Development Data Boot Camp
Relational Databases: Reshape and Collapse

Ge Sun

University of Notre Dame

May 23, 2023

Outline

Data types

Commands: *reshape*

Commands: *collapse*

Commands: *preserve* and *restore*

Outline

Data types

Commands: *reshape*

Commands: *collapse*

Commands: *preserve* and *restore*

Data types

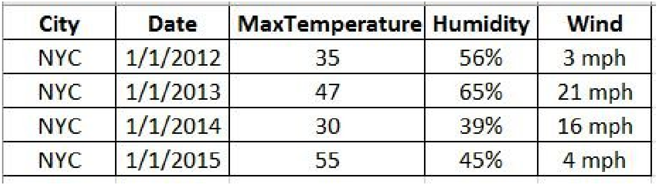
Another view to categorize data:
What is the connection between different observations?[1]

- Time-series data
- Cross-sectional data
- Panel data
- Pooled data

[1] I borrowed the examples from Quora, credit to Srikanth Varma

Time-series data

- Definition: Time-series data is a collection of observations for a single subject at different time intervals(generally equally spaced)
- Example: the max temperature, humidity and wind (three observations) in New York City(single entity) collected on First day of every year(multiple intervals of time)

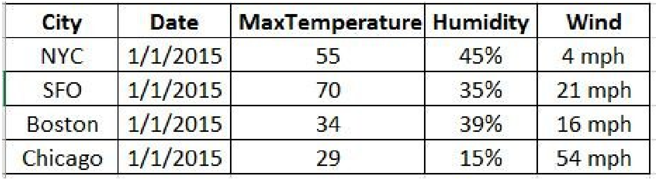


City	Date	MaxTemperature	Humidity	Wind
NYC	1/1/2012	35	56%	3 mph
NYC	1/1/2013	47	65%	21 mph
NYC	1/1/2014	30	39%	16 mph
NYC	1/1/2015	55	45%	4 mph

Figure 1: Example of time-series data

Cross-sectional data

- Definition: Cross-Sectional data is a collection of observations for multiple subjects(entities) at single point in time.
- Example: the max temperature, humidity and wind(all three observations) in New York City, SFO, Boston, Chicago(multiple entities) on 1/1/2015(single instance)



City	Date	MaxTemperature	Humidity	Wind
NYC	1/1/2015	55	45%	4 mph
SFO	1/1/2015	70	35%	21 mph
Boston	1/1/2015	34	39%	16 mph
Chicago	1/1/2015	29	15%	54 mph

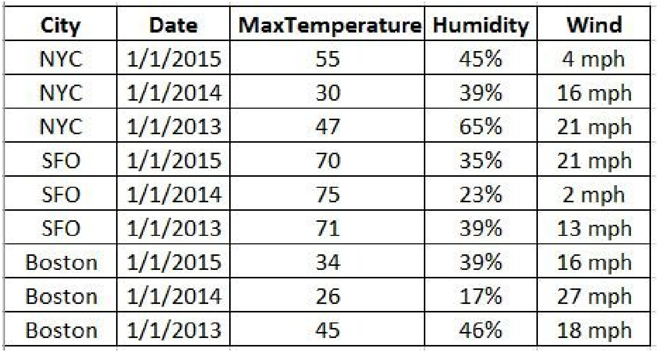
Figure 2: Example of cross-sectional data

Panel data (Longitudinal Data)

- Definition: We can interpret panel data as cross-sectional time-series data. It combines the types mentioned above, i.e., collection of observations for multiple subjects (usually same subjects) at multiple instances.
- Panel data consists of repeated observations for a single entity across different points in time, allowing for more comprehensive analysis. Consequently, panel data provides valuable insights, making it a preferred choice for researchers.

Panel data (Longitudinal data)

- Example: max temperature, humidity and wind (all three behaviors) in New York City, SFO, Boston, Chicago(multiple entities) on First day of every year(multiple intervals of time)



City	Date	MaxTemperature	Humidity	Wind
NYC	1/1/2015	55	45%	4 mph
NYC	1/1/2014	30	39%	16 mph
NYC	1/1/2013	47	65%	21 mph
SFO	1/1/2015	70	35%	21 mph
SFO	1/1/2014	75	23%	2 mph
SFO	1/1/2013	71	39%	13 mph
Boston	1/1/2015	34	39%	16 mph
Boston	1/1/2014	26	17%	27 mph
Boston	1/1/2013	45	46%	18 mph

Figure 3: Example of panel data

Pooled data

- Definition: Multiple cross-sectional data, i.e. collection of observations for multiple different subjects at multiple instances.
- Example: max temperature, humidity and wind (all three behaviors) in New York City and SFO on 1/1/2014, and max temperature humidity and wind in Boston and Chicago on 1/1/2015.

Data Structure

Why do we care about data structure?

- Different structure of data requires different kinds of analysis techniques.
- Stata knows how to handle different kinds of data correctly if you “tell” Stata and make the data structure recognizable to Stata.

Outline

Data types

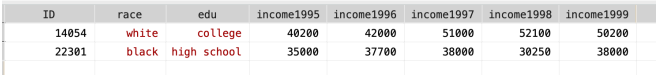
Commands: *reshape*

Commands: *collapse*

Commands: *preserve* and *restore*

Wide form and long form

- Sometime we need to deal with the case where one observation contains many sub-observations.
- Consider that we observe a worker's race and education, in addition, we can also observe his income in the last five years. How to organize this data structure in Stata? There are two ways:
- Wide form: combine several observations into a single observation.

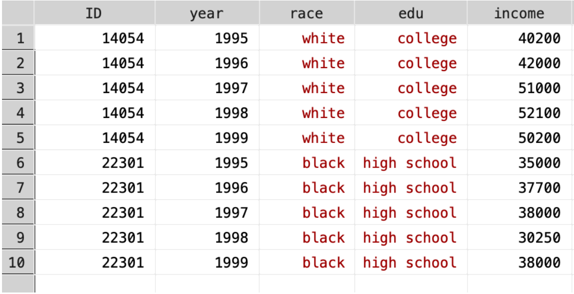


ID	race	edu	income1995	income1996	income1997	income1998	income1999
14054	white	college	40200	42000	51000	52100	50200
22301	black	high school	35000	37700	38000	30250	38000

Figure 4: Example of wide form

Wide form and long form

- Long form: each observation is for a distinct individual-year pair



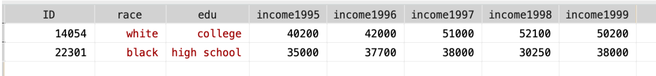
	ID	year	race	edu	income
1	14054	1995	white	college	40200
2	14054	1996	white	college	42000
3	14054	1997	white	college	51000
4	14054	1998	white	college	52100
5	14054	1999	white	college	50200
6	22301	1995	black	high school	35000
7	22301	1996	black	high school	37700
8	22301	1997	black	high school	38000
9	22301	1998	black	high school	30250
10	22301	1999	black	high school	38000

Figure 5: Example of long form

Command: reshape

- Analyzing long form data is more convenient than wide form data. And some estimation commands require data to be in long form, such as panel data analysis.
- However, some datasets are stored in wide form (which use less memory space), and we need to a command to convert long form data to wide form data, or convert wide form data to long form data.
- The syntax of *reshape* is complex, remember to type *help reshape* every time when you need to use the command.

Command: reshape



ID	race	edu	income1995	income1996	income1997	income1998	income1999
14054	white	college	40200	42000	51000	52100	50200
22301	black	high school	35000	37700	38000	30250	38000

- The syntax of *reshape* converting wide form data to long form data is:

```
reshape long var_name, i(ID_column) j(index_row)
```

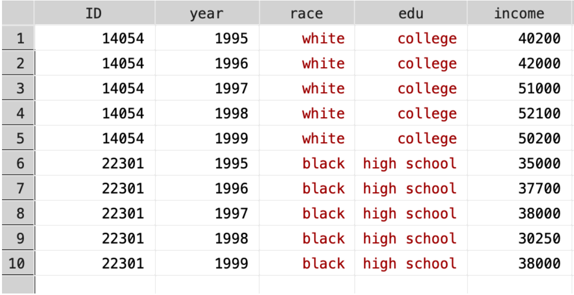
- In our example, *var_name* will be replaced by *income*.
- *ID_column* will be replaced by *ID*.
- *index_row* will be replaced by *year*. Note that the variable year a new variable that needed to be created in the command.

Command: reshape

- So type the following command

```
reshape long income, i(ID) j(year)
```

we will have the corresponding long form data:



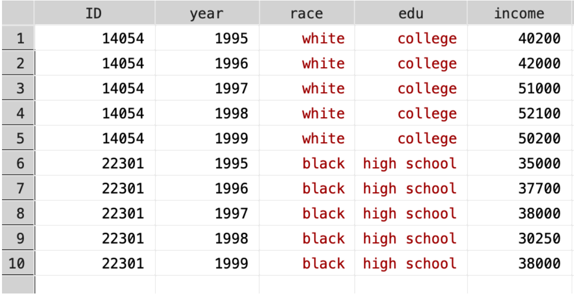
	ID	year	race	edu	income
1	14054	1995	white	college	40200
2	14054	1996	white	college	42000
3	14054	1997	white	college	51000
4	14054	1998	white	college	52100
5	14054	1999	white	college	50200
6	22301	1995	black	high school	35000
7	22301	1996	black	high school	37700
8	22301	1997	black	high school	38000
9	22301	1998	black	high school	30250
10	22301	1999	black	high school	38000

Command: reshape

- So type the following command

```
reshape long income, i(ID) j(year)
```

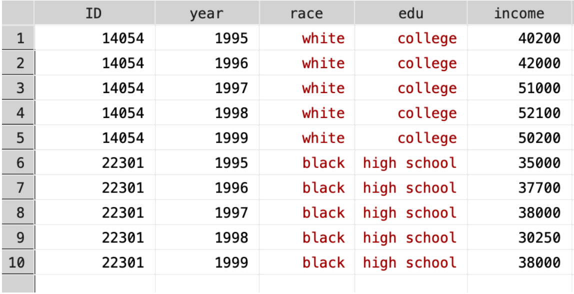
we will have the corresponding long form data:



	ID	year	race	edu	income
1	14054	1995	white	college	40200
2	14054	1996	white	college	42000
3	14054	1997	white	college	51000
4	14054	1998	white	college	52100
5	14054	1999	white	college	50200
6	22301	1995	black	high school	35000
7	22301	1996	black	high school	37700
8	22301	1997	black	high school	38000
9	22301	1998	black	high school	30250
10	22301	1999	black	high school	38000

- **Question:** How would Stata know where the information of “year” comes from?

Command: reshape



	ID	year	race	edu	income
1	14054	1995	white	college	40200
2	14054	1996	white	college	42000
3	14054	1997	white	college	51000
4	14054	1998	white	college	52100
5	14054	1999	white	college	50200
6	22301	1995	black	high school	35000
7	22301	1996	black	high school	37700
8	22301	1997	black	high school	38000
9	22301	1998	black	high school	30250
10	22301	1999	black	high school	38000

- The syntax of *reshape* converting long form data to wide form data is:

```
reshape wide var_name, i(ID_column) j(index_row)
```

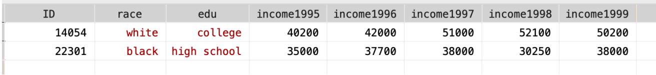
Command: reshape

- The syntax of *reshape* converting long form data to wide form data is:

  reshape wide var_name, i(ID_column) j(index_row)

- In our example, *var_name* will be replaced by *income*.
- *ID_column* will be replaced by *ID*.
- *index_row* will be replaced by *year*. Note that the variable year already exists.
- Type the following command and we will have

  reshape wide income, i(ID) j(year)



ID	race	edu	income1995	income1996	income1997	income1998	income1999
14054	white	college	40200	42000	51000	52100	50200
22301	black	high school	35000	37700	38000	30250	38000

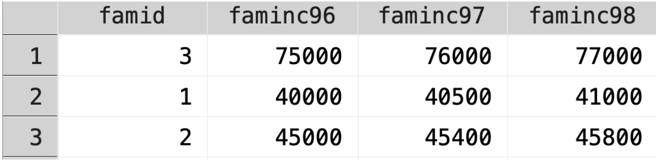
Command *reshape* : Exercise1[2]

- import the data from the website:
  use https://stats.idre.ucla.edu/stat/stata/modules/faminc, clear

[2] from UCLA Stata learning module

Command *reshape* : Exercise1[2]

- import the data from the website:
  use https://stats.idre.ucla.edu/stat/stata/modules/faminc, clear



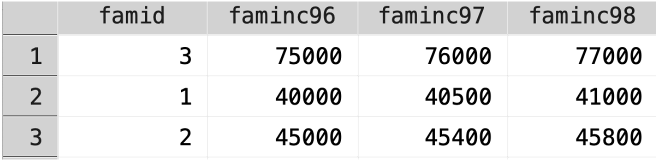
	famid	faminc96	faminc97	faminc98
1	3	75000	76000	77000
2	1	40000	40500	41000
3	2	45000	45400	45800

- What would a long-form look like for this dataset? How many observations and how many variables do we have? You can reshape it first on a piece of paper.

[2] from UCLA Stata learning module

Command *reshape* : Exercise1[2]

- import the data from the website:
  use https://stats.idre.ucla.edu/stat/stata/modules/faminc, clear



	famid	faminc96	faminc97	faminc98
1	3	75000	76000	77000
2	1	40000	40500	41000
3	2	45000	45400	45800

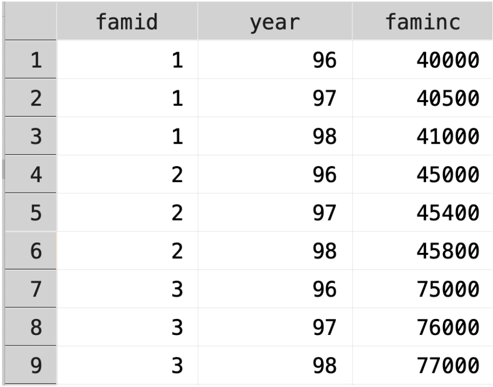
- What would a long-form look like for this dataset? How many observations and how many variables do we have? You can reshape it first on a piece of paper.
- How about the code? What would be the var_name, the ID and the index_row?

reshape long var_name, i(ID) j(index_row)

[2] from UCLA Stata learning module

Command *reshape* : Exercise1

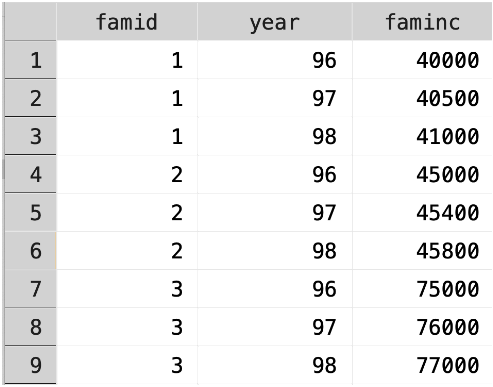
- The reshpaed dataset:



	famid	year	faminc
1	1	96	40000
2	1	97	40500
3	1	98	41000
4	2	96	45000
5	2	97	45400
6	2	98	45800
7	3	96	75000
8	3	97	76000
9	3	98	77000

Command *reshape* : Exercise1

- The reshpaed dataset:



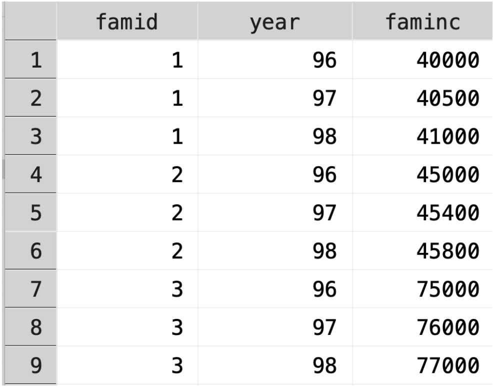
	famid	year	faminc
1	1	96	40000
2	1	97	40500
3	1	98	41000
4	2	96	45000
5	2	97	45400
6	2	98	45800
7	3	96	75000
8	3	97	76000
9	3	98	77000

- How about the code?

```
reshape long faminc, i(famid) j(year)
```

Command *reshape* : Exercise1

- The reshpaed dataset:



	famid	year	faminc
1	1	96	40000
2	1	97	40500
3	1	98	41000
4	2	96	45000
5	2	97	45400
6	2	98	45800
7	3	96	75000
8	3	97	76000
9	3	98	77000

- How about the code?

```
reshape long faminc, i(famid) j(year)
```

- Can you re-reshape it back to wide form?

Command *reshape* : Exercise2

- import the data from the website:
  use https://stats.idre.ucla.edu/stat/stata/modules/kidshtwt, clear
- reshape the data from wide form into long form.

Command *reshape* : Exercise2

- import the data from the website:
  use https://stats.idre.ucla.edu/stat/stata/modules/kidshtwt, clear
- reshape the data from wide form into long form.
- The code:

```
reshape long ht, i(famid birth) j(age)
```

Outline

Data types

Commands: *reshape*

Commands: *collapse*

Commands: *preserve* and *restore*

Introduction to command *collapse*

- Sometimes you have data in a finer level, but you want to have some aggregate level results
  - city-level population data ⇒ state-level sum of population
  - test score of each student ⇒ the average test score of the class
  - the kids' information ⇒ count the number of kids in each family

Introduction to command *collapse*

- Sometimes you have data in a finer level, but you want to have some aggregate level results
  - city-level population data ⇒ state-level sum of population
  - test score of each student ⇒ the average test score of the class
  - the kids' information ⇒ count the number of kids in each family
- We have the command :

  bys state: egen state_pop = sum(population)

  to help us, but we also need *drop* a lot of variables if we only want to keep the aggregate level information.

  drop if state_pop[_n] == state_pop[_n+1]

Introduction to command *collapse*

- Sometimes you have data in a finer level, but you want to have some aggregate level results
  - * city-level population data ⇒ state-level sum of population
  - * test score of each student ⇒ the average test score of the class
  - * the kids' information ⇒ count the number of kids in each family
- We have the command :

  $$bys\ state:\ egen\ state\_pop = sum(population)$$

  to help us, but we also need *drop* a lot of variables if we only want to keep the aggregate level information.

  $$\text{drop if state\_pop[\_n]} == \text{state\_pop[\_n+1]}$$
- *collapse* can accomplish these two steps at one time.

Command: *collapse*

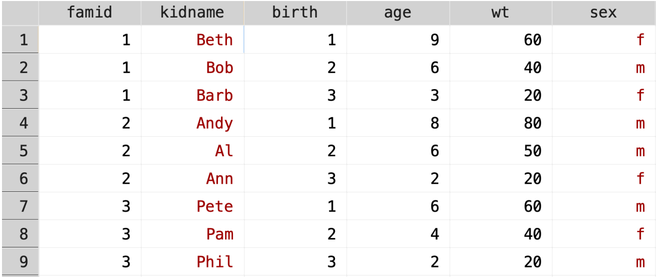
- Syntax:

  collapse (mean) var, by(categories)

- About the output: the default setting is *mean*, but you can change it to *sum*, *count*, *median* or other percentage quantiles.

Example for *collapse*

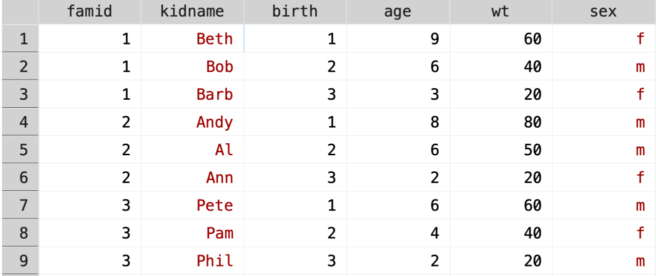
- use https://stats.idre.ucla.edu/stat/stata/modules/kids, clear
- Here is a file containing information about the kids in three families.
  - There is one record per kid. Birth is the order of birth (i.e., 1 is first), age wt and sex are the child's age, weight and sex.



	famid	kidname	birth	age	wt	sex
1	1	Beth	1	9	60	f
2	1	Bob	2	6	40	m
3	1	Barb	3	3	20	f
4	2	Andy	1	8	80	m
5	2	Al	2	6	50	m
6	2	Ann	3	2	20	f
7	3	Pete	1	6	60	m
8	3	Pam	2	4	40	f
9	3	Phil	3	2	20	m

Example for *collapse*

- use https://stats.idre.ucla.edu/stat/stata/modules/kids, clear
- Here is a file containing information about the kids in three families.
  - There is one record per kid. Birth is the order of birth (i.e., 1 is first), age wt and sex are the child's age, weight and sex.

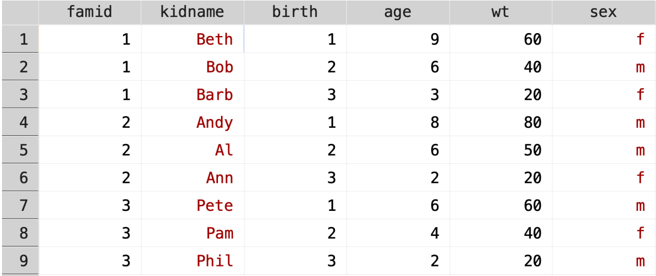


	famid	kidname	birth	age	wt	sex
1	1	Beth	1	9	60	f
2	1	Bob	2	6	40	m
3	1	Barb	3	3	20	f
4	2	Andy	1	8	80	m
5	2	Al	2	6	50	m
6	2	Ann	3	2	20	f
7	3	Pete	1	6	60	m
8	3	Pam	2	4	40	f
9	3	Phil	3	2	20	m

- To get to know the average age of the kids in each family:

collapse (mean) age, by(famid)
collapse age, by(famid)

Example for *collapse*



	famid	kidname	birth	age	wt	sex
1	1	Beth	1	9	60	f
2	1	Bob	2	6	40	m
3	1	Barb	3	3	20	f
4	2	Andy	1	8	80	m
5	2	Al	2	6	50	m
6	2	Ann	3	2	20	f
7	3	Pete	1	6	60	m
8	3	Pam	2	4	40	f
9	3	Phil	3	2	20	m

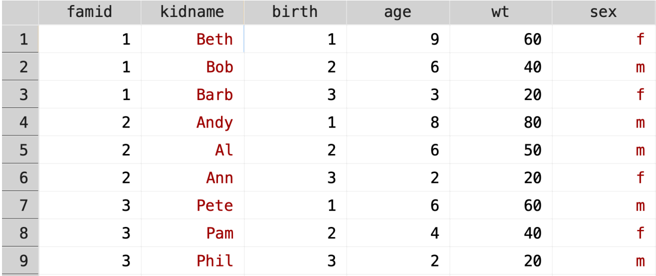
- Get the average level of children's age and weights in each family and change the generated variables as *ave_age* and *ave_wt*.

  collapse (mean) avgage=age avgwt=wt, by(famid)

- Count the number of kids in each family.

  collapse (mean) avgage=age avgwt=wt (count) numkids=birth, by(famid)

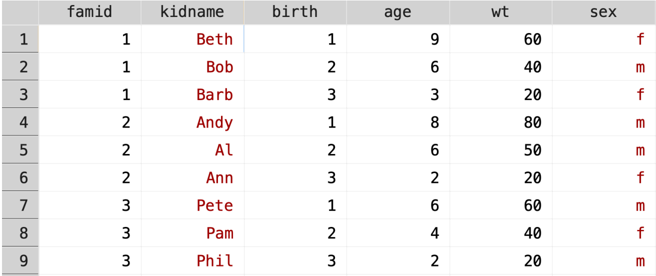
Exercise for *collapse*



	famid	kidname	birth	age	wt	sex
1	1	Beth	1	9	60	f
2	1	Bob	2	6	40	m
3	1	Barb	3	3	20	f
4	2	Andy	1	8	80	m
5	2	Al	2	6	50	m
6	2	Ann	3	2	20	f
7	3	Pete	1	6	60	m
8	3	Pam	2	4	40	f
9	3	Phil	3	2	20	m

- How would you count of the number of boys and girls in each family? Hint: tab,gen()

Exercise for *collapse*



	famid	kidname	birth	age	wt	sex
1	1	Beth	1	9	60	f
2	1	Bob	2	6	40	m
3	1	Barb	3	3	20	f
4	2	Andy	1	8	80	m
5	2	Al	2	6	50	m
6	2	Ann	3	2	20	f
7	3	Pete	1	6	60	m
8	3	Pam	2	4	40	f
9	3	Phil	3	2	20	m

- How would you count of the number of boys and girls in each family? Hint: tab,gen()

```
tabulate sex, generate(sexdum)
collapse (sum) girls=sexdum1 boys=sexdum2, by(famid)
```

Outline

Data types

Commands: *reshape*

Commands: *collapse*

Commands: *preserve* and *restore*

Introduction of *preserve* and *restore*

- In some cases, it is desirable to temporarily change the dataset, perform some calculation, then return to the original dataset.

Introduction of *preserve* and *restore*

- In some cases, it is desirable to temporarily change the dataset, perform some calculation, then return to the original dataset.
- Continue on our previous example:

```
preseve
collapse (mean) avgage=age, by(famid)
list
restore
```

- What do you find?

Commands: *preserve* and *restore*

When would we need *preserve* and *restore*?

It is useful when we handle big datasets and want to save the data memory we are asking for Stata.

- generate two-thousand new variables for analysis, but do not need them in the final dataset (gen)
- temporarily remove some variables (drop)
- generate pictures of aggregate trends (collapse)
- temporarily reshaping

Commands: *preserve* and *restore*

When would we need *preserve* and *restore*?

It is useful when we handle big datasets and want to save the data memory we are asking for Stata.

- generate two-thousand new variables for analysis, but do not need them in the final dataset (gen)
- temporarily remove some variables (drop)
- generate pictures of aggregate trends (collapse)
- temporarily reshaping

Drawbacks:

You need to first make sure the codes work before you *preserve* and *restore*.

How do we save the temporary results?

```
preserve
collapse (mean) avgage=age, by(famid)
list
cd ”/Users/gesun/Desktop/Bootcamp/04_Relational_datasets”
save kids_number.dta,replace
restore
```