16. Regression Discontinuity Design

When treatment is assigned exclusively on the basis of a cutoff value, then regression discontinuity (RD) design is a suitable alternative to randomized experiments or other quasi-experimental designs. Unlike randomized design, an eligible group need not be excluded from treatment just for the sake of impact assessment. Impact assessment can be implemented with RD design using the Bangladesh data because participation in microcredit programs is officially determined by a household's landholding; that is, a household is eligible to participate only if it has fewer than 50 decimals of land. Therefore, the cutoff point of 50 decimals in land assets fulfills the RD design criterion.

Impact Estimation Using RD

The impact assessment of RD is based on the idea that the sample in the neighborhood of the cutoff point (above and below) represents features of randomized design, because households in treatment and control groups are very similar in their characteristics and they vary only in their treatment status. So a difference in mean outcomes of treated and control groups restricted to the vicinity of the cutoff point (that is, local to the discontinuity) gives the impact of intervention. RD has two versions. In one, called *sharp discontinuity*, the cutoff point deterministically establishes treatment status. That is, everyone eligible gets the treatment, and no one ineligible gets it. In the other type of discontinuity, called *fuzzy discontinuity*, treatment status does not jump abruptly from zero to one as households become eligible from ineligible. This scenario is more realistic, particularly in this case, because some eligible households decide (for one reason or another) not to participate in microcredit, whereas some ineligible households do participate. In good RD design, eligible nonparticipants and ineligible participants remain low. The impact of microcredit participation, using RD design, can be given by following expression:

$$\mathrm{I} = (y^{+} - y^{-})/(s^{+} - s^{-}), \tag{16.1}$$

where, y^{+} is the mean outcome for microcredit participants whose landholding is in the vicinity of 50 decimals, y^{-} is the mean outcome for microcredit nonparticipants whose landholding is in the vicinity of 50 decimals, s^{+} is the mean treatment status for eligible households whose landholding is in the vicinity of 50 decimals, and s^{-} is the mean treatment status for ineligible households whose landholding is in the vicinity of 50 decimals.

In sharp discontinuity, $s^+ = 1$ and $s^- = 0$, and the difference in mean outcomes of participants and nonparticipants gives the impact.

In reality, instead of directly calculating means of outcome and treatment, one estimates their values from local linear (or kernel) regressions that are implemented in both sides of the cutoff point. Then these values are plugged into equation 16.1 to get estimated impacts.

Implementation of Sharp Discontinuity

Bangladesh data hh_91.dta or hh_98.dta do not satisfy the conditions to fulfill sharp discontinuity design because program participation is not deterministic based on the landholding cutoff point. In other words, some eligible households (land asset < 50 decimals) do not participate, and some ineligible households (land asset >= 50 decimals) do participate. Therefore, to demonstrate sharp discontinuity, hh_98.dta are adjusted by dropping these two types of households:

```
use ..\data\hh_98,clear;
gen lexptot=ln(1+exptot);
gen lnland=ln(1+hhland/100);

drop if (hhland<50 & (dmmfd==0|dfmfd==0))|(hhland>=50 &
(dmmfd==1|dfmfd==1));
```

The next step is to run the local linear regression for outcome (household per capita expenditure) against household's landholding for both eligible (participants) and ineligible (nonparticipants) households. As a result of the previous operation of dropping some households, eligible households are now deterministically participants and ineligibles deterministically nonparticipants. Local polynomial regression allows estimated outcomes to be stored for both participants and nonparticipants. The next step is to take means of those outcomes at the cutoff point. Because the cutoff point is a single value (50 decimals), it is better to specify a range of landholding values and take means of outcomes for households that are within that range. That range is set from 45 to 50 decimals for participants and from 50 to 55 for nonparticipants. With the means of outcomes computed, their difference can be taken to get estimated impacts of microcredit participation on per capita expenditure in the neighborhood of the cutoff point. This whole process is coded as follows within a Stata program called *rd_sharp*:

```
prog rd_sharp, rclass;
  version 8.2;
  args outcome;
  confirm var `outcome';
  tempname outrd1 outrd0 outcome1 outcome0;
  locpoly `outcome' lnland if hhland<50, gen(`outrd1')
at(lnland) nogr tri w(3) d(1);
```

```
  locpoly `outcome’ lnland if hhland>=50, gen(`outrd0’)
at(lnland) nogr tri w(3) d(1);
  sum `outrd1’ if hhland>=45 & hhland<50, meanonly;
  scalar `outcome1’=r(mean);
  sum `outrd0’ if hhland>=50 & hhland<55, meanonly;
  scalar `outcome0’=r(mean);
  return scalar diff_outcome=`outcome1’-`outcome0’;
end;
```

Although estimated impacts can be calculated this way, this process does not give a standard error that is used to calculate *t*-statistics. Standard error can be calculated by bootstrapping the preceding program. Bootstrapping runs a command (or set of commands) repeatedly by randomly drawing observations (with replacement) from the data, stores estimation results for each run, and then calculates standard error from the saved estimations. Each command need not be bootstrapped separately. Instead, the program that includes all the needed commands can be bootstrapped. For this reason, when multiple commands need to be bootstrapped together, writing a Stata program is extremely convenient. Programming also allows the same program to be run using different parameters. Look at different options of the “locpoly” command in the rd_sharp program, which runs the local linear regression of generic outcome variable against log of household land for both participants and nonparticipants:

gen()	stores the result of the estimation, that is, estimated value of outcome
at()	specifies a variable that contains the values at which the smooth of kernel regression should be evaluated
tri	specifies that the kernel type for local linear regression is triangle
w	specifies the half-width of the kernel, the bandwidth of the smoothing window around each point
nogr	suppresses graphs for each bandwidth
d()	specifies degree of polynomial to be used in the smoothing (1 implies linear regression)

In local linear regression, different bandwidths can produce different estimates, so testing with more than one bandwidth is recommended. Choice of kernel is less important, although trying different types can help check the robustness of estimates. An important observation to make here is that the rd_sharp program has no parameter to indicate microcredit program participation. That is because microcredit program participation has been made deterministic by landholding (by, as mentioned before, the “drop” command).

The following commands set a seed for random drawing for the bootstrapping and then do the bootstrapping. Bootstrapping is done by executing the Stata “bootstrap” command, which is followed by the command to be bootstrapped in double quotes (“ ”) and then the statistics or expression to be estimated. Here the “bootstrap” command runs the previously defined rd_sharp program with the "lexptot" argument, which replaces the generic argument "outcome" with lexptot (log of per capita annual expenditure). Consequently, lexptot is run against lnland (log of household landholding) using local linear regressions. At the end of execution, program rd_sharp returns the difference of means of lexptot (estimated impact), which the "bootstrap" command stores in a variable called "impact_sharp." Finally, "bootstrap" executes the rd_sharp program 100 times.

```
set seed 12345;
bootstrap "rd_sharp lexptot" impact_sharp=r(diff_outcome),
reps(100) nowarn;
```

The output of the "bootstrap" command is as follows. It shows that microcredit program participation has a negative impact on per capita expenditure (−12.6 percent) and standard error is 0.112:

```
command:       rd_sharp lexptot
statistic:     impact_s~p = r(diff_outcome)

Bootstrap statistics                                Number of obs  =      243
                                                    Replications   =      100
---------------------------------------------------------------------------
Variable     | Reps  Observed    Bias     Std. Err. [95% Conf. Interval]
-------------+-------------------------------------------------------------
impact_sharp |  92  -.1264224  .0023491   .1116639  -.3482292 .0953843  (N)
             |                                      -.3132059 .0937947  (P)
             |                                      -.3132059  .125849 (BC)
---------------------------------------------------------------------------

Note: N  = normal
      P  = percentile
     BC  = bias-corrected
```

The following commands create the *t*-statistics of the estimated impact and display them:

```
gen t_impact_sharp=_b[impact_sharp]/_se[impact_sharp];
sum t_impact_sharp;
```

After executing these commands, one can see that estimated impact is not significant ($t = -1.132$).

```
    Variable |  Obs    Mean      Std. Dev.    Min         Max
-------------+-----------------------------------------------
t_impact_s~p |  243  -1.132169      0      -1.132169  -1.132169
```

Implementation of Fuzzy Discontinuity

Unlike the implementation of sharp discontinuity, implementation of fuzzy discontinuity does not require dropping observations for eligible households' nonparticipation or ineligible households' participation. The program to estimate impacts for fuzzy discontinuity is very similar to the one used for sharp discontinuity. Here local polynomial regressions for treatment are included in addition to those for outcomes. Estimated impact is calculated using the formula specified in equation 16.1. The program to calculate fuzzy discontinuity follows:

```
prog rd_fuzzy, rclass;
  version 8.2;
  args treatment outcome;
  confirm var `treatment';
  confirm var `outcome';
  tempname treatrd1 treatrd0 outrd1 outrd0 treat1 treat0 out-
come1 outcome0;
  locpoly `treatment' lnland if hhland<50, gen(`treatrd1')
at(lnland) nogr tri w(3) d(1);
  locpoly `treatment' lnland if hhland>=50, gen(`treatrd0')
at(lnland) nogr tri w(3) d(1);
  locpoly `outcome' lnland if hhland<50, gen(`outrd1')
at(lnland) nogr tri w(3) d(1);
  locpoly `outcome' lnland if hhland>=50, gen(`outrd0')
at(lnland) nogr tri w(3) d(1);
  sum `treatrd1' if hhland>=45 & hhland<=55, meanonly;
  scalar `treat1'=r(mean);
  sum `treatrd0' if hhland>=45 & hhland<=55, meanonly;
  scalar `treat0'=r(mean);
  sum `outrd1' if hhland>=45 & hhland<=55, meanonly;
  scalar `outcome1'=r(mean);
  sum `outrd0' if hhland>=45 & hhland<=55, meanonly;
  scalar `outcome0'=r(mean);
  return scalar impact=(`outcome1'-`outcome0')/(`treat1'-
`treat0');
end;
```

The rd_fuzzy program, as opposed to rd_sharp, takes two arguments—one for treatment and one for outcome. Therefore, to estimate impacts of female microcredit participation on households' per capita expenditure, the "bootstrap" command executes the program rd_fuzzy with two arguments: dfmfd (women's microcredit participation) and lexptot (per capita annual expenditure). Here are the codes that run the relevant "bootstrap" command:

```
set seed 123;
bootstrap "rd_fuzzy dfmfd lexptot" impact_fuzzy_f=r(impact),
reps(100) nowarn;
```

The output of the "bootstrap" command shows that the sign of estimated impact is still negative:

```
command:      rd_fuzzy dfmfd lexptot
statistic:    impact_f~f = r(impact)
```

```
Bootstrap statistics                              Number of obs  =     1129
                                                  Replications   =      100
------------------------------------------------------------------------------
Variable     | Reps  Observed   Bias    Std. Err. [95% Conf. Interval]
-------------+----------------------------------------------------------------
impact_fuz~f | 100  -1.702198  1.92124   3.571683  -8.789193  5.384796  (N)
             |                                    -10.52238   9.24404  (P)
             |                                    -13.93708 -.0473376 (BC)
------------------------------------------------------------------------------

Note: N  = normal
      P  = percentile
     BC  = bias-corrected
```

The following commands create and display the *t*-statistics of the estimated impact:

```
gen t_impact_fuzzy_f=_b[impact_fuzzy_f]/_se[impact_fuzzy_f];
sum t_impact_fuzzy_f;
```

After executing these commands, one sees that estimated impact is insignificant ($t = -0.477$):

```
    Variable |  Obs     Mean    Std. Dev.     Min        Max
-------------+---------------------------------------------------
t_impact_f~f |  1129  -.4765815     0      -.4765815  -.4765815
```

Exercise

Estimate male program participation impacts on household per capita expenditure using fuzzy discontinuity design. Discuss your results.