

# Applied Epidemiology I: Data clearance

## A review of using Stata

Enoch Yi-Tung Chen

Department of Medical Epidemiology and Biostatistics, Karolinska Institutet

November 26, 2020

# Acknowledgements

This course material is based on my learning from Anastasia Lam's teachings in last year's Applied Epidemiology I lab sessions, and readings from *Epidemiology* by Gordis [1], *A First Course in Probability and Statistics* by Goldsman and Goldsman [2], *Principles of Biostatistics* by Pagano and Gauvreau [3], and *Biostatistics I* by Gabriel and Frumento [4]. I especially want to thank Marlene Stratmann for reviewing the slides and Prof. Paul Dickman for providing me with suggestions to improving the teaching.

# Outline (also learning outcomes)

## ① Set up working directory

## ② Import and save data

- Import

- Save

## ③ Manage datasets

- Merge

- Append

## ④ Get to know the data

- Summarize

- Describe

- Codebook

- List

## ⑤ Manage variables

- Numeric and string

- Drop/Keep

- Label

- Rename, recode, generate, replace

- Sort, by, if, in

- Operators

## ⑥ References

# Set up working directory

- Working directory is the folder where all your files are stored, and should be set each time you start.

# Set up working directory

- Working directory is the folder where all your files are stored, and should be set each time you start.

- Where is it?

```
. cd
```

```
/Users/Desktop
```

```
. pwd
```

```
/Users/Desktop
```

# Set up working directory

- Working directory is the folder where all your files are stored, and should be set each time you start.
- Where is it?
  - . cd  
/Users/Desktop
  - . pwd  
/Users/Desktop
- Change working directory
  - // Change working directory to Download
    - . cd "/Users/Download"
  - Click File - Change Working Directory

# Import and save data: Import

- **Excel (.xls or .xlsx)**

```
import excel filename, clear firstrow
```

- **Delimited (.csv) or text (.txt)**

```
import delimited filename, clear  
infile filename, clear
```

- **Stata (.dta)**

```
use filename, clear
```

- **SAS (.xpt)**

```
fdause filename, clear
```

- Save your dataset as a Stata file .dta
- The replace option lets you overwrite the existing dataset.  
`save "filename", replace`



# Manage datasets: Merge

merge adds new variables from a second dataset to your existing dataset.  
(Make the dataset wider)

```
. sysuse cancer, clear  
(Patient Survival in Drug Trial)
```

```
. gen id = _n
```

```
. keep id
```

```
. merge 1:1 id using cancer
```

Result	# of obs.
not matched	0
matched	48

(`_merge==3`)

# Manage datasets: Append

append adds new observations to existing variables in your current dataset.  
(Make the dataset longer)

```
. use cancer_drug12, clear
```

(Patient Survival in Drug Trial)

```
. append using cancer_drug3.dta // append patients using drug 3
```

# Get to know the data: Summarize

`summarize` gives summaries for all your variables, such as number of observations, mean, standard deviation, etc.

```
. sysuse cancer, clear  
(Patient Survival in Drug Trial)  
  
. keep if drug ==1 | drug == 2  
(14 observations deleted)  
  
. summarize age // One variable only (age)
```

Variable	Obs	Mean	Std. Dev.	Min	Max
age	34	56.41176	6.010686	47	67

# Get to know the data: Describe

describe gives descriptions for all your variables, such as storage type and labels.

```
. describe age
```

variable name	storage type	display format	value label
age	byte	%8.0g	Patient's age at start of exp.

# Get to know the data: Codebook

codebook is a combination of summarize and describe and will give a detailed summary of all your variables, including mean, sd, range, percentiles, missing, frequency, etc.

```
. codebook age
```

---

age	Patient's age at start of exp.
-----	--------------------------------

---

```
      type:  numeric (byte)
      range:  [47,67]
unique values: 15
      units:  1
      missing .: 0/34
      mean:   56.4118
      std. dev: 6.01069
percentiles:  10%      25%      50%      75%      90%
               49       51       56       61       65
```

# Get to know the data: List

`list` lists the observations of specified variables.

```
. list      age if age < 50
```

```
    +-----+  
    |  age  |  
    +-----+  
12. |   49  |  
15. |   49  |  
18. |   49  |  
25. |   49  |  
33. |   47  |  
    +-----+
```

# Manage variables: Numeric and string

**Numeric:** byte, integer, long, float, double – all types of numeric variables that just differ based on min and max length

**String:** character variables with a certain length (*str#*)

# Manage variables: Drop/Keep

- drop is used to delete variables or observations.
- keep is used to keep variables or observations.

```
. sysuse cancer, clear  
(Patient Survival in Drug Trial)  
. drop if drug ==1 | drug == 2  
(34 observations deleted)
```

```
. sysuse cancer, clear  
(Patient Survival in Drug Trial)  
. keep if drug ==1 | drug == 2 // So drug == 3 will be dropped  
(14 observations deleted)
```



# Manage variables: Label

- label helps you keep track of your dataset and variables, and helps others understand your data.

```
. // Label a dataset  
. label data "cancerdata"
```

Properties	
► Variables	
▼ Data	
Frame	default
► Filename	cancer.dta
<b>Label</b>	<b>cancerdata</b>
Notes	

```
. // Label variable in the "Variables" window  
. label variable drug "1=placebo, 2=mild, 3=strong"
```

Variables		
	Name	Label
<input checked="" type="checkbox"/>	studytime	Months to death or end of exp.
<input checked="" type="checkbox"/>	died	1 if patient died
<input checked="" type="checkbox"/>	drug	1=placebo, 2=mild, 3=strong

# Manage variables: Label

```
. // Label define claims the value label  
. label define drug 1 "placebo" 2 "mild" 3 "strong"  
  
. // Label value then assigns the label to the variables  
. label values drug drug
```

drug[1]			1
	studytime	died	drug
1	1	1	placebo
2	1	1	placebo

# Manage variables: Rename, recode, generate, replace

- `rename` changes the name of a variable.  
`. rename died death`
- `recode` changes variable values.  
`. recode drug (3=4)`
- `generate` creates a new variable.  
`. generate placebo = 1 if drug == 1`
- `replace` replaces existing variables (or variable values).  
`. replace placebo = 0 if drug != 1`

# Manage variables: Sort, by, if, in

- sort orders observations in ascending order.  
`. sort death`
- by executes a command within a specified variable (e.g. by age group), but data should be sorted first.  
`. by death: summarize`
- bysort combines the by and sort commands into one.  
`. bysort death: summarize // by death, sort: summarize`
- if is used to select by a condition.  
`. list age if death == 1`
- in is used to select by observations.  
`. gen id = _n`  
`. list id 1/10`

# Manage variables: Operators

Operator	Purpose	Example
==	Evaluates if true/false	summarize if sex==1
~= or !=	Indicates 'not equal'	summarize if sex!=0
<, <=	Less than (equal to) or greater	summarize if age<35
>, >=	than (equal to)	
&	Indicates 'and'	summarize outcome if sex==1 & age>=60
	Indicates 'or'	gen x=1 if a==1 & (b==1   c==1)

# References

1. Gordis L. *Epidemiology*. Philadelphia, PA: Elsevier/Saunders, 2014. ISBN 9781455737338.
2. David Goldsman PG. *A First Course in Probability and Statistics*. Georgia Institute of Technology, 2020.
3. Marcello Pagano KG. *Principles of Biostatistics*. Cengage Learning, Inc, 2000. ISBN 0534229026.
4. Erin Gabriel PF. Epidemiology PhD program, Karolinska Institutet, 2020.