STATA - Create the Sample for Analysis: Part 1

Prof. Tzu-Ting Yang

Institute of Economics, Academia Sinica

March 16, 2022

Create the Sample for Analysis

- Please see **C4_create_sample_part1.do**

gen: Creating new variables

STATA Command:gen

Creating new variables

Consolas

```
1 gen age_sq = age^2
2 gen log_incwage = log(incwage)
3 gen month=10 /* constant value of 10 */
4 gen id=_n    /* id number of observation */
5 gen total=_N /* total number of observations */
6 gen bigyear=year if incwage> 100000 /* show the year
      if income > 100000 */
```

- **gen**: creating new variables
- **Line 4:** *_n* represents the index for a specific observation
- **Line 5:** *_N* represents the total number of observations

STATA Command:gen

Creating new variables

id	total
1	60000
2	60000
3	60000
4	60000
5	60000
6	60000
7	60000
8	60000

STATA Command:gen

Creating new variables

Consolas

```
1 gen str6 source="CPS" /* string variable */
2 replace incwage =. if incwage==9999999  /* replace
      9999999 to missing */
```

- A couple of things to note:

  1. Stata's default data type is float, so if you want to create a variable in some other format (e.g. byte, string), you need to specify this (**Line 1**)

  2. missing numeric observations, denoted by a dot, are interpreted by Stata as a **very large positive number**.

- You need to pay special attention to such observations when using if statements

STATA Command:gen

Creating new variables

source
CPS
CPS
CPS
CPS
CPS
CPS
CPS
CPS

STATA Command:gen

Creating new variables

incwage
15000
42000
29000
9999999
9999999

STATA Command:gen

Creating new variables

incwage
15000
42000
29000
.
.

egen: Creating new variables based on summary measures

STATA Command:egen

Creating new variables based on summary measures

Consolas

```
1 egen year_inc=total(incwage), by(year)
2 egen state_inc=mean(incwage), by(statefip)
3 egen g_state_county=group(statefip county) /*
      generates numeric id variable for state and
      county */
```

- **egen**: this command typically creates new variables based on summary measures, such as sum, mean, min and max
- **Line 1:** Use function **total** to sum the wage income for each year
- **Line 2:** Use function **mean** to get average wage income for each state

STATA Command:egen

Creating new variables based on summary measures

statefip	state_inc
connect…	34858.93
connect…	34858.93

STATA Command:egen

Creating new variables based on summary measures

Consolas

```
1 egen count_id=count(id)
2 egen g_state_county=group(statefip county)
3 egen diff_v =  diff(incwage inctot)
```

- **Line 1:** Use function **count** to count number of observations
- **Line 2:** Use function **group** to generate numeric id variable for state and county
- **Line 3:** Use function **diff** to generate a variable indicating whether variables incwage and inctot are different or not

STATA Command:egen

Creating new variables based on summary measures

id	count_id
1	60000
2	60000
3	60000

STATA Command:egen

Creating new variables based on summary measures

	incwage	inctot	diff_v
1	0	5760	1
2	.	99999999	1
3	200000	200251	1
4	55000	55000	0

STATA Command:egen

Creating new variables based on summary measures

- The **egen** command is also useful if your data is in long format
- For example, you want to find the difference in sample *i*'s income and maximum of income within specific state

STATA Command:egen

Creating new variables based on summary measures

- The following routine will achieve this:

Consolas

```
gen temp1=incwage if year==2015
egen temp2=max(temp1), by(statefip)
gen temp3=incwage-temp2 if year==2015
egen diff=max(temp3), by(statefip)
drop temp*
```

STATA Command:egen

Creating new variables based on summary measures

	incwage	temp1	temp2	temp3	inctot
1	50000	50000	1099999	-1049999	50144
2	10000	10000	1099999	-1089999	10000
3	0	0	1099999	-1099999	40008

replace: Modifying existing variables

STATA Command:replace

Modifying existing variables

Consolas

```
1 gen ln_inctot=ln(inctot)
2 replace ln_inctot=0 if ln_inctot==.
3 gen byte yr=year-1900
4 replace yr=yr-100 if yr >= 100 /* 0,1,etc instead of
      100,101 for 2000 onwards */
```

- **replace**: this command modifies existing variables in exactly the same way as generate creates new variables
- **Line 2:** replace *ln_inctot* to zero if *ln_inctot* == .
  - Note that if you apply a transformation to missing data, the result will still be a missing value
  - A transformation that is undefined, e.g. taking the natural log of a negative number creates a missing value

STATA Command:replace

Modifying existing variables

	yr	year
1	114	2014
2	114	2014
3	114	2014

STATA Command:replace

Modifying existing variables

	yr	year
1	14	2014
2	14	2014
3	14	2014

label: Put labels on datasets, variables or values

STATA Command:label

Put labels on datasets, variables or values

Consolas

```
1 label data "Data from CPS 2014-2015"
2 label variable incwage "wage income"
```

- **label**: This command let you put labels on datasets, variables or values
- This helps to make it clear exactly what the dataset contains
- **Line 1:** put a label on the current data set
- **Line 2:** put a label on variable incwage

STATA Command:label

Put labels on datasets, variables or values

Consolas

```
1 tab sex
2 label define gendercode 1 "Male" 2 "Female"
3 label values sex gendercode
4 tab sex
5 codebook sex
```

- It can also be helpful to label different values
- Imagine states/countries were coded as numbers (which is the case in many datasets)
- It might be better to label exactly what each value represents
- **label define**: defining a label (giving it a name and specifying the mapping)
- **label values**: associating that label with a variable

STATA Command:label

Put labels on datasets, variables or values

```
. tab sex
```

sex	Freq.	Percent	Cum.
1	29,148	48.58	48.58
2	30,852	51.42	100.00
Total	60,000	100.00	

STATA Command:label

Put labels on datasets, variables or values

```
. tab sex
```

sex	Freq.	Percent	Cum.
Male	29,148	48.58	48.58
Female	30,852	51.42	100.00
Total	60,000	100.00	

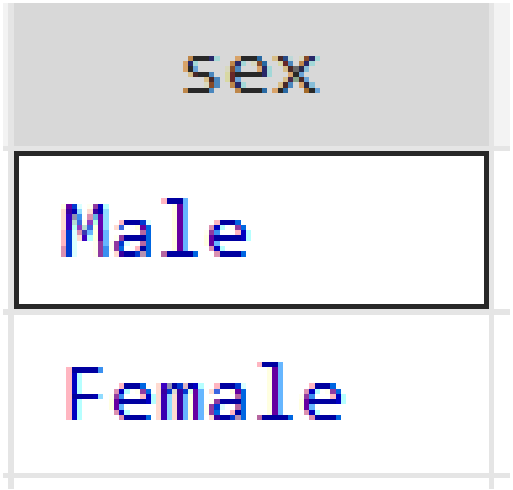
STATA Command:label

Put labels on datasets, variables or values

sex
1
2

STATA Command:label

Put labels on datasets, variables or values



rename: Change the names of your variables

STATA Command:rename

Change the names of your variables

Consolas

```
1 rename sex gender
2 rename gender sex
```

- **rename**: this command can change the names of your variables
- **Line 1:** rename sex to gender
- **Line 2:** rename gender to sex

recode: Change the values that certain variables take

STATA Command:recode

Change the values that certain variables take

Consolas

```
recode sex (1=0) (2=1)
recode incwage inctot (0 = .)
```

- **recode**: this command can change the values that certain variables take
- This command can also be used to recode missing values to the dot that Stata uses to denote missing
- And you can recode several variables at once

STATA Command:recode

Change the values that certain variables take

Consolas

```
recode incwage (0 / 50000 = 1) (50001 / 100000 = 2)
    (100000 / 7000000 = 3)
```

- **recode** can also make several changes at the same time
- We could, for example, use recode to generate a new variable with categorical income values

STATA Command:recode

Change the values that certain variables take

incwage
15000
42000
29000

STATA Command:recode

Change the values that certain variables take

incwage
1
1
1

tostring/destring: Change variable to string/numeric variables

Numeric and String Variables

- Stata stores or formats data in either of two ways – **numeric** or **string**
- Numeric will store numbers while string will store text
- **Numeric variables** are in **black/blue color** and **string variables** are in **red color**
- String variale can also be used to store numbers, but you will not be able to perform numerical analysis on those numbers

STATA Command:tostring

Change variable to string

Consolas

```
1 tostring year, replace
```

- change variable to string

STATA Command:tostring

Change variable to string

	year
1	2014
2	2014
3	2014

STATA Command:tostring

Change variable to string

	year
1	2014
2	2014
3	2014

STATA Command:destring

Change variable to numeric

Consolas

```
1 destring year, gen(year1)
```

- change variable to numeric

STATA Command:destring

Change variable to numeric

	year	year1
1	2014	2014
2	2014	2014
3	2014	2014

decode: Create a string variable from a numerical code

STATA Command:decode

Create a string variable from a numerical code

Consolas

```
1 decode statefip,gen(state_name)
```

- **decode:** Create a string variable from a numerical code, as long as the numeric variable has labels attached to each value
- **Line 1:** This creates a new variable `state_name`, which use label of numerical code as string variable

STATA Command:decode

Create a string variable from a numerical code

	statefip	state_name
1	maine	maine
2	maine	maine

encode: Converting strings to numerical code

STATA Command:encode

Converting strings to numerical code

Consolas

```
1 encode state_name, gen(state_id)
```

- **encode:** Converting strings to numerical code
- **Line 1:** This creates a new variable `state_id`, which takes a value of 1 for alabama, 2 for alaska, and so on.

STATA Command:encode

Converting strings to numerical code

	state_name	state_id
1	maine	maine
2	maine	maine

substr: Dividing string variables

Combining String Variables

Consolas

```
1 tostring year, gen(yearcode)
2 gen yearcode1 = string(year)
3 gen stateyear = state_name + yearcode1
```

- We can create a new string variable whose data is a combination of the data values of other variables
- **Line 2:** This creates a new variable "stateyear", which combine two string variable "`state_name`" and "yearcode1"

Combining String Variables

	yearcode1	state_name	stateyear
1	2014	maine	maine2014
2	2014	maine	maine2014

STATA Command:substr

Dividing String Variables

Consolas

```
1 gen yr1 = substr(yearcode,3,2)
```

- **substr**: Divide up a variable or to extract part of a variable to create a new one
  - The first term in parentheses is the string variable that you are extracting from
  - The second term (3) is the position of the first character you want to extract
  - The third term (2) is the number of characters to be extracted

STATA Command:substr

Dividing String Variables

Consolas

```
1 gen yr2 = substr(yearcode,-2,2)
```

- Alternatively, you can select your starting character by counting from the end (2 positions from the end instead of 3 positions from the start)

STATA Command:substr

Dividing String Variables

	yr2	yr1	yearcode
1	14	14	2014
2	14	14	2014

Create dummy variables

Create dummy variables - Method 1

Consolas

```
1 gen largewage1=(incwage_test>=30000)
```

- **Line 1:** create a variable largewage1 if the statement within parentheses is true ($incwage\_test >= 30000$)

Create dummy variables - Method 1

	largewage1	incwage
1	0	15000
2	1	42000
3	0	29000

Create dummy variables - Method 2

Consolas

```
1 tab statefip, gen(state_d)
```

- This creates:
  - a dummy variable "`state_d1`" equal to 1 if the state is "alabama" and zero otherwise
  - a dummy variable "`state_d2`" if the state is "alaska" and zero otherwise and so on
- We can use "`state_d*`" to represent the all dummy variables

Create dummy variables - Method 2

```
. tab statefip, gen(state_d)
```

state (fips code)	Freq.	Percent	Cum.
connecticut	2,799	4.67	4.67
delaware	2,113	3.52	8.19
illinois	4,393	7.32	15.51
indiana	2,137	3.56	19.07
iowa	2,486	4.14	23.21
kansas	1,984	3.31	26.52
maine	1,794	2.99	29.51
maryland	2,319	3.87	33.38

Create dummy variables - Method 2

	statefip	state_d1	state_d2	state_d3	state_d4	state_d5	state_d6	state_d7
1	maine	0	0	0	0	0	0	1
2	maine	0	0	0	0	0	0	1
3	maine	0	0	0	0	0	0	1

Create dummy variables - Method 3

Consolas

```
1 reg incwage i.sex#i.year
2 reg incwage i.sex##i.year
```

- Sometimes we just need to include these dummies in our regression and do not want to create these variable permanently
- Using one # tells Stata to ignore the level for each variable and only report the interaction
- Doubling up the #, **`i.region##i.year`** makes the command account for both dummies for each region, each year and each interaction.

Create dummy variables - Method 3

```
. reg incwage i.sex#i.year  //i.region#i.year
```

Source	SS	df	MS
Model	1.6571e+15	5	3.3141e+14
Residual	1.0495e+18	59,994	1.7494e+13
Total	1.0512e+18	59,999	1.7520e+13

Number of obs	60,000
F(5, 59994)	18.94
Prob > F	0.0000
R-squared	0.0016
Adj R-squared	0.0015
Root MSE	4.2e+06

incwage	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
sex#year						
1#2015	223840.4	60074.95	3.73	0.000	106093.2	341587.5
1#2016	222358.2	60117.63	3.70	0.000	104527.4	340188.9
2#2014	-220957.9	59198.05	-3.73	0.000	-336986.3	-104929.5
2#2015	-110505.1	59445.27	-1.86	0.063	-227018	6007.824
2#2016	109416.8	59404.48	1.84	0.065	-7016.214	225849.8
_cons	2271781	42675.02	53.23	0.000	2188138	2355424

Create dummy variables - Method 3

```
. reg incwage i.sex##i.year
```

Source	SS	df	MS		
Model	1.6571e+15	5	3.3141e+14	Number of obs	= 60,000
Residual	1.0495e+18	59,994	1.7494e+13	F(5, 59994)	= 18.94
				Prob > F	= 0.0000
				R-squared	= 0.0016
				Adj R-squared	= 0.0015
Total	1.0512e+18	59,999	1.7520e+13	Root MSE	= 4.2e+06

incwage	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
sex						
2	-220957.9	59198.05	-3.73	0.000	-336986.3	-104929.5
year						
2015	223840.4	60074.95	3.73	0.000	106093.2	341587.5
2016	222358.2	60117.63	3.70	0.000	104527.4	340188.9
sex#year						
2#2015	-113387.5	83694.88	-1.35	0.175	-277429.8	50654.72
2#2016	108016.5	83696.57	1.29	0.197	-56029.05	272062.1
_cons	2271781	42675.02	53.23	0.000	2188138	2355424