Lecture 4: Data Cleaning

Yi Chen

Institute for Economic and Social Research

September, 2019

Outline

Outline

Data Input

- If a data file is .dta format, we can easily load the data with `use` command.
- But the world is not designed specifically for economists/Stata-users. Data can be in various format.
  - SPSS/SAS
  - Excel/.csv/.txt
  - Free format

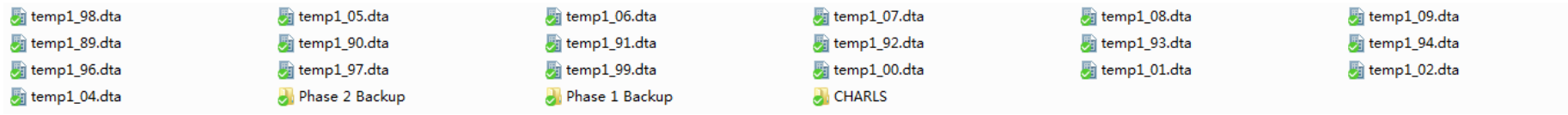
Input by Hand—Difficulty 0

- Stata allows for manual input.
  - Although sounds stupid, sometimes it could be useful.
  - A (small) portable data can be embedded in do-file. (Recall the portability principle)
- See the Stata example for inputting time series of CPI.

.dta—Difficulty 0

- use [*varlist*] [if *exp.*] [in *range*] [using *filename*] [, clear]
- `if` and `in` can be used for preliminary investigation if the data is extremely large
  - An alternative is to work on a random subsample first.
  - e.g., always working with large census data can be time consuming. You may consider:
    1. Generate a 1% random subsample.
    2. Program and debug with the small sample.
    3. When the program is ready, run on the full sample to see the results.

- Q: how to read following data sets with a loop?

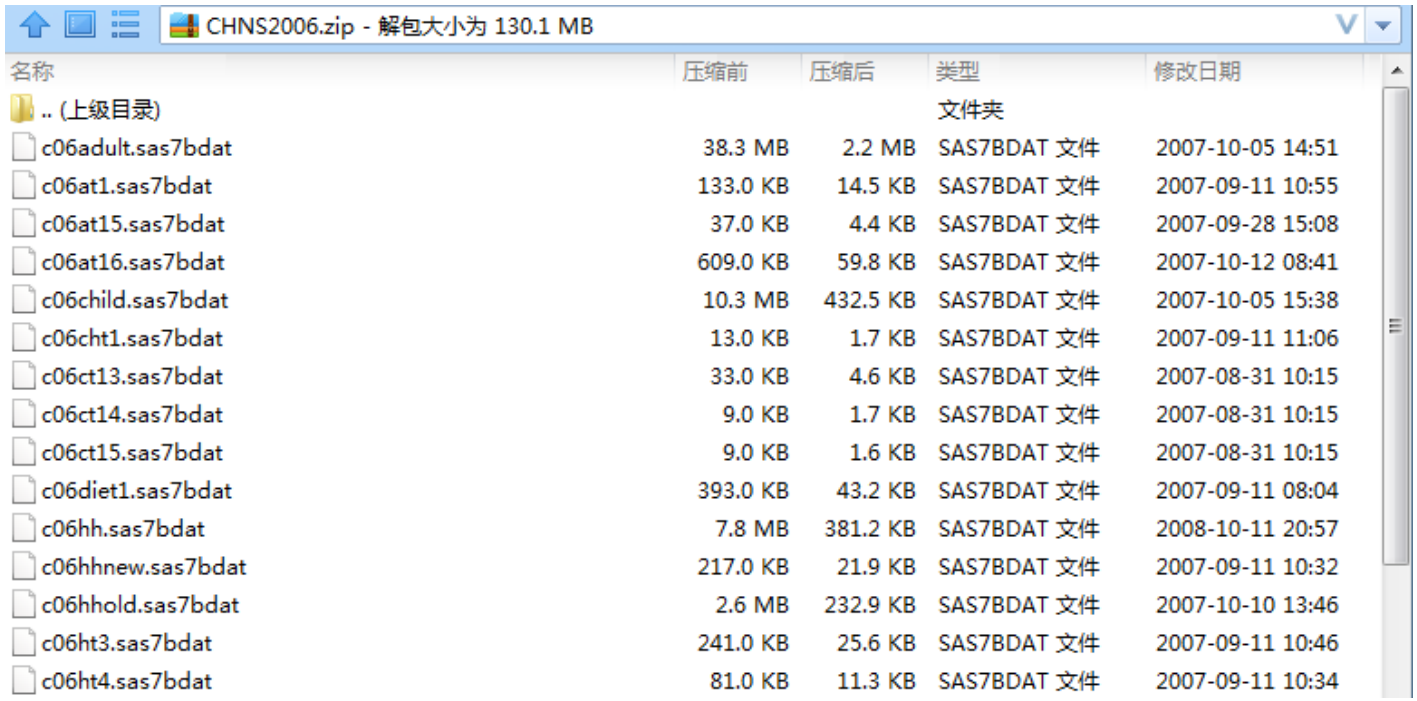


- A: use `substr` function

```
forvalues y = 1988/2009 {
local name = substr("`y'",3,2)
use "temp1_`name'.dta",clear
```

SPSS/SAS—Difficulty 1

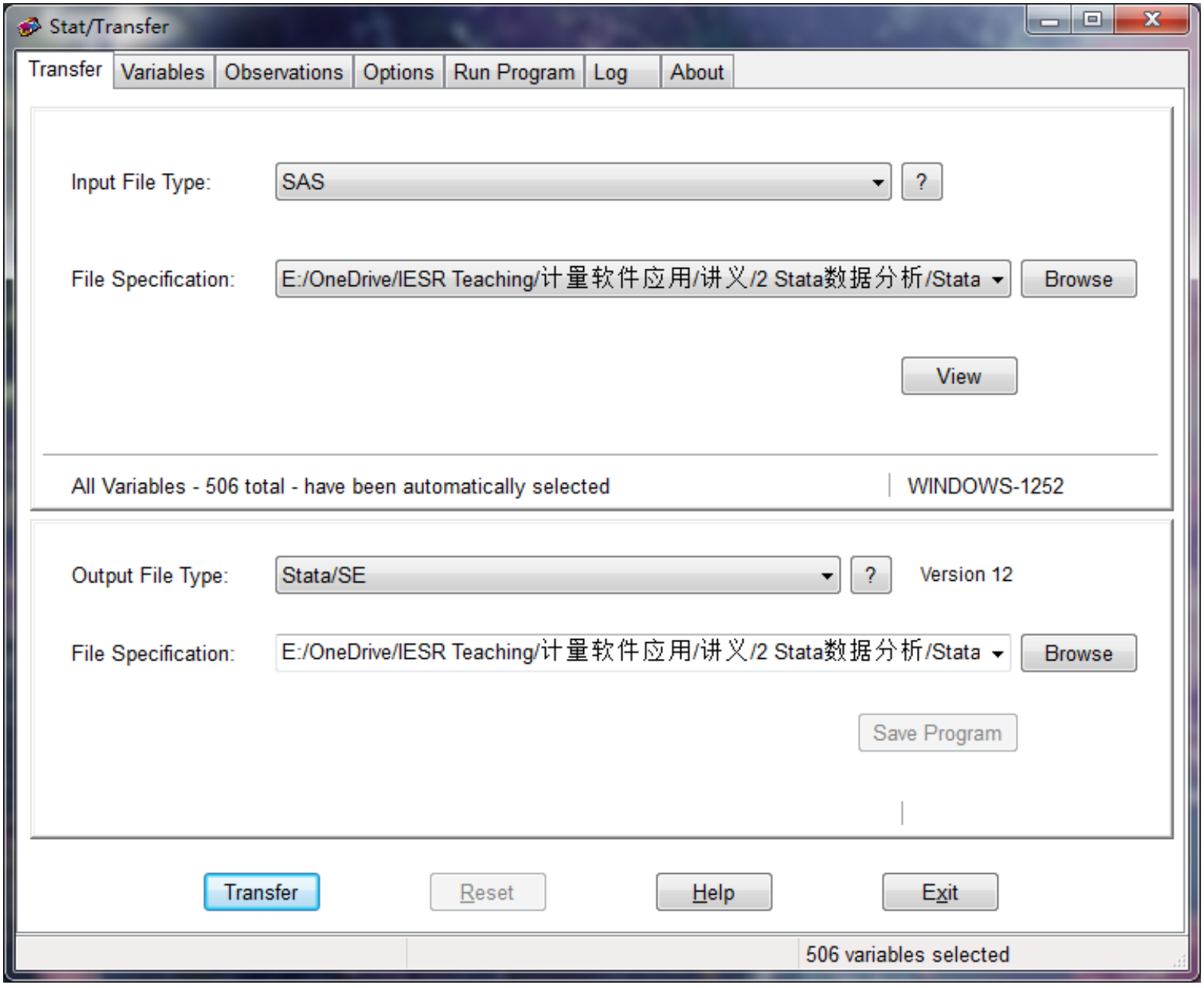
- Some data sets are given in other format. e.g., CHNS is originally in SAS format.
  - These types of data are still designed for statistics software. So they would be easier to be transformed to Stata format.



CHNS2006.zip - 解包大小为 130.1 MB

名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
c06adult.sas7bdat	38.3 MB	2.2 MB	SAS7BDAT 文件	2007-10-05 14:51
c06at1.sas7bdat	133.0 KB	14.5 KB	SAS7BDAT 文件	2007-09-11 10:55
c06at15.sas7bdat	37.0 KB	4.4 KB	SAS7BDAT 文件	2007-09-28 15:08
c06at16.sas7bdat	609.0 KB	59.8 KB	SAS7BDAT 文件	2007-10-12 08:41
c06child.sas7bdat	10.3 MB	432.5 KB	SAS7BDAT 文件	2007-10-05 15:38
c06cht1.sas7bdat	13.0 KB	1.7 KB	SAS7BDAT 文件	2007-09-11 11:06
c06ct13.sas7bdat	33.0 KB	4.6 KB	SAS7BDAT 文件	2007-08-31 10:15
c06ct14.sas7bdat	9.0 KB	1.7 KB	SAS7BDAT 文件	2007-08-31 10:15
c06ct15.sas7bdat	9.0 KB	1.6 KB	SAS7BDAT 文件	2007-08-31 10:15
c06diet1.sas7bdat	393.0 KB	43.2 KB	SAS7BDAT 文件	2007-09-11 08:04
c06hh.sas7bdat	7.8 MB	381.2 KB	SAS7BDAT 文件	2008-10-11 20:57
c06hhnew.sas7bdat	217.0 KB	21.9 KB	SAS7BDAT 文件	2007-09-11 10:32
c06hhold.sas7bdat	2.6 MB	232.9 KB	SAS7BDAT 文件	2007-10-10 13:46
c06ht3.sas7bdat	241.0 KB	25.6 KB	SAS7BDAT 文件	2007-09-11 10:46
c06ht4.sas7bdat	81.0 KB	11.3 KB	SAS7BDAT 文件	2007-09-11 10:34

- Stat/Transfer is a very useful software. If you are interested, try search `stcmd` to run Stat/Transfer with Stata commands.


Stat/Transfer
Transfer
Variables
Observations
Options
Run Program
Log
About
Input File Type:
SAS
?
File Specification:
E:/OneDrive/IESR Teaching/计量软件应用/讲义/2 Stata数据分析/Stata
Browse
View
All Variables - 506 total - have been automatically selected
WINDOWS-1252
Output File Type:
Stata/SE
?
Version 12
File Specification:
E:/OneDrive/IESR Teaching/计量软件应用/讲义/2 Stata数据分析/Stata
Browse
Save Program
Transfer
Reset
Help
Exit
506 variables selected

Excel/.csv/.txt—Difficulty 2

- insheet [*varlist*] [using *filename*] [, clear [no]names tab/comma/delimiter("char")]
- import excel [*varlist*] [using *filename*] [, clear sheet("*sheetname*") cellrange([*start*][:*end*]) firstrow]
  - option *sheet* makes Excel suitable to manage multiple .txt file.

"Pitfall" in *insheet*

- `insheet` is a very simple command to load data, but it has one caveat—it automatically assigns variable type.
  - See the Stata example.
- `infile`—more complicated but safer

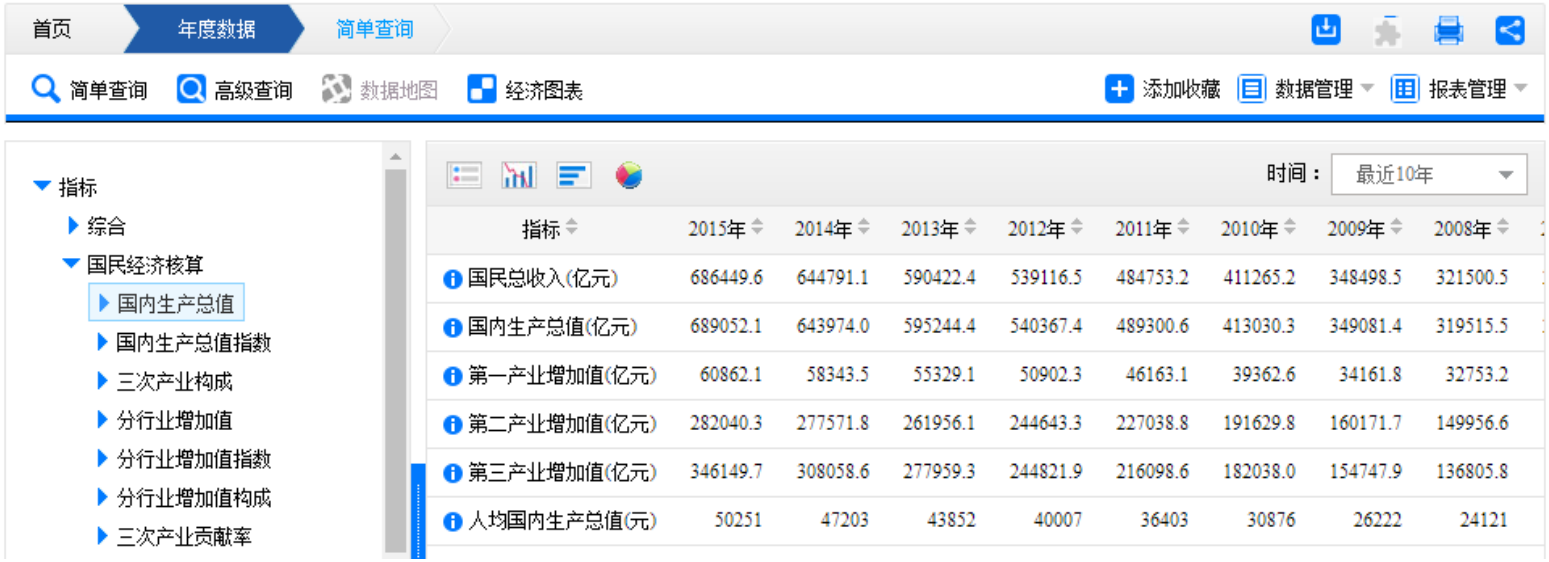
```
*- 整数的存储类型
   * byte      字节型       (-100, +100)
   * int       一般整数型   (-32000, +32000)
   * long      长整数型     (-2.14*10^10, +2.14*10^10), 即, 正负21亿

*- 小数的存储类型
   * float     浮点型    8  位有效数字
   * double    双精度    16 位有效数字

*- 字符型变量
   * str#      如 str20 表示该变量最多包含 20 个字符
   *          每个汉字占两个字符
```

Excel & Transpose

- In Stata, rows represent observations and columns represent variables.
- But in other data format this is now always the case.



指标	2015年	2014年	2013年	2012年	2011年	2010年	2009年	2008年
国民总收入(亿元)	686449.6	644791.1	590422.4	539116.5	484753.2	411265.2	348498.5	321500.5
国内生产总值(亿元)	689052.1	643974.0	595244.4	540367.4	489300.6	413030.3	349081.4	319515.5
第一产业增加值(亿元)	60862.1	58343.5	55329.1	50902.3	46163.1	39362.6	34161.8	32753.2
第二产业增加值(亿元)	282040.3	277571.8	261956.1	244643.3	227038.8	191629.8	160171.7	149956.6
第三产业增加值(亿元)	346149.7	308058.6	277959.3	244821.9	216098.6	182038.0	154747.9	136805.8
人均国内生产总值(元)	50251	47203	43852	40007	36403	30876	26222	24121

- Command `xpose` can be very helpful in this case.

Exercise 1: Import Data from Excel

One often-encountered conflict when important data from excel is—the column heads in excel are often in years (e.g., 1990, 2019) but Stata (and many other softwares) does not allow variable names to start with a number. How to solve this conflict?

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**SA51-53 Disposable personal income summary**														
2	Disposable personal income (thousands of dollars)														
3	Bureau of Economic Analysis														
4	State or DC														
5															
6	**GeoFips**	**GeoName**	**1948**	**1949**	**1950**	**1951**	**1952**	**1953**	**1954**	**1955**	**1956**	**1957**	**1958**	**1959**	**1960**
7	00000	United Sta	1.93E+08	1.93E+08	2.14E+08	2.36E+08	2.49E+08	2.64E+08	2.71E+08	2.9E+08	3.1E+08	3.28E+08	3.38E+08	3.59E+08	3.72E+08
8	01000	Alabama	2471947	2382967	2662225	3047380	3234463	3374775	3285389	3705568	3960676	4193495	4374707	4593637	4770021
9	02000	Alaska	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	724993
10	04000	Arizona	856875	892996	978633	1198121	1338733	1440476	1499279	1641513	1840024	2011001	2140105	2343756	2540833
11	05000	Arkansas	1529473	1443263	1545153	1738594	1803288	1807509	1783027	1951398	2003164	2069298	2152168	2343926	2361214
12	06000	California	16328095	16930872	18884915	21407772	23483086	25383334	26392235	28848373	31709645	33987167	35836292	39020143	41109939
13	08000	Colorado	1706932	1742194	1907826	2229155	2369643	2395991	2487957	2717047	2968282	3295732	3356561	3609123	3813206
14	09000	Connecticu	3127814	3120784	3481317	3900173	4192305	4583649	4732791	5114012	5556616	5953327	5762963	6143872	6348259
15	10000	Delaware	435656	481420	553910	595260	634145	684774	716885	815169	960967	962796	1004246	1042272	1096119
16	11000	District of	1632778	1689982	1790171	1840925	1890377	1867970	1851862	1826556	1919532	1987674	2013821	2042146	2055996
17	12000	Florida	2942337	3095135	3520448	3899139	4377369	4893982	5193561	5933568	6824376	7549472	8277135	9087567	9510181
18	13000	Georgia	3048038	3080560	3562973	4078257	4349731	4515538	4521434	4977742	5317198	5481685	5723536	6080316	6309543
19	15000	Hawaii	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	1416342
20	16000	Idaho	689809	684967	741924	819019	890716	860570	876662	927298	1004086	1075804	1085677	1146590	1167352
21	17000	Illinois	13992219	13398699	14696564	15891440	16524568	17734981	18120771	19427214	20922155	21836490	22190810	23533117	24114591
22	18000	Indiana	5181118	4973942	5621879	6382212	6641141	7322346	7008221	7539316	8092990	8338189	8427298	8912073	9301579
23	19000	Iowa	3850529	3264637	3776405	3961572	4186452	3990428	4383449	4160718	4403187	4886884	4946743	5080797	5161941

Exercise: transform the Excel data "disposable_income.xls" into Stata without any change in Excel file.

Exercise 1: Solution

- If you can modify the Excel file, one straight forward solution would be manually add a letter to all the column headers.
  - If there are numerous such Excel file...
  - If the years are not organized in a regular way...
- Therefore, we wish to automatize the above manual process without using Excel.

infile with a dictionary file—Difficulty 5

- Many data exists before the modern statistical software appears. So they are presented in the simplest form — numbers only.
- You will be given the information about the positions of each variable.
- e.g. Consumer Expenditure Survey, SEER (Surveillance, Epidemiology, and End Results Program)
  `https://seer.cancer.gov/popdata/popdic.html`

ffile001

1	10581912122220200	33755.352	0.0001	0.2	1.0	1.0	1.2	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15702.00	0.00	0.00	0.00	0.00	0.00	0.00
2	10582014122110200	71892.711	131834.7031	1.0	0.0	0.0	1.0	1	22500.00	0.00	0.00	0.00	0.00	1000.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	105821122 1220300	39505.320	0.0001	1.0	1.0	1.0	2.0	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	10582212122220200	48588.270	0.0001	1.5	1.2	2.8	5.0	6	54000.00	0.00	0.00	0.00	0.00	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	10582322122220200	54737.715	0.0001	2.0	0.0	2.2	4.0	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	10582414422110200	64205.008	118793.4841	0.5	2.0	2.0	3.0	3	4129.00	0.00	0.00	0.00	0.00	0.00	0.00	8850.00	0.00	0.00	0.00	0.00	0.00	2712.00
7	10582613412120200	46450.977	0.0002	1.0	0.0	0.0	1.0	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	10582813422110200	47881.789	0.0002	1.0	1.0	2.0	4.0	4	20000.00	0.00	0.00	0.00	0.00	0.00	0.00	5363.00	0.00	0.00	0.00	0.00	0.00	0.00
9	10583013122220100	51780.762	0.0001	2.8	2.0	3.2	3.0	3	3200.00	40000.00	0.00	0.00	0.00	0.00	0.00	9786.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10583323122220200	135007.375	0.0001	1.0	1.0	2.2	1.0	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	10583913122110100	56717.930	104696.2581	1.0	2.8	3.8	3.0	3	100000.00	-2200.00	0.00	0.00	0.00	0.00	0.00	5766.00	1164.00	0.00	0.00	1164.00	0.00	0.00
12	10584013222220200	62922.219	0.0001	0.0	0.0	1.0	2.0	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1092.00	0.00	0.00	0.00	0.00	0.00	0.00
13	10584111422110200	38031.102	0.0002	1.0	1.0	1.0	3.5	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	10584214422110200	58679.547	108570.1561	1.8	0.5	0.5	1.8	3	38000.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	10584312122220200	73178.953	0.0001	1.5	1.0	2.0	7.0	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	10584413122120100	63688.086	0.0002	1.0	1.0	1.0	1.0	1	37000.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	105846144 1220300	16570.883	0.0002	1.0	0.0	0.0	1.0	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	10584712222120200	57429.582	93968.8591	0.0	2.0	2.0	2.0	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	21108.00	0.00	0.00	0.00	0.00	0.00	0.00
19	10584813122110300	56878.094	104991.9061	2.0	1.0	2.0	2.0	2	181000.00	0.00	0.00	0.00	600.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	10584913222120300	52700.426	96640.4611	3.0	0.0	1.2	3.0	3	12000.00	0.00	110001.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	105851112 1220200	46245.332	0.0001	0.0	2.0	2.0	2.0	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	19728.00	0.00	0.00	0.00	0.00	0.00	0.00
22	10585213422110300	23172.014	0.0002	1.0	1.0	1.0	2.0	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	10585411222220200	54941.656	0.0001	2.0	0.0	0.0	2.0	2	110000.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	10585522122110300	57653.891	0.0002	4.0	4.0	5.0	4.0	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	10585611222110100	78770.188	141147.6091	1.0	1.2	1.2	2.0	2	11000.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

A. Family Records

Variable	Type	Position
NEWID	I7	1-7
BLSURBN	A1	8
REGION	A1	9
CUTENUR	A1	10
GOVHOUS	A1	11
PUBHOUS	A1	12
REPSTAT	A1	13
SREPSTAT	A1	14
INTMO	A2	15-16
INTYR	A2	17-18
TOTWT	F11.3	19-29
ADJWT	F11.3	30-40
FULLYR	I1	41
NUMEARN	F4.1	42-45
NUMAUTO	F4.1	46-49
VEHQ	F4.1	50-53
FAMSIZE	F4.1	54-57
MEMBCNT	I2	58-59
VAR(109)	109F10.2	60-1149
LAGINC(22)	22F10.2	1150-1369
REPFLAG(109)	109A1	1370-1478

What is a dictionary file?

- Dictionary file is in .dct format. But fundamentally it is a text file
- A dictionary file specifies
  - The length of a record
  - The position/length/type of a variable
- `infile using dfilename [, clear using(filename)]`
- In most cases, the dictionary file is provided together with the data.
  - But in some cases, you may need to write your own dictionary.

```
dictionary {
_lrecl(1478)
_column(1) NEWID %7f
_column(8) BLSURBN %1f
_column(9) REGION %1f
_column(10) CUTENUR %1f
_column(13) REPSTAT %1f
_column(19) TOTWT %11.3f
_column(30) ADJWT %11.3f
_column(41) FULLYEAR %1f
_column(54) FAMSIZE %4.1f
.
.
.
}
```

Storage Type v.s. Output Format

- There are two confusing "formats" in Stata. Storage type and output format.
  - Storage type: how a variable is stored in computer?
  - Output format: how a variable displays on the screen?
- Changing the storage type would actually change the size of the file.
  - Use `recast` command to change the storage type of a variable.
  - Use `compress` command to reduce the size of the file.
- Use `format` command to change the "appearance" of a variable.

Outline

Starting from One Component

- In most cases, data does not come as a complete file.
  - Different waves.
  - Different components (individual/household/community).
  - Different module (demographics/education/work).
- In this section, we focus on one single file.

Know your Data & Variables before your Start!

- Know your data—browse the website, read questionnaires and user guide, also look at previous studies using the same data.
  - Actually, these should be done before you decide to use the data!
- Know your input questions
  - Target sample; preceding questions; way of coding
- Understand your output variables
  - Education: level or years? How to treat dropouts? Adult education?
  - Income: monthly or hourly? Labor income or total income? How to treat income obtained at the household level (e.g., agricultural income, government subsidy)?

Missing Values

- Correctly understanding how missing values work in Stata is VERY VERY VERY IMPORTANT!
- "." is treated as a value larger than any number
  - Some commands would neglect missing values, such as `sum/regress/generate`
  - Some commands would treat missing values as infinitely large numbers, such as `count/keep`
  - There is no general rule, only experience would help.

Missing Values

- `keep if x > c` will keep the sample that x is greater than c OR x is missing.
- There can be different type of "missing"
  - Missing by construction, don't know, refuse to answer.
  - In CHARLS, ".d" is used to represent "don't know", ".r" is used to represent "refuse to answer"
- $. < .a < .b < ...$
- Use `keep if x > c & x < .` instead of `keep if x > c & x !=.`
- Use `drop if x >= .` instead of `drop if x == .`

```
. use "E:\Data Sets\CHARLS\2011\DATA\health_status_and_functioning.dta"

. tab da001, missing
```

Self Comment of Your Health	Freq.	Percent	Cum.
1 Excellent	63	0.36	0.36
2 Very good	748	4.25	4.61
3 Good	1,427	8.11	12.72
4 Fair	4,031	22.91	35.63
5 Poor	2,444	13.89	49.52
.	8,878	50.45	99.97
.d	4	0.02	99.99
.r	1	0.01	100.00
Total	17,596	100.00	

```
. count if da001>=4
  15,358

. count if da001>=4 & da001!=.
  6,480

. count if da001>=4 & da001<.
  6,475
```

How to Deal with Missing Values?

- Consequence of missing values
  - In a regression, you would lose the observation as long as any variable shown up is missing.
  - Non-random missing (Heckman Two-Step)
- First of all, you need to think about why the variable is missing. Maybe you can properly compute the value based on other variables.
  - e.g., in a panel survey, some questions will only be asked once in the baseline survey.
  - Therefore, in the follow-up surveys, these variables might be missing. But actually, you should back them up from previous surveys.
- If a value is truly missing,
  - In a time-series/panel data, you can use interpolation/extrapolation to replace the missing values.
  - In a cross-sectional data, you can consider replacing the missing values with group means or predicted value based on other observables.

$EXP = EXP1 + EXP2 + . + EXP4$

- In many cases, the variable of interest is composed of several sub-categories.
  - e.g., total household expenditures can be decomposed into: food expenses, cloth expenses, educational expenses, medical expenses, et al.
- Strictly speaking, we need all the sub-categories to be non-missing to construct the aggregate measure.
- But admittedly, it is kind of "wasteful" to discard an entire observation just because one sub-category is missing.
- See how CEX computes income. (Reading Material 4.1)
  - Before 2004—introduce the concept of "complete income reporter," if their respondents provide values for at least one of the major sources of income, such as wages and salaries, self-employment income, and Social Security income.
  - This method is similar to listwise deletion/complete-cases analysis.
  - Starting from 2004—multiple imputation. (Reading Material 4.2)

Multiple Imputation

- In statistics, there is a method named "multiple-imputation" to impute the missing values based on some Bayesian method.
- While the technical details will be skipped, the general idea is
  - Instead of generating a single predicted value, multiple predicted values will be generated for each missing value following certain rules.
  - A set of statistical method is developed to draw influence from such data structure.
- Why multiple predicted value instead of a single one?
  - "It accounts for missing-data uncertainty and, thus, does not underestimate the variance of estimates like single imputation methods."
  - A similar spirit: there are many "two-step" estimators, use the "twostep" option instead of doing the two-step estimation by hand! You will underestimate the standard error.

- Personally recommendations,
  - Exam the variables with lots of missing values VERY carefully.
  - Do the imputation/interpolation/extrapolation for the time-series data or time-predictable variables in a panel data.
  - For the cross-sectional data
    - Do the imputation if you miss some sub-categories of a larger categories. But first make sure it is truly "missing", not "zero"!
    - For descriptive purpose, multiple imputation method can be better.
    - For analysis purpose, single imputation is sufficient in most cases. Multiple imputation combined with more advanced econometric model can be technically difficult.
    - Do not do the imputation in other scenarios.
- Be VERY VERY VERY careful about the missing values in the dependent variable and the key independent variable.
  - In most cases, I would recommend delete observations containing missing values in key variables.
  - Imputation with predicted value could be dangerous. By construction, it violates the i.i.d. assumption. (Why?)

Several Useful Commands

- `lookfor`: search for variables
  - Useful for preliminary investigation. But ultimately you need to look into the questionnaires & user guide.
- `codebook`/`summarize`/`tabulate`: know your data well before your start.
- `renvars`
- `clonevar`
  - What's the difference from `gen` ?
- `recode`
- `egen`

Use `help` for further details.

The Art of Stata: collapse

- Here is how Stata officially describes `collapse`
  - Make dataset of summary statistics
  - collapse [(stat)] *varlist1* [[(stat)] ...] [*if*] [*in*] [*weight*] [, by(*varlist2*)]
- `collapse` is way more powerful and flexible than you could expect.
- If `help` is the start of *learning* Stata, `collapse`/`egen`/`reshape` is the start of *mastering* Stata.
  - We would see numerous applications later in this course.

Exercise 2: Compute Total Transfer from Children

[PROCEDURE: Skip to the procedure before CE011 if the respondent and spouse have no non-coresident children.] [程序：如果受访者及配偶没有不住在一起的孩子，则跳至CE011 前的程序控制.]

CE007. In the past year, did you or your spouse receive any economics supports from your non-coresident children? 过去一年，您或您的配偶从您的没住在一起的孩子那里收到过任何经济支持吗？

(1) Yes 是

(2) No 否→ Skip to CE011 请跳至CE011

CE008: Which child（ren）?那是哪一个孩子（哪些孩子）？ (choose all that apply可多选)

[Provide the list of children]

For each of the child checked in CE008, ask CE009. 对CE008 里选中的每一个孩子，询问CE009 。

*******CE009.** How much of the following did you receive from this child [CHILDNAME] in the past year?(specify the amount of each type of economics transfers). 过去一年你们从这个孩子【孩子名】那里获得了多少以下各种帮助？（请回答每一类经济帮助的数额）

(1) Regular monetary or in-kind support (e.g., money or in-kind support every month/quarter/half year/year, at fixed time) 定期给钱或实物（比如按月、按季度、按半年、按年给钱或实物，时间上大致固定）

Regular monetary support 定期给钱_____Yuan 元（CE009_1）

Regular in-kind support 定期给实物_____Yuan 元（CE009_2）

(2) Non-regular monetary or in-kind support (e.g., money or in-kind support at Spring Festival or/and Mid-Autumn Festival or/and birthday or/and wedding or/and funeral or/and others) 不定期给钱或实物 (比如逢年过节、各种节假日、生日、婚丧事、教育、医疗等情况下不定期给钱或实物)

Non-regular monetary support 不定期给钱_____Yuan 元（CE009_3）

Non-regular in-kind support 不定期给实物_____Yuan 元（CE009_4）

CE010: add unfolding brackets （100/200/400/800/1600yuan）for each type of money.请在此处给每种费用添加分级展开问题（100/200/400/800/1600 元）。

- Task: generate household-level aggregate transfer from children
- Input: for each households, there are 10*(4+4+3+3) variables
- Output: combine 10*(4+4+3+3) variables into one single variables
- Pay attention to the loop structure

Exercise 3: Work on a County-Level Data

- In a yearbook, the data structure often looks as follows
  - Province
    City
    County
- Another similar example is firm and branches.
- While such structure looks clear in a book, it is very difficult to work with a statistical software.

- Task: determine which province each county belongs to (`county_prov_match.xls`)
  - For one sheet it may seem easier to just do it manually.
  - But imagine if you are assigning cities instead of provinces for multiple years...
- Input: names in various format & knowledge about what are the "provinces"
- Output: mapping from counties to provinces

Short Summary

- As you can see, seemingly simple task like data input can be not so "trivial."
- Some variables can be very difficult to generate.
- Keep practicing! The marginal cost of doing a second time is much smaller than doing for the first time.

Outline

Combine Multiple Data Sets

- Combine data "vertically."
  - Multiple waves
  - Different surveys
- Combine data "horizontally."
  - Combine at the same level (e.g, individual–individual)
  - Combine at different levels (e.g., individual–family)

Combine Data "Vertically"

- append using *filename* [*filename* ...] [, generate(*newvar*)]
- Be VERY careful: the same variable should have the same type in different data sets
  - Recall the example in Section 1, if you append a string variable to a numerical one, the string value will be transited to missing.
- HOWEVER, the difficult part is not using the `append` command. It is to make the data sets ready for `append`
- Make sure variables with the same name is actually the "same."

Question on self-rated health in CFPS 2010:

P3 您认为自己的健康状况如何？

1. 健康　　2. 一般　　3. 比较不健康　　4. 不健康　　5. 非常不健康

Self-rated health in CFPS 2012:

P201 您认为自己的健康状况如何？

1.非常健康　　2.很健康　　3.比较健康

4.一般【不读出】　　5.不健康

Wage income in CFPS 2010:

K 部分　个人收入

【CAPI】#01 G4 选择“1”，则跳至 K2。

#02 如果 G3 选择“5”，则跳至 K3；否则提问 K1。

K1 下面的问题涉及去年您个人的各项非经营性收入情况**【出示卡片】**

F1：“非经营性收入”指通过贡献自己的体力、智力，从非自己具有产权的机构或/和个体中获得的收入，如工资、奖金等；农民通过经营自己的土地或其他资产如水面等获得收入被计入了家庭收入或/和经营性收入，故不在此列。

K101 去年您平均每月工资有多少？_____0..50000 元

访员注意：

（1）如果没有，请输入“0”；

（2）如果工资奖金无法分开就在本题中录入总数，下一题选择“不清楚”。

K102 去年您平均每月的浮动工资、加班费以及各种补贴和奖金有多少？_____0..50000 元

访员注意：如果没有，请输入“0”。

K103 去年您的年终奖金等有多少？_____0..500000 元

Wage income in CFPS 2012:

G403 您这“【CAPI】加载 G402”份工作单位名称分别是：_____(JobBName1)、____(JobBName2)、…_____ (JobBNamen)

.

.

.

【CAPI】从 JobBName1 开始，依次引语开始，询问 G408 至 G424，直至最后一份工作结束。

.

.

.

G417 把奖金等各种收入以及您刚刚所说的现金福利都算在内，您过去一年税后从这份工作中总共拿到多少钱？凡是以现金形式发放的或打在工资卡里的收入都算。_____0..10,000,000 元

【CAPI】

#1 Soft Check：<=200,000。“访员注意：受访者过去一年税后收入超过二十万。”

#2 如果 G417=“不知道”或“拒答”，继续回答 G418；否则跳至 G419。

Even if the survey conductor prepared the panel for you, still be very careful!

```
. use "E:\Data Sets\CHNS\CHNS Longitudinal Data\m12jobs.dta"

. tab b6 wave, missing
```

primary occupation : type of work unit	survey year									Total
	1989	1991	1993	1997	2000	2004	2006	2009	2011	
-9	0	0	0	0	0	0	0	0	101	101
1	2,242	2,105	1,856	1,501	1,483	174	147	177	288	9,973
2	1,167	788	780	547	369	474	524	560	833	6,042
3	732	712	673	404	401	384	296	285	506	4,393
4	4,851	26	3,165	4,510	4,639	175	172	131	249	17,918
5	2,639	5,432	2,112	1,143	1,489	109	90	104	138	13,256
6	947	284	239	54	96	2,509	2,593	2,608	2,442	11,772
7	87	31	0	264	356	1,421	1,618	1,757	2,426	7,960
8	0	0	0	0	0	67	65	80	177	389
9	0	0	137	126	135	187	208	188	293	1,274
.	2,798	1,262	1,225	2,578	3,612	4,517	4,199	4,274	5,550	30,015
Total	15,463	10,640	10,187	11,127	12,580	10,017	9,912	10,164	13,003	103,093

Type of work unit in CHNS 2000

Type of work unit in CHNS 2004

3. 工作单位是何种类型？ □□B6a

01 政府机关
02 国有事业单位和研究所
03 国有企业
04 小集体(如乡镇所属)
05 大集体(县、市、省所属)
06 家庭联产承包农业
07 私营、个体企业
08 三资企业（属于外商、华侨和合资）
09 其他（具体说明：_____）
-9 不知道

How to Assure the Variable is compatible in Different Waves?

- The idea has been mentioned repeatedly in this lecture: know your input & output.
- *Input*: reading the user-guide and questionnaire carefully, know all your variables (in each wave) carefully before you proceed!
- *Output*: describe your compiled variables by wave, check whether there is any unexpected sudden change.
  - `collapse (mean/count) varlist, by(wave)`
  - For discrete variables, `tab var wave, missing nolabel`

Set up the Panel

- If the data is pooled cross-sectional data, we are done after `append`. If the data structure is panel, we need one additional step to link individuals across different waves `xtset`
- xtset *panelvar timevar*
- *panelvar*—only one single numeric variable is allowed
  - Sometimes individuals are identified by multiple variables, e.g. hhid + pid
  - egen `group()` function can be very helpful in constructing *panelvar*.
  - But make sure you use it AFTER `append`, not `BEFORE`! See the Stata example.

- *timevar*—by default, the lag operator refers to the time value minus one.
  - e.g., there are two waves, 2000 and 2002. The "lag" of 2002 is 2001, not 2000.
  - Although the `delta(#)` option in `xtset` can change the default gap, but it is not recommended because the gap is often irregular.
  - Again, `egen group()` function is extremely helpful in this scenario, especially when you wish to construct *timevar* from multiple time variables (e.g., year + month)

Update the Missing Values

- As mentioned in the previous section, after constructing the panel you should be able to confidently update some missing values.
- For example, if you know an interviewee's mother is junior high graduate and age 62 in the year 2010, but those two variables are missing in the year 2012. In this case, you can infer that in 2012, mother's education should still be junior high graduate and the age should be 64.
- What should we do if we draw different answers from different observable values? e.g., 2010 primary, 2012 missing, 2014 junior high
  - No general solution. Measurement error is also inevitable in a survey data.
  - Personally, I refer to the latest survey.
- See the Stata example.

Combine Data "Horizontally"

- merge [1:1 m:1 1:m] *varlist* using *filename* [, options]
- *varlist* should be able to uniquely identify the observation on the "1" side (including missing values!)
- After each merge, a variable *_merge* will be generated. Tabulate this variable to guarantee that you have an ideal match

numeric code	equivalent word (*results*)	description
1	master	observation appeared in master only
2	using	observation appeared in using only
3	match	observation appeared in both
4	match update	observation appeared in both, missing values updated
5	match conflict	observation appeared in both, conflicting nonmissing values

- Similar to append. Also pay attention to the name of the variable before `merge`! See the Stata example for an application.

"Uniquely Identified"

In practice, getting "uniquely identified" can be very painful...

- Missing values in merge *varlist*
  - Probably no alternatives other than dropping those observations.
- Duplicates in merge *varlist*
  - A commonly used approach is `duplicates drop varlist, force`
  - My suggestion is... wait a second!
- `duplicates drop varlist, force` keep the first observation that meets the *varlist* requirement
  - During the merge process, the order of the observations can be messed up. (see the previous example)
  - Even if you are forced to use this approach as the nuclear option, `sort varlist, stable` before `duplicates drop` to guarantee replicability

Duplicated Observations

Probably one of the most hated sentence in Stata:
variable *varlist* does not uniquely identify observations in the *data*

What should we do in this case?

1. Use `duplicates report` to check frequencies of duplications. If there are too many duplications...
   - Your choice of *varlist* maybe not appropriate for identification.
   - 1:1? m:1? 1:m?
   - Long ID without *double* type?
2. Use `duplicates list` and `duplicates tag` to further look at duplicated observations.
3. Use `duplicates drop` or by hand to delete the duplicated observation.

Many-to-Many Merge

- Imagine you want to create a parent-child pair data from following two datasets:
  1. Child data (hid, cid)
  2. Parent data (hid, pid)

  The two data should be linked by *hid*, which does not uniquely identify child/parent in each data.
- You may wish to use `merge m:m` in this scenario. But Stata manual says:
  - "Because m:m merges are such a bad idea, we are not going to show you an example."
  - "Use of merge m:m is not encouraged."
- A more suitable command is `joinby` (See Stata example)
  - "Form all pairwise combinations within groups"

Outline

What's Remaining?

- Sample selection
- String values
- Log transformation
- Label the variables

Sample Selection

The final analysis rarely uses the full sample.

- Group of interest
- Complete information
- Drop outliers

Keep Track of Observations

- Try to delete the observations after combining the data.
- Record how many observation you lose in each step. This is relevant to
  - How strong the restriction is?
  - How special the group of interest is?
  - The external validity of the subsample

Drop Outliers

- It is well-known that least square estimates are sensitive to extreme values.
- The external command `winsor2` is very helpful.
  - Q: What is the difference between winsorizing and trimming?
- Describe and know your data well BEFORE you decide to drop outliers!
  - `summarize varlist, detail`, or `histogram varname`
  - Some variables naturally bunch at certain values, e.g., zero wage. In this case, left truncation is not a good idea.
  - Sometimes extreme values are of interest, e.g., medical expenses.
  - Admittedly, it is very hard to distinguish extreme values from measurement error without further information.
- When there are multiple criteria for outliers
  - Better to mark them first and then drop them than to drop them step by step. (See Stata example)

String Values appear more Often than You might Expect

- 3000 is a value, "3000" is a string. So does $3000 and 3,000.
- ID number (don't forget the last digit could be "X"!)

- Nonstandard date format such as "1997/07/01"
- Nonstandard Stata type missing, such as " ", "N/A"
- Reading material 4.3

Fetch Information from String Values

- String values can never be directly used in an econometric model. Therefore, somehow they need to transformed to a numerical value.
- `substr(s,n1,n2)` function: extract information from string values
  - Recall the example of reading data named "temp99", "temp00", "temp01" et al.
- `strpos(s1,s2)` function: the position in *s1* at which *s2* is first found; otherwise, 0
  - Exercise 3 as an example

Transition between String Values and Numerical Values

- `destring`/`tostring`: Convert string variables to numeric variables and vice versa
- `encode`/`decode`: Encode string into numeric and vice versa
- `real(s)`/`string(n)`: s converted to numeric or missing; n converted to a string
- So... What's their difference?

encode

- Three string values, “100”, “200”, “1500”. If encoded with default option, you will obtain numerical values of 1, 3, 2.
- From the above example, you can see two important features of `encode`
  - It automatically “recodes” string values to numerical values.
  - The order of `encode` follows the alphabetical order, which is not necessarily what we want.
  - The order issue can be solved combined with `label values`

real(*s*) versus destring

- Since `real()` is a function, it can be applied to both variables and scalar/local. -destring- is a command and can only be applied to a variable.
- `destring` is "slightly smarter" in the sense that they will optimally decide the storage type of the new variable.
- Why only "slightly smarter"? Both methods cannot recognize special characteristics, such as "$1500" and "1,500".
- Solution: `subinstr(s1,s2,s3,n)`
  - "The first *n* occurrences in *s1* of *s2* have been replaced with *s3*"
  - Tip 1: *s3* could be null ("", not " "). This means simply deleting *s2*
  - Tip 2: *n* could be ".", recall missing implies infinitely large in Stata.

Log Transformation

- Log transformation is a very useful and convenient method for analysis.
- I will say something about its disadvantage.
- Disadvantage 1—will not match the mean
  - In principles of econometrics, we know $\overline{\widehat{y}} = \overline{y}$
  - But such equation does not hold after log transformation, $y \to \log(y) \to \widehat{\log(y)} \to \widehat{y} = \exp\left(\widehat{\log(y)}\right)$
  - Then $\overline{\widehat{y}} \neq \overline{y}$
- Disadvantage 2—zero values and negative values
  - A commonly used practice is to drop negative values and use $\log(1+X)$ transformation.

Log Transformation with Zeros

- In principle, $\log(C + X)$ works for any positive value of C.
  - The question is, how to choose C? Alternatively, is $C = 1$ a good choice?
- If $C = 1$ is too large, imagine income X is measured in thousands. Then you are implicitly giving 1,000 for free to everyone.
- If $C = 1$ is too small, imagine income X is measured in 0.01 cents. Then you are adding an outlier to the regression.
  e.g., $\log\left(\frac{1}{1 \text{ billion}}\right) = -\log(1 \text{ billion})$
- As a rule of thumb, C equals to one half of the smallest, non-zero value.

Extensive Margin and Intensive Margin

The most rigorous approach of handling zero values is to explicitly discuss the extensive margin (zero versus non-zero) and the intensive margin (conditional on non-zero).

- Check the frequency of zeros. If there are lots of zeros, then you should consider either explicitly discussing the extensive margin or focusing on the subsample with positive values.
- If there are not so many zeros, probably how you transform X will not affect the result.

Labeling

- Why labeling is important?
  - For your advisor/coauthor
  - For your future self
  - Convenient for results output, such as `estout`
- There are three kinds of "labels" in Stata
  - Labeling data
  - Labeling variables
  - Labeling values

```
. tab qa2
```

Label Variables

您现在的户口状况是	Freq.	Percent	Cum.
不知道	2	0.24	0.24
农业户口	598	71.19	71.43
非农业户口	238	28.33	99.76
没有户口	1	0.12	99.88
非中国国籍	1	0.12	100.00
Total	840	100.00	

Label Values

Labeling

- label variable *varname* "*label*"
- Labeling values is slightly more complicated. You need to distinguish *varname* and *lblname*
  - *lblname* can be the same as *varname*
  - The original data often contains pre-defined *lblname* (`label dir`, `label list`). Therefore, when defining your own value label, name it like "L_edu"
  - The same value label can be assigned to multiple variables.
- label define *lblname* # "*label*" [# "*label*" ...] [, add modify replace]
  - modify: change the label for one specific value #
  - replace: re-define the entire *lblname*
- label values *varlist lblname*
- It's possible to transform between "values" and "labels" using `encode`/`decode`.

Exercise 4: Repeated Labeling

- You are given a text file named "stdpop.18ages". With a dictionary file given in the following slide,
- Your task
  1. Load the data
  2. Assign proper label values to the age group. Try your best to do "automatically."

Variable Name and Values	Start Column	Length	Data Type
Standard 006 = World (Segi 1960) Std Million (19 age groups) 007 = 1991 Canadian Std Million (19 age groups) 005 = European (Scandinavian 1960) Std Million (19 age groups) 008 = 1996 Canadian Std Million (19 age groups) 010 = World (WHO 2000–2025) Std Million (19 age groups) 141 = 1940 US Std Million (19 age groups) 151 = 1950 US Std Million (19 age groups) 161 = 1960 US Std Million (19 age groups) 171 = 1970 US Std Million (19 age groups) 181 = 1980 US Std Million (19 age groups) 191 = 1990 US Std Million (19 age groups) 201 = 2000 US Std Million (19 age groups) 203 = 2000 US Std Population (19 age groups - Census P25-1130) 202 = 2000 US Std Population (single ages to 84 - Census P25-1130) 205 = 2000 US Std Population (single ages to 99 - Census P25-1130) 011 = World (WHO 2000–2025) Std Million (single ages to 84) 012 = World (WHO 2000–2025) Std Million (single ages to 99) 001 = World (Segi 1960) Std Million (18 age groups) 002 = 1991 Canadian Std Million (18 age groups) 003 = European (Scandinavian 1960) Std Million (18 age groups) 004 = 1996 Canadian Std Million (18 age groups) 009 = World (WHO 2000–2025) Std Million (18 age groups) 140 = 1940 US Std Million (18 age groups) 150 = 1950 US Std Million (18 age groups) 160 = 1960 US Std Million (18 age groups) 170 = 1970 US Std Million (18 age groups) 180 = 1980 US Std Million (18 age groups) 190 = 1990 US Std Million (18 age groups) 200 = 2000 US Std Million (18 age groups) 204 = 2000 US Std Population (18 age groups - Census P25-1130)	1	3	numeric
Age Age group data: 000 = 0 years 001 = 1–4 years (or 001 = 0–4 years for 18 age groups) 002 = 5–9 years 003 = 10–14 years 004 = 15–19 years ... 017 = 80–84 years 018 = 85+ years Single age data: 000 = 0 years 001 = 1 years 002 = 2 years ... 085 = 85+ years (for single ages to 84) 100 = 100+ years (for single ages to 99)	4	3	numeric
Standard Population	7	8	numeric