`Stata` Tutorial: Assembling a Data Set

Data Management

- Importing data
- Specialized download commands
- Clean & reshape data
- Merging datasets

Goals:

1) Read Data into `Stata`
2) Clean & Reformat Data
3) Combine data from difference sources
4) Do it all via `.do` files

Why #4?

- Proper research methodology
- Correct errors later on

Goals:

1) Read Data into `Stata`
2) Clean & Reformat Data
3) Combine data from difference sources
4) Do it all via `.do` files

Why #4?

- Proper research methodology
- Correct errors later on

Download files for website:

```
Data Tutorial.do
FH1972_2019 raw.xls
us_foreignaid_greenbook.xlsx
EMDAT2020.csv
```

Open `.do` file and with to command window

1) Read Data into `Stata`

- Start with Excel or .csv (comma separated values) formats
- Several other formats supported

Start with menu: `File > Import`

- Specify sheet
- Specify data range
- Specify 1st row as variable names

Excel: Import `FH1972_2021 raw.xls` & save
Import `us_foreignaid_greenbook.xlsx` & save
Text: Import `EMDAT2020.csv` & save

1) Read Data into `Stata`

- Start with Excel or .csv (comma separated variable) formats
- Several other formats supported

Start with menu: `File > Import`

- Specify sheet
- Specify data range
- Specify 1st row as variable names

Excel: Import `FH1972_2021 raw.xls` & save
Import `us_foreignaid_greenbook.xlsx` & save
Text: Import `EMDAT2020.csv` & save

Copy code to `Data Tutorial.do` file

2) Specialized Download Commands

`wbopendata` World Development Indicators (WDI)

```
ssc install wbopendata                          /*once to install*/
wbopendata, indicator(SP.POP.TOTL) long clear
```

(get codes from https://databank.worldbank.org/wdi)

`freduse` Federal Reserve Economic Database (FRED)

```
ssc install freduse                             /*once to install*/
freduse UNRATE CPIAUCSL, clear
```

(get codes from https://fred.stlouisfed.org/)

`wid` World Inequality Database (WID)

```
ssc install wid                                 /*once to install*/
wid, indicators(shweal) areas(FR US) perc(p90p100 p99p100)
     year(1950/2015) ages(992) pop(j) clear
```

2) Specialized Download Commands

`wbopendata` World Development Indicators (WDI)

```
ssc install wbopendata                          /*once to install*/
wbopendata, indicator(SP.POP.TOTL) long clear
```

(get codes from https://databank.worldbank.org/wdi)

You try but also include GDP (constant 2015 US$)

[Separate codes with ;]

Save resulting data as `wdi.dta`

`fre`

```
ss                                                          stall*/
fr
```

`wid`

```
ssc install wid                                 /*once to install*/
wid, indicators(shweal) areas(FR US) perc(p90p100 p99p100)
     year(1950/2015) ages(992) pop(j) clear
```

2) Specialized Download Commands

`wbopendata` World Development Indicators (WDI)

```
ssc instal                ndata                    /*once to install*/
wbopen                    icator(SP.POP.TOTL) long clear
```

(g s from https://databank.worldbank.org/wdi)

`freduse` Federal Reserve Economic Database (FRED)

```
ssc install freduse                              /*once to install*/
freduse UNRATE CPIAUCSL, clear
```

(get codes from https://fred.stlouisfed.org/)

`wid` World Inequality Database (WID)

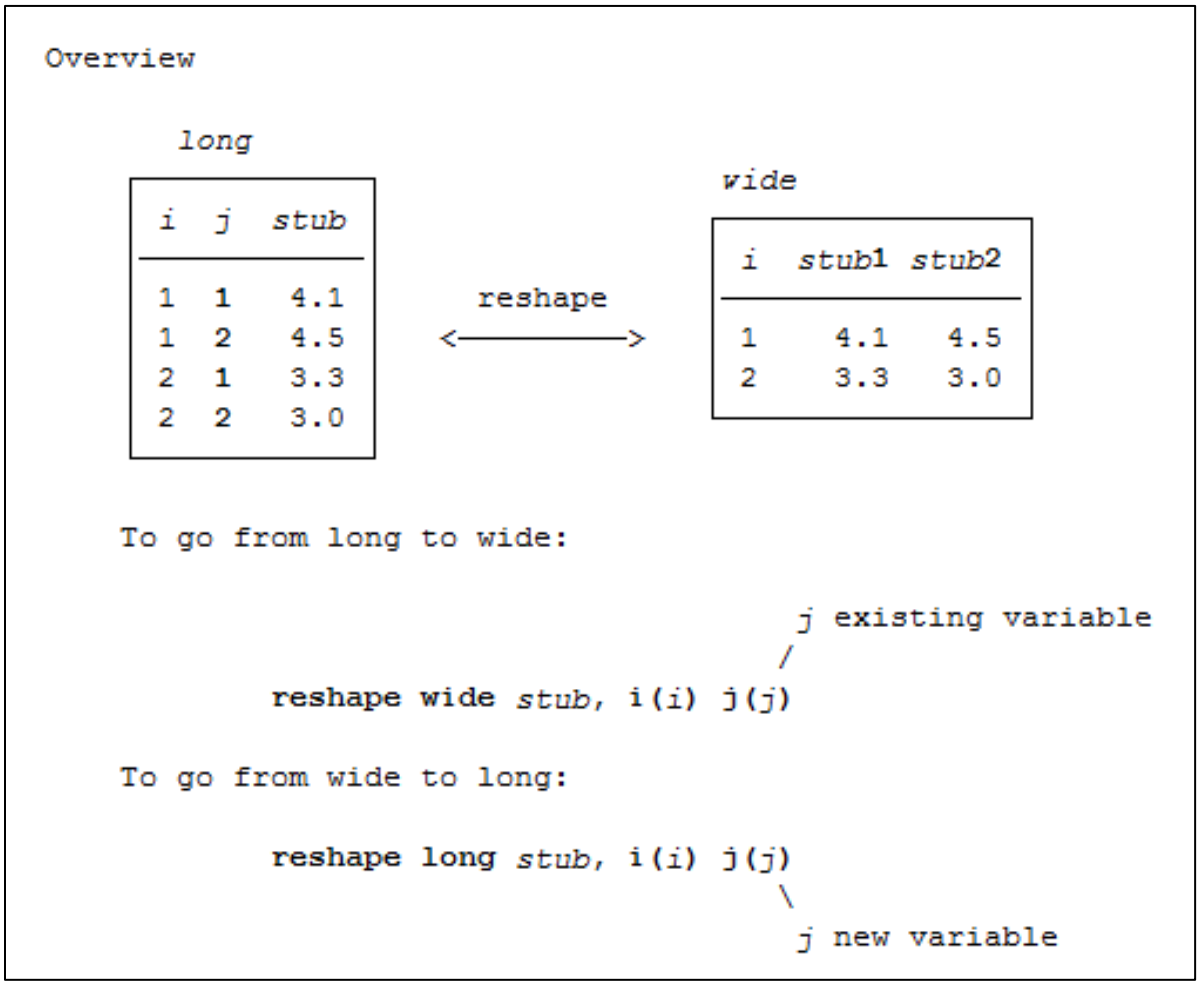
```
ssc install wid                                  /*once to install*/
wid, indicators(shweal) areas(FR US) perc(p90p100 p99p100)
     year(1950/2015) ages(992) pop(j) clear
```

3) Clean & Reformat Data

- Reshape if needed
- Drop unwanted variables / observations
- Identify & correct errors in data
- Collapse to average, sum, etc., as needed

3) Clean & Reformat Data

Reshape:



Options:

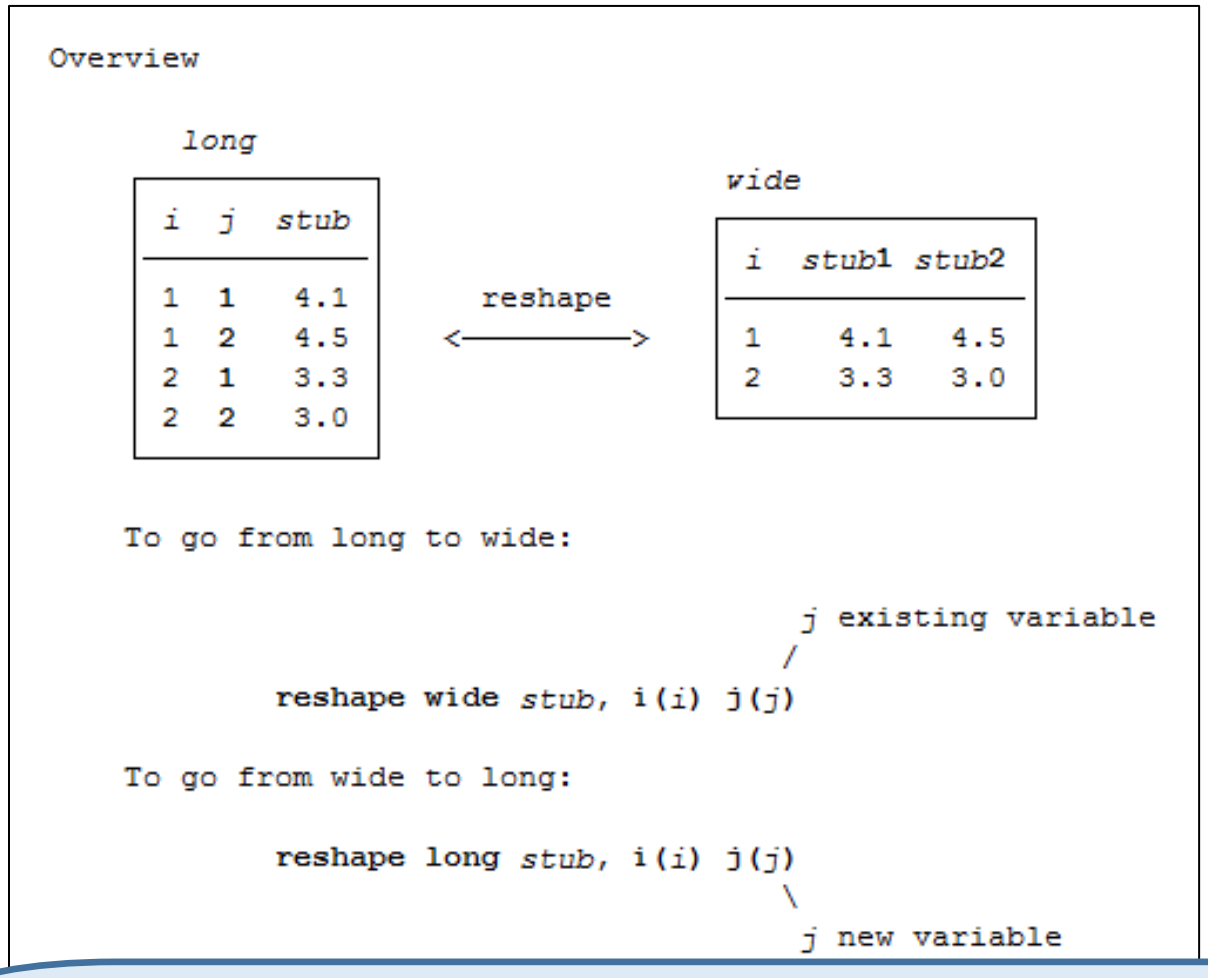
```
string
```

Placing # with @:

```
@stub
ctry@var
```

3) Clean & Reformat Data

Reshape:



Options:

```
string
```

Placing # with @:

```
@stub
ctry@var
```

Reshape data for FH

3) Clean & Reformat Data

Dropping variables/observations

```
drop GDP Pop          /*drop columns=variables*/
drop if year<2001     /*drops rows=data points*/
```

or

```
keep Country countrycode region year
keep if year>2000
```

Text variables – "Strings"

`destring`	convert from text to number
`trim`	trim blank spaces from start & finish
`substr`	keep part of longer string of text
`subinstr`	replace part of text
`strpos`	returns location of text

3) Clean & Reformat Data

Dropping variables/observations

```
drop GDP Pop          /*drop columns=variables*/
drop if year<2001     /*drops rows=data points*/
```

or

```
keep Country countrycode region year
keep if year>2000
```

Text variables – “Strings”

`destring`	convert from text to number
	finish
	t
`subinstr`	replace part of text
`strpos`	returns location of text

Convert data to #s for FH

3) Clean & Reformat Data

Dates: `Stata` counts from January 1, 1960

- earlier dates are negative #s
- count normally in days but can be in seconds, months, etc.

`date` converts text to date

```
gen mydate=date(adate,"MDY")
```

`year` extracts year from date

```
gen myyear=year(mydate)
```

`month` extracts month of year from date

```
gen mymonth=month(mydate)
```

`day` extracts day of month from date

```
gen myday=day(mydate)
```

3) Clean & Reformat Data

Cleaning might need loops to repeat tasks:

```
foreach i in var1 var2 var3 {
   replace `i'=0 if missing(`i')
}

forvalues i=1/3 {
   replace var`i'=0 if missing(var`i')
}
```

Another useful commands:

```
bysort Country (year): gen count=_n
egen country_mean=mean(GDP), by(Country)
egen MySum=rowtotal(var1 var2 var3)
```

3) Clean & Reformat Data

Cleaning might need loops to repeat tasks:

```
foreach i in var1 var2 var3 {
   replace `i'=0 if missing(`i')
}
```

These do the same thing as above (assuming variable order is var1 var2 var3 and no other names start with var):

```
foreach i of varlist var1-var3 {…
foreach i of varlist var* {…
```

```
egen country_mean=mean(GDP), by(Country)
egen MySum=rowtotal(var1 var2 var3)
```

Numbers don't have to go up by 1 each time:

```
forvalues i=3(-1)1 {…
forvalues i=0(2)6 {…
```

```
forvalues i=1/3 {
   replace var`i'=0 if missing(var`i')
}
```

Another useful commands:

```
bysort Country (year): gen count=_n
egen country_mean=mean(GDP), by(Country)
egen MySum=rowtotal(var1 var2 var3)
```

3) Clean & Reformat Data

Cleaning might need loops to repeat tasks:

```
foreach i in var1 var2 var3 {
   replace `i'=0 if missing(`i')
}

forvalues i=1/3 {
   replace var`i'=0 if missing(var`i')
}
```

Another useful commands:

```
bysort Country (year): gen count=_n
egen country_mean=mean(GDP), by(Country)
egen MySum=rowtotal(var1 var2 var3)
```

3) Clean & Reformat Data

Collapse Command

- Collapse large data set into smaller data set by
  - sum across group
  - average across group
  - count non-missing elements group
- Examples

```
collapse (mean) GDP, by(year)
collapse (max) max_y=GDP (min) min_y=GDP, by(Country)
collapse (count) year (sum) GDP, by(Country)
```

3) Clean & Reformat Data

Collapse Command

- Collapse large data set into smaller data set by
  - sum across group
  - average across group
  - count non-missing elements group
- Examples

```
collapse (mean) GDP, by(year)
collapse (max) max_y=GDP (min) min_y=GDP, by(Country)
collapse (count) year (sum) GDP, by(Country)
```

Use `collapse` to put `us_foreignaid_greenbook` and `EMDAT2020` data in correct format (each row is one country-year)

4) Merging Datasets

- Procedure to match names for merging
- Merge command

```
merge 1:1 varlist using filename [, options]
```

```
use wdi.dta, clear
merge 1:1 Country year using …
```

File in memory = “master file”
File on hard drive = “using file”
`1:1` if 1 row in master matches with 1 row in using
`m:1` if several rows in master match with 1 row in using
`1:m` if one in master matches with several rows in using

```
merge 1:1 varlist using filename [, options]
```

```
use wdi.dta, clear
merge 1:1 Country year using …
```

File in memory = “master file”
File on hard drive = “using file”
`1:1` if 1 row in master matches with 1 row in using
`m:1` if several rows in master match with 1 row in using
`1:m` if one in master matches with several rows in using

```
merge 1:1 varlist using filename [, options]
```

```
use wdi.dta, clear
merge 1:1 Country year using …
```

`options` control whether

- tracking variable `_merge` is created: `nogen`
- mismatches kept/dropped: `keep(master using match)`

Exercise: Put it all together!

Create a data set that includes macro data (GDP & Population), Freedom House Democracy scores, US military aid & disaster information. Reformat as needed.

1) Download *Data Tutorial.do* from my website; use as starting point
2) Download WDI data for *GDP (constant 2015 US$)* & *Population*
3) Import Freedom House data from *FH1972_2021 raw.xls*
4) Import military aid data from *us_foreignaid_greenbook.xlsx*
5) Import Disaster Data from *EMDAT2020.csv*
   - create vars *totaldeaths* & *totalaffected* across all disaster types

   Use `reshape` & `collapse` as needed
6) Merge data sets, correcting for differences in country names.
   - use `namecheck` procedure *before* merging
7) Run regression to see how/if US military aid depends on GDP, Population, Democracy & Disasters! ***

* * * Three other issues:

1) North Korea problem in Greenbook data…
2) Serbia problem in Greenbook data…
3) Replace missing with zero where appropriate