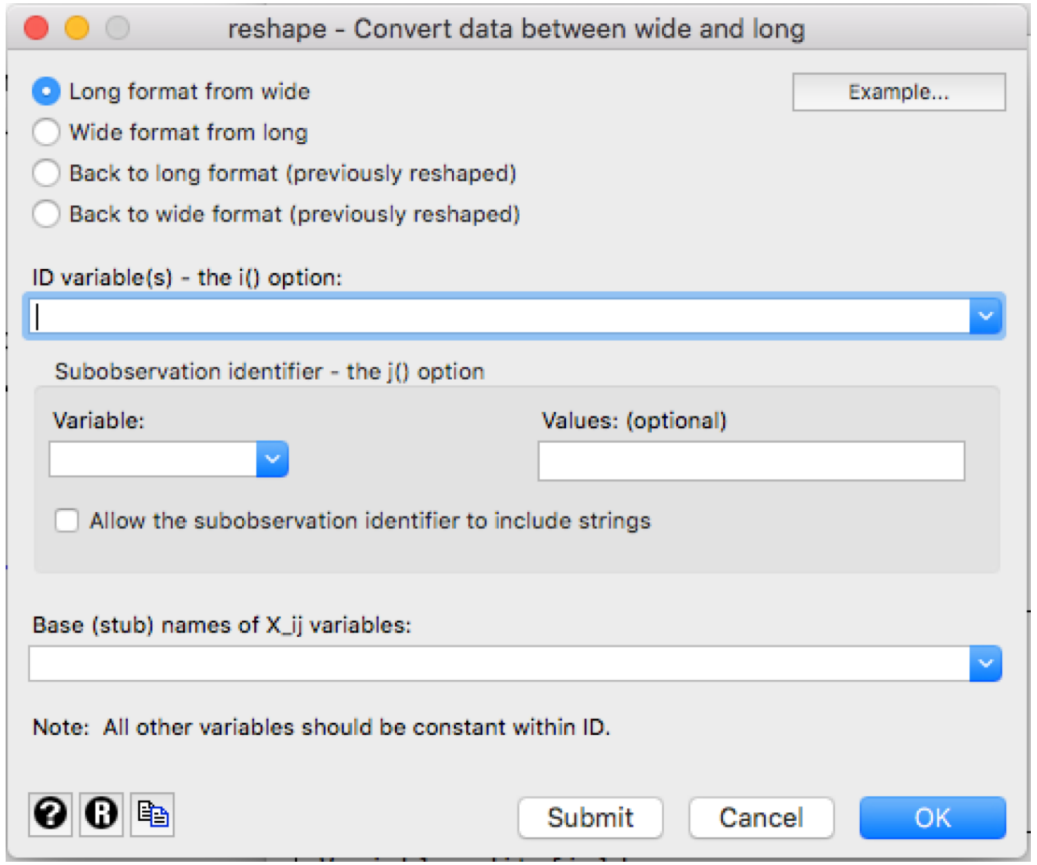
id	male	visit	weight
1	0	1	158
1	0	2	155
1	0	3	155
1	0	4	156
2	1	1	289
2	1	2	293
2	1	3	290
2	1	4	287
3	1	1	318
3	1	2	315
3	1	3	315
3	1	4	312

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

Outline for today

- Name and label variables
- Checking ranges and removing outliers
- Missing data
- Generating variables
- Convert string variables to numeric
- Subset data
- Adding new data (append & merge)
- Reshape
- **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 the date format

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

Dates

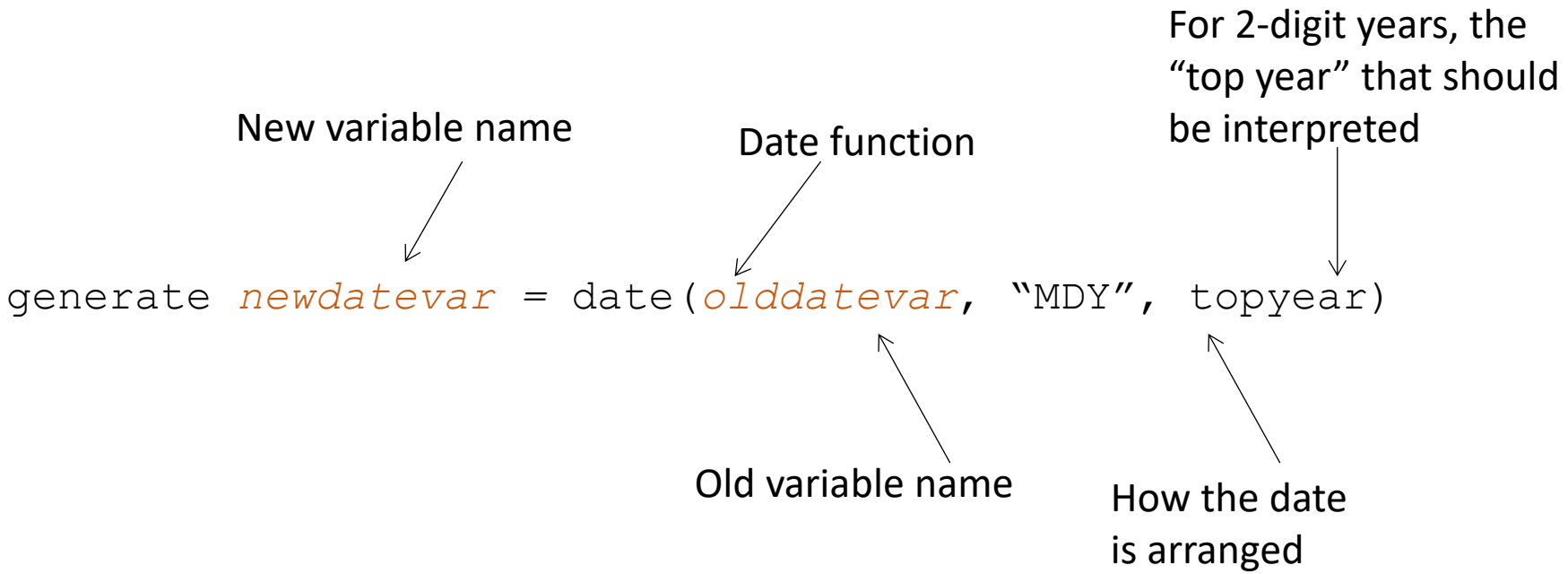
- Stata recodes dates as # of days elapsed from a reference date
  - January 1, 1960 = 0
  - January 2, 1960 = 1
  - January 3, 1960 = 2
  - December 31, 1960 = 364

- Makes it “easy” for Stata to manipulate dates mathematically

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

Dates

Generate new date variable



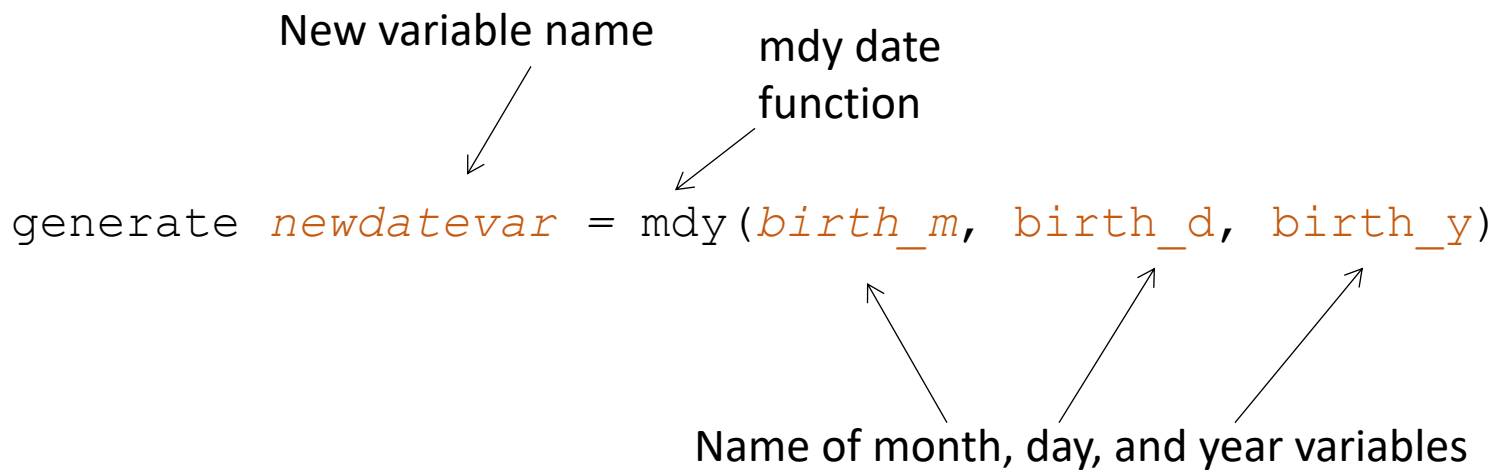
Credit: Biostat 212 TAs Sankari Ayyaluru & Yiwey Shieh, MD

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 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 reformat using the following command:

```
format newdatevar %td
```

  - %td = regular date (other formats available using STATA help)

Dates — Mathematical operations

- We can add and subtract numbers to dates and find time 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

```
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)
```

*0(Sunday) - 6(Saturday)

- Month

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

- Year

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