Regression II
Vanderbilt University
Prof. Sean P. Corcoran

December 1, 2022

Lecture 6 In-Class Example *Solutions*

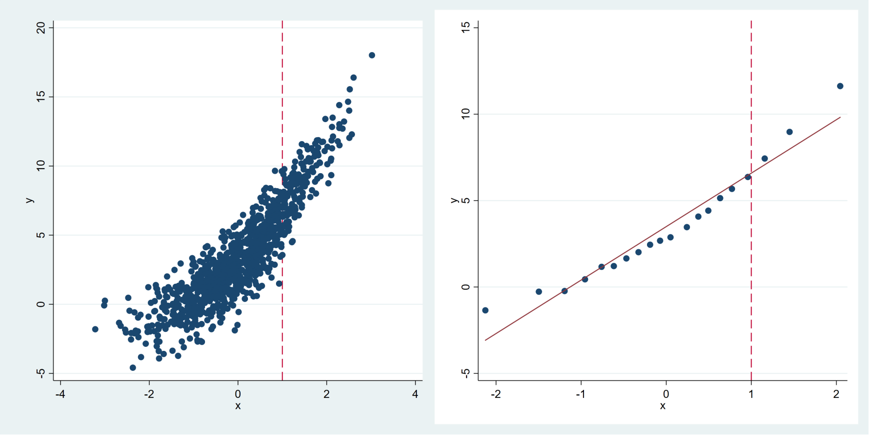
Example 1. This example will generate data with a known discontinuity in y at a threshold level of x, and then estimate a RD model. It also illustrates the McCrary test. (Adapted from Ballou).

1. First produce simulated data using the syntax below. Notice that x is the running variable. What is the functional relationship between the outcome y and the running variable? What is the cut score? What is the treatment effect? Is this a strict or fuzzy regression discontinuity?

```
clear
set seed 1234
drawnorm x w e u, n(1000)
gen y = 3 + 3*x + .5*x^2 + w + u
gen t = (x > 1)
replace y = y + .5*t
```

   There is a quadratic relationship between the running variable x and the outcome y, as seen in line 4. The cut score is x=1 (line 5). The treatment effect is 0.5 (line 6): cases where t=1 have a value of y that is 0.5 higher than what it would be otherwise. This is a *strict* regression discontinuity since all cases where x $\leq$ 1 are untreated, and all cases where x $>$ 1 are treated.

2. Produce a scatterplot of y against x. Do you see evidence of a discontinuity? Try using `binscatter`. Do you see a discontinuity?



   Figures shown above. It is difficult to see any discontinuity in the scatter plot. The discontinuity in the binned scatter plot is evident, but slight.

3. Now estimate a parametric RD model assuming a linear relationship with the running variable (with the same slope on either side of the cut score). How close does it get to estimating the true treatment effect? Provide an intuitive explanation for your finding.

   Results below. The treatment effect estimate is 2.1, quite a bit larger than the known effect of 0.5. The reason is that the functional relationship between *y* and *x* is misspecified. It is known to be quadratic, and we fit a linear model. The increasing slope of the relationship between *y* and *x* is mistakenly subsumed into the treatment effect.

```
. // Estimate RD model assuming linear relationship between y and x
. reg y t x

      Source |       SS           df       MS      Number of obs   =     1,000
-------------+----------------------------------   F(2, 997)       =   2240.02
       Model |  10240.5295         2  5120.26473   Prob > F        =    0.0000
    Residual |   2278.9563       997  2.28581375   R-squared       =    0.8180
-------------+----------------------------------   Adj R-squared   =    0.8176
       Total |  12519.4858       999  12.5320178   Root MSE        =    1.5119

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           t |   2.070868   .1711322    12.10   0.000     1.735047    2.406689
           x |   2.598992   .0622848    41.73   0.000     2.476767    2.721216
       _cons |   3.153382   .0554548    56.86   0.000      3.04456    3.262203
------------------------------------------------------------------------------
```

4. Repeat but using a quadratic function of the running variable. Does this help?

   Indeed it does. The treatment effect estimate is now 0.447, much closer to the known effect of 0.5. (Note 0.5 is within the 95% confidence interval).

```
. // Estimate RD model using a quadratic
. reg y t c.x##c.x

      Source |       SS           df       MS      Number of obs   =     1,000
-------------+----------------------------------   F(3, 996)       =   1761.62
       Model |  10534.1874         3   3511.3958   Prob > F        =    0.0000
    Residual |  1985.29839       996  1.99327147   R-squared       =    0.8414
-------------+----------------------------------   Adj R-squared   =    0.8409
       Total |  12519.4858       999  12.5320178   Root MSE        =    1.4118

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           t |   .4473678   .2083961     2.15   0.032     .0384221    .8563135
           x |   3.027445   .0680363    44.50   0.000     2.893934    3.160956
             |
     c.x#c.x |   .5029089   .0414335    12.14   0.000     .4216019    .5842158
             |
       _cons |   2.899438   .0558514    51.91   0.000     2.789838    3.009038
------------------------------------------------------------------------------
```

5. Obtain some nonparametric estimates using the Stata command `rd`. `rd` estimates treatment effects using local linear or kernel regression models on both sides of the cut score. Note the option `z0( )` provides Stata the known cutoff value. The option `strineq` (strict inequality) tells Stata that treatment is assigned *above* the cut score—observations *at* the cut score are not treated. (The default assumes treatment begins *at* the cut score). The option `bwidth( )` allows you to select a bandwidth for local linear regression. There are also lots of options for `rd` that produce graphs.

   ```
   rd y x, z0(1) strineq bwidth(.4)
   ```

   The estimate `lwald` (Local Wald) is your baseline result. The additional estimates `lwald50` and `lwald200` are robustness checks at other bandwidths (50% and 200% of your specified bandwidth).

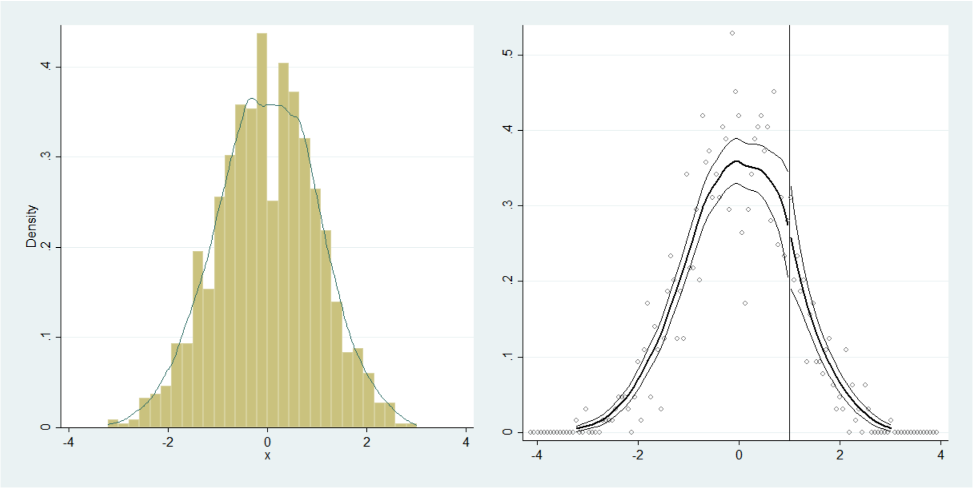
   ```
   . rd y x, z0(1) strineq bwidth(.4)
   Two variables specified; treatment is
   assumed to jump from zero to one at Z=1.

    Assignment variable Z is x
    Treatment variable X_T unspecified
    Outcome variable y is y

   Estimating for bandwidth .4
   Estimating for bandwidth .2
   Estimating for bandwidth .8
   ------------------------------------------------------------------------------
              y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
   -------------+----------------------------------------------------------------
          lwald |   .4172043   .4876325     0.86   0.392    -.5385378    1.372946
        lwald50 |    .550193   .7137631     0.77   0.441     -.848757    1.949143
       lwald200 |   .2259569   .3320834     0.68   0.496    -.4249146    .8768285
   ------------------------------------------------------------------------------
   ```

6. Check for manipulation in the running variable in two ways: by inspection using `histogram`, and using McCrary's `DCdensity` command. The option `breakpoint` tells Stata the known cutoff value. What does the latter test conclude?

   ```
   histogram x, kdens
   DCdensity x, breakpoint(1) gen(Xj Yj r0 fhat se_fhat)
   ```



The histogram looks a bit jumpy around x=0, but not around the actual cutpoint of x=1. The McCrary test does not detect any manipulation around the cutpoint of 1. The test statistic is 0.021 with a standard error of 0.202. We cannot reject the null hypothesis of no manipulation.

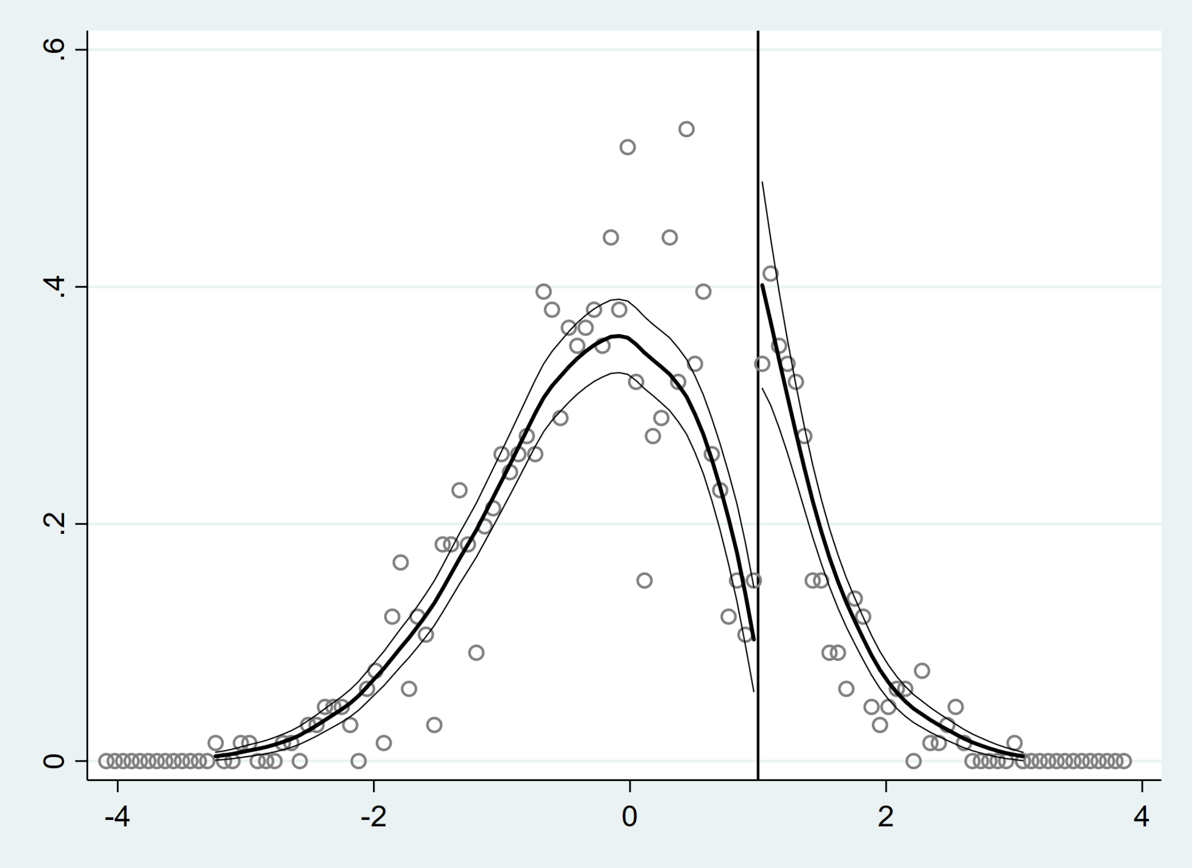
```
. DCdensity x, breakpoint(1) gen(Xj Yj r0 fhat se_fhat) graphname(mccrary.png)
Using default bin size calculation, bin size = .064375706
Using default bandwidth calculation, bandwidth = .880663554

Discontinuity estimate (log difference in height): .020945661
                                                   (.202476704)
Performing LLR smoothing.
98 iterations will be performed
.........
Exporting graph as mccrary.png
```

7. Now modify the data a bit to introduce manipulation in x. Try the syntax below and explain in words what the first line is doing. Then, re-do the McCrary test.

```
replace x = x + .4 if x < 1 & x > .65 & e > 0
drop Xj Yj r0 fhat se_fhat
DCdensity x, breakpoint(1) gen(Xj Yj r0 fhat se_fhat)
```

The code above is manipulating values of x between 0.65 and 1, giving them an additional 0.4 to put them over the threshold. In this case the McCrary test is clear in showing the manipulation. The test statistic is 1.65 with a standard error of 0.295, so we can reject the null hypothesis of no manipulation.



```
. DCdensity x, breakpoint(1) gen(Xj Yj r0 fhat se_fhat) graphname(mccrary2.png)
Using default bin size calculation, bin size = .065660873
Using default bandwidth calculation, bandwidth = .814820157

Discontinuity estimate (log difference in height): 1.65136255
                                                   (.295439284)
Performing LLR smoothing.
97 iterations will be performed
.........
Exporting graph as mccrary2.png
```

8. Now that we know there is manipulation, try estimating the parametric and non-parametric RD models in (3) and (5). How do the estimates compare?

   For this step it is worth thinking about how *y* should be changed, if at all. If we assume that cases manipulated into the treatment group get the same effect from being exposed to the treatment, then we can add the 0.5 to these cases (as below). One could also leave the original *y*'s intact, but this would be assuming no treatment effect for these manipulated cases. The OLS estimate of the treatment effect is too large, at 1.0, while the RD estimate is statistically insignificant and negative.

```
gen manipulated=(x>0.65 & x<1 & e>0)
replace x = x + .4 if manipulated==1
// For half of the observations in the interval (.65, 1), we increase x by .4,
// which is enough to get them into the eligible group
replace y = y+0.5 if manipulated==1

. reg y t c.x##c.x

      Source |       SS           df       MS      Number of obs   =     1,000
-------------+----------------------------------   F(3, 996)       =   1711.51
       Model |  10581.1583         3  3527.05276   Prob > F        =    0.0000
    Residual |  2052.54309       996  2.06078623   R-squared       =    0.8375
-------------+----------------------------------   Adj R-squared   =    0.8370
       Total |  12633.7014       999  12.6463477   Root MSE        =    1.4355

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           t |   1.017063   .1949417     5.22   0.000     .6345197    1.399607
           x |   2.825773   .0627783    45.01   0.000      2.70258    2.948966
             |
     c.x#c.x |    .416721   .0396139    10.52   0.000     .3389846    .4944574
             |
       _cons |   2.844725   .0575166    49.46   0.000     2.731858    2.957593
------------------------------------------------------------------------------

. // Non-parameteric estimates using RD (note manipulation is present)
. rd y x, z0(1) strineq bwidth(.4)
Two variables specified; treatment is
assumed to jump from zero to one at Z=1.

 Assignment variable Z is x
 Treatment variable X_T unspecified
 Outcome variable y is y

Estimating for bandwidth .4
Estimating for bandwidth .2
Estimating for bandwidth .8
------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       lwald |  -.3772306   .4564508    -0.83   0.409    -1.271858    .5173965
     lwald50 |   .1260595   .6495859     0.19   0.846    -1.147105    1.399224
    lwald200 |  -.7943974   .3468673    -2.29   0.022    -1.474245   -.1145499
------------------------------------------------------------------------------
```

Example 2. This example, based on an example created by Celeste Carruthers, also uses simulated data to estimate the effect of participation in a gifted and talented (G&T) program.

1. Generate 10,000 student observations. The data will include a measure of students' "true ability," $trueability \sim N(50, 4)$, and their 3rd grade test score, which is a noisy measure of their true ability $grade3test = trueability + u$ where $u \sim N(0, 1)$. To add a bit of realism, we will round test scores to the nearest 0.25 to create a discrete scale.

```
clear
set seed 195423
set obs 10000
gen id=_n
gen trueability = 50 + 4*rnormal()
gen grade3test  = trueability + rnormal()
replace grade3test = round(grade3test, 0.25)
```

2. Suppose 3rd graders scoring at or above 56 are eligible for the G&T program. Create a treatment assignment variable re-centered at zero, and a "gap" variable that contains the distance between the running variable and the cut score.

```
gen above56 = (grade3test>=56)
gen gap = grade3test-56
```

3. Assume perfect compliance. Create an indicator variable for G&T participation *inGT*, that equals one for treated students and zero otherwise. What fraction of students participate in G&T? Try estimating a regression for G&T participation where *inGT* is regressed on the *gap* and the threshold indicator *above56*. What happens?

   7.59% of students participated in G&T. When you regress the treatment *inGT* on the *gap* and threshold indicator *above56*, Stata cannot produce estimates. This is because *above56* perfectly determines the outcome *inGT*. (It is a strict, not fuzzy, discontinuity).

```
. gen inGT=(above56==1)

. sum inGT

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        inGT |     10,000       .0759    .2648513          0          1

. reg inGT gap above56

      Source |       SS           df       MS      Number of obs   =    10,000
-------------+----------------------------------   F(2, 9997)      =         .
       Model |   701.3919         2   350.69595   Prob > F        =         .
    Residual |          0     9,997           0   R-squared       =    1.0000
-------------+----------------------------------   Adj R-squared   =    1.0000
       Total |   701.3919     9,999  .070146205   Root MSE        =         0
```

```
------------------------------------------------------------------------------
        inGT |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         gap |          0  (omitted)
     above56 |          1          .        .       .            .           .
       _cons |          0  (omitted)
------------------------------------------------------------------------------
```

4. Create the outcome variable (grade 4 test score) such that G&T participation has a positive treatment effect of 3 points. Assume that test growth from 3rd to 4th grade would be 5 points in the absence of treatment. As before, we will include some random noise, and round the test scale to the nearest 0.25.

   ```
   gen grade4test = round(trueability + 5 + rnormal() + (3*inGT), 0.25)
   ```

5. Estimate a parametric RD model assuming a linear relationship with the running variable. First do this assuming the same slope on either side of the cut score. Then, allow the slope to vary on either side. Is there evidence of a change in slope beyond the cut score? Does this finding make sense to you?

   Results below. There is no evidence of a change in slope beyond the cut score. This makes sense since we know the original data generating process, in which *grade4test* is linear in the running variable.

   ```
   . reg grade4test gap inGT

         Source |       SS           df       MS      Number of obs   =    10,000
   -------------+----------------------------------   F(2, 9997)      =  48332.69
          Model |  187161.691         2  93580.8457   Prob > F        =    0.0000
       Residual |  19356.0026     9,997  1.93618111   R-squared       =    0.9063
   -------------+----------------------------------   Adj R-squared   =    0.9063
          Total |  206517.694     9,999  20.6538348   Root MSE        =    1.3915

   ------------------------------------------------------------------------------
     grade4test |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
   -------------+----------------------------------------------------------------
            gap |   .9392992   .0040454   232.19   0.000     .9313694     .947229
           inGT |   3.029696   .0624409    48.52   0.000       2.9073    3.152093
          _cons |   60.61507   .0304137  1993.02   0.000     60.55545    60.67468
   ------------------------------------------------------------------------------

   . reg grade4test c.gap##i.inGT

         Source |       SS           df       MS      Number of obs   =    10,000
   -------------+----------------------------------   F(3, 9996)      =  32218.65
          Model |  187161.733         3  62387.2445   Prob > F        =    0.0000
       Residual |  19355.9606     9,996   1.9363706   R-squared       =    0.9063
   -------------+----------------------------------   Adj R-squared   =    0.9062
   ```

```
       Total |  206517.694     9,999  20.6538348   Root MSE        =    1.3915

------------------------------------------------------------------------------
  grade4test |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         gap |    .939221   .0040802   230.19   0.000     .9312229    .9472191
      1.inGT |    3.02236   .0798731    37.84   0.000     2.865792    3.178927
             |
  inGT#c.gap |
          1  |   .0046223   .0313781     0.15   0.883    -.0568851    .0661296
             |
       _cons |   60.61455   .0306168  1979.78   0.000     60.55453    60.67456
------------------------------------------------------------------------------
```

6. Drop the existing *inGT* and *grade4test* variables and re-create them assuming a "fuzzy" GT treatment that increases smoothly with grade 3 test scores and then jumps discontinuously (by about 70 percentage points) at the cut score. This might arise if G&T placement is dependent on the grade 3 test score as well as other factors (e.g., parental input, teacher recommendation). Use the syntax below. What fraction of students are treated, overall? Below the cutoff? Above?

   ```
   drop inGT grade4test
   gen inGT=round(-.77+.007*grade3test+0.7*above56+runiform())
   gen grade4test = round(trueability + 5 + rnormal() + (3*inGT), 0.25)
   ```

   13.3% of students are treated, overall. Below the cutoff, 7.6% of students are selected for the G&T program. Above the cutoff, 83.4% are selected.

7. As in (3), estimate a regression for G&T placement where *inGT* is regressed on the *gap* and the threshold indicator *above56*. Interpret your results. (Try estimating this in two ways: first assuming the slope is constant on either side of the cutoff, and then allowing the slope to change). With a discrete running variable, it is usually advisable that you adjust the standard errors for clustering by that variable. For later reference, use the `predict` command to get predicted values for treatment (placement in G&T) given the 3rd grade score. Call this variable *hat_trt*.

   Results below. The first regression tells us that the probability of selection for G&T increases with *gap* (the student's score minus 56). There is also a discontinuous jump in the probability of selection at the cut score, of 70.3 percentage points.

   ```
   . reg inGT gap above56, cluster(grade3test)

   Linear regression                               Number of obs     =     10,000
                                                   F(2, 112)         =    2089.23
                                                   Prob > F          =     0.0000
   ```

```
                                                R-squared         =     0.3533
                                                Root MSE          =     .27348

                             (Std. Err. adjusted for 113 clusters in grade3test)
------------------------------------------------------------------------------
             |               Robust
        inGT |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         gap |   .0066524   .0008072     8.24   0.000     .0050531    .0082516
     above56 |   .7026519   .0139781    50.27   0.000      .674956    .7303478
       _cons |   .1198432   .0067453    17.77   0.000     .1064782    .1332082
------------------------------------------------------------------------------

. reg inGT c.gap##i.above56, cluster(grade3test)

Linear regression                               Number of obs     =     10,000
                                                F(3, 112)         =    1492.34
                                                Prob > F          =     0.0000
                                                R-squared         =     0.3533
                                                Root MSE          =     .27348

                              (Std. Err. adjusted for 113 clusters in grade3test)
-------------------------------------------------------------------------------
              |               Robust
         inGT |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
--------------+----------------------------------------------------------------
          gap |   .0065711   .0008155     8.06   0.000     .0049553    .0081868
    1.above56 |    .695023   .0151054    46.01   0.000     .6650936    .7249523
              |
above56#c.gap |
           1  |   .0048064   .0058532     0.82   0.413     -.006791    .0164038
              |
        _cons |   .1193058   .0068076    17.53   0.000     .1058173    .1327943
-------------------------------------------------------------------------------

. predict hat_trt
(option xb assumed; fitted values)
```

8. Re-estimate the parametric RD model assuming a linear relationship with the running variable. Assume the discontinuity is "sharp," even though we know otherwise. Again, cluster the standard errors by the grade 3 score. How does the estimated treatment effect differ from the known treatment effect of 3 points? Repeat using the non-parametric `rd`. How does the point estimate compare?

   Results below. The estimated treatment effect is smaller, at 2.0 versus the known 3 points. This is not surprising: when the discontinuity is fuzzy, the difference in outcomes around the cutoff will be smaller, since not all above the cut score were treated, and some of those below the cut score were treated.

```
. reg grade4test c.gap##i.above56, cluster(grade3test)

Linear regression                               Number of obs     =     10,000
                                                F(3, 112)         =   30362.39
                                                Prob > F          =     0.0000
                                                R-squared         =     0.8752
                                                Root MSE          =     1.6064

                             (Std. Err. adjusted for 113 clusters in grade3test)
--------------------------------------------------------------------------------
               |               Robust
    grade4test |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------------+----------------------------------------------------------------
           gap |   .9611196   .0041402   232.14   0.000     .9529163    .9693228
     1.above56 |   2.010342   .0880435    22.83   0.000     1.835895    2.184789
               |
above56#c.gap |
             1 |   .0359043   .0311577     1.15   0.252    -.0258307    .0976393
               |
         _cons |   61.00139   .0346558  1760.21   0.000     60.93272    61.07006
--------------------------------------------------------------------------------

. // non-parametric version - local linear regression with an optimal
. // bandwidth (Imbens and Kalyanaraman, 2009) and triangle kernel weights
. rd grade4test grade3test, z0(56) graph noscatter
Two variables specified; treatment is
assumed to jump from zero to one at Z=56.

 Assignment variable Z is grade3test
 Treatment variable X_T unspecified
 Outcome variable y is grade4test

Command used for graph: lpoly; Kernel used: triangle (default)
Bandwidth: 3.087786; loc Wald Estimate: 2.1632103
Bandwidth: 1.543893; loc Wald Estimate: 2.2975457
Bandwidth: 6.175572; loc Wald Estimate: 2.0204368
Estimating for bandwidth 3.087786015142091
Estimating for bandwidth 1.543893007571046
Estimating for bandwidth 6.175572030284182
------------------------------------------------------------------------------
  grade4test |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       lwald |    2.16321   .1667734    12.97   0.000      1.83634     2.49008
     lwald50 |   2.297546   .2355847     9.75   0.000     1.835808    2.759283
    lwald200 |   2.020437    .126864    15.93   0.000     1.771788    2.269086
------------------------------------------------------------------------------
```

9. We will now estimate the treatment effect using `rd` but allowing for non-compliance. Because of the fuzzy RD, you need to modify the `rd` command to include the treatment variable ($inGT$), otherwise it assumes $inGT = 0$ below the cut score and $inGT = 1$

above it. Notice the running variable goes last in the list of variables.

```
rd grade4test inGT grade3test, z0(56) graph noscatter
```

Note that `rd` gives you more estimates when the treatment assignment is fuzzy. The `numer` line here is nearly identical to the sharp RD result from (8). This is the "reduced form." The `denom` line is roughly equivalent to the effect of exceeding the threshold on treatment from (7). This is the "first stage." `lwald` is the reduced form divided by the first stage.

See below. Note the Wald estimates are all close to 3.

```
. rd grade4test inGT grade3test, z0(56) graph noscatter
Three variables specified; jump in treatment
at Z=56 will be estimated. Local Wald Estimate
is the ratio of jump in outcome to jump in treatment.

 Assignment variable Z is grade3test
 Treatment variable X_T is inGT
 Outcome variable y is grade4test

Command used for graph: lpoly; Kernel used: triangle (default)
Bandwidth: 3.087786; loc Wald Estimate: 3.0404888
Bandwidth: 1.543893; loc Wald Estimate: 2.9764419
Bandwidth: 6.175572; loc Wald Estimate: 2.943203
Estimating for bandwidth 3.087786015142091
Estimating for bandwidth 1.543893007571046
Estimating for bandwidth 6.175572030284182
------------------------------------------------------------------------------
  grade4test |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       numer |    2.16321   .1666626    12.98   0.000     1.836558    2.489863
       denom |   .7114679   .0329481    21.59   0.000     .6468909     .776045
       lwald |   3.040489   .1850535    16.43   0.000     2.677791    3.403187
     numer50 |   2.297546   .2352639     9.77   0.000     1.836437    2.758654
     denom50 |   .7719102   .0433517    17.81   0.000     .6869424    .8568779
     lwald50 |   2.976442   .2457943    12.11   0.000     2.494694     3.45819
    numer200 |   2.020437    .126827    15.93   0.000      1.77186    2.269013
    denom200 |   .6864755   .0257945    26.61   0.000     .6359192    .7370318
    lwald200 |   2.943203   .1424133    20.67   0.000     2.664078    3.222328
------------------------------------------------------------------------------
```

10. To connect to the lecture on IV, try the syntax below. (We cannot use the Stata factor variables for the two *gap=* slopes, since *above56* cannot be included in the list of regressors—it is the exogenous instrument). Compare the first stage and final point estimates to (9).

```
gen gapabove = gap*above56
gen gapbelow = gap*(1-above56)
ivregress 2sls grade4test (inGT=above56) gapbelow gapabove , first cluster(grade3test)

. ivregress 2sls grade4test (inGT=above56) gapbelow gapabove , first robust cluster(grade3test)

First-stage regressions
-----------------------

                                                Number of obs     =     10,000
                                                N. of clusters    =        113
                                                F(   3,   9996)   =    1492.34
                                                Prob > F          =     0.0000
                                                R-squared         =     0.3533
                                                Adj R-squared     =     0.3531
                                                Root MSE          =     0.2735

------------------------------------------------------------------------------
             |               Robust
        inGT |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    gapbelow |   .0065711   .0008155     8.06   0.000     .0049726    .0081696
    gapabove |   .0113775   .0057961     1.96   0.050     .0000159     .022739
     above56 |    .695023   .0151054    46.01   0.000     .6654134    .7246326
       _cons |   .1193058   .0068076    17.53   0.000     .1059615    .1326502
------------------------------------------------------------------------------


Instrumental variables (2SLS) regression          Number of obs   =     10,000
                                                  Wald chi2(3)    =  130628.34
                                                  Prob > chi2     =     0.0000
                                                  R-squared       =     0.9066
                                                  Root MSE        =     1.3895

                           (Std. Err. adjusted for 113 clusters in grade3test)
------------------------------------------------------------------------------
             |               Robust
  grade4test |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        inGT |   2.892483   .0920942    31.41   0.000     2.711981    3.072984
    gapbelow |   .9421128   .0039748   237.02   0.000     .9343223    .9499033
    gapabove |   .9641147   .0257027    37.51   0.000     .9137383    1.014491
       _cons |    60.6563   .0381409  1590.32   0.000     60.58154    60.73105
------------------------------------------------------------------------------
Instrumented:  inGT
Instruments:   gapbelow gapabove above56
```