Lecture 4: Data Cleaning Part 2

BIOSTAT 212

Crystal Langlais, MPH, PhD(c)
Crystal.Langlais@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

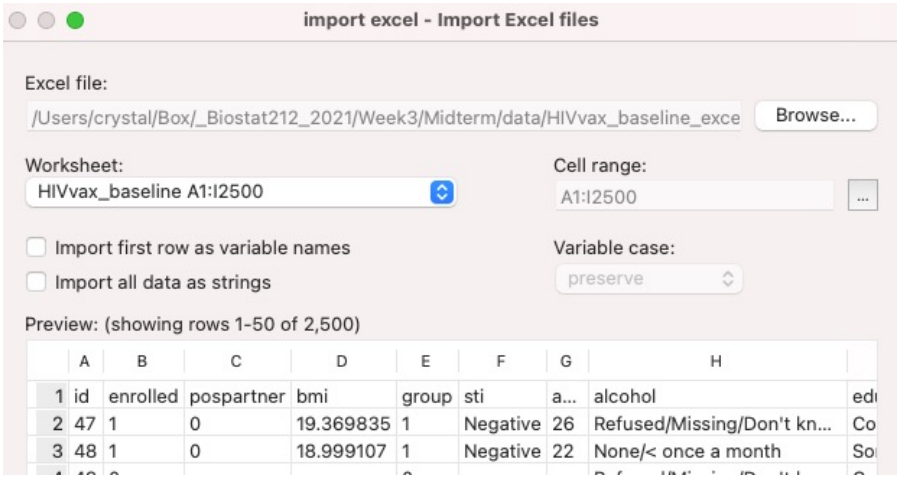
Announcements – Clarifications to Midterm

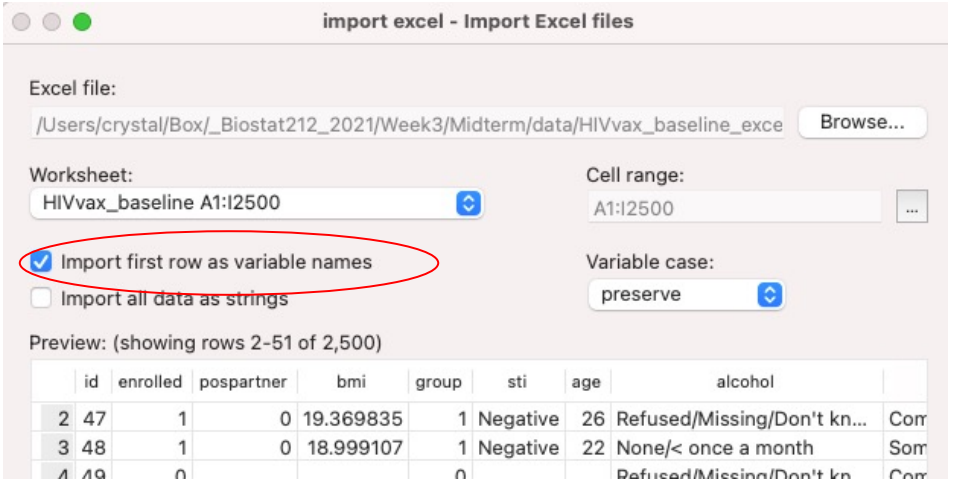
- **Part 2, question e:**
  - Purpose: clean a string variable using the numeric assignments indicated.
  - Not being asked to label missing (.); only being asked to assign character value associated with missing data to missing.

- **Part 2, question i:**
  - "Generate a variable that indicates whether each participant ever ~~seroconverted~~ was HIV+…"
    - i.e., include patients who were HIV+ at baseline as "ever HIV+"

Announcements

- Set your working directory and then use relative file paths
  - Drop down will always use full file path so will need to edit
- Identify variable labels when importing data





Today

Data Cleaning Part 2:

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

Dates

Dates

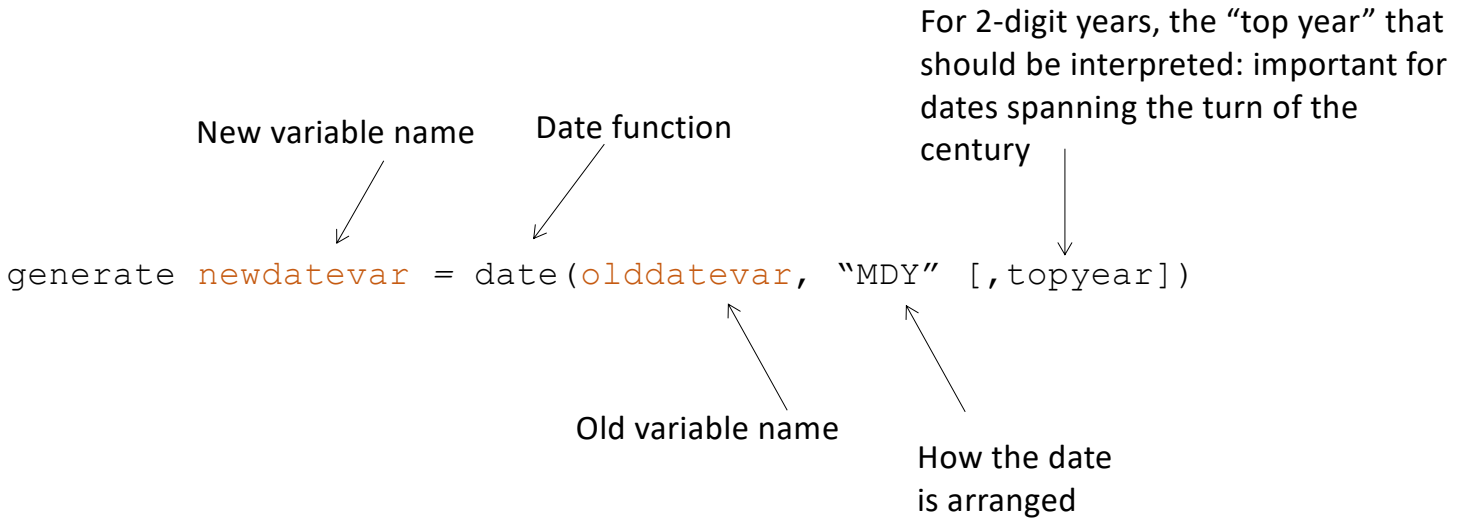
- Stata recodes dates as # of days elapsed from a reference date
  - December 31, 1959 = -1
  - January 1, 1960 = 0
  - January 2, 1960 = 1
  - January 3, 1960 = 2

- Makes it easy for Stata to manipulate dates mathematically

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

Dates

Generate new date variable



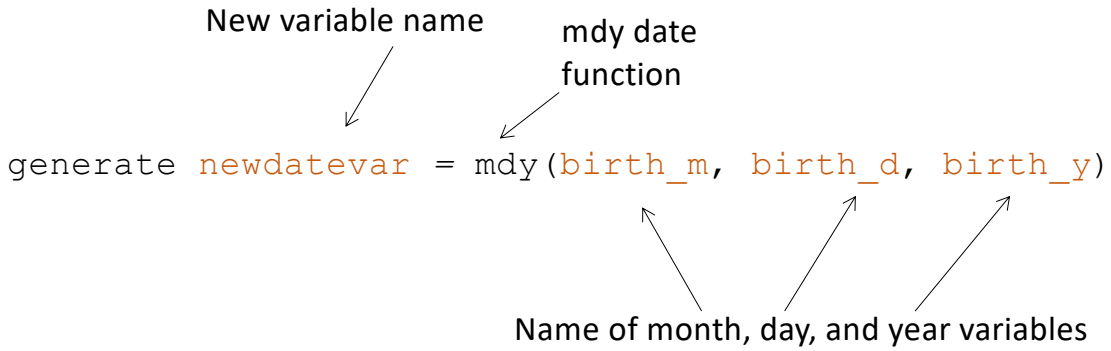
Credit: Biostat 212 TAs Sankari Ayyaluru & Yiwey Shieh, MD

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 MDY command

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

```
generate newdatevar = mdy(birth_m, birth_d, birth_y)
```

```
gen birthdate = mdy(birth_m, birth_d, birth_y)
format birthdate %td
```

birth_m	birth_d	birth_y	birthdate
7	4	1953	-2372
1	18	1948	-4366
6	28	1949	-3839
5	30	1960	150

birth_m	birth_d	birth_y	birthdate
7	4	1953	04jul1953
1	18	1948	18jan1948
6	28	1949	28jun1949
5	30	1960	30may1960

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

```
gen birthdate = date(birth, "MD20Y")
format birthdate %td
```

birth	birthdate
8/3/17	03aug2017
9/10/10	10sep2010
3/22/16	22mar2016
9/10/09	10sep2009

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 3, 1917
9/10/10 = September 10, 2010 ← top year
03/22/16 = March 22, 1916
9/10/09 = September 10, 2009

```
gen birthdate = date(birth, "MDY", 2010)
format birthdate %td
```

birth	birthdate
8/3/17	03aug1917
9/10/10	10sep2010
3/22/16	22mar1916
9/10/09	10sep2009

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)

birth_m	birth_d	birth_y	birthdate
7	4	1953	-2372
1	18	1948	-4366
6	28	1949	-3839
5	30	1960	150

birth_m	birth_d	birth_y	birthdate
7	4	1953	04jul1953
1	18	1948	18jan1948
6	28	1949	28jun1949
5	30	1960	30may1960

Missing Data

Missing data

- 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, varname`

```
. tab hospital, mi

    hospital |      Freq.     Percent        Cum.
-------------+-----------------------------------
             |        125       98.43       98.43
        UCLA |          1        0.79       99.21
        UCSF |          1        0.79      100.00
-------------+-----------------------------------
       Total |        127      100.00
```

```
. misstable sum lungcapacity

                                                              Obs<.
                                                +------------------------------+
    Variable |     Obs=.     Obs>.     Obs<.  |Unique values       Min       Max
-------------+-------------------------------+------------------------------+
lungcapacity |         20                107  |          95       -98  .9982018
```

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

Stata codes for missing data

lungcapacity
-98
.5365901
-98
.8213475
.8651673
.
.5226901
.7988783
.

- "." 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

```
mvdecode lungcapacity, mv(-98 = .a)

label define refused .a "refused to answer"
label values lungcapacity refused
```

lungcapacity
.a
.5365901
.a
.8213475
.8651673
.
.5226901
.7988783
.

lungcapacity[96] .a

lungcapacity
refused to answer
.536590099334717
refused to answer
.82134747505188
.865167260169983
.
.522690117359161
.798878252506256
.

Convert -99 to missing values

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

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

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

Stata codes for missing data

- Relationship to arithmetic operators
  - Non-missing observation < **.** < **.**a < **.**b < … < .z
  - Implications for how you specify non-missing values in your dataset

Variable	Obs	Mean	Std. dev.	Min	Max
lungcapacity	104	.7650329	.1621906	.2949074	.9982018

```
sum lungcapacity
```

				Obs<.		
Variable	Obs=.	Obs>.	Obs<.	Unique values	Min	Max
lungcapacity	20	3	104	94	.2949074	.9982018

```
misstable sum lungcapacity
```

Stata codes for missing data

- Non-missing observation < . < .a < .b < … < .z
  - Implications for how you specify non-missing values in your dataset

CAUTION when creating new variables:

```
gen lungcap80 = (lungcapacity >=.80)
     tab lungcap80, mi
     tab lungcap80 if lungcapacity == .
     replace lungcap80 = . if lungcapacity >=.
     tab lungcap80, mi
```

lungcap80	Freq.	Percent	Cum.
0	55	43.31	43.31
1	72	56.69	100.00
Total	127	100.00	

```
. tab lungcap80 if lungcapacity == .
```

lungcap80	Freq.	Percent	Cum.
1	20	100.00	100.00
Total	20	100.00	

lungcap80	Freq.	Percent	Cum.
0	55	43.31	43.31
1	49	38.58	81.89
.	23	18.11	100.00
Total	127	100.00	

`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

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
```

```
. isid ptid
variable ptid does not uniquely identify the observations
```

Tag Duplicates

Tag duplicates so you can look at them

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

Drop all but first observation **CAUTION**

```
duplicates drop
```

- Sorting matters here: “first” observation per unique identifier is determined by how the data are sorted!
- cannot undo “drop” - recommend saving data file prior to running

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

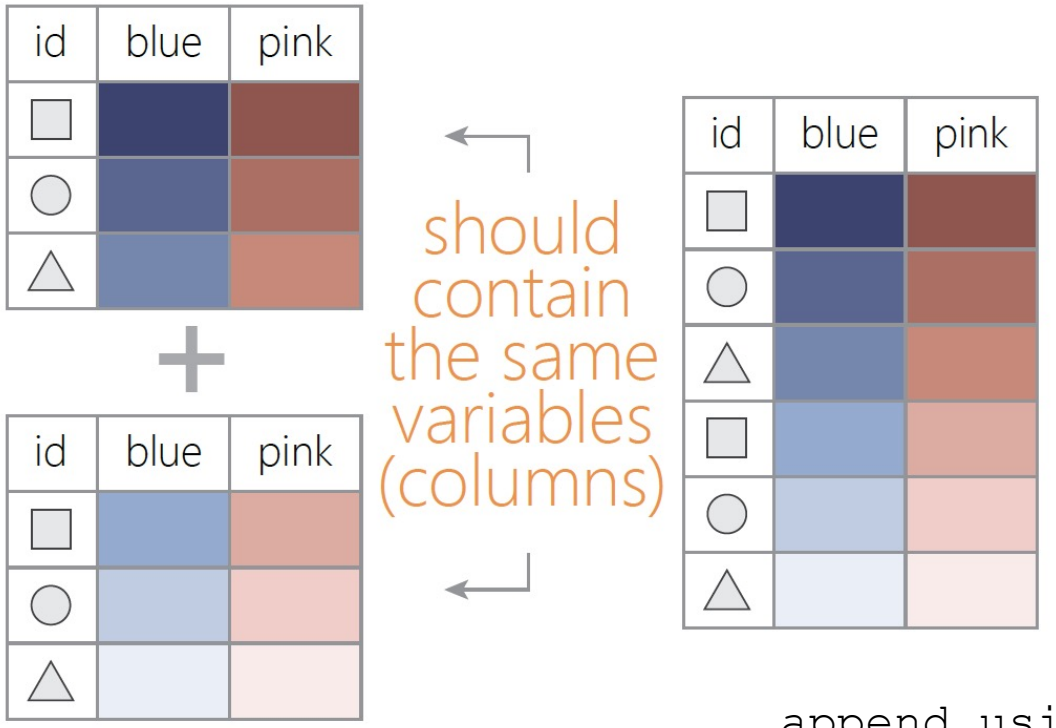
`quietly run "dofile.do"`

- `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

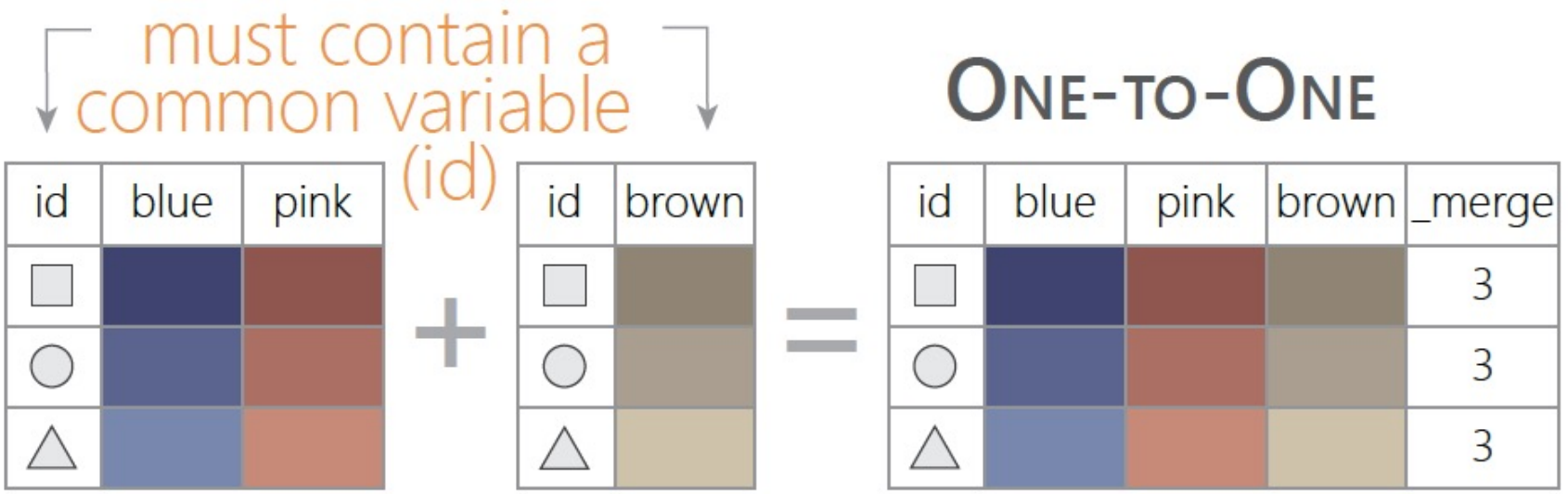
Combining Datasets

Adding (appending) new data: `append`




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

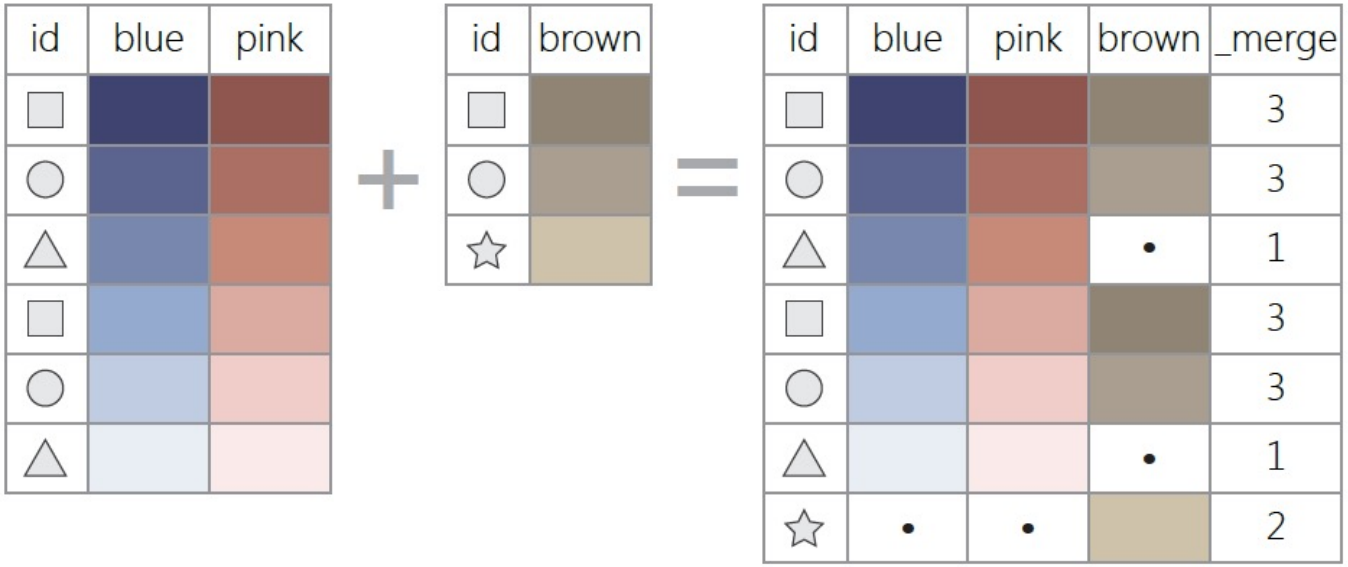
Merge data: One-to-One




```
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` variable

- Generated automatically with the `merge` command
  - Use "`, gen(newmergename)`" option to name "_merge" variable
- 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`

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
```

Next Week

Data Visualization

Midterm due Tuesday @ 1PM PST