Dis 1: General Info; Get Started with Stata

1 General Info

1. **Contact Me**

   You can reach me by **sending me an email** or **attending my (virtual) office hours**.

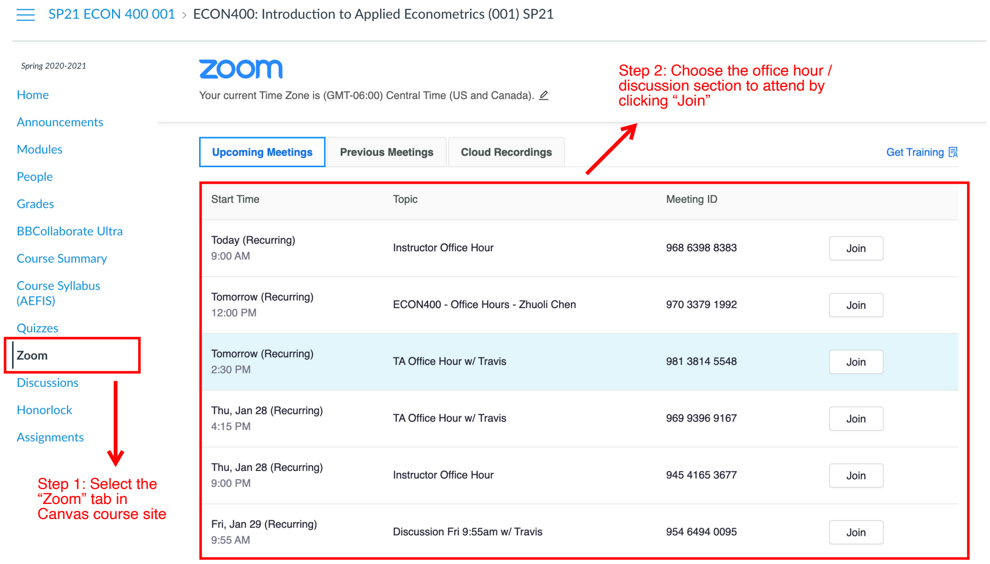
   - Email me at **zchen596@wisc.edu** (please start the subject line with "Econ 400").
   - Virtual office hours take place on Zoom at the following times:
     - Mondays, noon - 1:00pm
     - Wednesdays, noon - 1:00pm
     - Or by appointment

2. **Discussion Sections**

   - Attendance
     - Live attendance is not required, but strongly encouraged.
     - Recordings will be posted (I tape both of my sections, but only the best one will be posted; both sections cover the same material)
     - Sections will take place on Zoom at the following times:
       * Fridays, 1:20 - 2:10pm
       * Fridays, 2:25 - 3:15pm

       Feel free to attend the one that works the best with your schedule.
   - Handouts
     - Handouts (and dataset used) will be available by Thursday midnight on Travis's (the other Econ 400 TA) website.
     - Solutions and section recordings will be posted by Friday night. Same place.

3. **Where to Find These Zoom Sessions?**



Your current Time Zone is (GMT-06:00) Central Time (US and Canada).

Step 1: Select the "Zoom" tab in Canvas course site

Step 2: Choose the office hour / discussion section to attend by clicking "Join"

Start Time	Topic	Meeting ID	
Today (Recurring) 9:00 AM	Instructor Office Hour	968 6398 8383	Join
Tomorrow (Recurring) 12:00 PM	ECON400 - Office Hours - Zhuoli Chen	970 3379 1992	Join
Tomorrow (Recurring) 2:30 PM	TA Office Hour w/ Travis	981 3814 5548	Join
Thu, Jan 28 (Recurring) 4:15 PM	TA Office Hour w/ Travis	969 9396 9167	Join
Thu, Jan 28 (Recurring) 9:00 PM	Instructor Office Hour	945 4165 3677	Join
Fri, Jan 29 (Recurring) 9:55 AM	Discussion Fri 9:55am w/ Travis	954 6494 0095	Join

2 Get Started with Stata

2.1 What is Stata?

Stata is a statistical software that we use in this course to analyze data and perform econometric analysis.

Side note: Stata is a proprietary software, and it has limitations. For the purpose of this class, Stata is more than sufficient. But for more challenging projects that you may encounter in the future, you might want to look into some open-source programming languages that are friendly for data analysis (such as Julia, R, Python; UW stats department has a course on R, and the econ department offers course on machine learning using Python).

2.2 How to Access Stata?

You can access Stata using either one of the two following methods:

- **Installing it onto your personal laptop**:

  Visit Campus Software Library for installation guide and license & activation key. The Stata version licensed by the university is Stata/SE.

- **Logging into Winstat (i.e. a remote server)**:

  To use Winstat, you must install a software package called **Citrix Receiver** on your computer. The following link has more information on Citrix Receiver and Winstat: `https://kb.wisc.edu/sscc/using-winstat`

2.3 Use Help

- **Google** is always a good starting point
- Alternatively, try using the `help` function in Stata. For example, if you want to know how to use the `describe` function, try

```
help describe
```

2.4 Tips to Help You Master Stata

- Start off by working in the command window and typing individual commands to test out whether the command is doing the right thing for you.
- When you settled on the right series of commands, use a do-file to record them, so you can easily re-run this series of commands (a do-file is the same as a script you've encountered in other programming languages / software).
- Make sure to log or to export your result, so that you have something to refer to at a later date (& to turn in for your problem sets).
- Google and the `help` function are always your friend.

3 Problems

1. For our first exercise, let's try loading different types of data into Stata:

   (a) Download this discussion's dataset, and unzip them to some place on your computer.

   (b) Before we can load any data file, we have to first tell Stata where to look for. To do this, we need to change our working directory. Let's go through two different ways on how to do that.

   Method 1: Go to Stata's menu bar, select File → Change working directory..., and locate whatever folder you unzipped the dataset into.

   Method 2: In Stata's command window, use the `cd` command to change the working directory. For example, if your dataset is unzipped into the path /Users/username/Desktop, you will type the following in Stata's command line:

   ```
   cd "/Users/username/Desktop"
   ```

   Notice that there are quotation marks around the path. This is important – the quotation marks tell Stata that the thing surrounded by the quotation marks is a String, so Stata can know where to begin finding this path and where to end. This is also emphasized on the help document of `cd` function.

   (c) Import `gss2014.xls` into Stata (hint: try `import`)

   ```
   import excel "gss2014.xls"
   ```

   (d) Import the first row of data from `gss2014.csv` into Stata

   `*.csv` is a file containing comma-separated-values. Upon inspecting the help document of `import`, the following command should be used:

   ```
   import delimited "gss2014.csv", rowrange(1:2) clear
   ```

   Notice that the `clear` option is used at the end. This is because Stata can only read one set of data at a time. In part (c), we already imported a set of excel data, so for Stata to release its memory and allowing for importing a new set of data, we need to attach the `clear` option at the end of the `import` command to clear out Stata's memory.

   (e) Save the imported data as `gss2014.dta`

   ```
   save "gss2014.dta", replace
   ```

   (f) Load the saved `gss2014.dta` (hint: try `use`)

   ```
   use "gss2014.dta", clear
   ```

   For any file of Stata's data format (`.dta`), `use` (or its variant) is used to load the data.
   For other data formats (such as `.csv`, `.xls`), we first need to `import` it into Stata.

   (g) Stata also has built-in dataset that you can use. Try

   ```
   sysuse "census.dta", clear
   ```

   See `https://www.stata-press.com/data/r16/g.html` for the list of dataset included in Stata.

   (h) Stata can also directly access `.dta` files via a url. Try

   ```
   use "http://www.stata-press.com/data/r13/sp500.dta", clear
   ```

2. Load `panel2007.dta` into Stata. Let's now do some basic data analysis.

   (a) Browse the dataset in Stata using `browse`. Is there anything jumping out to you in this dataset?

   Upon first look, the `id` column looks a bit strange. If you look at properties panel (usually located at the bottom right of your browse window), you'll see that in the "Type" field, `id` has "str2" listed, while `score` says "byte".

   Here, "str2" implies that the `id` column has data of String type. This could be a problem, since `id` clearly is meant to store ID numbers, so we want to convert the String into numerical value somehow.

   (b) Obtain a review of the variables that are included in your dataset using the `describe` command.

   (c) Based on your observation in (a) & (b), convert the `id` column data into the correct data format.

   We want to convert `id` column data from String into numerical values. The function we need is `destring`. Try looking up the help document of `destring` and see what you should do. The command used here should be

```
destring(id), replace
```

   Side note: If you need to convert some variable into String format, try

```
tostring(variable), replace
```

   instead.

   (d) Replace the original `panel2007.dta` file with the current one correcting the `id` variable.

```
save "panel2007.dta", replace
```

   (e) Obtain some summary statistics for our data (such as mean, minimum, maximum of each variable) by using the `summarize` command.

3. Continue from question 2. Now let's try to clean and merge `panel2007.dta` with `panel2008.dta`

   (a) Since the currently loaded data, `panel2007.dta`, are from 2007, let's make this clear by creating a variable called `year`.

   ```
   gen year=2007
   ```

   (b) Again, since Stata can only have one set of data present in its memory, we need to first save the 2007 panel data before merging it with the 2008's.

   ```
   save "panel2007_append", replace
   ```

   (c) Now let's append the 2007 data to the 2008's. As in (a), create a `year` variable for the 2008 data prior to appending. Save the merged data as `appended_data.dta`

`use "panel2008.dta", clear`	(Load `panel2008` data into Stata)
`gen year=2008`	(Generate the `year` variable for the 2008 data)
`append using "panel2007_append.dta"`	(Grab the 2007 data we saved before, and appending it to the 2008 dataset)
`save "appended_data.dta", replace`	(Save the appended data)

   (d) Another way to combine these two datasets is to merge them together (i.e. horizontally instead of vertically appending). To do so, take our 2007 and 2008 datasets and rename the score variables to score2007 and score2008. Then, use the `merge` command to horizontally merge the two datasets by their `id`. Save the merged data as `appended_data.dta`

   ```
   use "panel2007"
   rename score score2007
   save "panel2007_merge", replace

   use "panel2008"
   rename score score2008

   merge 1:1 id using "panel2007_merge"
   save "merge_data.dta", replace
   ```

   For the last step, since each row corresponds to a unique `id` value in both data sets, we use the 1:1 option in `merge`. Other options are available if this isn't true, but let's not worry about that for now. The next term in the `merge` function specifies the variable that we're merging on. The final term says "merge my 2007 data into my 2008 data".

   As requested, `merge` stacked our data horizontally, but it also ensures that we're stacking correctly by matching the 2007 and 2008 scores to the right individual (using the `id` variable).

   (e) Last step: Put everything we did in question 2 and 3 into a do-file, and wrap around your commands in a log.

   We demonstrated in section on how to put what we did in problem 3 into a do-file. To wrap around everything in a log, type

   ```
   log using "logFilename.log", replace
   ```

   at the beginning of your commands, and

```
log close
```

at the end.

(f) * For people interested in making their Stata log file fancy, or if they want to export regression table results into LaTeXor some other format, feel free to check out the following resources:

   i. Stata Markup and Control Language (SMCL)
   ii. estout: Making regression tables in Stata