`rdlocrand`: Inference in RD Designs Under Local Randomization

Matias Cattaneo
University of Michigan

Rocío Titiunik
University of Michigan

Gonzalo Vazquez-Bare
University of Michigan

https://sites.google.com/site/rdpackages

Stata Conference - Chicago

July 28, 2016

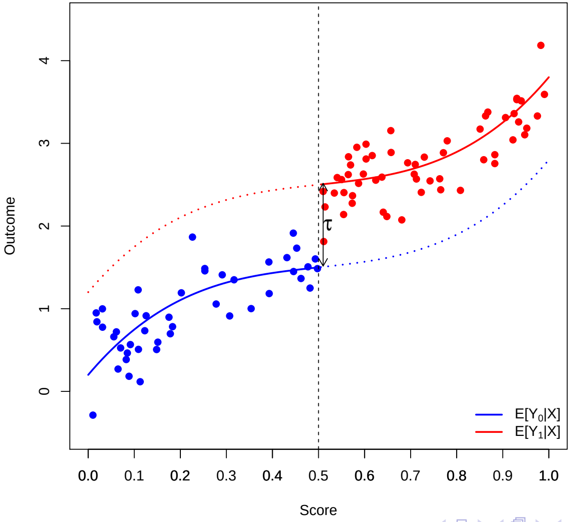
Introduction

- Regression discontinuity designs (RDDs) are one of the most popular methods for causal inference.
- RDDs can be interpreted as a local experiment in a window around the cutoff.
- `rdlocrand` analyzes RDDs using tools from classical randomized experiments literature:
  - `rdwinselect`: window selection
  - `rdrandinf`: randomization inference
  - `rdsensitivity`: sensitivity analysis
  - `rdrbounds`: Rosenbaum bounds

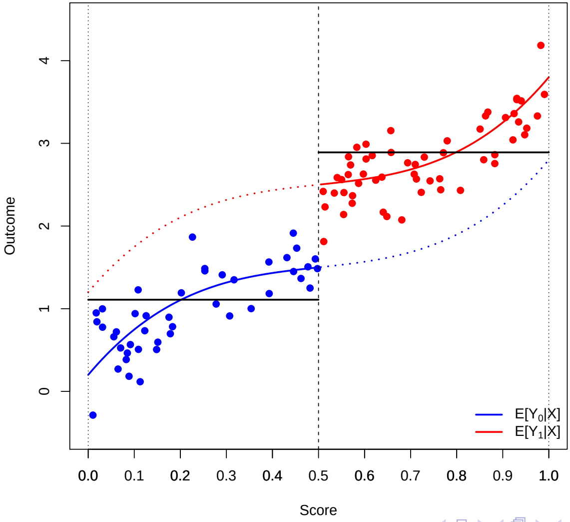
Regression Discontinuity Designs: motivation

- Many programs or policies are assigned based on whether a score (running variable) X exceeds a threshold c:
  - Scholarship to students above a certain test score.
  - Subsidy to households above a poverty threshold.
- RDDs exploit the discontinuity in the probability of treatment assignment at the cutoff.
  - Sharp design: $D_i = \mathbb{1}(X_i \geq c)$.
- Intuition: in a "small" window around the cutoff, units above and below are comparable.

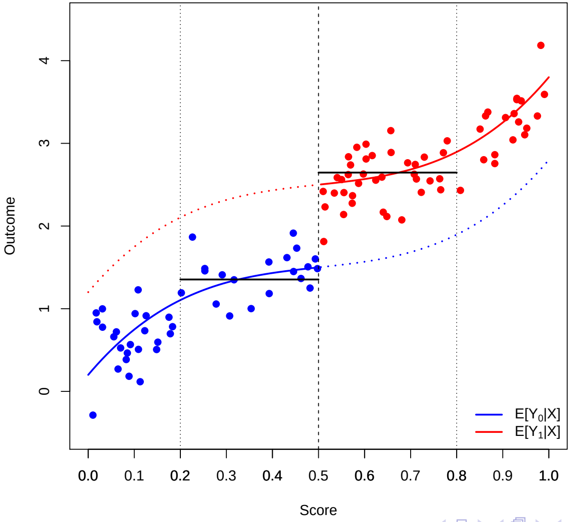
RDD: intuition



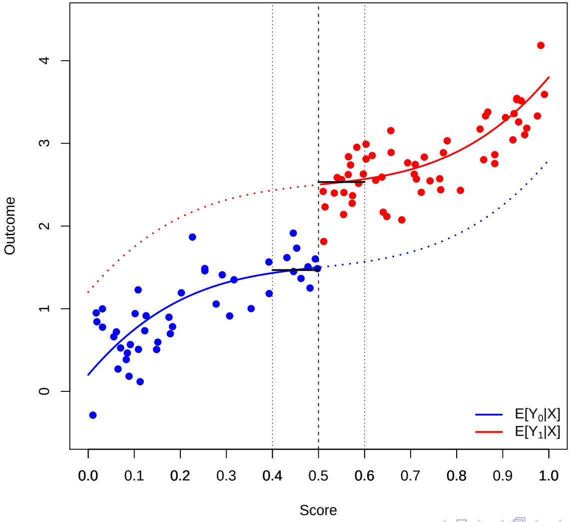
RDD: intuition



RDD: intuition



RDD: intuition



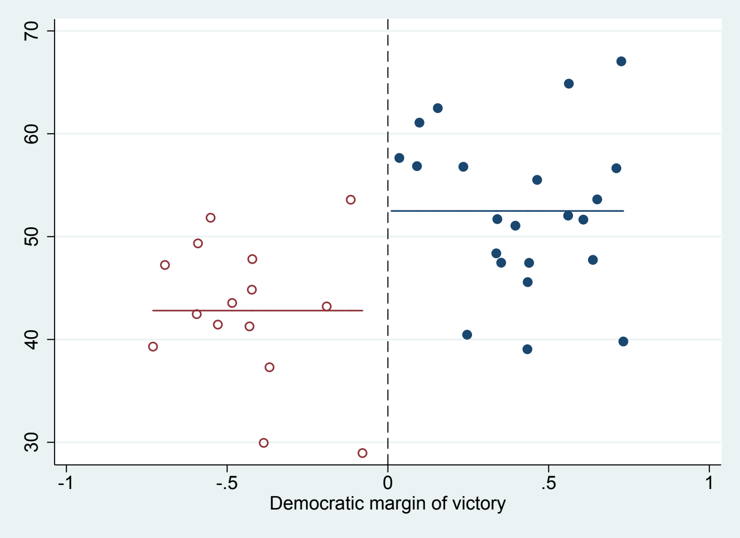
Inference in classical randomized experiments

- RDDs as randomized experiments around the cutoff.
  - Key assumption: existence of a window in which this is true.
- Inference in classical experiments:
  - Fixed (nonrandom) potential outcomes.
  - Known assignment mechanism.
- Randomization (finite sample) p-value:
  - Choose a statistic T (e.g. difference in means),
  - Calculate T for all permutations of treatment assignment,
  - Find $\mathbb{P}(T \geq T_{obs})$.
- In Stata:
  `permute d stat = (r(mu_1)-r(mu_2)): ttest y, by(d).`

Randomization inference with rdrandinf

```
. rdrandinf demvoteshfor2 demmv, wl(-.75) wr(.75)
Selected window = [-.75 ; .75]

Running permutation test...
Permutation test complete.

Inference for sharp design

Cutoff c = 0.00  | Left of c  Right of c         Number of obs =        1390
-----------------+----------------------         Order of poly =           0
   Number of obs |       595         702         Kernel type   =     uniform
Eff. Number of obs |      15          22         Reps          =        1000
  Mean of outcome |   42.808      52.497         Window        = set by user
  S.D. of outcome |    7.042       7.742         H0:       tau =       0.000
          Window |    -0.750       0.750         Randomization = fixed margins

Outcome: demvoteshfor2. Running variable: demmv.
--------------------------------------------------------------------------------
                 |           Finite sample              Large sample
                 |           -------------      ----------------------------------
       Statistic |        T       P>|T|         P>|T|    Power vs d =       3.52
-----------------+--------------------------------------------------------------
  Diff. in means |    9.689       0.001         0.000                       0.300
--------------------------------------------------------------------------------
```

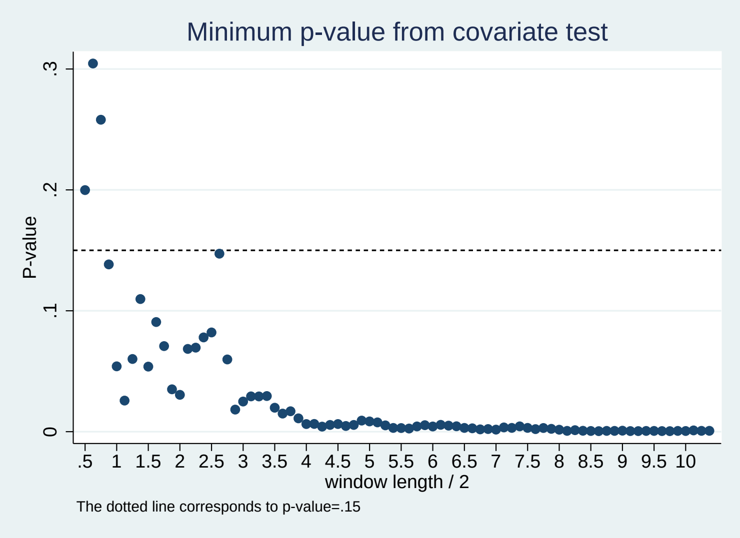
Randomization inference with `rdrandinf`



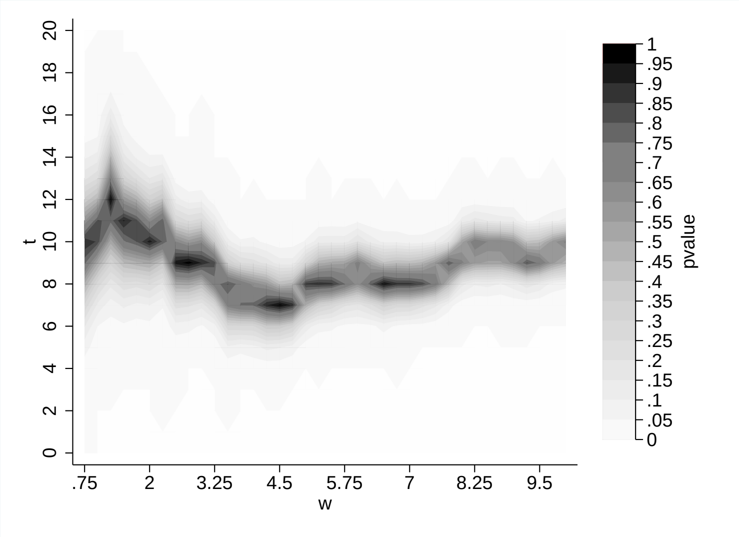
Choosing the window with rdwinselect

```
. rdwinselect demmv $covariates, wmin(.5) wstep(.125) reps(10000)

Window selection for RD under local randomization

Cutoff c = 0.00 | Left of c   Right of c            Number of obs  =        1390
----------------+-----------------------            Order of poly  =           0
  Number of obs |      640         750              Kernel type    =     uniform
 1th percentile |        6           8              Reps           =       10000
 5th percentile |       32          38              Testing method =   rdrandinf
10th percentile |       64          75              Balance test   =       ttest
20th percentile |      128         150

                |  Bal. test       Var. name        Bin. test
Window length /2|   p-value    (min p-value)         p-value     Obs<c     Obs>=c
----------------+------------------------------------------------------------------
          0.500 |     0.268    demvoteshlag2           0.230         9         16
          0.625 |     0.435            dopen           0.377        13         19
          0.750 |     0.268            dopen           0.200        15         24
          0.875 |     0.150            dopen           0.211        16         25
          1.000 |     0.069            dopen           0.135        17         28
          1.125 |     0.037            dopen           0.119        19         31
          1.250 |     0.062            dopen           0.105        21         34
          1.375 |     0.141         dmidterm           0.539        30         36
          1.500 |     0.092         dmidterm           0.640        34         39
          1.625 |     0.113         dmidterm           0.734        37         41
Variable used in binomial test (running variable): demmv
Covariates used in balance test: presdemvoteshlag1 population demvoteshlag1 demvoteshlag2
> demwinprv1 demwinprv2 dopen dmidterm

Largest recommended window is [-.75; .75] with 39 observations (15 below, 24 above).
```

Choosing the window with `rdwinselect`



Sensitivity analysis with `rdsensitivity`



Rosenbaum bounds with rdrbounds

```
. rdrbounds demvoteshfor2 demmv, gammalist(.8 1 1.2) wlist(.5 .75 1) reps(1000)
Calculating randomization p-values...
                           w =      0.500      0.750      1.000
-----------------------------------------------------------------
           Bernoulli p-value |      0.012      0.001      0.000
-----------------------------------------------------------------

Running sensitivity analysis...
gamma  exp(gamma)          w =      0.500      0.750      1.000
-----------------------------------------------------------------
 0.80      2.23    lower bound |      0.006      0.001      0.000
                   upper bound |      0.068      0.015      0.002
-----------------------------------------------------------------
 1.00      2.72    lower bound |      0.004      0.001      0.000
                   upper bound |      0.106      0.034      0.006
-----------------------------------------------------------------
 1.20      3.32    lower bound |      0.003      0.001      0.000
                   upper bound |      0.168      0.060      0.017
-----------------------------------------------------------------
```

Other features of `rdlocrand`

- Alternative statistics: Kolmogorov-Smirnov, rank sum.
- Polynomial adjustment of potential outcomes.
- Randomization-based confidence intervals for treatment effect.
- Companion `R` functions with same capabilities.
- See Cattaneo, Titiunik and Vazquez-Bare (2016): Inference in Regression Discontinuity Designs under Local Randomization. *Stata Journal 16(2): 331-367*.

Thank you!

Additional material

Other issues: multiple testing

- `rdwinselect` performs hypothesis tests for a large set of covariates.
- Multiple testing leads to overrejection $\rightarrow$ "err on the safe side" (smaller windows).
    - Local randomization assumption only credible in a small window.
- `rdwinselect` can also test all covariates jointly using Hotelling's T^2 test.
    - Typically leads to much larger windows.

Other issues: outcome model adjustment

- Strongest version of local randomization assumption states that potential outcomes do not depend on the score inside the window:
    - Exclusion restriction: $Y_i(d, x) = Y_i(d)$.
- This assumption may be too strong in some scenarios.
- `rdlocrand` allows the user to state a polynomial model for the potential outcomes to eliminate the dependence on X.
    - E.g. use a linear model to remove the slope.