Intro to Longitudinal Data Analysis using Stata (Version 10)

Part A: Overview of Stata

I. Reading Data:
   • `use`
     Read data that have been saved in Stata format.

   • `infile`
     Read “.raw” and “.data” data and “dictionary” files.

   • `insheet`
     Read spreadsheets saved as “CSV” files from a package such as Excel.

II. Do Files
   • What is a do file?

     A “do” file is a set of commands just as you would type them in one-by-one during a regular Stata session. Any command you use in Stata can be part of a do file. Do files are very useful, particularly when you have many commands to issue repeatedly, or to reproduce results with minor or no changes.

     Example: `lab1.do`
     *the path and name of the files are specific to your computer;
     *change the directory to where you have saved the files for use in lab 1
     `cd "C:\Documents and Settings\Sandrah Eckel\Desktop\LDA lab1"

     `log using "lab1.log"
     `infile week1-week9 using "pigs.data"
     `save "pigs.dta"
     .
     .
     .

     You can edit a do file anywhere then save as a file with the extension “.do”. In Windows or Mac, you can type `doedit` in Stata to open and edit any do files.

   • Where to put a do file?
     Put the do file in the working directory of Stata.

   • How to run a do file?

     `do mydofile`

     Example: `do lab1`
III. Ado files

• What is an ado file?

An ado file is just a Stata program. You can use it as a command. A *.ado file usually contains a program called * in it. For example, the first non-comment line “autocor.ado” is

```
program define autocor
```

• Where do I save ado files?

Save the .ado files and the corresponding .hlp files in your current directory, in your personal Stata "ado" directory, or in a directory where Stata will know where to look for them. Use “adopath” to find out where Stata is looking for ado files. Here is an example in a Windows PC (Ado directory may be different among different platforms).

```
. adopath
[1] (UPDATES)   "C:\Program Files\Stata10\ado\updates/"
[2] (BASE)      "C:\Program Files\Stata10\ado\base/"
[3] (SITE)      "C:\Program Files\Stata10\ado\site/"
[4]              "."  
[5] (PERSONAL)  "c:\ado\personal/"
[6] (PLUS)      "c:\ado\plus/"
[7] (OLDPLACE)  "c:\ado/"
```

• How do I run an ado file?

Use the name of the program as a command as you use other default Stata commands. For example:

```
. autocor outcome time id
```

IV. Convert data from wide to long or vice versa

• Two forms of data: wide and long

Different models may require different forms of data in Stata. For instance, “logit” or “logistic” model in Stata prefers a wide format.
Example: Incomes of 3 individuals in 1980-1982

(wide form)  (long form)

-1-       -i-       -j-       -x_ij-
   id   sex  inc80 inc81 inc82   id   year  sex  inc

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>5000</td>
<td>5500</td>
<td>6000</td>
<td>1</td>
<td>80</td>
<td>0</td>
<td>5000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2000</td>
<td>2200</td>
<td>3300</td>
<td>1</td>
<td>81</td>
<td>0</td>
<td>5500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3000</td>
<td>2000</td>
<td>1000</td>
<td>1</td>
<td>82</td>
<td>0</td>
<td>6000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2000</td>
<td>2200</td>
<td>3300</td>
<td>2</td>
<td>80</td>
<td>1</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3000</td>
<td>2000</td>
<td>1000</td>
<td>3</td>
<td>81</td>
<td>0</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• reshape converts data from one form to the other:
  • From Wide to Long
    . reshape long inc, i(id) j(year)
  • From Long to Wide
    . reshape wide inc, i(id) j(year)

• Examples: Cows Data

. infile prot1-prot19 using cows.lupins.data
. gen id = _n
. order id
. list in 1/2

<table>
<thead>
<tr>
<th></th>
<th>id</th>
<th>prot1</th>
<th>prot2</th>
<th>prot3</th>
<th>prot4</th>
<th>prot5</th>
<th>prot6</th>
<th>prot7</th>
<th>prot8</th>
<th>prot9</th>
<th>prot10</th>
<th>prot11</th>
<th>prot12</th>
<th>prot13</th>
<th>prot14</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>3.69</td>
<td>3.38</td>
<td>3.05</td>
<td>3.5</td>
<td>3.09</td>
<td>3.3</td>
<td>3.07</td>
<td>3.22</td>
<td>2.97</td>
<td>3.42</td>
<td>3.59</td>
<td>3.77</td>
<td>3.74</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>4.2</td>
<td>3.35</td>
<td>3.37</td>
<td>3.07</td>
<td>2.82</td>
<td>3.05</td>
<td>3.12</td>
<td>2.85</td>
<td>3.2</td>
<td>3.38</td>
<td>3.25</td>
<td>3.26</td>
<td>3.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.78</td>
<td>3.78</td>
<td>3.22</td>
<td>3.22</td>
<td>3.25</td>
<td>3.25</td>
<td></td>
<td></td>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>prot15</th>
<th>prot16</th>
<th>prot17</th>
<th>prot18</th>
<th>prot19</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.78</td>
<td>3.78</td>
<td>3.77</td>
<td>3.77</td>
<td>3.55</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td>3.41</td>
<td>3.28</td>
<td>3.25</td>
<td></td>
</tr>
</tbody>
</table>

. reshape long prot , i(id) j(week)
(note: j = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19)

Data	wide -> long

Number of obs.	27 -> 513
Number of variables	20 -> 3
j variable (19 values)	-> week
xij variables:
prot1 prot2 ... prot19 -> prot
. list in 1/20

    id  week  prot
  1.   1    1   3.69
  2.   1    2   3.38
  3.   1    3   3.09
  4.   1    4   3.50
  5.   1    5   3.09
  6.   1    6   3.32
  7.   1    7   3.07
  8.   1    8   3.22
  9.   1    9   2.97
 10.   1   10   3.6
 11.   1   11   3.42
 12.   1   12   3.59
 13.   1   13   3.77
 14.   1   14   3.74
 15.   1   15   3.7
 16.   1   16   3.78
 17.   1   17   3.78
 18.   1   18   3.77
 19.   1   19   3.53
 20.   2    1   4.2

. reshape wide prot, i(id) j(week)
   (note: j = 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19)

Data                               long   ->   wide
------------------------------------------------------------------------------
Number of obs.                      513   ->      27
Number of variables                   3   ->      20
j variable (19 values)             week   ->   (dropped)
xij variables:
   prot   ->   prot1 prot2 ... prot19
------------------------------------------------------------------------------
    id  prot1  prot2  prot3  prot4  prot5  prot6  prot7  prot8  prot9  prot10  prot11  prot12  prot13  prot14  prot15  prot16  prot17  prot18  prot19
1.   1   3.69   3.38   3.09   3.32   3.07   3.22   3.09   3.38   3.32   3.42   3.59   3.77   3.74   3.74   3.74   3.74   3.74   3.74
<table>
<thead>
<tr>
<th></th>
<th>prot15</th>
<th>prot16</th>
<th>prot17</th>
<th>prot18</th>
<th>prot19</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7</td>
<td>3.7</td>
<td>3.78</td>
<td>3.78</td>
<td>3.77</td>
<td>3.53</td>
</tr>
</tbody>
</table>
------------------------------------------------------------------------------
    id  prot1  prot2  prot3  prot4  prot5  prot6  prot7  prot8  prot9  prot10  prot11  prot12  prot13  prot14  prot15  prot16  prot17  prot18  prot19
2.   2   4.2   3.35   3.37   3.07   2.82   3.05   3.12   2.82   3.05   3.12   3.2   3.38   3.25   3.26   3.3   3.17   3.4   3.41
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>3.4</td>
<td>3.41</td>
<td>3.28</td>
<td>3.42</td>
<td>3.25</td>
</tr>
</tbody>
</table>
Part B: Longitudinal data analysis in Stata

I. Convert an ordinary dataset into a longitudinal dataset: use *xtset*
   - “xtset” declares ordinary data to be panel data,
     - Cross-sectional data: one panel
     - Longitudinal (cross-sectional time-series) data: multi-panel
       - Each observation in a cross-sectional time-series (xt) dataset is an observation of x for unit i (panel) at time t.
   - For this course, we use cross-sectional time-series data.
   - Syntax for “xtset” for cross-sectional time-series data:

     . xtset panelid timevar

Example:
   . use cd4.dta, clear
   . xtset

panel variable not set, use -xtset varname ...-
r(459);

. xtset id time
   time variable must contain only integer values
   r(451);

. list time in 1/10

|---------+
<table>
<thead>
<tr>
<th>time</th>
</tr>
</thead>
</table>
1. | -741958 |
2. | -246407 |
3. | 243669  |
4. | -2729637|
5. | -2250513|
6. | -221766 |
7. | 221766  |
8. | 774812  |
9. | 1256673 |
10. | -1240246|
+---------+

. gen timedays=round(time*365.25,1)

. list time timedays in 1/10

+---------------------------+
<table>
<thead>
<tr>
<th>time   timedays</th>
</tr>
</thead>
</table>
1. | -741958   -271          |
2. | -246407   -90           |
3. | 243669    89            |
4. | -2729637  -997          |
+---------------------------+
\[
\begin{array}{cc}
5. & -2.250513 \\
6. & -0.221766 \\
7. & 0.221766 \\
8. & 0.774812 \\
9. & 1.256673 \\
10. & -1.240246 \\
\end{array}
\]

\[
\begin{array}{cc}
| & -822 |
\mid----------------------|
| & -81 |
| & 81 |
| & 283 |
| & 459 |
| & -453 |
\end{array}
\]

\[
+----------------------+
\]

\[
\begin{align*}
\text{. xtset id timedays} \\
\text{\hspace{1cm} panel variable: id (unbalanced)} \\
\text{\hspace{1cm} time variable: timedays, -1092 to 1994, but with gaps} \\
\text{\hspace{1cm} delta: 1 unit}
\end{align*}
\]

\[
\begin{align*}
\text{. xtset} \\
\text{\hspace{1cm} panel variable: id (unbalanced)} \\
\text{\hspace{1cm} time variable: timedays, -1092 to 1994, but with gaps} \\
\text{\hspace{1cm} delta: 1 unit}
\end{align*}
\]

- Most built in xt commands require that you first specify \texttt{xtset}
- In older versions of Stata, you would use \texttt{tsset}, which is very similar to \texttt{xtset} (same syntax) or \texttt{iis} and \texttt{tis} to set each part of \texttt{tsset}

II. xt commands
The xt series of commands provide tools for analyzing cross-sectional time-series (panel) datasets:

- \texttt{xtdes}  \hspace{1cm} Describes pattern of xt data

Example: Cows data
\[
\begin{align*}
\text{. use "cows (long).dta", clear} \\
\text{\hspace{1cm} * look at only the cows on Barley diet (coded as 1)} \\
\text{\hspace{1cm} . keep if (diet==1)} \\
\text{\hspace{1cm} (1026 observations deleted)} \\
\text{\hspace{1cm} . drop if (protein==.)} \\
\text{\hspace{1cm} (50 observations deleted)} \\
\text{\hspace{1cm} . xtset id time} \\
\text{\hspace{1cm} * could have also used tsset} \\
\text{\hspace{1cm} . xtdes, patterns(0)} \\
\end{align*}
\]

\[
\begin{align*}
\text{id: } & 1, 2, \ldots, 25 \hspace{1cm} \text{n} = 25 \\
\text{time: } & 1, 2, \ldots, 19 \hspace{1cm} \text{T} = 19 \\
\text{Delta(time) = 1 unit} \\
\text{Span(time) = 19 periods} \\
\text{(id*time uniquely identifies each observation)}
\end{align*}
\]

\[
\begin{array}{cccccccc}
\text{Distribution of T_i:} & \text{min} & \text{5\%} & \text{25\%} & \text{50\%} & \text{75\%} & \text{95\%} & \text{max} \\
\text{12} & \text{14} & \text{15} & \text{18} & \text{19} & \text{19} & \text{19} \\
\end{array}
\]
. xtdes, patterns(5)

id: 1, 2, ..., 25
n = 25
time: 1, 2, ..., 19
T = 19
Delta(time) = 1 unit
Span(time) = 19 periods
(id*time uniquely identifies each observation)

Distribution of T_i: min 5% 25% 50% 75% 95% max
12 14 15 18 19 19 19

<table>
<thead>
<tr>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>44.00</td>
<td>44.00</td>
<td>1111111111111111111</td>
</tr>
<tr>
<td>5</td>
<td>20.00</td>
<td>64.00</td>
<td>1111111111111111111.....</td>
</tr>
<tr>
<td>2</td>
<td>8.00</td>
<td>72.00</td>
<td>1111111111111111111.....</td>
</tr>
<tr>
<td>2</td>
<td>8.00</td>
<td>80.00</td>
<td>1111111111111111111.....</td>
</tr>
<tr>
<td>2</td>
<td>8.00</td>
<td>88.00</td>
<td>1111111111111111111.....</td>
</tr>
<tr>
<td>3</td>
<td>12.00</td>
<td>100.00</td>
<td>(other patterns)</td>
</tr>
</tbody>
</table>

25 100.00 | xxxxxxxxxxxxxxxxxxxxxxx

. xtdes
* default is 9 patterns

id: 1, 2, ..., 25
n = 25
time: 1, 2, ..., 19
T = 19
Delta(time) = 1 unit
Span(time) = 19 periods
(id*time uniquely identifies each observation)

Distribution of T_i: min 5% 25% 50% 75% 95% max
12 14 15 18 19 19 19

<table>
<thead>
<tr>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>44.00</td>
<td>44.00</td>
<td>1111111111111111111</td>
</tr>
<tr>
<td>5</td>
<td>20.00</td>
<td>64.00</td>
<td>1111111111111111111.....</td>
</tr>
<tr>
<td>2</td>
<td>8.00</td>
<td>72.00</td>
<td>1111111111111111111.....</td>
</tr>
<tr>
<td>2</td>
<td>8.00</td>
<td>80.00</td>
<td>1111111111111111111.....</td>
</tr>
<tr>
<td>2</td>
<td>8.00</td>
<td>88.00</td>
<td>1111111111111111111.....</td>
</tr>
<tr>
<td>1</td>
<td>4.00</td>
<td>92.00</td>
<td>1.11111111111111111111</td>
</tr>
<tr>
<td>1</td>
<td>4.00</td>
<td>96.00</td>
<td>1111111.11111111111111</td>
</tr>
<tr>
<td>1</td>
<td>4.00</td>
<td>100.00</td>
<td>11111111.11111111111111</td>
</tr>
</tbody>
</table>

25 100.00 | xxxxxxxxxxxxxxxxxxxxxxx

Other xt commands:
• xtsum Summarize xt data
  We provide an improved version: xtsumcorr with the course .ado files
• xttab Tabulate xt data
• xtreg Fixed-, between- and random-effects, and population-averaged linear models
• xtdata Faster specification searches with xt data
• **xtlogit**  Fixed-effects, random-effects, & population-averaged logit models
• **xtprobit** Random-effects and population-averaged probit models
• **xttobit** Random-effects tobit models
• **xtpois**  Fixed-effects, random-effects, & population-averaged Poisson models
• **xtnbreg** Fixed-effects, random-effects, & population-averaged negative binomial models
• **xtclog**  Random-effects and population-averaged cloglog models
• **xtintreg** Random-effects interval data regression models
• **xtrchh** Hildreth-Houck random coefficients models
• **xtgls**  Panel-data models using GLS
• **xtgee** Population-averaged panel-data models using GEE

Look at “help xt” in Stata

### III. Graphs for longitudinal data

• **xtgraph**
A new command for summary graphs of xt data (cross-sectional time series data). Download the xtgraph.ado file from course website.

**Syntax:**
```
xtgraph varname [if] [in] , group(groupvar) av(avtype) bar(bartype)
graph options xt options
```

**Choice of average**
```
xtgraph , av(avtype)
```
The average types are
- am - arithmetic mean, the default
- gm - geometric mean
- hm - harmonic mean
- median - only with bars ci - default, iqr or rr.

**Choice of error bars**
```
xtgraph , bar(bartype)
level(significance level)
```
The bar types are
- ci - the default, significance set by level()
- se - standard error
- sd - standard deviation
- rr - reference range, level set by level()
- iqr -same as bar(rr) level(50)
- no - no bars

Examples (still using cows (long).dta):

```
xtgraph protein, av(median) bar(iqr) tl("median, iqr")
```

![Median and IQR graph](image)

```
xtgraph protein, av(am) bar(se) tl("arithmetic mean, se")
```

![Arithmetic Mean and SE graph](image)

Refer to xtgraph.pdf or xtgraph.hlp for help.
• How to graph trajectories
An example that we’ve seen in class of drawing trajectories used subjects picked based on ranking of within-subject statistics (the difference in the medians before and after HIV seroconversion).

Other examples:
  • A random set (trajectory1.do)

*trajectory1.do file for Stata 10.0
use cd4.dta, clear
egen newid=group(id)
sum newid
drop id
rename newid id
gen timedays=round(time*365.25,1)
sort id timedays
gen pick = 0
local i=1
while `i' < 8{
  set seed `i'
  local r = round(1+uniform()*369,1)
  gen cd4l`i' = count if (id == `r')
  local i=`i'+1
}
twoway (scatter count timedays) (scatter cd4smth timedays) (line cd4l1-cd4l7 timedays)
- Ranking with the individual mean CD4 counts (trajectory2.do)

*trajectory2.do file for Stata 10.0
use cd4.dta, clear
egen newid=group(id)
sum newid
drop id
rename newid id
gen timedays=round(time*365.25,1)
sort id timedays
egen cd4mean = mean(count), by(id)
list id count cd4mean in 1/10
sort id
quietly by id: replace cd4mean=. if (_n > 1)
egen rnk=rank(cd4mean)
local i = 1
while `i' <= 7{
gen sub`i' = (rnk == `i'*25)
sort id timedays
quietly by id: replace sub`i'=sub`i'[1]
gen cd4l`i' = count if (sub`i')
drop sub`i'
local i=`i'+1
}
ksm count timedays, lowess gen(cd4smth) nograph
twoway (scatter count timedays) (scatter cd4smth timedays) (line cd4l1-cd4l7 timedays)