Advanced Methods in Impact Assessment Workshop

Day 1: The Creation of Knowledge through Impact Assessments

This is the first of our data exercises that will give you a chance to work with real data and apply the techniques you learned during the morning lecture sessions. For most of the days we will be using the Village Dynamics of South Asia (VDSA) panel data set collected by ICRISAT. We will utilize the recent high frequency parcel level production data from households in India. For the data exercises concerning RCTs we will use data from a real RCT on the effects of marketing in encouraging households to purchase index insurance. This RCT was conducted in conjunction with ICRISAT, again in India.

To get started, we will take the "raw" VDSA data and prepare it for analysis. This process will be useful for two reasons. First, it will provide you with a chance to familiarize (or re-familiarize) yourself with **Stata**. Second, since we will be using this data throughout the workshop, these initial exercises will get the data "regression ready" so that we will not need to spend time on Day 3 or Day 4 preparing a data set for regressions.

There are three objectives for today's exercises:

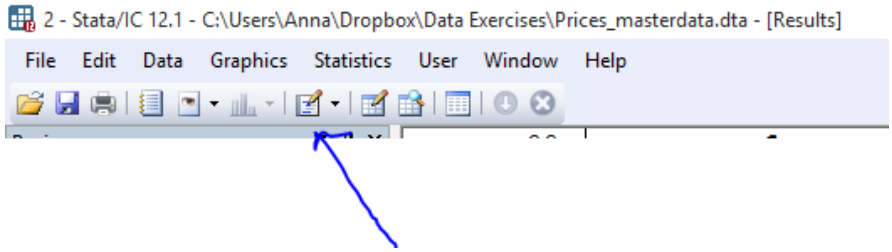
1. Become familiar with **Stata** and the data we will analyze throughout the workshop.
2. Prepare data for analysis on subsequent days.
3. Use real data to illustrate the role of confounding factors in impact assessment.

Introduction to Stata

To get started, load into **Stata** the data set `VDSA_Prod_Data.dta`. These data are parcel level production data. The data set contains parcel level observations on inputs and outputs for wheat. It also contains observations on household level characteristics and a few observations of village level characteristics. As you move through the data set can you identify which observations are at the parcel level, household level, and village level?

Before we begin to modify or analyze the data, open a `.log` file and create a `.do` file. It is good practice to record how you manipulate or clean a data set so that others can replicate your work, or so that you can reconstruct your work in case you find later that you have made a mistake.

First, create a `.do` file. Using a `.do` file will help track what you have done, if you want to return to you work later. These are the **Stata** equivalent of macros. To do create a `.do` file, click on the "notepad" button.



Instead of entering the commands for the exercises in the command window in **Stata**, type them in the `.do` file that we opened. This approach will allow you to return to your work at a later date.

Now, create a `.log` file. Using a log-file will create log of your commands and **Stata**'s outputs which can later be translated to `.pdf`. This can help you track commands and results. To do this, type in the `.do` file `log using "data_exercise_1.smcl", replace`. Remember you will need to close the `.log` file at the end either by typing `log close` at the end of the `.do` file.

When you're done with your `.log` file, you can translate it to a `.pdf` file by typing:

```
translate "data_exercise_1.smcl" "data_exercise_1.pdf", replace
```

Now that you have started a log, let's look at the data set. To see all the variables in a data set, use the `describe` command. This command provides information about the data set, including the name, size, and number of observations. It also lists all variables, including name, storage format, display format, and label. To see a subset of the variables, use the `describe` command, followed by the variable name or names.

The command `summarize` displays a few summary statistics, including means and standard deviations. If no variable is specified, summary statistics are calculated for all variables in the data set.

If you are interested in looking at summary statistics for a subset of observations, not for the entire data set, it is necessary to "sort" the data. Sort the data by the group variable(s) of interest. Then, it is possible to compare your variable of interest by each group. For example, we can look at output and input use by the gender of the household head:

```
sort genderH
by genderH: sum output lab_q fert_q irr_q mech_v pest_v
```

There are three additional ways to consider summary statistics:

- `tabstat` allows you to list the statistics for a variable of interest that you want to consider in a single table. It is also possible to group by another variable.
- `tabulate` allows for consideration of frequency distributions for categorical variables such as gender or farm category

Using the same variables we used to summarize the data, examine the data using `tabstat`. Use `tab` to tabulate the categorical variables `genderH` and `farm_cat`.

There are three ways that can be useful for considering / learning more about your data:

- `count` is used to count the number of observations in a data set. It is also possible to group by another variable.
- `duplicates` highlights potential duplicate observations in the data. You can report, tag, or drop duplicate observations (using commands with the same name).
- `distinct` tells you the number of observations and the total number of distinct observations for a variable of interest.

To use the `distinct` command you will most likely need to install a new package. **Stata** has many additional commands written by third-parties that require installation. To install `distinct` (or any other external package) type `ssc install distinct`. This command will download and install the package.

Now that you have `distinct` installed, let's use the command. Simply type `distinct prcl_id`. How many observations are in the data and how many distinct parcels are there in the data? How many distinct households and villages?

Variables and observations of a data set can be selected using the keep or drop commands. You can also condition these on another variable, for example:

```
keep var1 if var1 > 100
```

Let's begin to manipulate the data. First, tabulate survey year. Notice that survey years 2009 and 2013 have fewer observations than the other years. This is because the VDSA survey only captured half a year of data in 2009 and only recently published the remainder of 2013. So, drop all observations that come from 2009 and 2013.

Stata is able to produce a number of basic graphs. A histogram can be created for a single variable using the command `hist`. **Stata** can also graph multiple variables. As an example, scatterplots can be created using the command: `twoway (scatter var1 var2)`.

1. Check the production relationships in the data by creating scatter plots that show the relationship between `output` and the inputs `plot_area`, `lab_q`, `irr_q`, and `pest_v`. Create scatter plots for each of the four relationships. What do these plots tell you?

One of the commands you will use most frequently in **Stata** is the `gen` command, short for `generate`. It allows you to create a new variable. You can either set the value of the new variable to some number (or text string). Or you can set the value of the variable equal to another, existing variable. Or you can set the value equal to some complicated function included numbers and variables. The format is

```
gen <<newvar>> = <<exp>>
```

Where `<<newvar>>` is your new variable name and `<<exp>>` is whatever you want that variable to equal.

In production economics, we often want to take log transformations of the data. This allows us to estimate Cobb-Douglas (or Translog) production functions while also allowing us to interpret coefficient estimates as elasticities. However, as you may have noticed from the scatter plot of pesticide, there are a lot of zeros in the data. This creates a problem since $\log(0)$ is undefined. One way to deal with the problem of taking the log of a zero is to use the Inverse Hyperbolic Sine Transformation. The equation for this transformation is: $\log(x) = \log(x + \sqrt{(x^2 + 1)})$. This solves the problem of $\log(0)$ without adding an arbitrating number to the value of the variable.

Create variables for output and each input in per hectare terms using the Inverse Hyperbolic Sine Transformation. **Stata** has a built in command to calculate this value: `asinh`. Call the output variable and each input variable `lny`, `lnl`, `lnf`, `lni`, `lnm`, and `lnp`. As an example:

```
gen lny = asinh(output/plot_area)
```

To label data, variables, or values of variables, use the command `label`.

```
label variable var1 "<<label1>>"
```

Label the variables you just created with labels that include the unit of measure, i.e., (kg/ha) in the case of fertilizer.

Finally we can now run regressions in **Stata**. Regressions in **Stata** are in the following form:

```
reg <<independent variable>> <<dependent variable(s)>>
```

Note that **Stata** requires no punctuation in the regression equation.

2. Run a simple regression with log labor, fertilizer, irrigation, mechanization, and pesticide as independent variables and log yield as the dependent variable. How do you interpret the point estimates on each variable?

Before we move on, let's create a binary indicator for irrigation use called `irr` that equals 1 if the parcel under observation had irrigation greater than zero and equals zero if the parcel had no irrigation. This will be our "treatment" variable. Let's also create log transformed variables of `aindex`, `lindex`, `tot_acre`, and `dist` using the Inverse Hyperbolic Sine. Call them `lnaindex`, `lnlindex`, `lntot_acre`, and `lndist`.

Save this data file as `VDSA_Prod_Data_Ref.dta`. This will be the file we return to throughout the week.

The Challenge of Establishing a Causal Effect

To demonstrate the difficulty in establishing causal effects, we are going to look at our data in two different ways. First, we are going to compare the effect of a hypothetical irrigation intervention on those who received the irrigation treatment versus those who did not. This is our Within/Without comparison. Second, we are going to compare the effect of the irrigation treatment on households before they received the treatment and after they received the treatment. This is our Before/After comparison.

Within/Without Comparison

Using the transformed data set that you just saved, create a new data set that contains only data from `sur_yr=2011`. Save this data set using a different file name.

3. Generate a table (using the `summarize` command) that shows the yields for those with irrigation treatment and those without the irrigation treatment. What do you learn from the table about the impact of the irrigation treatment on the log of yields?
4. Do a `ttest` to compare the mean yield by households who received the irrigation treatment with that of the control. What does the test indicate? Is this estimate the intention to treat effect, the effect of treatment on the treated, or the average treatment effect? Explain. The code is:

```
ttest lny, by(irr)
```

5. Run a regression that includes only the irrigation treatment variable and log of crop yield as the outcome. What is the result? What is the marginal effect of having irrigation on crop yield? Is this the impact of the irrigation treatment? Why or why not?
6. What control variables might we want to include in a regression to determine the impact of the irrigation project?

Now add the following control variables: `lnl`, `lnf`, `lnm`, `lnp`, `ageH`, `genderH`, `sizehh`, `lnaindex`, `lnlindex`, `lntot_acre`, and `lndist`. Our dependent variable should be `lny`. Note that we will refer to this set of 11 variables as our "standard set of control variables" in subsequent exercises.

7. After adding the control variables, how do our results change? What does the point estimate tell us? Is the coefficient on `irr` the unbiased effect of the treatment? What else could it be?

Before/After Comparison

Return to the data set you saved before the Within/Without comparison. Using these data, create a new data set that contains only households that received the irrigation treatment. To do this, run the following code on the data set you just loaded.

```
drop if sur_yr==2012
drop if sur_yr==2011 & irr==0
duplicates tag prcl_id, generate(dup)
drop if dup==0
```

Before proceeding, check to make sure observations are balanced by typing `tab irr`.

8. Generate a table (using the `summarize` command) that shows the average yields for households before they received the irrigation treatment and average yields after the irrigation treatment. What do you learn from the table about the impact of the irrigation treatment on yields?
9. Do a `ttest` to compare the mean yield by households before and after the irrigation treatment. What does the test indicate? Is this estimate the intention to treat effect, the effect of treatment on the treated, or the average treatment effect? Explain.
10. Run a regression that includes only the irrigation treatment variable and the log of crop yield as the outcome. What is the result? What is the marginal effect of having irrigation on crop yield? Is this the impact of the irrigation treatment?
11. Now add the standard set of control variables. How do our results change? What does the point estimate tell us? Is the coefficient on `irr` the unbiased effect of the treatment? What else could it be?