IMPACT EVALUATION USING STATA ▶

Habiba Djebbari and María Adelaida Lopera

This material is a complement to the book Impact Evaluation in Practice. It consists of a series of examples illustrating the basic analysis required in a rigorous program evaluation report. Nonetheless, each evaluation has particularities that need to be addressed and researchers are expected to further explore the caveats of their own program. The required background is a basic knowledge of the software Stata and some familiarity with statistical terminology.

The document is organized as follows. Chapter 1 is a quick introduction to Stata and its programming language. Chapter 2 illustrates the randomization process and how to compute basic power calculations. Chapter 3 shows how to estimate simple program effects. This is an interactive document. The data sets and Stata exercises are available by clicking on the links, complementary videos are also accessible by clicking on the icon ▶. Your comments or suggestions can be addressed to maria.lopera.1@ulaval.ca.

Contents

Chapter 1

Introduction to Stata

1.1 Getting Started

This chapter is a review of the Stata software. This is not an extensive manual but an overview of some of the elements required in a program evaluation.

1.1.1 Interface ▶

The Stata interface is composed of four windows. There is a menu and a toolbar at the top. Use either of these to open the datasets and the command files. You can use the *Command* window to enter commands, although most of the time we will directly execute them from the do-file. To execute a single command from a do-file, highlight it and click on the *Do* icon.

1.1.2 File Structure ▶

There are four types of Stata files. Those identified by the suffix `*.dta` contain data. The `*.do` files contain Stata commands. You can replicate the examples in each section by executing the corresponding do-file. The `*.log` files record the output. Finally, the `*.ado` files are fancy do-files that you do not want to deal with for now.

1.2 Introduction to Programming

Programming has quite a high fixed cost. However, if you are planning to use Stata often, your investment will totally pay off. Your do-files will be concise, comprehensible, replicable and easily modifiable.

1.2.1 Macros ▶

When programming you manipulate a special type of object named *macro*. It holds information in memory aside from the dataset; the concept is similar to a variable in other programming languages. There are two type of macros: `local` and `global`. Our first local macro is named *world* and contains the word *mundo*. Use the symbols ` ' to recall a local macro and the command `display` to printout its content.

```
local world = "mundo"
display "hola `world'"

(output)
hola mundo
```

Macros can also contain numbers or a list of elements:

```
local numerator   = 1+sqrt(5)
local denominator = 2
display "golden number = " `numerator'/`denominator'

(output)
golden number = 1.618034
```

```
local list1 = "0 1 1 2 3 5 8 13 21 34 55 89 144 233 377"
local list2 = "El ingenioso hidalgo don Qvixote de la Mancha, 1605"
display "`list1'"
display "`list2'"

(output)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
El ingenioso hidalgo don Qvixote de la Mancha, 1605
```

Global macros are not very different from local ones. All you need to know at this stage is that global macros stay in memory for longer. To recall a global macro, use the dollar sign (`$`).

```
global golden = (1+sqrt(5)) / 2
display $golden
```

```
(output)
1.618034
```

1.2.2 Loops ▶

A loop executes the commands enclosed in its braces many times. There are three main types of loops: `forvalues`, which iterates over a series of numbers, `foreach`, which iterates over the elements of a list, and `while`, which iterates until a condition is evaluated as false. These three loops are useful for all sorts of repetitive tasks. Our first loop sets the local macro *n* to values from 1 to 10. Each loop sets to the power of *n* the *golden number* defined above.

```
forvalues n = 1 (1) 10 {
        display $golden^`n'
        }
```

```
(output)
1.618034
2.618034
4.236068
6.854102
11.09017
17.944272
29.034442
46.978714
76.013156
122.99187
```

The second loop iteratively sets the local macro *word* to each element of the *list2* defined above. The `_continue` command suppresses the automatic newline at the end of each display.

```
foreach word in `list2' {
        display "`word' " _continue
        }
```

```
(output)
El ingenioso hidalgo don Qvixote de la Mancha, 1605
```

For the third loop, we set the starting values of two counters (`i=0`, `j=1`) and define a stopping condition (`i<1500`). The loop displays the counters' values at each round until the condition is evaluated as false.

```
local i = 0
local j = 1
while `i' < 1500 {
        display `i' " " _continue
        display `j' " " _continue
        local i = `i' + `j'
        local j = `i' + `j'
        }
```

```
(output)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

1.2.3 Programs ▶

Programming is an advanced topic. After reading this section, you will be able to read and modify basic Stata programs according to your particular needs.

Our first program is called *greetings*. It gives a particular value to the local macro *hello* and displays it. Before creating the program we have to erase any other program with the same name. Any program code has to be enclosed between two commands: `program (...) end`. To run it, we simply write the programs' name.

```
capture program drop greetings
program greetings
        local hello = "hola"
        display "`hello'"
end
greetings
```

```
(output)
hola
```

Whatever follows the program's name will be taken by Stata as its arguments. In the example below, the program *ratio* has two arguments. Stata will automatically name all the arguments with positional macros, in this case: `` `1' `` and `` `2' ``. When we execute the program *ratio* followed by the numbers 6 and 3, the program divides the first argument by the second.

```
capture program drop ratio
program ratio
        display "ratio = " `1'/`2'
end
ratio 6 3
```

```
(output)
ratio = 2
```

The `rclass` option saves in `r()` any value preceded by the command `return`. The output `r()` can be used as an argument in other commands.

```
local numerator   = 1+sqrt(5)
local denominator = 2

capture program drop divide
program divide, rclass
        return local ratio = `1'/`2'
end

divide `numerator'  `denominator'
display "golden number = " r(ratio)
```

```
(output)
golden number = 1.618033988749895
```

1.2.4 Help ▶

The *Help* tool contains precious information on commands and is often recommended as great leisure reading. One of Stata's strengths is the consistency of its command syntax. Most commands share the following structure:

```
[prefix:]  command [varlist][if][in][weight][, options]
```

The square brackets `[ ]` indicate a non-mandatory argument. The suffix `varlist` is a list of variables; when not specified, all the variables are used. The suffixes `if` and `in` restrict the set of observations used by the command. Most commands accept prefixes that modify their task, one of the most common being `by`. The `options` are always specified after a semicolon and modify what the command does.

1.3 Working with Data ▶

The examples in this document are based on a subsample from the Bangladesh Household Survey 1991/92-1998/99 (Khandker et al., 2010). This sections uses the file `hh_91_practice.dta`.

Start by resetting the memory with the `clear all` command. Then, specify the current directory path on the computer using the `cd` command. Obviously, you need to indicate your own computer path.

```
clear all
cd "C:\Users\your path...\Stata Practice\data\"
```

1.3.1 Summary Statistics and Tables

The command `use` opens the dataset.

```
use "hh_91_practice.dta"
```

Explore the data using the commands `describe`, `summarize`, `tabulate`, and `codebook`. Try to add some options to the commands using the Stata help.

```
describe
summarize
sum exptot
sum exptot, detail
```

```
(output omitted)
```

The variable *exptot* measures total household expenditures and the variable *sexhead* identifies the gender of the household heads. This last one takes a value of 1 if the head of the household is a man and 0 if it is a woman.

```
summarize sexhead
tabulate  sexhead
```

```
(output)
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     sexhead |       583    .9519726    .2140079          0          1

  Gender of |
   HH head: |
   1=M, 0=F |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         28        4.80        4.80
          1 |        555       95.20      100.00
------------+-----------------------------------
      Total |        583      100.00
```

The following command uses the `bysort` prefix to compare statistics between the male and the female household heads.

```
bysort sexhead:sum exptot
```

```
    (output)
-> sexhead = 0

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      exptot |        28    3964.517    2206.474   1371.211   11970.69

-> sexhead = 1

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      exptot |       555    3786.956    1614.144    1352.46   18859.47
```

1.3.2 Graphics ▶

Figure 1.1 shows the distribution of the total household expenditures *exptot*. Panel (a) uses the `kdensity` command to draw a non-parametric distribution of the variable (*exptot*). The `title()` option identifies the graphic.

```
kdensity exptot, title((a) distribution of household expenditures)
```

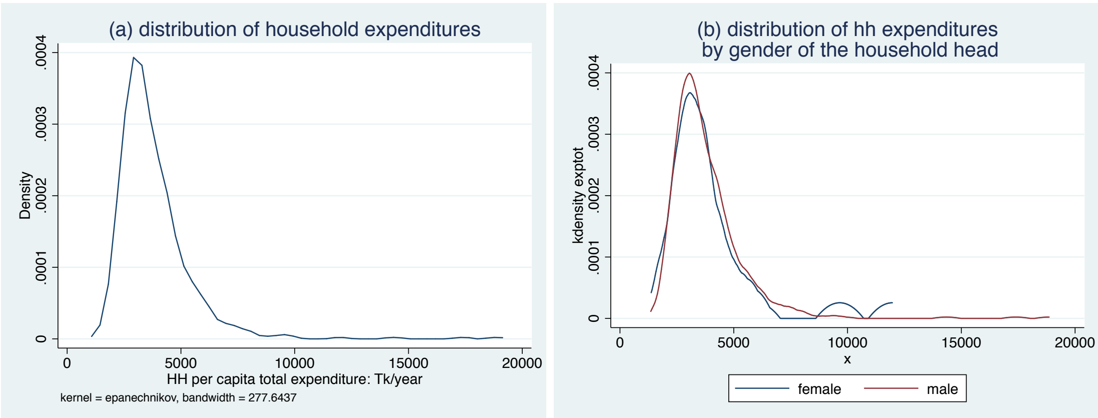

Figure 1.1: Distributional graphics of *exptot*

In panel (b), we use the `twoway` command to overlap the graphic of two groups: male household heads and female household heads. We draw the distribution of the household expenditures for each group conditioning with the `if` suffix. We label the two distributions using the `legend` option. To break up the lines of a single command we use `///`.

```
twoway  (kdensity exptot if sexhead==0)                    ///
        (kdensity exptot if sexhead==1),                   ///
        title("(b) distribution of hh expenditures"       ///
              " by gender of the household head")          ///
        legend(label(1 "female") label(2 "male"))
```

Quite often, for statistical reasons and interpretation purposes, economists scale and transform the variables of interest. We create a new variable with the natural logarithm of the total household expenditures.

```
generate lnexptot = ln(exptot)
```

As an exercise, draw the distribution of the transformed variable *lnexptot*.

Chapter 2

Random Assignment to the Treatment ▶

Suppose that we are asked to evaluate an upcoming microcredit program for Bangladesh. Ideally, we would like to compare two identical groups, with the only difference that one group would participate in the program (treatment group) and the other would not (control group). Since the program has not yet been implemented, we can use random assignment to create the two groups. When judiciously implemented, this experimental approach guarantees that any significant difference between the future outcomes of the two groups is caused by the program. This chapter explains how to create those two groups before the program starts and includes some basic power calculations. To illustrate the randomization procedure we use the baseline dataset `hh_91_practice.dta`. This is a fictitious baseline survey with information about the target population before the microcredit program is implemented.

2.1 Experimental Sample ▶

Our baseline data contains information on 583 households from 87 villages. For the purpose of our evaluation, we want to select an experimental sample of 300 households, so that it is representative of the surveyed population. A simple way to select a representative experimental sample is to implement a virtual lottery. We distribute at random one "lottery ticket" to each household in the survey and select those with the lowest 300 numbers. The question of how many households to select will be addressed later.

```
use "data/hh_91_practice.dta"
describe
```

```
(output)
Contains data from data/hh_91_practice.dta
  obs:           583
 vars:            22                          21 Apr 2014 17:40
 size:        55,968
-----------------------------------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-----------------------------------------------------------------------------------------------------------
nh              float  %9.0g                  HH ID
year            float  %9.0g                  Year of baseline observation
villid          double %1.0f                  Village ID
thanaid         double %2.0f                  Thana ID
agehead         float  %3.0f                  Age of HH head: years
sexhead         float  %2.0f                  Gender of HH head: 1=M, 0=F
educhead        float  %2.0f                  Education of HH head: years
famsize         float  %9.0g                  HH size
hhland          float  %9.0g                  HH land asset: decimals
hhasset         float  %9.0g                  HH total asset: Tk.
expfd           float  %9.0g                  HH per capita food expenditure: Tk/year
expnfd          float  %9.0g                  HH per capita nonfood expenditure: Tk/year
exptot          float  %9.0g                  HH per capita total expenditure: Tk/year
vaccess         float  %9.0g                  Village is accessible by road all year: 1=Y, 0=N
pcirr           float  %9.0g                  Proportion of village land irrigated
rice            float  %9.3f                  Village price of rice: Tk./kg
wheat           float  %9.3f                  Village price of wheat: Tk./kg
milk            float  %9.3f                  Village price of milk: Tk./liter
potato          float  %9.3f                  Village price of potato: Tk./kg
egg             float  %9.3f                  Village price of egg: Tk./4 counts
oil             float  %9.3f                  Village price of edible oil: Tk./kg
vill            float  %9.0g                  village identification number
-----------------------------------------------------------------------------------------------------------
Sorted by:  nh  vill  thanaid
```

The code below creates the variable *random* which represents the lottery tickets. To draw the actual numbers we use the `runiform()` command. It assigns a number to each household from the uniform distribution in the interval $[0, 1]$. Then, we `sort` all the households in increasing order with respect to their lottery ticket and select the first 300. In order to identify the experimental sample, we create the dummy variable *experiment*. It takes a value of 1 for households participating in the evaluation and 0 for the rest.

```
generate random = runiform()
sort random
generate experiment = (_n <= 300)
tabulate experiment
```

```
(output)
 experiment |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        283       48.54       48.54
          1 |        300       51.46      100.00
------------+-----------------------------------
      Total |        583      100.00
```

The selected households are considered a random subsample and thus should be representative of the survey sample.

2.1.1 Replicability of Random Draws ▶

Once we have randomly selected the households that will participate in our evaluation, the experimental sample should not change. To make sure that the same households are selected every time we execute our code, we shall use the `set seed` command before we draw the lottery tickets. Intuitively, this command anchors the random process to a particular algorithm to create those random numbers. It allows us to obtain the same results every time we run the program. In practice, the value of the seed does not matter as long as there is no obvious pattern.

```
set seed 19320419
```

2.1.2 External Validity ▶

External validity means that the experimental sample is representative of the target population. When there is external validity, the conclusions from the experimental sample can be extrapolated to the target population from which this sample was drawn. When the experimental sample is large enough, the average of its variables tends toward the population mean (law of large numbers).[1] The code below explores the representativeness of an experimental sample of 20 households compared to the larger experimental sample of 300 households.

```
generate experiment_20 = (_n <= 20)
twoway  (kdensity exptot)                                              ///
        (kdensity exptot if experiment    ==1, lpattern(dash))         ///
        (kdensity exptot if experiment_20 ==1, lpattern(shortdash)),   ///
legend(label(1 "survey") label(2 "large sample") label(3 "small sample"))
```

The variable *experiment_20* selects an experimental sample of 20 households instead of 300. We plot 3 densities of the *exptot* variable: one with all of the baseline data, one with the large experimental sample of 300 households,

[1] See Impact Evaluation in Practice Ch.11 for discussion of the sampling strategy.

and one with the small experimental sample of 20 households. The `lpattern()` option allows us to draw different line patterns.

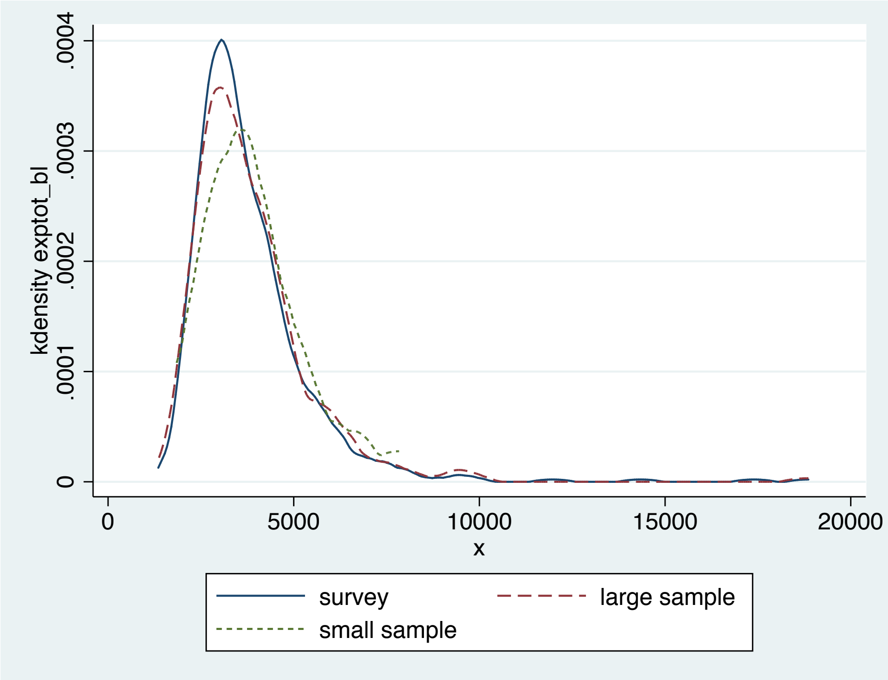

Figure 2.1: Sample size and representativeness

From Figure 2.1, we conclude that larger experimental samples are closer (more similar) to the original survey data.

2.2 Treatment and Control Group ▶

The second step consists in separating the experimental sample into two identical groups: treatment and control. Again, the variable *random_T* simulates the draw of a lottery number for each household in the experimental sample. Households with numbers below 0.5 are assigned to the treatment group (*T=1*), and the rest to the control group (*T=0*). The *missing* option shows missing values for the assignment of the non-experimental households.

```
generate random_T = runiform()
generate T = (random_T < 0.5) if experiment == 1
tabulate T, missing
```

```
(output)
          T |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        157       26.93       26.93
          1 |        143       24.53       51.46
          . |        283       48.54      100.00
------------+-----------------------------------
      Total |        583      100.00
```

2.2.1 Stratification ▶

Suppose that we want to make sure that we have the same number of female household heads in the control and the treatment groups. We do this by running a separate lottery for the households with a female household head. For each value of *sexhead* we run a lottery to assign half of the households to the treatment group and half of the households to the control group. This procedure requires the `bysort` prefix.

```
sort sexhead random
bysort sexhead: generate random_T_strat = runiform()
bysort sexhead: generate T_strat = (random_T_strat < 0.5) if experiment == 1
tabulate T_strat sexhead
```

```
(output)
           |       1=M, 0=F
   T_strat |         0          1 |     Total
-----------+----------------------+----------
         0 |         7        141 |       148
         1 |         7        145 |       152
-----------+----------------------+----------
     Total |        14        286 |       300
```

2.2.2 Level of Randomization ▶

The level of randomization is mainly guided by the nature of the intervention.[2] In our microcredit example, the random assignment can be done among households, villages or thanas (sub-districts). To avoid contamination, Hawthorne effects or John Henry effects, it could be useful to randomize at the village level. For this, we need a unique village identifier (*vill*). We create the variable *random_village*, a random number (a lottery ticket) linked to each village with households in the experimental sample. The households in villages with a random number below 0.5 will be part of the treatment group (*T_village*=1), and

[2] See Impact Evaluation in Practice Ch.4 for discussion of the adequate level of randomization.

the other households in the experimental sample will be part of the control group (*T_village* = 0).

```
sort vill random_T
bys vill: egen random_vill = mean(random)
gen T_vill = (random_vill < 0.5)
```

The table of the assignment variable *T_village* shows the result of the random assignment. There are 45 villages in the treatment group and 42 in the control group.

```
egen tag =  tag(vill)
tabulate T_vill if tag == 1

(output)
     T_vill |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         42       48.28       48.28
          1 |         45       51.72      100.00
------------+-----------------------------------
      Total |         87      100.00
```

2.2.3 Internal Validity ▶

Internal validity means that the control group provides a valid counterfactual for the treatment group. When an assignment process is random, we obtain two groups that have a high probability of being statistically identical, so long as the size of the experimental group is sufficiently large. We can test the similarity between two groups simply by comparing the variables' means prior to the program.

2.2.4 Validation of the Research Design ▶

A test of equality of means gives us the probability that the observed differences in means between the treatment and the control groups prior to the program are due to random chance and not to systematic differences.

We use the `ttest` command to test the equality of gender proportions between the treatment and the control group. Around 96% of the household heads are males in the treatment group. In the control group, this proportion is close to 95%. The $t-$test indicates a 71% probability that this difference is due to random selection of the treatment and the control group. Thus, we do not reject the null hypothesis that the two groups have the same mean.

```
ttest sexhead, by(T)
```

```
(output)
Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |     157    .9490446    .0176066    .2206104    .9142664    .9838227
       1 |     143     .958042     .016825     .201198    .9247821    .9913019
---------+--------------------------------------------------------------------
combined |     300    .9533333     .012198    .2112763    .9293284    .9773382
---------+--------------------------------------------------------------------
    diff |           -.0089974    .0244581               -.0571298    .0391351
------------------------------------------------------------------------------
    diff = mean(0) - mean(1)                                      t =  -0.3679
Ho: diff = 0                                     degrees of freedom =      298

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.3566         Pr(|T| > |t|) = 0.7132          Pr(T > t) = 0.6434
```

2.2.5 Clustering ▶

When the randomization is done at an aggregate level, the error terms are not independent. Individuals in the same group may be subject to common shocks. In our case, households from the same village may be have similar unobserved characteristics. We can measure the intra class correlation within villages.

```
loneway exptot vill
global rho = r(rho)
```

```
(output)
   One-way Analysis of Variance for exptot: HH per capita total expenditure:

                                          Number of obs =       583
                                              R-squared =    0.2441

    Source                    SS         df      MS            F     Prob > F
-------------------------------------------------------------------------
Between vill               3.846e+08     86    4471667.9      1.86     0.0000
Within vill                1.191e+09    496    2401516.9
-------------------------------------------------------------------------
Total                      1.576e+09    582    2707415.5
```

```
Intraclass       Asy.
correlation      S.E.       [95% Conf. Interval]
------------------------------------------------
   0.11412     0.03647       0.04264     0.18560

Estimated SD of vill effect             556.2043
Estimated SD within vill                1549.683
Est. reliability of a vill mean          0.46295
     (evaluated at n=6.69)
```

Here, the intraclass correlation is 11%, which can be considered small. In general, to account for the correlation of households within villages, we use a technique called clustering. Most of the time, this is easily done by specifying the `cluster()` option after a Stata command. For example, when validating the research design, the test of equality of means can also be implemented running a regression. Adding the `cluster()` option adjusts the standard errors when there is a potential correlation. Again, if the assignment is truly random, we should not reject the null hypothesis of equality of means. It is possible to regress the treatment variable on a set of pre-treatment variables that we want to test. Including many regressors allows us to perform several tests at a time. As a rule of thumb, the randomization can be considered successful if we do not reject the null hypothesis for 90% of the baseline regressors tested.

```
reg T sexhead agehead educhead famsize hhland hhasset, cluster(vill)

(output)
Linear regression                                      Number of obs =      300
                                                       F(  6,     83) =     2.89
                                                       Prob > F      =  0.0133
                                                       R-squared     =  0.0303
                                                       Root MSE      =  .49767

                                  (Std. Err. adjusted for 84 clusters in vill)
------------------------------------------------------------------------------
             |               Robust
           T |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     sexhead |   .0364026   .1234966     0.29   0.769    -.2092272    .2820323
     agehead |   .0036785    .003167     1.16   0.249    -.0026205    .0099776
    educhead |    .017239   .0106379     1.62   0.109    -.0039194    .0383974
     famsize |   .0044245   .0146219     0.30   0.763    -.0246579    .0335069
      hhland |   .0000311   .0026367     0.01   0.991    -.0052132    .0052753
     hhasset |   5.81e-07   2.55e-07     2.28   0.025     7.38e-08    1.09e-06
       _cons |   .2218458    .172411     1.29   0.202    -.1210727    .5647643
------------------------------------------------------------------------------
```

2.3 Power Calculations ▶

Power calculations are a major component of a program evaluation and should be computed regardless of the evaluation technique, experimental or not. Power calculations indicate the sample size required to detect a given program impact. Calculations can be done using a variety of (free) software such as Optimal Design. We use Stata to illustrate the basic procedure. Let's take a minute to discuss the intuition of power calculations. Figure 2.2 presents fictitious follow-up data from four different experimental evaluations. Each panel shows the outcome variable after the program, and the treatment and control groups are shown separately. From panel (a) we can conclude that the mean outcome is different in the two groups. The conclusions are not so evident in panel (b), because the data are quite "spread out" (large variance). In panel (c), the variance is also large, but the sample size of the experimental group is much larger as well. As a result, the program effect becomes easier to detect. Although the data in panel (d) also has a large variance and there are few observations (small sample size), we can easily conclude that the mean outcome is different. The reason is that the program effect is much larger.

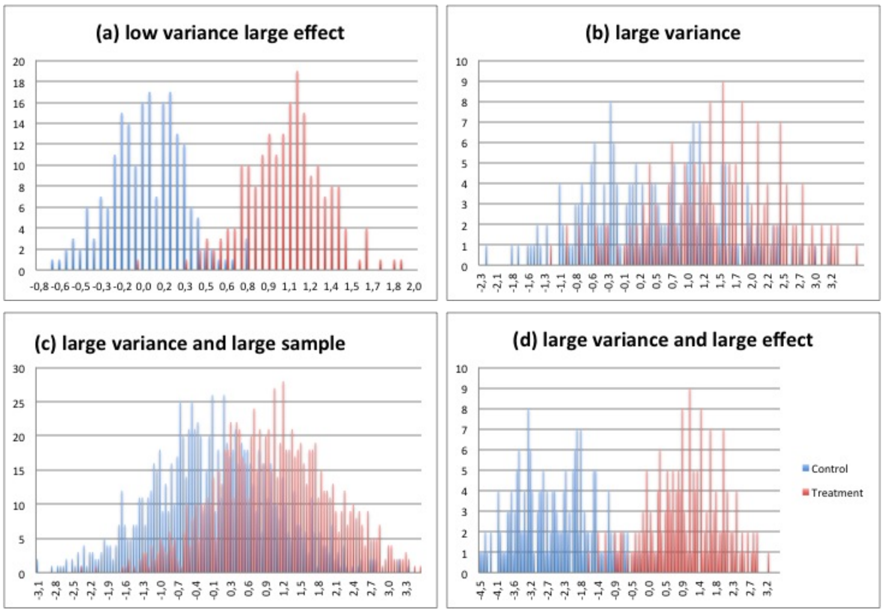

Figure 2.2: Outcome variable by treatment status

There are two important messages embedded in this figure: first, depending on the characteristics of the data, a large sample size could be necessary to capture the program impact. Second, large program effects can be detected with small experimental samples.

2.3.1 Required Sample Size ▶

We use the baseline data (before the program is implemented) to run power calculations. Suppose that your theory of change predicts that the microcredit program will increase household consumption (*exptot*). The first step to calculate the required sample size is to propose expected outcome values for the counterfactual. We create the local macros `mean_0` and `sd_0`, which contain our expectations about the mean and the standard deviation of the outcome variable in the absence of the program. In practice, we approximate those values using the pre-intervention average of the outcome variable (*lnexptot*) in the baseline dataset.

```
summarize exptot
local mean_0 = r(mean)
local sd_0   = r(sd)
```

Often, the baseline data is not available prior to the study and you need to determine sample size before you collect your own baseline data. For this, you can rely on other sources of data from the population that you are interested in. These sources must contain the information that you need to estimate the mean and the standard deviation of the outcome variable in the absence of the program.

The second step to calculate the required sample size consists in proposing the expected outcome values of treatment group in the future. We set the macro `mean_1` to our expected outcome in the treatment group. To determine this value, you need to dig into the literature to find similar program effects that have been estimated before. You should be as conservative as possible when setting this value; it should correspond to the minimum program effect that you are willing to detect. In our case, we assume that on average the microcredit program increases the total household expenditures by 600 taka.

```
local mean_1 = `mean_0' + 600
sampsi `mean_0' `mean_1', sd(`sd_0') power(0.8)
```

```
(output)
Estimated sample size for two-sample comparison of means

Test Ho: m1 = m2, where m1 is the mean in population 1
                    and m2 is the mean in population 2
Assumptions:

         alpha =   0.0500  (two-sided)
         power =   0.8000
            m1 =  3795.48
            m2 =  4395.48
           sd1 =  1645.42
           sd2 =  1645.42
         n2/n1 =     1.00

Estimated required sample sizes:

            n1 =      119
            n2 =      119
```

Under the assumptions stated above, we require 120 households in the treatment group and 120 households in the control group to detect any program effect larger than 600 taka.

2.3.2 Clustering and Required Sample Size ▶

When calculating the required sample size, we also need to account for potential correlations between participants. In our example, this corresponds to the correlation of households within a village. The higher the correlation, the larger the

```
findit sampclus
sampclus, numclus(87) rho($rho)
```

```
(output)
Sample Size Adjusted for Cluster Design

   n1 (uncorrected) = 119
   n2 (uncorrected) = 119

   Intraclass correlation     = .1141191

   Average obs. per cluster   = 4
   Minimum number of clusters = 87
```

```
Estimated sample size per group:

   n1 (corrected) = 160
   n2 (corrected) = 160
```

2.3.3 Power of the Evaluation ▶

Once we have randomized our experimental sample, we should always report the power calculations of the impact evaluation. It is suitable for an experimental design to have a high probability of concluding that a program has an impact when it has one. This is called the *power* of the evaluation. The standard benchmark is 80 percent, which means that when there is an impact, we will capture it 80 percent of the time. It would be convenient to have higher power but there is a trade-off, because power comes at the cost of increasing the size of the sample. In order to calculate the power, we create the macros `n_0` and `n_1` equal to the number of observations in the control and treatment groups.

```
count if T == 0
local n_0 = r(N)
count if T == 1
local n_1 = r(N)
```

The mandatory arguments of the Stata command to calculate power are: the expected mean outcome in the control group (`mean_0`), the expected mean outcome in the treatment group (`mean_1`), the standard deviation in the control group (`sd_0`), the number of households in the control group (`n_0`) and the number of households that benefit from the program (`n_1`).

```
sampsi 'mean_0' 'mean_1', sd1('sd_0') n1('n_0') n2('n_1')

(output)
Estimated power for two-sample comparison of means

Test Ho: m1 = m2, where m1 is the mean in population 1
                    and m2 is the mean in population 2
Assumptions:

         alpha =   0.0500  (two-sided)
            m1 =  3795.48
```

```
             m2 =  4395.48
            sd1 =  1645.42
            sd2 =  1645.42
 sample size n1 =      157
             n2 =      143
          n2/n1 =     0.91

Estimated power:

          power =   0.8839
```

Under the stated assumptions, if we randomize at the household level assigning 143 households to the treatment group and 157 households to the control group, and if the program increases the total expenditures by 600 taka, our evaluation has a power of 88%.

Chapter 3

Program Impact Estimation ▶

This chapter demonstrates how to estimate a program impact using follow-up data. You can download the dataset hh_follow-up.dta, which contains information about a fictitious randomized evaluation of a microcredit program. The variables are similar to the variables in the baseline dataset from the previous chapter.

```
use "data/hh_follow-up.dta"
describe
```

```
(output)
Contains data from data/hh_follow-up.dta
  obs:           300
 vars:            26                          21 Apr 2014 17:40
 size:        33,000
-------------------------------------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------------------------------------
T               float  %9.0g                  T=0: control; T=1: treatment
nh              double %7.0f                  HH ID
year            float  %9.0g                  Year of observation
villid          double %9.0g                  Village ID
thanaid         double %9.0g                  Thana ID
agehead         float  %3.0f                  Age of HH head: years
sexhead         float  %2.0f                  Gender of HH head: 1=M, 0=F
educhead        float  %2.0f                  Education of HH head: years
famsize         float  %9.2f                  HH size
hhland          float  %9.0g                  HH land: decimals
hhasset         float  %9.0g                  HH total asset: Tk.
expfd           float  %9.0g                  HH per capita food expenditure: Tk/year
expnfd          float  %9.0g                  HH per capita nonfood expenditure: Tk/year
exptot          float  %9.0g                  HH per capita total expenditure: Tk/year
```

```
(output break)
```

The dataset contains 300 observation and has the same variables as the Bangladeshi household survey. Additionally, the treatment variable *T* takes a value of 1 if a household benefited from the microcredit program (treatment group) and 0 otherwise (control group). There are 150 treated households and 150 in the control group.

```
tabulate T

(output)
        T=0: |
    control; |
        T=1: |
   treatment |      Freq.     Percent        Cum.
-------------+-----------------------------------
           0 |        150       50.00       50.00
           1 |        150       50.00      100.00
-------------+-----------------------------------
       Total |        300      100.00
```

3.1 Outcome variable ▶

We expect the microcredit program to increase household consumption in the short run. Some of the resources offered by microcredit could have been invested into physical or human capital. Our selected outcome variable is the natural logarithm of the total expenditures. This transformation facilitates the interpretation of the estimates.

```
gen lnexptot = ln(exptot)
```

3.2 Average Treatment Effect ▶ ▶ ▶

In our sample, the assignment to treatment and control is perfectly random. This means that the probability of participating in the program for any household in the population of interest is independent of the potential gain from the program. In this case, the estimation of the *Average Treatment Effect* (ATE) among the potential beneficiaries is a simple difference of means between the treatment group and the control group. As discussed before, we can test the difference of means between two groups using a linear regression and clustering by village.

```
ttest lnexptot, by(T)
```

```
   (output)
Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |     150    8.271546    .0403408    .4940721    8.191832     8.35126
       1 |     150    8.439759    .0391743    .4797853     8.36235    8.517168
---------+--------------------------------------------------------------------
combined |     300    8.355652    .0284871    .4934116    8.299592    8.411713
---------+--------------------------------------------------------------------
    diff |           -.1682131    .0562317               -.2788748   -.0575515
------------------------------------------------------------------------------
    diff = mean(0) - mean(1)                                      t =  -2.9914
Ho: diff = 0                                     degrees of freedom =      298

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.0015         Pr(|T| > |t|) = 0.0030          Pr(T > t) = 0.9985
```

The estimation results suggest that, on average, households who benefit from microcredit increase their expenditures by 17%. Moreover, this increase is statistically significant.

We can also run the regression clustering by village.

```
regress lnexptot T, cluster(vill)

(output)
Linear regression                                      Number of obs =     300
                                                       F(  1,    83) =    7.11
                                                       Prob > F      =  0.0092
                                                       R-squared     =  0.0292
                                                       Root MSE      =  .48698

                                  (Std. Err. adjusted for 84 clusters in vill)
------------------------------------------------------------------------------
             |               Robust
    lnexptot |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           T |   .1682131   .0630851     2.67   0.009     .0427395    .2936868
       _cons |   8.271546   .0476477   173.60   0.000     8.176776    8.366315
------------------------------------------------------------------------------
```

On average, households that benefit from microcredit increase their expenditures by 1016 taka compared to non-beneficiaries. In order to improve the precision of our estimates, we can add other exogenous variables to the regression. When those variables are correlated to the outcome variable (*lnexptot* or *exptot*) but uncorrelated to the treatment (*T*), the confidence interval of the estimated program impact becomes smaller. However, if the assignment is truly random, the program impact estimate itself should remain unchanged.

3.2.1 Heterogenous Impact

It is possible that the program impact depend on the characteristics of the microcredit beneficiaries. The evidence of heterogeneous effect could help to unravel the channels through which the impact is generated.

```
reg lnexptot T sexhead educhead famsize, cluster(vill)

(output)
Linear regression                                      Number of obs =     300
                                                       F(  4,    83) =   11.65
                                                       Prob > F      =  0.0000
                                                       R-squared     =  0.1812
                                                       Root MSE      =   .4495

                                  (Std. Err. adjusted for 84 clusters in vill)
------------------------------------------------------------------------------
             |               Robust
    lnexptot |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           T |   .2200043   .0599744     3.67   0.000     .1007177     .339291
     sexhead |  -.0472284    .100097    -0.47   0.638    -.2463173    .1518605
    educhead |   .0613267   .0099378     6.17   0.000     .0415609    .0810925
     famsize |  -.0344101   .0124714    -2.76   0.007    -.0592152   -.009605
       _cons |   8.366611   .1049292    79.74   0.000     8.157911     8.57531
------------------------------------------------------------------------------
```

Households where the household head is more educated benefit more from the microcredit program. This is shown by a positive and significant estimate of the variable *educhead*. One possibility is that educated beneficiaries invest resources from the credit into more highly income-generating activities. On average, each additional year of education of the household head is associated with a 6% increase of the program impact. Larger families appear to benefit less from the program. Each additional family member reduces the program impact by 4%. Moreover, the gender of the household head is not related to the program impact. The p-value associated with the variable *sexhead* is larger than 0.05; therefore, its estimate is not significant.[1]

3.3 Quantile Treatment Effect

We measure the effect of a program on the mean outcome because we expect the program to shift the distribution of this variable. Nonetheless, it is possible that the program affect the outcome at other points of its distribution. For

[1]The interpretation of treatment estimate is not straightforward, so we omit it here.

example, it could affect its median (50th percentile) or any other percentile. The quantile regression estimates the program effect at any percentile of the outcome variable distribution.

```
qreg lnexptot T, quantile(0.5)

(output)
Median regression                                    Number of obs =        300
  Raw sum of deviations 110.6147 (about 8.3203945)
  Min sum of deviations 109.2565                     Pseudo R2     =     0.0123

------------------------------------------------------------------------------
    lnexptot |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           T |   .1600161     .06413     2.50   0.013     .0338111    .2862211
       _cons |    8.22173   .0453467   181.31   0.000      8.13249    8.310971
------------------------------------------------------------------------------
```

In our example, the microcredit program increases median household consumption by 18%. This suggests that the distribution of *lnexptot* is symmetric because the mean effect and the median effect are similar.

3.4 Unintended Effects

The program impact on intermediary outcomes and unintended effects are important to evaluate. Aside from the outcome variable, other variables may be worth looking at. The intermediary outcomes should be determined before the empirical work starts. To select them, you can use insights about the program or expectations about its impact, for example, the context in which the program takes place and what we expect from economic theory. In the case of a microcredit program, intermediary outcomes include expenses on productive assets and spending on health of adults and children. Besides looking at intermediary outcomes, one may want to test whether the program has some unintended effects. For example, if women are the recipients of aid the program may impact domestic violence; this effect could be positive or negative.

Bibliography

Angrist, J. and Pischke, J. (2009). *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press.

Cameron, A. and Trivedi, P. (2005). *Microeconometrics: Methods And Applications*. Cambridge University Press.

Duflo, E., Glennerster, R., and Kremer, M. (2006). Using Randomization in Development Economics Research: A Toolkit. Working Paper 333, National Bureau of Economic Research.

Gertler, P., Martinez, S., Premand, P., Rawlings, L., and Vermeersch, C. (2011). *Impact Evaluation in Practice*. World Bank Training Series. World Bank.

Khandker, S. R., Koolwal, G. B., and Samad, H. A. (2010). *Handbook on Impact Evaluation : Quantitative Methods and Practices*. Number 2693 in World Bank Publications. The World Bank.

Morra-Imas, L. and Rist, R. (2009). *The Road to Results: Designing and Conducting Effective Development Evaluations*. World Bank Training Series. World Bank.

Pickering, A. J. (2014). (Photographer). Copyright and courtesy of selected photos. From: http://www.pbase.com/amyjanel/profile.

StataCorp. (2009). *Stata: Release 12*. Statistical Software. College Station, TX: StataCorp LP.

Last modified: Québec, April 24, 2014.