Fuzzy Regression Discontinuity

Megha

Regression Discontinuity

Assignment to the treatment group is determined through a cut score (e.g., 4000 on Algebra I end of course STAAR scale scores). People above the cut score take and complete Algebra II (treatment) and people below don't (control). The idea behind RDD is that people right around the cut-score are randomly above or below—like they have been randomly assigned to treatment or control. External validity will be limited because we can only estimate average treatment effect for people near the cut-off. We can't say what the effect of the treatment is for all the students without extrapolating.

Sharp RDD

If all people above cut-score took and completed Algebra II, and all those below didn't we would have perfect compliance. We basically need to worry only about who is assigned to treatment versus control.

Fuzzy RDD

However, in our case, you can imagine that not all those who scored above the 4000 cut-point would take and complete Algebra II. They are only more likely to do so. And, not all who scored below 4000 never took Algebra II. So we have non-compliance issue.

To deal with non-compliance we combine regular RDD with instrumental variables (IV) regression.

Example

The following is an example data from Causal class. In the data below (from James's notes):

- `id`: unique identifier for each family
- `school`: unique identifier for the student's school
- `male`: indicator variable equal to one if the student is male
- `FRL`: indicator variable equal to one if the student is eligible for the free/reduced-price lunch program
- `ITBS_read_96`: student's score on the reading portion of the Iowa Test of Basic Skills in 3rd grade in 1996. Students scoring less than -1 were assigned to remedial summer school.
- `attend_SS`: indicator variable equal to one if the student attended summer school
- `ITBS_read_97`: student's score on the reading portion of the Iowa Test of Basic Skills in 1997

```r
library(tidyverse)
library(rddensity)
library(rdrobust)
library(AER)
library(clubSandwich)
library(kableExtra)

summer_school <- read_csv("data/summer_school.csv")

glimpse(summer_school)
```

```
Rows: 4,678
Columns: 7
$ id           <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
$ school       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ male         <dbl> 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, ~
$ FRL          <dbl> 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, ~
$ ITBS_read_96 <dbl> -0.18207764, 0.07645412, -0.99741097, -1.19872460, -1.641~
$ attend_SS    <dbl> 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, ~
$ ITBS_read_97 <dbl> -0.364692207, 0.252508945, -0.850805689, 0.002452996, -0.~
```

Sharp RDD

For now lets assume that everyone below the cut score who were assigned to remedial summer school went to summer school.

Assumptions

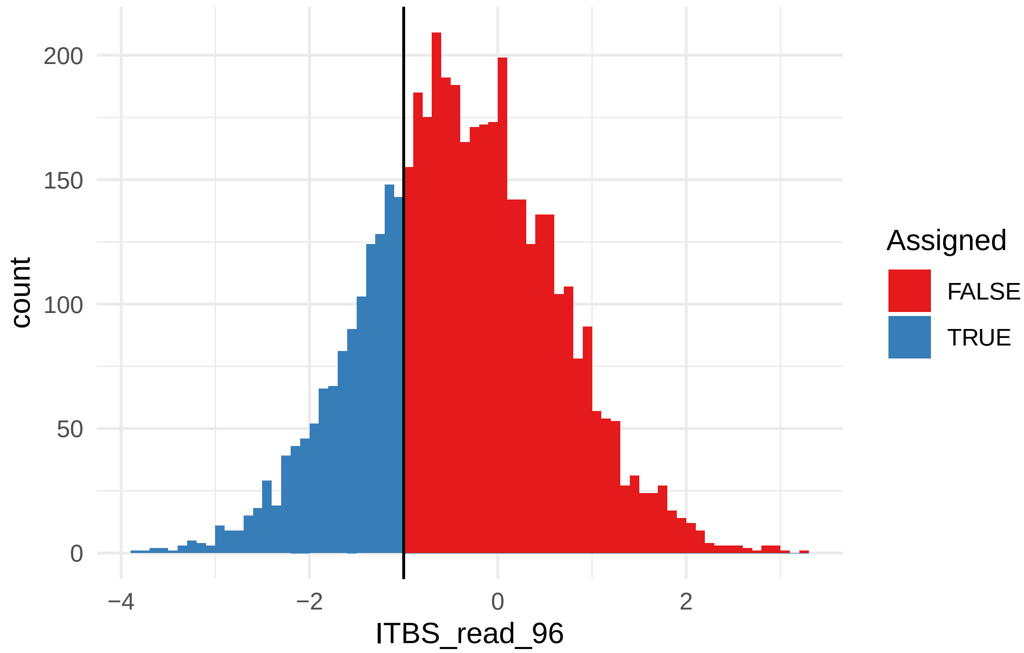
Before we run RDD, we should check some assumptions.

1. No tampering of forcing variable. If people who have power to score know about the cut-score beforehand, they could make tamper with the scoring and make more people pass the Algebra I STAAR test (so that districts look good, for example). That is not good for causal inference. So the first step is to check if there is any weird jumps around the cut-off. We create histogram like below and can conclude that there is no evidence of tampering- density is continuous at -1. And, we conduct a formal test using `rddensity()` function and note that the p value greater than .05 so there is not much evidence that density is discontinuous at the cut score.

```
# the cutscore is -1
cutscore <- -1

# centering the forcing variable on the cutscore so the cutscore is now 0
summer_school <- summer_school %>%
  mutate(Fi = ITBS_read_96 - cutscore,
         assigned = factor(Fi < 0, exclude = NA))

# create a histogram
ggplot(summer_school, aes(x = ITBS_read_96, fill = factor(assigned))) +
  geom_histogram(binwidth = .1, center = .15) +
  geom_vline(xintercept = c(-1), colour = "black") +
  scale_fill_brewer(type = "qual", palette = 6) +
  labs(fill = "Assigned") +
  theme_minimal()
```




```r
# formal test
summary(rddensity(summer_school$ITBS_read_96, c = cutscore))
```

```
Manipulation testing using local polynomial density estimation.

Number of obs =       4678
Model =               unrestricted
Kernel =              triangular
BW method =           estimated
VCE method =          jackknife

c = -1                Left of c           Right of c
Number of obs         1262                3416
Eff. Number of obs    965                 1279
Order est. (p)        2                   2
Order bias (q)        3                   3
BW est. (h)           0.927               0.708

Method                T                   P > |T|
Robust                0.2755              0.7829
```

```
P-values of binomial tests (H0: p=0.5).

Window Length / 2          <c     >=c    P>|T|
0.008                      10      10    1.0000
0.016                      19      24    0.5424
0.024                      36      36    1.0000
0.032                      44      53    0.4168
0.039                      52      66    0.2313
0.047                      67      77    0.4534
0.055                      79      90    0.4419
0.063                      89      99    0.5117
0.071                     103     109    0.7314
0.079                     111     117    0.7406
```

2. Check if people near cut-off are close to randomized. Check baseline equivalence. Skipping this for now but basically need to run smds on any baseline covariates and compare treatment and control group within some bandwidth.

Analysis

For sharp RDD, the following code will select optimal bandwidth and run the RDD analysis. I am specifying `-Fi` below because in our cause people below the cut-point are treatment, and the default for `rdrobust` is that people above the cut-off is treatment (which is true for HB5 as people above 4000 score is treatment - taking and completing Algebra II).

```
mitt <- with(summer_school,
             rdrobust(y = ITBS_read_97, # the outcome variable
                      x = -Fi, # cut score or forcing variable
                      c = 0, # the cut point which is 0 because we centered it
                      vce = "hc2", # heteroskedasticity consistent
                      cluster = school)) # cluster robust

summary(mitt)
```

```
Call: rdrobust

Number of Obs.                 4678
BW type                       mserd
Kernel                   Triangular
VCE method                      HC2
```

```
Number of Obs.                 3416         1262
Eff. Number of Obs.            1141          748
Order est. (p)                    1            1
Order bias  (q)                   2            2
BW est. (h)                   0.616        0.616
BW bias (b)                   1.006        1.006
rho (h/b)                     0.613        0.613
Unique Obs.                    3416         1262

=============================================================================
        Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional     0.090     0.067     1.330     0.183    [-0.042 , 0.221]
        Robust         -         -     0.905     0.366    [-0.084 , 0.228]
=============================================================================
```

Fuzzy RDD

If there isn't prefect compliance, you would run fuzzy RDD like below. With the cut score as the instrument in the instrumental variable regression, actually attending the summer school as the treatment. The `rdrobust()` function runs two stage least squares regression.

Assumptions

Here for fuzzy you need to check assumptions for instrumental variables regression as well as those for RDD. The assumptions are local because for RDD we are only estimating effects for students near the cutoff.

1. Local exclusion restriction - Instrument does not have any effect on outcome other than through treatment. This one we check theoretically.
2. Local monotonicity - No defiers. We also check this theoretically.
3. Treatment effectiveness - Being assigned to treatment increases the probability of treatment receipt. You check that by regressing treatment receipt `attend_SS` on treatment assignment `-Fi`, the forcing variable. Below is the first stage test:

```
first_stage <- with(summer_school,
                    rdrobust(y = attend_SS, # attend_SS instead of the outcome
                             x = -Fi,
                             c = 0,
                             vce = "hc2",
```

```r
                                    cluster = school))

summary(first_stage)
```

```
Call: rdrobust

Number of Obs.                   4678
BW type                         mserd
Kernel                     Triangular
VCE method                        HC2

Number of Obs.                   3416         1262
Eff. Number of Obs.              1012          685
Order est. (p)                      1            1
Order bias  (q)                     2            2
BW est. (h)                     0.549        0.549
BW bias (b)                     0.874        0.874
rho (h/b)                       0.628        0.628
Unique Obs.                      3416         1262

=============================================================================
        Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional     0.759     0.031    24.445     0.000     [0.698 , 0.819]
        Robust         -         -    20.670     0.000     [0.680 , 0.823]
=============================================================================
```

Analysis

Below is the code to run fuzzy RDD using `rdrobust()`:

```r
mcate <- with(summer_school,
              rdrobust(y = ITBS_read_97, # the outcome is back to the 97 test score
                       x = -Fi,
                       c = 0,
                       fuzzy = attend_SS, # here we are adding attend to fuzzy
                       vce = "hc2",
                       cluster = school))


summary(mcate)
```

```
Call: rdrobust

Number of Obs.                 4678
BW type                       mserd
Kernel                   Triangular
VCE method                      HC2

Number of Obs.                 3416         1262
Eff. Number of Obs.            1347          852
Order est. (p)                    1            1
Order bias  (q)                   2            2
BW est. (h)                   0.747        0.747
BW bias (b)                   1.063        1.063
rho (h/b)                     0.703        0.703
Unique Obs.                    3416         1262

=============================================================================
        Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional     0.127     0.078     1.639     0.101    [-0.025 , 0.280]
        Robust         -         -     0.968     0.333    [-0.096 , 0.283]
=============================================================================
```

Fuzzy RDD by Hand

The following function is what happens under the hood for fuzzy RDD:

```
estimate_mcate <- function(dat = summer_school,
                           poly = 1,
                           factor_mag = 1){

  # select optimal bandwidth for any polynomial
  b <- with(dat,
            rdbwselect(y = ITBS_read_97,
                       x = Fi,
                       vce = "hc2",
                       p = poly,
                       fuzzy = attend_SS))

  # extract the bandwidth
  bandwidth <- b$bws[1]
```

```
# multiply for sensitivity analysis
# this will make the bandwidth bigger or smaller
bandwidth <- bandwidth * factor_mag

# upper and lower bandwidth
b_lower <- -bandwidth
b_upper <- bandwidth

# subset the data to include units within the bandwidth and create weights
dat_bw <- subset(dat, b_lower < Fi & Fi < b_upper)
dat_bw$tri_wt <- 1 - abs(dat_bw$Fi) / b_upper

#  run ivreg based on order of polynomial
# however this assumes the outcome is continuous
# like rdrobust
# but for HB5 most of our outcomes are binary
# which is why I think we are getting some wonky estimates


if(poly == 1){

model <- ivreg(ITBS_read_97 ~ Fi + attend_SS + Fi:assigned | Fi + assigned + Fi:assigned

}  else if(poly == 2){

model <- ivreg(ITBS_read_97 ~ Fi + I(Fi^2) + attend_SS + Fi:assigned + I(Fi^2):assigned|
                 Fi + I(Fi^2) + assigned + Fi:assigned + I(Fi^2):assigned, data = dat_bw

} else if (poly == 3){

  model <- ivreg(ITBS_read_97 ~ Fi + I(Fi^2) + I(Fi^3) + attend_SS + Fi:assigned + I(Fi^

}

# cluster the se by school

res <- coef_test(model,
                 vcov = "CR2",
                 cluster = dat_bw$school,
                 tidy = TRUE) %>%
  mutate(bandwidth = bandwidth)
```

Table 1: MCATE Results

	Coef	beta	SE	tstat	df_Satt	p_Satt	bandw
(Intercept)	(Intercept)	-0.8567310	0.0402097	-21.3065627	35.44731	0.0000000	0.710
Fi	Fi	0.8014338	0.1059066	7.5673675	36.71125	0.0000000	0.710
attend_SS	attend_SS	0.1242942	0.0805343	1.5433703	38.41937	0.1309388	0.710
Fi:assignedTRUE	Fi:assignedTRUE	-0.0276748	0.1703650	-0.1624444	38.21932	0.8718117	0.710

```
  return(res)


}
```

```
mcate_hand <- estimate_mcate()

mcate_hand %>%
  kable(format = "latex", caption = "MCATE Results") %>%
  kable_styling(c("striped", "bordered"),  full_width = F)
```

The `attend_SS` coef is very similar to the one from `rdrobust()` above. Some small discrepancy in estimation probably - if you take out the cluster in `rdrobust()` the coefs match like below (something to look into because clustered se's shouldn't impact the coefficient but don't worry about this now :D):

```
mcate_2 <- with(summer_school,
              rdrobust(y = ITBS_read_97, # the outcome is back to the 97 test score
                       x = -Fi,
                       c = 0,
                       fuzzy = attend_SS, # here we are adding attend to fuzzy
                       vce = "hc2"))

summary(mcate_2)
```

```
Call: rdrobust

Number of Obs.                 4678
BW type                       mserd
Kernel                   Triangular
VCE method                      HC2
```

```
Number of Obs.                 3416        1262
Eff. Number of Obs.            1283         826
Order est. (p)                    1           1
Order bias  (q)                   2           2
BW est. (h)                   0.711       0.711
BW bias (b)                   1.085       1.085
rho (h/b)                     0.655       0.655
Unique Obs.                    3416        1262

=============================================================================
        Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional     0.124     0.073     1.703     0.089    [-0.019 , 0.267]
        Robust         -         -     1.054     0.292    [-0.080 , 0.266]
=============================================================================
```

Sensitivity

When running RDD, we need to check if results are robust to different bandwidths and different polynomial specifications. Below is the code to do that:

```r
# different polynomial and bandwidth specifications
design_factors <- list(
  poly = c(1, 2, 3),
  factor_mag = c(.5, 1, 2)

)

# combine into a design set
params <-
  cross_df(design_factors)

# run sensitivity analysis
mcate_results <-
    params %>%
    mutate(
      res = pmap(., .f = estimate_mcate)
    ) %>%
    unnest(cols = res)

# clean the results
```

Table 2: Sensitivity for MCATE

poly	factor_mag	Coef	beta	SE	tstat	df_Satt	p_Satt	bandwidth
1	0.5	attend_SS	0.1768081	0.1239215	1.426776	36.18805	0.1622161	0.3553168
1	1.0	attend_SS	0.1242942	0.0805343	1.543370	38.41937	0.1309388	0.7106335
1	2.0	attend_SS	0.1797898	0.0571533	3.145746	39.58091	0.0031408	1.4212670
2	0.5	attend_SS	0.2174620	0.1590693	1.367090	33.46836	0.1807098	0.3903672
2	1.0	attend_SS	0.1293982	0.1233622	1.048929	36.89573	0.3010334	0.7807343
2	2.0	attend_SS	0.1343744	0.0849269	1.582235	38.76238	0.1217223	1.5614686
3	0.5	attend_SS	0.2259102	0.1768711	1.277259	32.06215	0.2106829	0.4940174
3	1.0	attend_SS	0.1714527	0.1475946	1.161646	35.81678	0.2530623	0.9880348
3	2.0	attend_SS	0.1104628	0.0996686	1.108300	38.19759	0.2746621	1.9760695

```r
mcate_results %>%
  filter(Coef == "attend_SS") %>%
  arrange(poly) %>%
  kable(format = "latex", caption = "Sensitivity for MCATE") %>%
  kable_styling(c("striped", "bordered"),  full_width = F)
```

HB5

For HB5, the coefficients are looking wonky. The outcomes are binary so the linear probability model should give difference in proportion (e.g., of those who graduate after HB5 vs those who do before). But, I am getting proportions greater than 1 or less than 0 for some models. So need to figure out what is going wrong. One possibility is that linear probability ivreg is not working for the binary outcomes so we need to mess with the function above to run iv probit or iv logit.