Data Science for Economists

Regression Discontinuity

Kyle Coombs, adapted from Nick Huntington-Klein + Raj Chetty
Bates College | ECON/DCS 368

Table of contents

Prologue

Prologue

- We've covering difference-in-differences, which is one way of estimating a causal effect using observational data
- DID is *very* widely applicable, but it relies on strong assumptions like parallel trends
- Today we'll cover another causal inference method: Regression Discontinuity
    - This method can sometimes be easier to defend
    - But it is rarer to find situations where it applies
    - There's also plenty of room for "snake oil" here as with all causal inference
- Today I intentionally use simulated data to illustrate concepts to simplify the presentation
- But the fundamentals are the same no matter what you're studying and I don't want that lost in the econometric sauce
- As always, there's a ton here and we're just scratching the surface

Regression Discontinuity

Line up in height order

1. Line up in height order
2. Those below the median height get a pill to increase their basketball ability
3. Those above the median height don't
4. We want to know the effect of the pill on free throw percentage

- Can we compare the average height of the treated and untreated groups after a year?

Line up in height order

1. Line up in height order
2. Those below the median height get a pill to increase their basketball ability
3. Those above the median height don't
4. We want to know the effect of the pill on free throw percentage

- Can we compare the average height of the treated and untreated groups after a year?
- Nope! Heights and the rate of growth is different for other reasons than the pill
- But what if we compared people right around 5'6"? They're basically the same, except for random chance

Regression Discontinuity

The basic idea is this:

- We look for a treatment that is assigned on the basis of being above/below a *cutoff value* of a continuous variable
- For example, if your GPA exceeds a 3.0 in Florida, you're more likely to attend college (Zimmerman, 2014),
- Or if you are just on one side of a time zone line, your day starts one hour earlier/later
- Or if a candidate gets 50.1% of the vote they're in, 40.9% and they're out
- Or if you're 65 years old you get Medicare, if you're 64.99 years old you don't
- Class size must be below 40 students, so there are small classes when a grade reaches 41, 81, 121, etc. students

We call these continuous variables "Running variables" because we *run along them* until we hit the cutoff

Running variables

There is a relationship between an outcome (Y) and a running variable (X)

There is also a treatment that triggers if $X < c$, a cutoff.

- Let's do the wrong thing

1. Assign $Treatment = 1$ if running variable above c and $Treatment = 0$ if below
2. Regress $y = \beta_0 + \beta_1 Treatment + \varepsilon$
3. Get a biased estimate. Why?

Running variables

There is a relationship between an outcome (Y) and a running variable (X)

There is also a treatment that triggers if $X < c$, a cutoff.

- Let's do the wrong thing

1. Assign $Treatment = 1$ if running variable above c and $Treatment = 0$ if below

2. Regress $y = \beta_0 + \beta_1 Treatment + \varepsilon$

3. Get a biased estimate. Why?

- The running variable is omitted, so we have endogeneity!

    - e.g. Older people receive Medicare, but they're also more likely to be sick
    - Shoot! Our treatment is endogenous! We have to control for the running variable

Regression Discontinuity

- So what does this mean?
- If we can control for the running variable *everywhere except the cutoff*, then...
  - We will be controlling for the running variable, removing endogeneity
  - But leaving variation at the cutoff open, allowing for variation in treatment
- We focus on variation around the treatment, zooming in so sharply that it's basically controlled for.
  - Then the effect of cutoff on treatment is like an experiment!
- How so?
  - If your $GPA > 3$, you're more likely to attend college, but also more likely to excel otherwise

Regression Discontinuity

- The idea is that *right around the cutoff*, treatment is randomly assigned
- If you have a GPA over 2.99 (below standard FIU admission), you're basically the same as someone who has a GPA of 3.01 (just barely high enough)
- So if we just focus around the cutoff, we remove endogeneity because it's basically random which side of the line you're on
- But we get variation in treatment!
- This specifically gives us the effect of treatment *for people who are right around the cutoff* a.k.a. a "local average treatment effect"
    - We don't know the effect of being in college for someone with a GPA of 2.0

Terminology

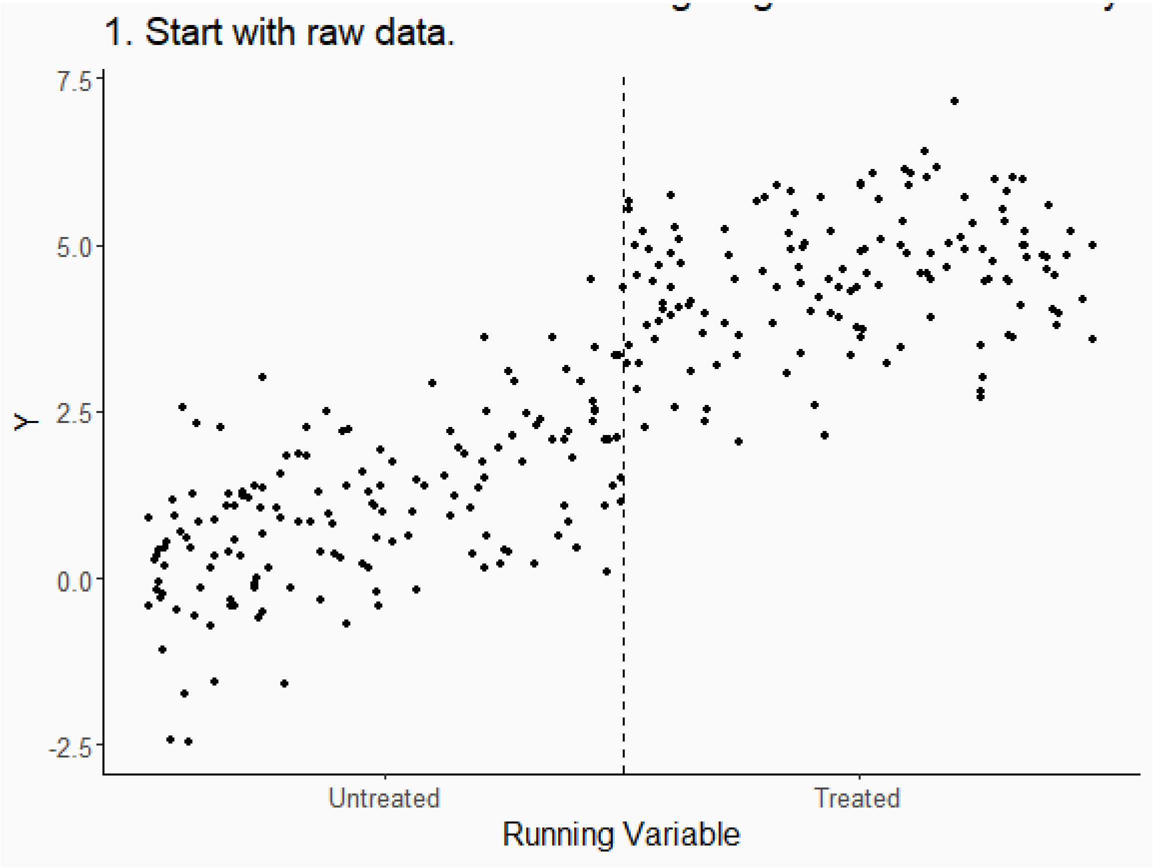
- Some quick terminology before we go on

1. **Running Variable**: The continuous variable that triggers treatment, sometimes called the **forcing variable**
2. **Cutoff**: The value of the running variable that triggers treatment
3. **Bandwidth**: The range of the running variable we use to estimate the effect of treatment -- do we look at everyone within .1 of the cutoff? .5? 1? The whole running variable?

Regression Discontinuity

- A very basic idea of this, before we even get to regression, is to create a *binned scatterplot*
- And see how the bin values jump at the cutoff
- A binned chart chops the Y-axis up into bins
- Then takes the average Y value within that bin. That's it!
- Then, we look at how those X bins relate to the Y binned values.
- If it looks like a pretty normal, continuous relationship... then JUMPS UP at the cutoff X-axis value, that tells us that the treatment itself must be doing something!

Regression Discontinuity



Fitting Lines in RDD

- Looking only at the cutoff ignores useful information from data further away
- Data away from the cutoff helps predict values at the cutoff more accurately
- Simplest approach uses OLS with an interaction term

$$Y = \beta_0 + \beta_1 Treated + \beta_2 XCentered + \beta_3 Treated \times XCentered + \varepsilon$$

Fitting Lines in RDD

- Looking only at the cutoff ignores useful information from data further away
- Data away from the cutoff helps predict values at the cutoff more accurately
- Simplest approach uses OLS with an interaction term

$$Y = \beta_0 + \beta_1 Treated + \beta_2 XCentered + \beta_3 Treated \times XCentered + \varepsilon$$

- First, we need to *transform our data*:
    - Create a "Treated" variable when treatment is applied (one side of cutoff)
    - Then, we are going to want a bunch of things to change at the cutoff.
- This will be easier if the running variable is *centered around the cutoff*.
- So we'll turn our running variable X into $X - cutoff$ and call that $XCentered$

```
cutoff = .5
df <- df %>%
  mutate(treated = X >= cutoff,
         X_centered = X - cutoff) # center at cutoff
```

Centering the Running Variable

Why do we center the running variable (subtract the cutoff)?

1. **Direct Treatment Effect Interpretation**
    - Without centering: intercept = outcome at X = 0 (usually meaningless)
    - With centering: intercept = treatment effect at the cutoff
2. **Numerical Stability with Polynomials**
    - Uncentered X^2: values get very large
    - Centered $(X - c)^2$: values stay closer to zero
3. **Clearer Interactions**
    - β_1: "jump" exactly at cutoff
    - β_3: how treatment effect changes away from cutoff

Varying Slope

- Let the slope vary to either side, i.e. fit a different regression on each side of the cutoff
- We can do this by interacting both running variable and intercept with $Treated$!
    - β_1 estimates the intercept jump at treatment (RDD effect), β_3 is the slope change.[1]

$$Y = \beta_0 + \beta_1 Treated + \beta_2 XCentered + \beta_3 Treated \times XCentered + \varepsilon$$

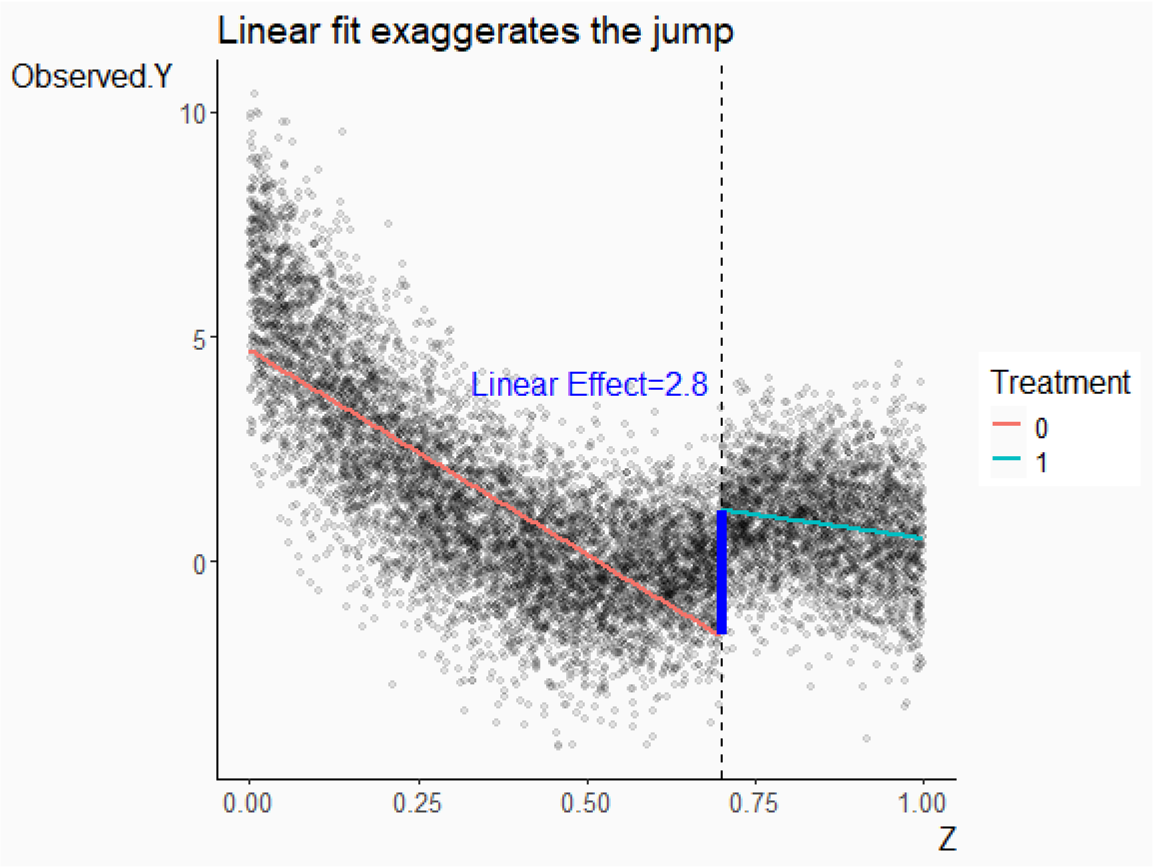
```
etable(feols(Y ~ treated*X_centered, data = df,fitstat='N',vcov='HC1')) # True treatment is 0.7
```

```
##                          feols(Y ~ tr..
## Dependent Var.:                       Y
##
## Constant                  -0.01 (0.03)
## Treated                  0.75*** (0.04)
## X_centered               0.98*** (0.09)
## Treated x X_centered     0.45*** (0.13)
## ____________________ ______________
## S.E. type                Heterosk.-rob.
## Observations                      1,000
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[1] Sometimes the change in slope is the effect of interest -- this is called a "regression kink" design, which measures how the relationship between `(X)` and `(Y)` changes at the cutoff.

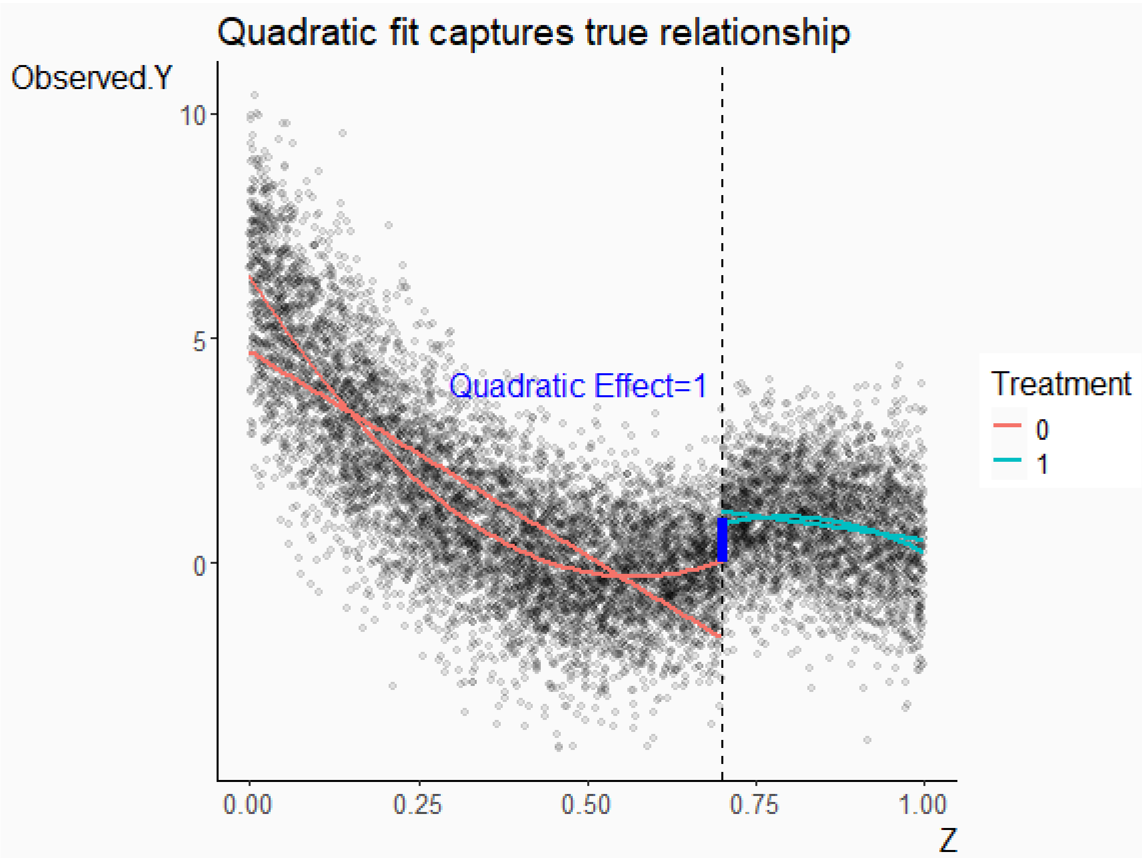
Fitting Lines in RDD

- Visualizations can help! What's going on here?



Non-linearities

- Key Point: The functional form matters!



Polynomial Terms

- We can add quadratic (or higher-order) terms to better fit curved relationships
- Key points:
    1. Center X (as before)
    2. Add squared terms: $(X - c)^2$
    3. Interact everything with treatment
- Keep it simple: you could overfit with more complex polynomials
    - Squares are usually enough -- though there's a subfield dedicated to optimizing polynomial terms
    - The "jump" at cutoff is still our RDD estimate

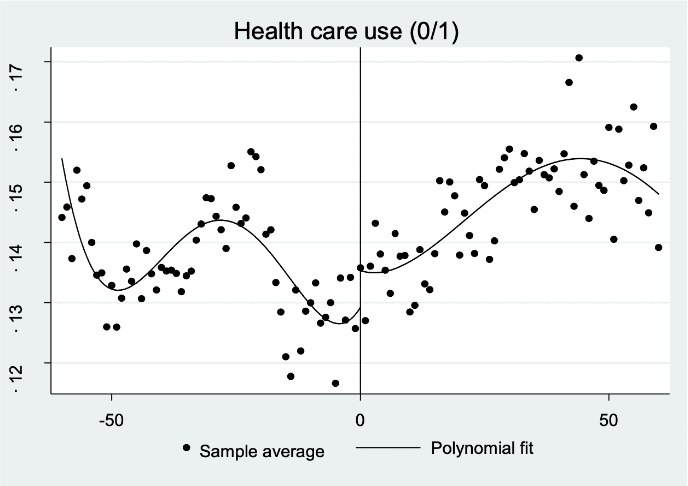
$$Y = \beta_0 + \beta_1 XC + \beta_2 XC^2 + \beta_3 Treated + \beta_4 Treated \times XC + \beta_5 Treated \times XC^2 + \varepsilon$$

```
etable(feols(Y ~ X_centered*treated + I(X_centered^2)*treated, data = df,vcov='HC1'))
```

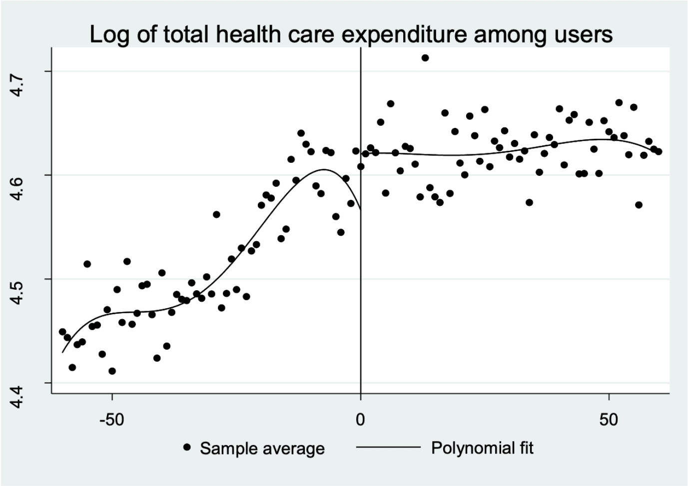
```
##                              feols(Y ~ X_..
## Dependent Var.:                           Y
##
## Constant                       -0.03 (0.04)
## X_centered                     0.70. (0.37)
## Treated                      0.77*** (0.06)
## X_centered square              -0.57 (0.72)
## X_centered x Treated            0.75 (0.56)
## Treated x X_centered squared     0.53 (1.1)
```

Careful with higher order polynomials

- Sometimes higher order polynomials can be a little too flexible and make it look like there's an effect where there isn't one
- "Overfitting" where your model too flexibly follows the data points can lie to you!



Health care use (0/1)



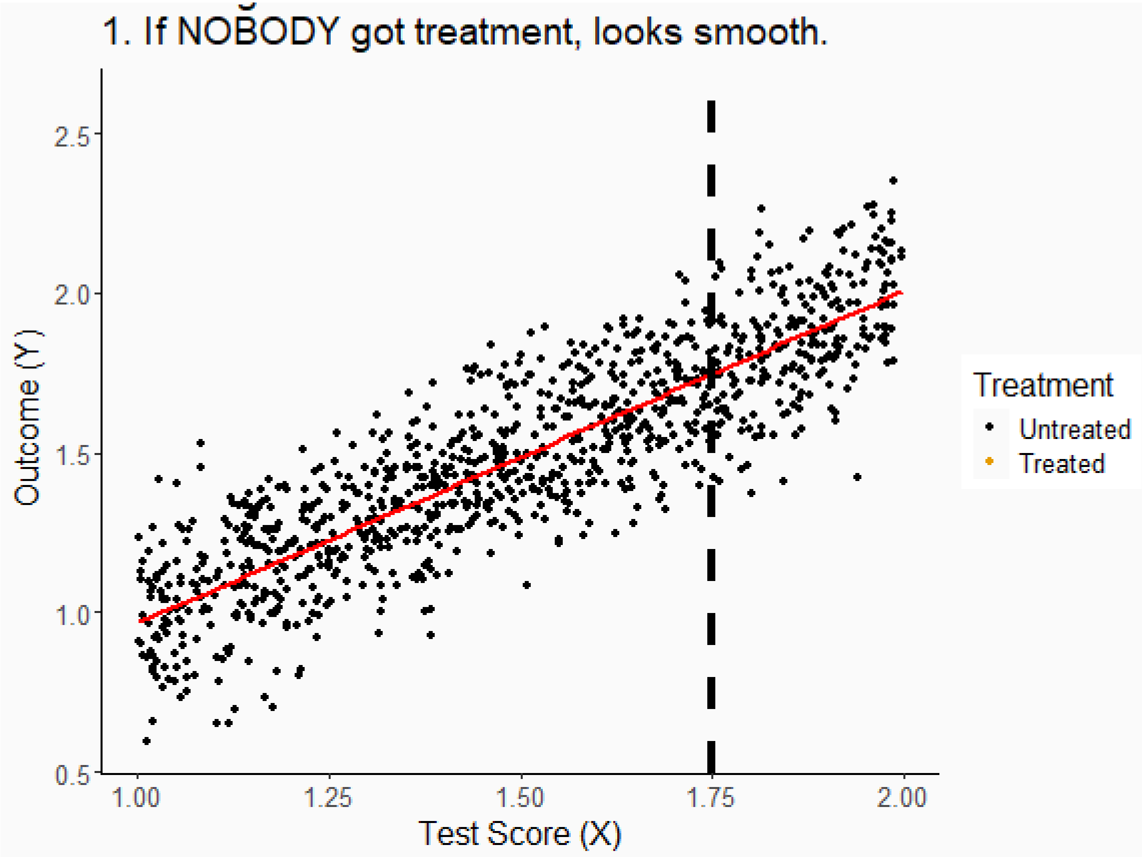
Log of total health care expenditure among users

Running variable is age with cutoff at age 20 (voting eligibility). Chang & Meyerhoefer (2020) on whether voting makes you sick via Andrew Gelman

Assumptions

- There must be some assumptions lurking around here
- Some are more obvious (we use the correct functional form)
- Others are trickier. What are we assuming about the error term and endogeneity here?
- Specifically, we are assuming that *the only thing jumping at the cutoff is treatment*
- Sort of like parallel trends, but maybe more believable since we've narrowed in so far
- For example, if earning below 150% of the poverty line gets food stamps AND job training, then we can't isolate the effect of just food stamps
    - Or if the proportion of people who are self-employed jumps up just below 150% (based on *reported* income), that's endogeneity!
- The only thing different about just above/just below should be treatment

Graphically



RDD Challenges

Windows

- The basic idea of RDD is that we're interested in *the cutoff*
- The points away from the cutoff are only useful to help predict values at the cutoff
- Do we really want that full range? Is someone's test score of 30 really going to help us much in predicting Y at a test score of 89?
- So we might limit our analysis within just a narrow window around the cutoff, just like that initial animation we saw!
- This makes the exogenous-at-the-jump assumption more plausible, and lets us worry less about functional form (over a narrow range, not too much difference between a linear term and a square), but on the flip side reduces our sample size considerably

Windows

- Pay attention to the sample sizes, accuracy (true value .7) and standard errors!

```
m1 <- feols(Y~treated*X_centered, data = df)
m2 <- feols(Y~treated*X_centered, data = df %>% filter(abs(X_centered) < .25))
m3 <- feols(Y~treated*X_centered, data = df %>% filter(abs(X_centered) < .1))
m4 <- feols(Y~treated*X_centered, data = df %>% filter(abs(X_centered) < .05))
m5 <- feols(Y~treated*X_centered, data = df %>% filter(abs(X_centered) < .01))

etable(list('All'=m1,'|X|<.25'=m2,'|X|<.1'=m3,'|X|<.05'=m4,'|X|<.01'=m5),
  fitstat='N',vcov='HC1',keep='Treated$',se.below=TRUE) # robust standard errors
```

```
##                            All    |X|<.25     |X|<.1   |X|<.05    |X|<.01
## Dependent Var.:              Y          Y          Y         Y          Y
##
## Treated               0.75***    0.77***    0.71***   0.61***      0.56
##                      (0.04)     (0.06)     (0.10)    (0.15)      (0.36)
## _______________      ________   ________   ________  ________   ________
## S.E. type            Het.-rob.  Het.-rob.  Het.-rob. Het.-rob.  Het.-rob.
## Observations            1,000        492        206        93         15
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Granular Running Variable

- We assume that the running variable varies more or less *continuously*
- That makes it possible to have, say, a test score of 89 compared to a test score of 90 it's almost certainly the same as except for random chance
- But what if our data only had test score in big chunks? i.e. I just know those earning "80-89" or "90-100"
  - Much less believable that groups only separated by random chance
- There are some fancy RDD estimators that allow for granular running variables
- But in general, if this is what you're facing, you might be in trouble
- Before doing an RDD, ask:
  - Is it plausible that someone with the highest value just below the cutoff, and someone with the lowest value just above the cutoff are only at different values because of random chance?

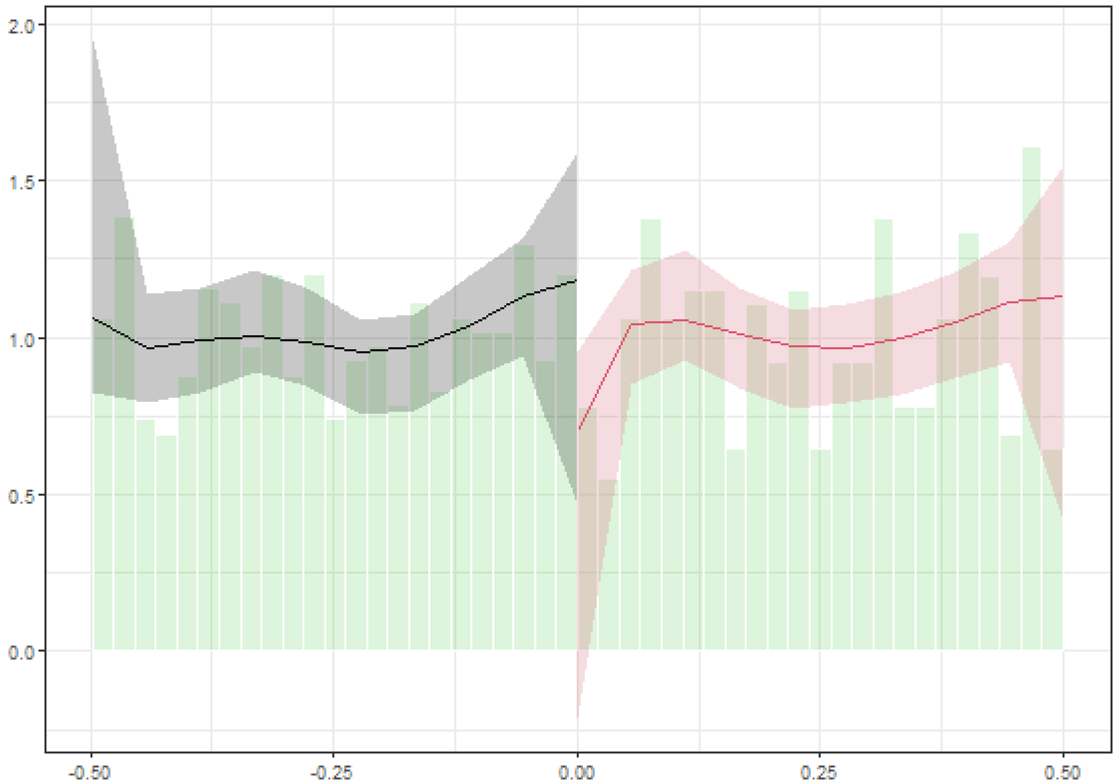
Looking for Lumping

- Ok, now let's go back to our continuous running variables
- What if the running variable is *manipulated*?
  - Imagine you're a teacher you learn a "B-student" needs a B+ for a 3.0 and admitted to FIU, you might fudge the numbers a bit
- Suddenly, that treatment is a lot less randomly assigned around the cutoff!
- If there's manipulation of the running variable around the cutoff, we can often see it in the presence of *lumping*
  - i.e. if there's a big cluster of observations to one side of the cutoff and a seeming gap missing on the other side
- "Bin" the running variable and plot a histogram of it to check for clustering at the cutoff

Testing for Manipulation

- McCrary test checks if people manipulate the running variable
- Key idea: The density should be smooth at the cutoff: sharp change = 🚩

```
mccrary ← rddensity(df$X_centered)
rdplotdensity(mccrary,df$X_centered)
```



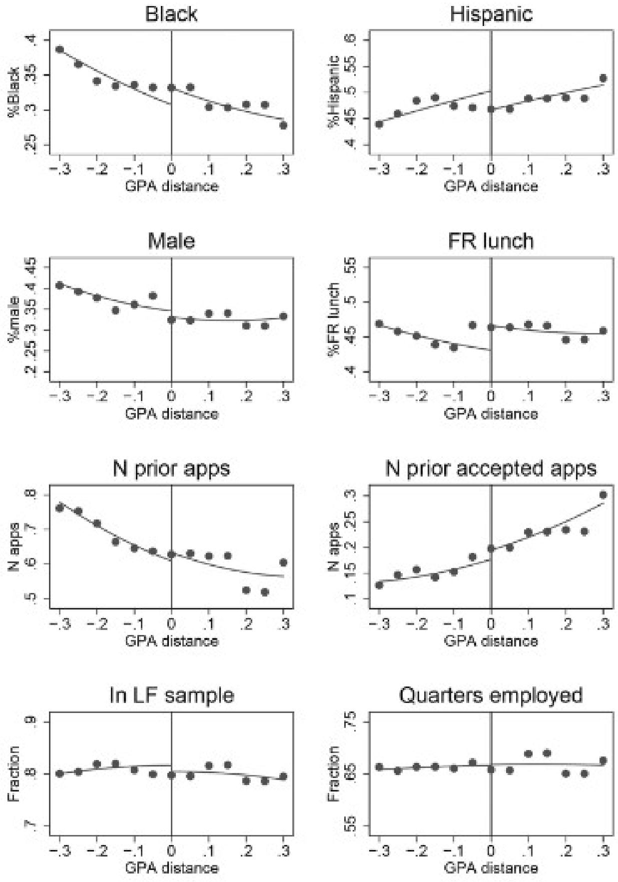
Placbo Tests for Lumping

- Check if variables *other than treatment or outcome* vary at the cutoff

- We can do this by re-running our RDD but switching our outcome with another variable

- If we get a significant jump, that's bad! That tells us that *other things are changing at the cutoff* which implies some sort of manipulation (or just super lousy luck)

- If all placebo tests are passed, that's great, but doesn't prove zero manipulation



Placebo balance tests from Zimmerman (2014)

That's it!

- That's what we have for RDD
- Go explore the regression discontinuity activity on class sizes
- There's more neat details in the appendix -- check it out if you're curious!
  - If we have time, I'll show you how to do this stuff!

Appendix

Fuzzy Regression Discontinuity and RDD Standard Errors

Fuzzy Regression Discontinuity

- So far, we've assumed that you're either on one side of the cutoff and untreated, or the other and treated
- What if it isn't so simple? What if the cutoff just *increases* your chances of treatment?
- For example, what if 30% of schools with fewer than 40 students make smaller classrooms for whatever reason
    - It can get more complicated than this -- it always can
- This is a "fuzzy regression discontinuity" (yes, that does sound like a bizarre Sesame Street episode)
- Now, our RDD will understate the true effect, since it's being calculated on the assumption that we added treatment to 100% of people at the cutoff, when really it's 70%. So we'll get roughly only about 70% of the effect

Fuzzy Regression Discontinuity

- We can account for this with a model designed to take this into account
- Specifically, we can use something called two-stage least squares (or Wald instrumental variable estimator) to handle these sorts of situations
- Basically, two-stage least squares estimates how much the chances of treatment go up at the cutoff, and scales the estimate by that change
- So it would take whatever result we got on the previous slide and divide it by 0.7 (the increased in treated share) to get the true effect

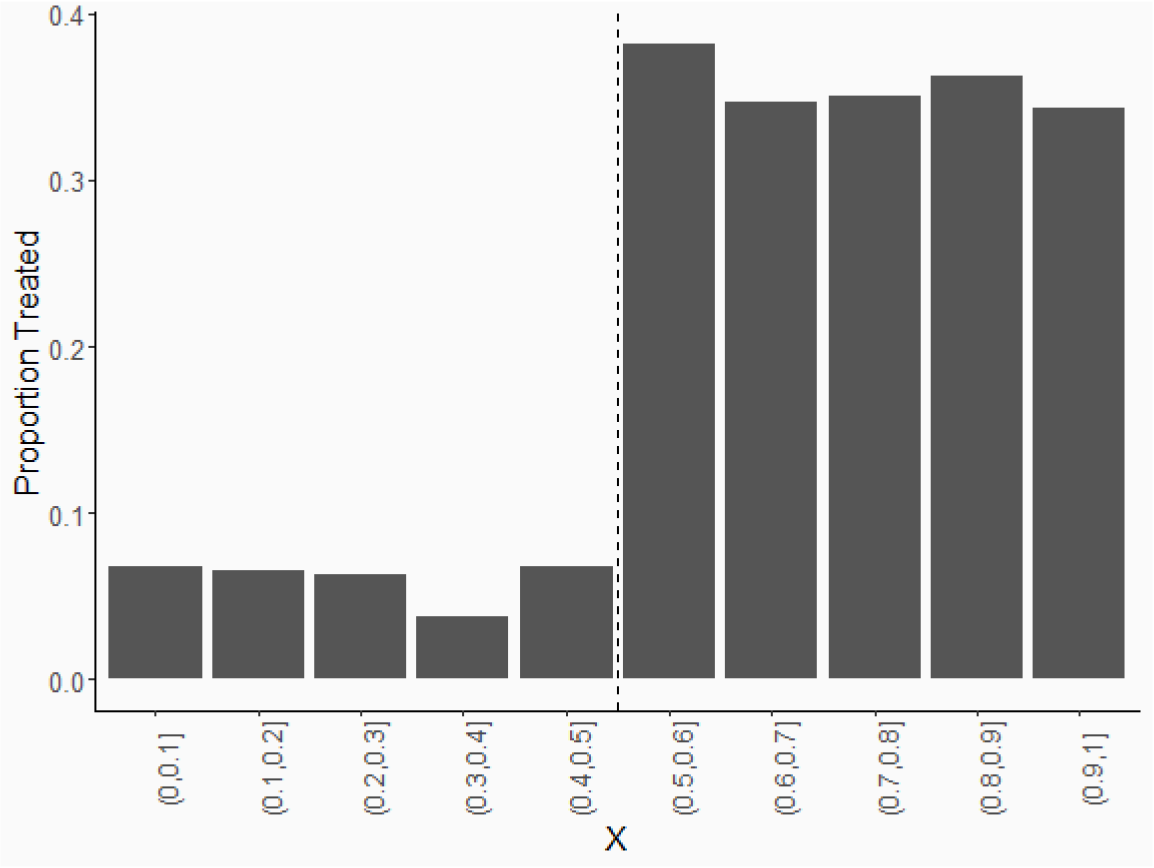
Fuzzy Regression Discontinuity

First let's make some fake data:

```
set.seed(1000)
df <- tibble(X = runif(1000)) %>%
  mutate(treatassign = .05 + .3*(X > .5)) %>%
  mutate(rand = runif(1000)) %>%
  mutate(treatment = treatassign > rand) %>%
  mutate(Y = .2 + .4*X + .5*treatment + rnorm(1000,0,0.3)) %>% # True effect .5
  mutate(X_center = X - .5) %>%
  mutate(above_cut = X > .5)
```

Fuzzy Regression Discontinuity

- Notice that the y-axis here isn't the outcome, it's "proportion treated"



Fuzzy Regression Discontinuity

- We can perform this using the instrumental-variables features of `feols`
- The first stage is the interaction between the running variable and whether treated regressed on the interaction of the running variable and the "sharp" cutoff
- `feols(outcome ~ controls | XC*treated ~ XC*above_the_cutoff)`

Fuzzy Regression Discontinuity

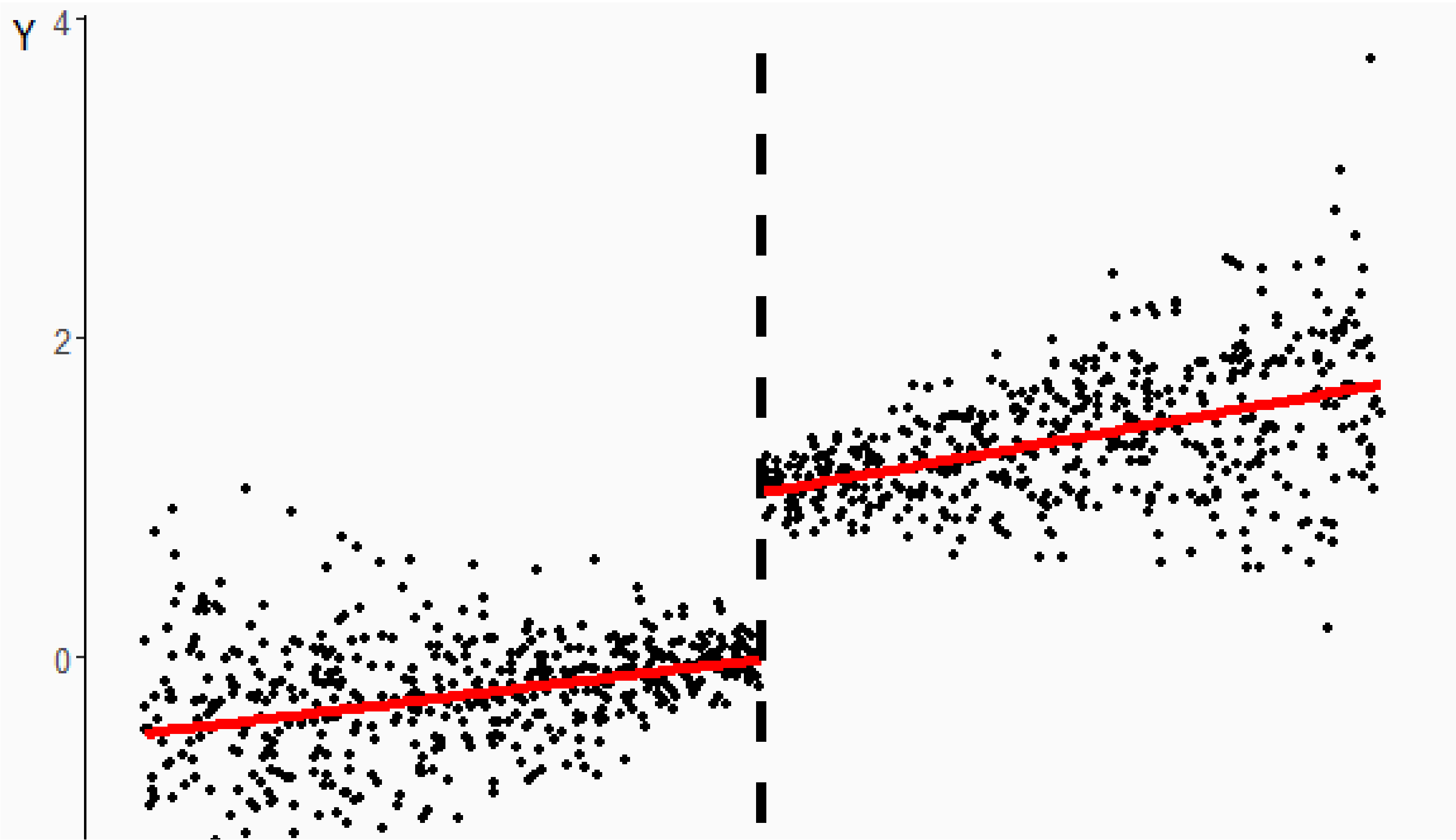
- (the true effect of treatment is .5 - okay, it's not perfect)

```
predict_treatment <- feols(treatment ~ X_center*above_cut, data = df)
without_fuzzy <-feols(Y ~ X_center*treatment, data = df)
fuzzy_rdd <- feols(Y ~ 1 | X_center*treatment ~ X_center*above_cut, data = df)
etable(predict_treatment, without_fuzzy, fuzzy_rdd,
  dict=c('above_cutTRUE'='Above Cut','treatmentTRUE'='Treatment'))
```

```
##                      predict_trea..  without_fuzzy      fuzzy_rdd
## Dependent Var.:           treatment              Y              Y
##
## Constant              0.06. (0.04) 0.41*** (0.01) 0.41*** (0.03)
## X_center              0.004 (0.12) 0.40*** (0.04) 0.45*** (0.12)
## Above Cut           0.31*** (0.05)
## X_center x Above Cut  -0.04 (0.17)
## Treatment                          0.45*** (0.03) 0.48*** (0.12)
## X_center x Treatment                  0.07 (0.10)   -0.25 (0.48)
## ____________________ ______________ ______________ ______________
## S.E. type                       IID            IID            IID
## Observations                  1,000          1,000          1,000
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Standard Errors in RDD

- Oftentimes the error term is likely correlated with the running variable
- People tend to use "robust" standard errors, `vcov='HC1'` in R or `, r` in Stata
- Other times, it makes sense to clustered standard errors by a running variable bin



How professionals do it

How professionals do it

- We've gone through all kinds of procedures for doing RDD in R already using regression
- But often, professional researchers won't do it that way because it's a bit too easy to mess up details
- Instead, they use packages like **rdrobust** (available in R, Stata, and Python) and written by a team of econometricians
- It abstracts the tedious stuff, like bandwidth selection and standard errors, and gives you loads of customization options for your RDD
- In general, packages like these written by experts who are well-published in discussing a method are a good idea to try

RDrobust

- There are three major functions in RD robust:

1. `rdrobust()` - the main estimation approach, it returns info about the regression and you can customize a variety of complex RD stuff
2. `rdplot()` - a plotting function that shows the jump at the cutoff and let's you customize much of the complexities
3. `rdbwselect()` - a bandwidth selection tool that helps you pick the best bandwidth for your RDD

Basics of **rdrobust**

- We can specify an RDD model by just telling it the dependent variable Y, the running variable X, and the cutoff c.
- We can also specify how many polynomials to use with `p`, defaults to 1
  - (it applies the polynomials more locally than our linear OLS models do - a bit more flexible)
- Use `c` to specify the cutoff (no need to center the running variable manually)
- Pick the bandwidth with `h` or use a data-driven technique with `rdbwselect()`
- Including a `fuzzy` option to specify actual treatment outside of the running variable/cutoff combo
- And many other options
- But output is pretty nasty, so you'll need to do some work to get it into a readable format

rdrobust

```r
summary(rdrobust(df$Y, df$X, c = .5))
```

```
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                 1000
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                  488          512
## Eff. Number of Obs.             135          162
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                   0.152        0.152
## BW bias (b)                   0.229        0.229
## rho (h/b)                     0.666        0.666
## Unique Obs.                     488          512
##
## =============================================================================
```

rdrobust

```r
summary(rdrobust(df$Y, df$X, c = .5, fuzzy = df$treatment))
```

```
## Fuzzy RD estimates using local polynomial regression.
##
## Number of Obs.                 1000
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                  488          512
## Eff. Number of Obs.             117          154
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                   0.141        0.141
## BW bias (b)                   0.206        0.206
## rho (h/b)                     0.685        0.685
## Unique Obs.                     488          512
##
## First-stage estimates.
```

rdrobust

- We can even have it automatically make plots of our RDD! Same syntax

```
rdplot(df$Y, df$X, c = .5)
```