Big Data and Economics

Regression Discontinuity by Kyle Coombs

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

Table of contents

Prologue

Prologue

- We've just finished covering difference-in-differences, which is one way of estimating a causal effect even if you can't measure and control for everything you need to control for
- DID is *very* widely applicable, but it relies on some pretty strong assumptions like parallel trends
- We want to have some other designs in mind for how we can estimate effects in these settings that might be a little easier to swallow!

Regression Discontinuity

Regression Discontinuity

- Regression discontinuity design (RDD) is currently the darling of the econometric world for estimating causal effects without running an experiment
- It doesn't apply everywhere, but when it does, it's very easy to buy the identification assumptions
- Not that it doesn't have its own issues, of course, but it's pretty good!

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 you get above a certain test score they let you into a "gifted and talented" program
- 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 Medicaid, if you're 64.99 years old you don't

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

Regression Discontinuity

- But wait, hold on, if treatment is driven by running variables, won't we have endogeneity going through the running variables?? Yes!
- And we can't just control for RunningVar because that's where all the variation in treatment comes from. Uh oh!

Regression Discontinuity

- The key here is realizing that the running variable affects treatment *only when you go across the cutoff*
- So really the diagram looks like this!

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 just the variation around the treatment, narrowing the range of the running variable we use so sharply that it's basically controlled for. Then the effect of cutoff on treatment is like an experiment!

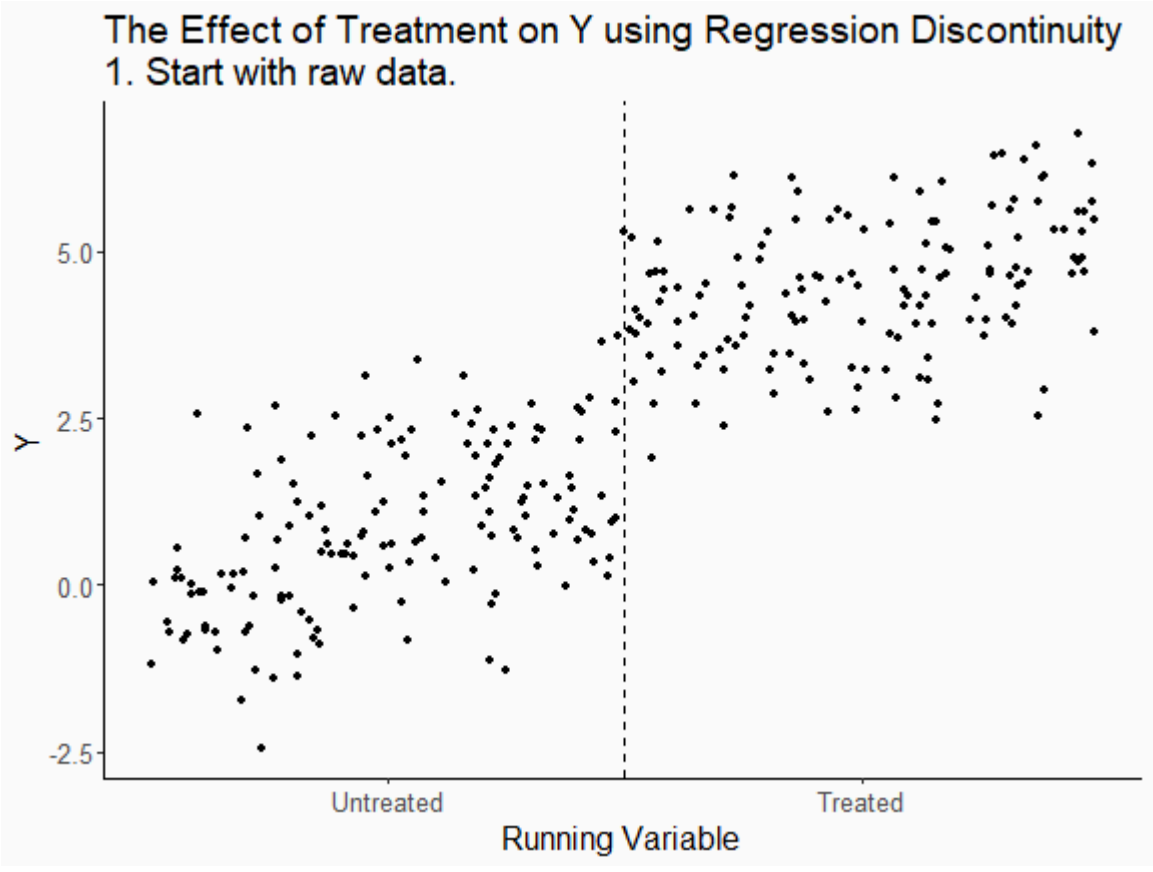
Regression Discontinuity

- Basically, the idea is that *right around the cutoff*, treatment is randomly assigned
- If you have a test score of 89.9 (not high enough for gifted-and-talented), you're basically the same as someone who has a test score of 90.0 (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 still won't know the effect of being put in gifted-and-talented for someone who gets a 30)

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



Concept Checks

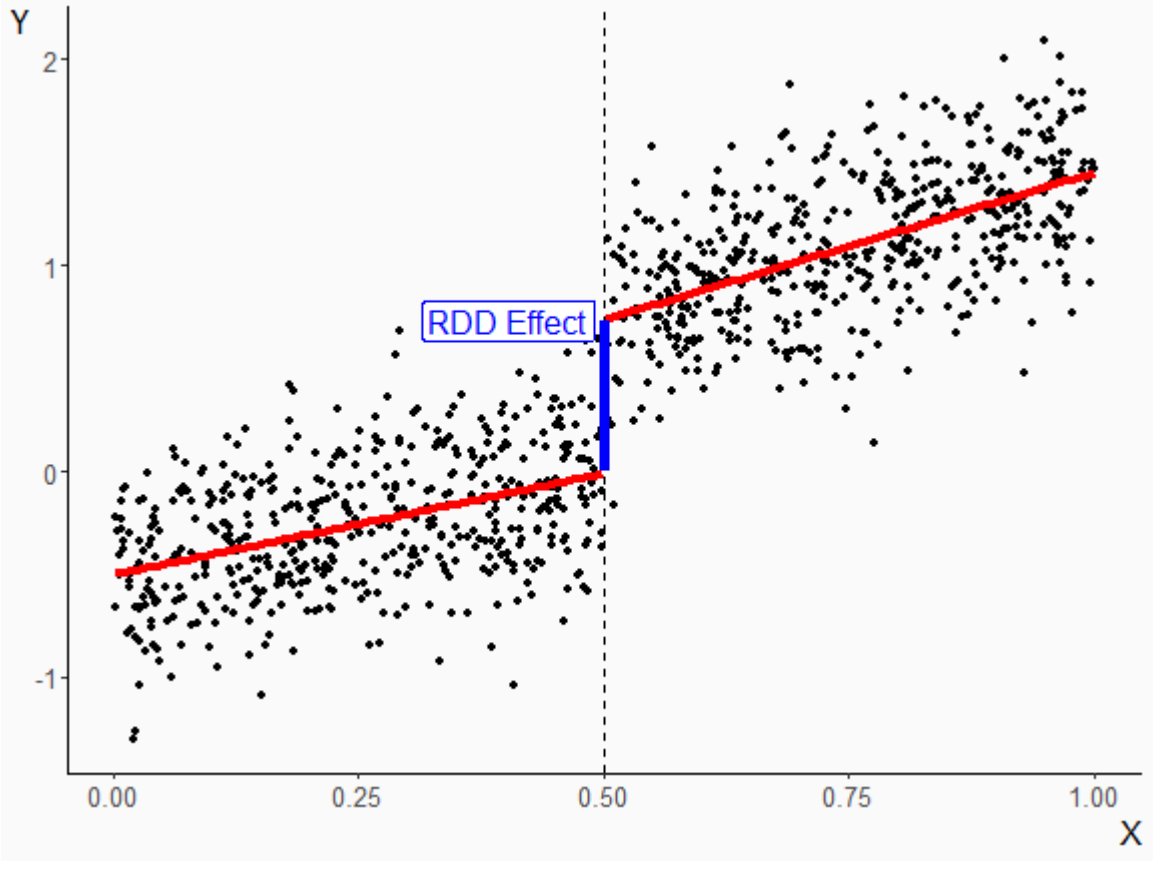
- Why is it important that we look as narrowly as possible around the cutoff? What does this get us over comparing the entire treated and untreated groups?
- Can you think of an example of a treatment that is assigned at least partially on a cutoff?
- Why can't we just control for the running variable as we normally would to solve the endogeneity problem?

Fitting Lines in RDD

- Looking purely just at the cutoff and making no use of the space *away* from the cutoff throws out a lot of useful information
- We know that the running variable is related to outcome, so we can probably improve our *prediction* of what the value on either side of the cutoff should be if we *use data away from the cutoff to help with prediction* than if we *just use data near the cutoff*, which is what that animation does
- We can do this with good ol' OLS.
- The bin plot we did can help us pick a functional form for the slope

Fitting Lines in RDD

- To be clear, producing the line(s) below is our goal. How can we do it?
- The true model I've made is an RDD effect of .7, with a slope of 1 to the left of the cutoff and a slope of 1.5 to the right



Regression in RDD

- First, we need to *transform our data*
- We need a "Treated" variable that's `TRUE` when treatment is applied - above or below the 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 ⩾ .5,
         X_centered = X - .5)
```

Varying Slope

- Typically, you will want to let the slope vary to either side
- In effect, we are fitting an entirely different regression line on each side of the cutoff
- We can do this by interacting both slope and intercept with $treated$!
- Coefficient on Treated is how the intercept jumps - that's our RDD effect. Coefficient on the interaction is how the slope changes

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

```
etable(feols(Y ~ treated*X_centered, data = df))
```

```
##                       feols(Y ~ treate..
## Dependent Var.:                        Y
##
## Constant                 -0.0111 (0.0260)
## Treated                 0.7467*** (0.0376)
## X_centered              0.9825*** (0.0907)
## Treated x X_centered    0.4470*** (0.1296)
## ______________________ __________________
## S.E. type                             IID
## Observations                        1,000
## R2                                0.84769
## Adj. R2                           0.84723
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Varying Slope

(as an aside, sometimes the effect of interest is the interaction term - the change in slope! This answers the question "does the effect of X on Y change at the cutoff? This is called a "regression kink" design. We won't go more into it here, but it is out there!)

Polynomial Terms

- We don't need to stop at linear slopes!
- Just like we brought in our knowledge of binary and interaction terms to understand the linear slope change, we can bring in polynomials too. Add a square maybe!
- Don't get too wild with cubes, quartics, etc. - polynomials tend to be at their "weirdest" near the edges, and we don't want super-weird predictions right at the cutoff. It could give us a mistaken result!
- A square term should be enough

Polynomial Terms

- How do we do this? Interactions again. Take *any* regression equation...

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$$

- And just center the X (let's call it XC, add on a set of the same terms multiplied by $Treated$ (don't forget $Treated$ by itself - that's $Treated$ times the interaction!))

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

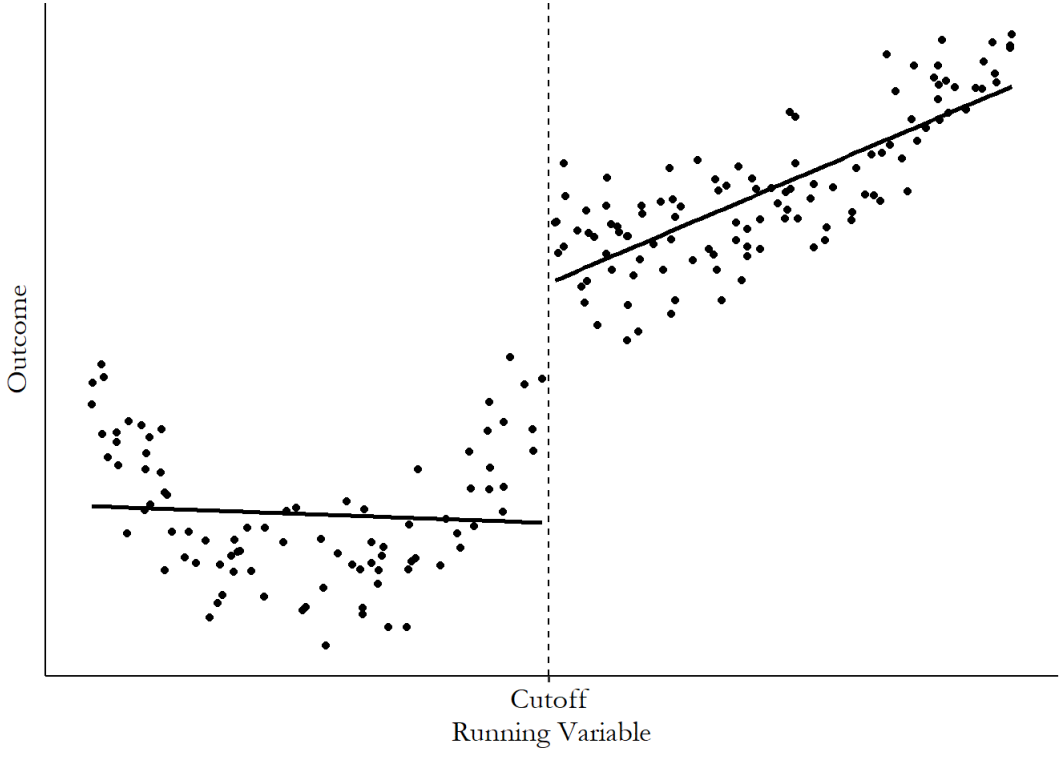
- The coefficient on $Treated$ remains our "jump at the cutoff" - our RDD estimate!

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

```
##                              feols(Y ~ X_cent..
## Dependent Var.:                               Y
##
## Constant                        -0.0340 (0.0385)
## X_centered                       0.6990. (0.3641)
## treatedTRUE                    0.7677*** (0.0577)
## X_centered square                -0.5722 (0.7117)
## X_centered x treatedTRUE          0.7509 (0.5359)
## treatedTRUE x I(X_centered^2)      0.5319 (1.034)
## ____________________________ _________________
## S.E. type                                   IID
## Observations                              1,000
```

Fitting Quadratic Lines in RDD

- Sometimes it can be hard to tell if a quadratic (or higher-order) term is really necessary
- Visualizations can help!



A linear slope makes the jump much bigger than it really is! (From the Effect Chapter 20)

Concept Checks

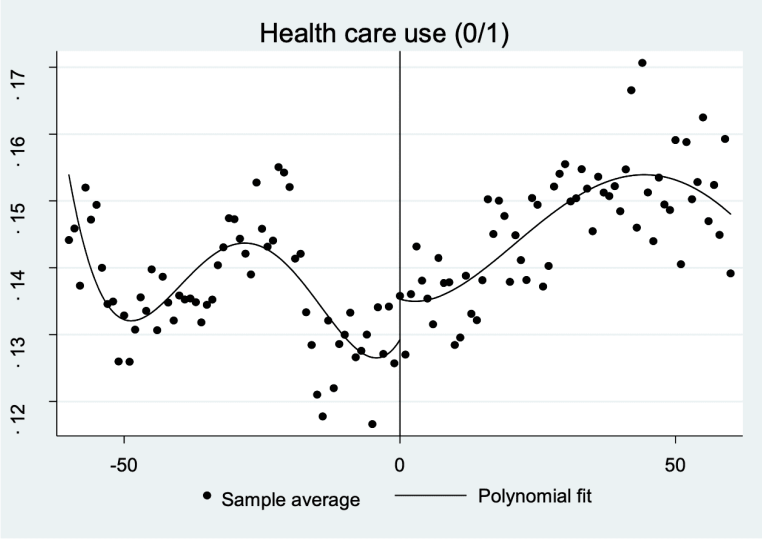
- Would the coefficient on $Treated$ still be the regression discontinuity effect estimate if we hadn't centered X? Why or why not?
- Why might we want to use a polynomial term in our RDD model?
- What relationship are we assuming between the outcome variable and the running variable if we choose not to include $XCentered$ in our model at all (i.e. a "zero-order polynomial")

Careful with higher order polynomials

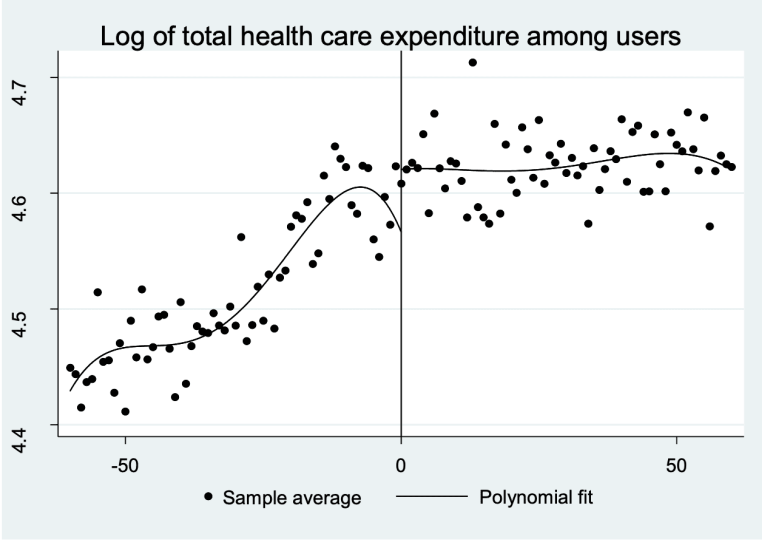
- Sometimes higher order polynoials can be a little too flexible
- They make it look like there is an effect where none exist
- "Overfitting" where your model too flexibly follows the data points can lie to you!
- Read Andrew Gelman's blog for more info

Does voting make you sick?

Or did the researchers just overfit their model?



Health care use (0/1)



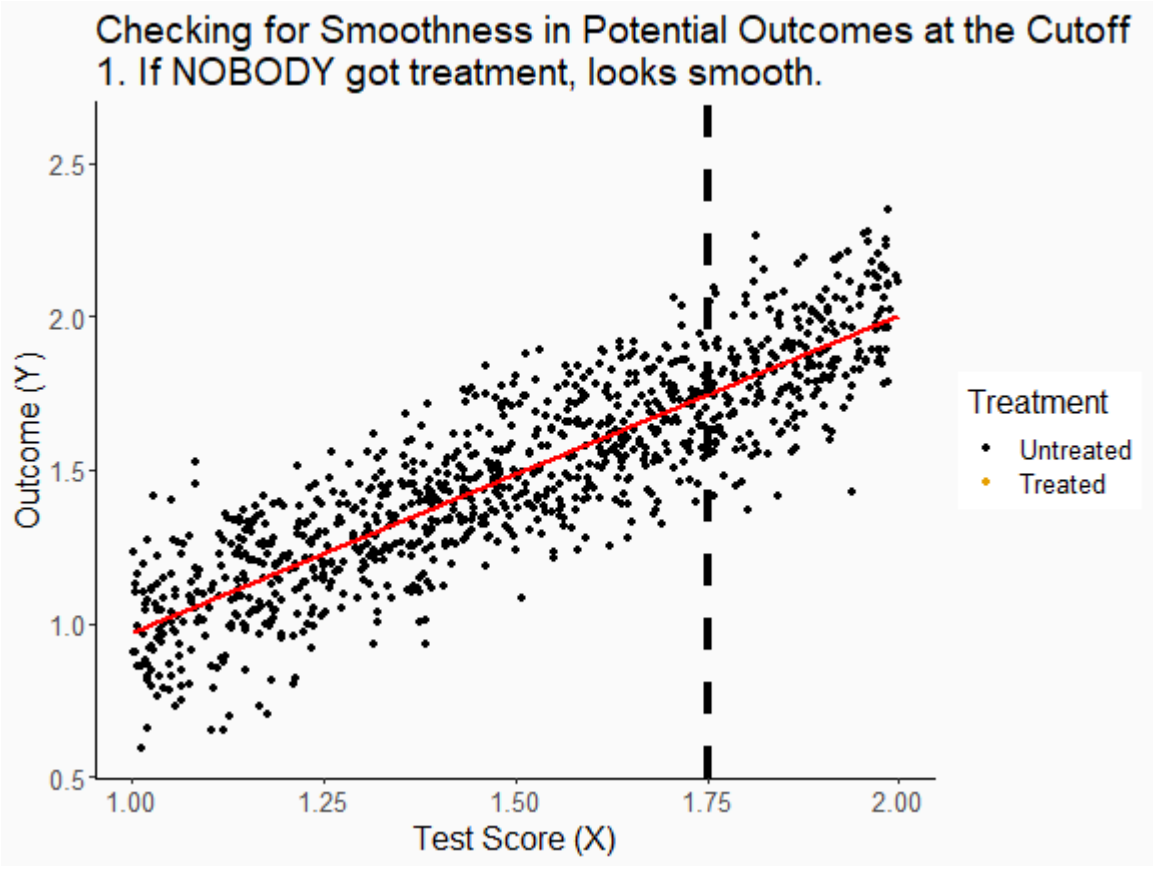
Log of total health care expenditure among users

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

Assumptions

- We knew there must be some assumptions lurking around here
- Some are more obvious (we should be using 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 having an income below 150% of the poverty line gets you access to food stamps AND to job training, then we can't use that cutoff to get 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

Other Difficulties

More assumptions, limitations, and diagnostics!

- Windows
- Granular running variables
- Manipulated running variables
- Fuzzy regression discontinuity

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(m1,m2,m3,m4,m5, keep = 'treatedTRUE')
```

```
##                                      m1                 m2
## Dependent Var.:                       Y                  Y
##
## treatedTRUE              0.7467*** (0.0376) 0.7723*** (0.0566)
## treatedTRUE x X_centered 0.4470*** (0.1296)   0.6671. (0.4022)
## _______________________ __________________ __________________
## S.E. type                              IID                IID
## Observations                         1,000                492
## R2                                 0.84769            0.74687
## Adj. R2                            0.84723            0.74531
##
##                                      m3                 m4                m5
## Dependent Var.:                       Y                  Y                 Y
##
## treatedTRUE              0.7086*** (0.0900) 0.6104*** (0.1467) 0.5585 (0.4269)
## treatedTRUE x X_centered     -1.307 (1.482)      6.280 (4.789)   41.21 (72.21)
##
```

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 don't know you're 89 or 90, I just know you're "80-89" or "90-100"
- A lot less believable that the only difference between these groups is random chance
- Plenty of other things change between 80 and 100! That's not "smooth at the cutoff"

Granular Running Variable

- Not a whole lot we can do about this
- 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, think "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

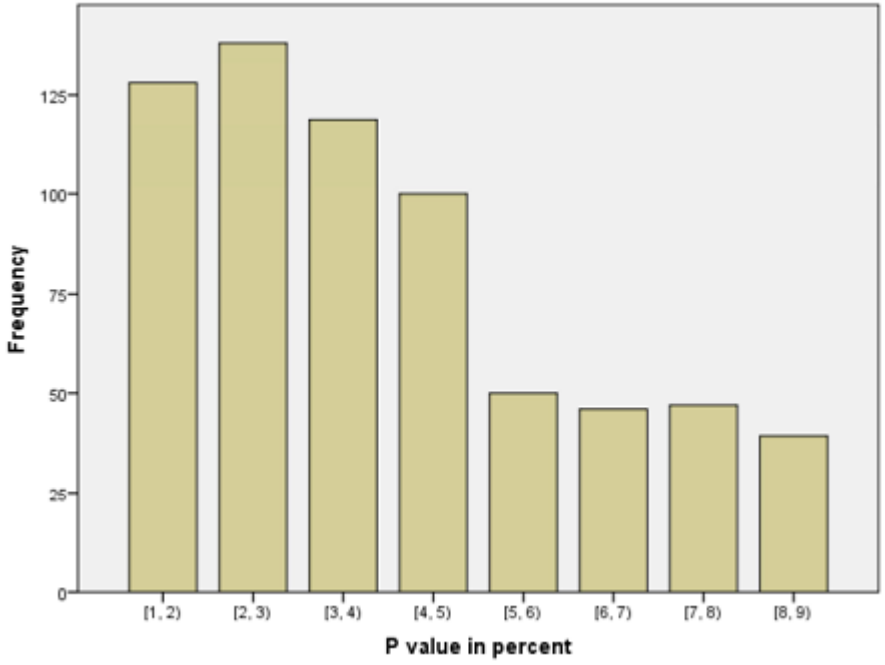
- Ok, now let's go back to our continuous running variables
- What if the running variable is *manipulated*?
- Imagine you're a teacher grading the gifted-and-talented exam. You see someone with an 89 and think "aww, they're so close! I'll just give them an extra point..."
- Or, if you live just barely on one side of a time zone line, but decide to move to the other side because you prefer waking up later
- Suddenly, that treatment is a lot less randomly assigned around the cutoff!

Looking for Lumping

- 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

Looking for Lumping

- Here's an example from the real world in medical research - statistically, p-values *should* be uniformly distributed
- But it's hard to get insignificant results published in some journals. So people might "p-hack" until they find some form of analysis that's significant, and also we have heavy selection into publication based on $p < .05$. Can't use that cutoff for an RDD!



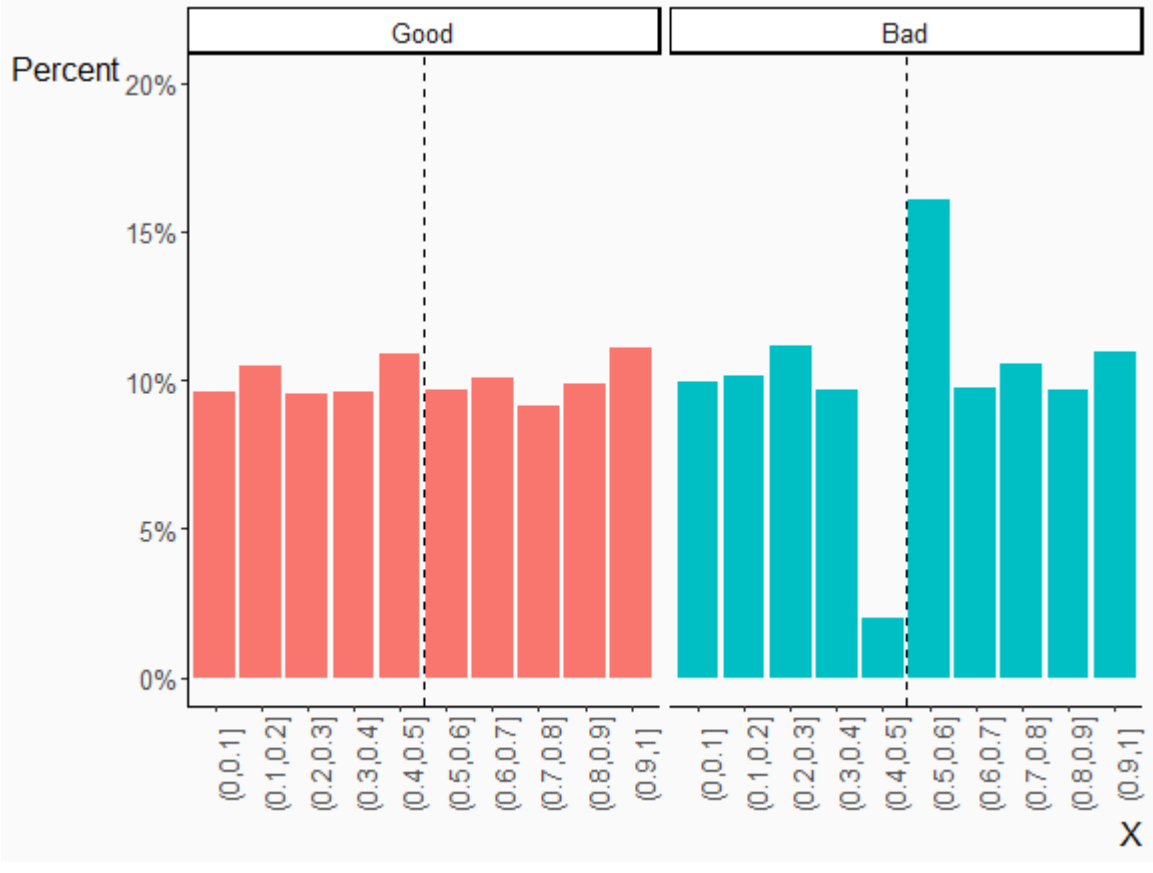
Looking for Lumping

- How can we look for this stuff?
- We can look graphically by just checking for a jump at the cutoff in *number of observations* after binning

```
df_bin_count ← df %>%
  # Select breaks so that one of hte breakpoints is the cutoff
  mutate(X_bins = cut(X, breaks = 0:10/10)) %>%
  group_by(X_bins) %>%
  count()
```

Looking for Lumping

- The first one looks pretty good. We have one that looks not-so-good on the right



Looking for Lumping

- Another thing we can do is do a "placebo test"
- 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)

Concept Checks

- Why does using a narrow window make the effect estimate noisier?
- Intuitively, why would we be skeptical that a regression discontinuity run on a very granular running variable is valid?

Fuzzy Regression Discontinuity

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, maybe about 10% of kids with too-low test scores get into gifted-and-talented, and 80% of kids with high-enough scores do
- For whatever reason!
- 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 .7 to get the true effect

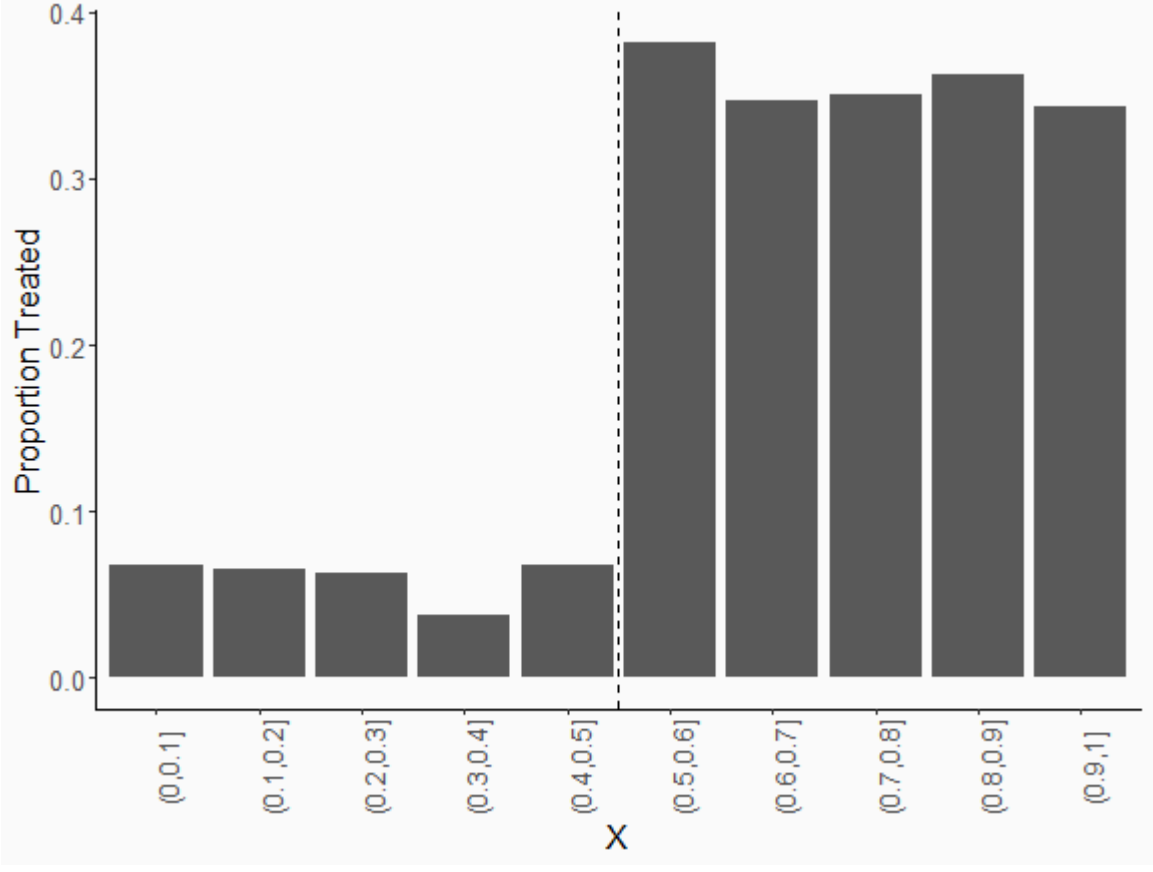
Fuzzy Regression Discontinuity

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 "percentage 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','treatmentT
```

```
##                        predict_treatment      without_fuzzy          fuzzy_rdd
## Dependent Var.:                treatment                  Y                  Y
##
## Constant                0.0605. (0.0354) 0.4263*** (0.0359) 0.4429*** (0.1149)
## X_center                 0.0044 (0.1215)  0.4099** (0.1250)    0.5802 (0.4089)
## Above Cut             0.3053*** (0.0484)
## X_center x Above Cut   -0.0392 (0.1687)
## Treatment                                0.3414*** (0.0937)    0.4458 (0.4158)
## X_center x Treatment                        0.2199 (0.3315)    -0.8367 (1.610)
## ____________________ _________________ _________________ _________________
## S.E. type                         IID               IID               IID
## Observations                    1,000             1,000             1,000
## R2                            0.13289           0.04967           0.03968
## Adj. R2                       0.13028           0.04681           0.03679
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Concept Checks

- How do we know that, if our treatment variable is fuzzily assigned, we will *underestimate* the effect if we just run a regular RDD, rather than overestimate it?

Regression Discontinuity in R

- 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!
- We'll use packages and formulas that do things like "picking a bandwidth (window)" for us in a smart way, or not relying so strongly on linearity
- The **rdrobust** package does just that!
- Let's look at `help(rdrobust, packge = 'rdrobust')`

Regression Discontinuity in R

- 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 us with `p`
- (it applies the polynomials more locally than our linear OLS models do - a bit more flexible without weird corner predictions)
- It will also pick a window for us with `h`
- Plenty of other options
- Including a `fuzzy` option to specify actual treatment outside of the running variable/cutoff combo

rdrobust

```
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.            120         156
## Order est. (p)                   1           1
## Order bias  (q)                  2           2
## BW est. (h)                  0.142       0.142
## BW bias (b)                  0.213       0.213
## rho (h/b)                    0.668       0.668
## Unique Obs.                    488         512
##
## ===============================================================================
```

rdrobust

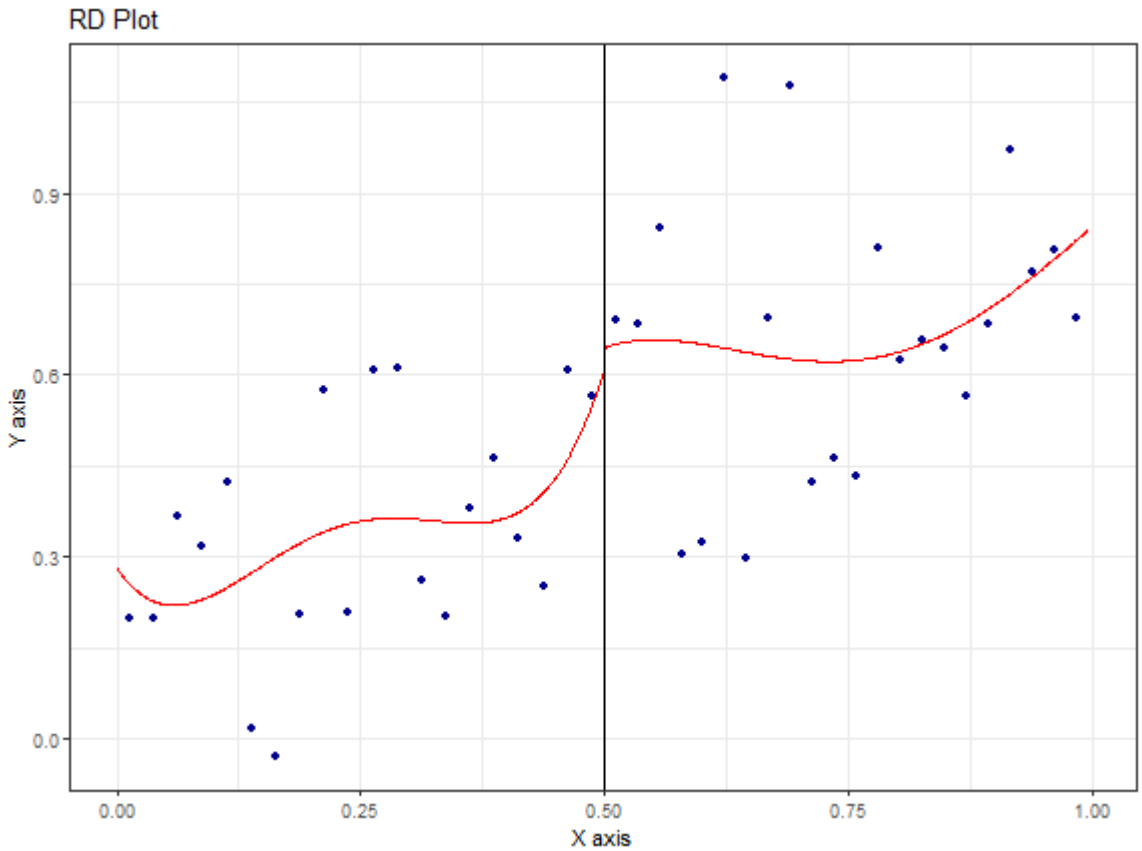
```
summary(rdrobust(df$Y, df$X, c = .5, fuzzy = df$treatment))
```

```
## 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.            119          156
## 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)                    0.687        0.687
## 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)
```



That's it!

- That's what we have for RDD
- Go explore the regression discontinuity activity on class sizes

Next lecture: SQL