APPENDIX: Description of example code and Software Packages

A1. Walkthrough of RDD Application with R Code

In this appendix, we walk through a sample statistical R code for an RD Design analysis that can help to guide investigators interested in utilizing this method for their research. The R code is available at https://github.com/AdinaZekiAlHazzouri/RDD_Paper . Required R packages include ***rdrobust***[1,2] and ***rddensity***[3] (available at https://rdpackages.github.io/rdrobust/). Our code can be used to run a RD model on a teaching version of the Framingham Heart Study (FHS) dataset, which is available upon request at https://biolincc.nhlbi.nih.gov/teaching/. Please note that this dataset, which contains redacted data and is intended for teaching purposes only, should *not* be used for publishing or interpreting actual results.

The purpose of this walk-through is to provide a practical sequence of steps that can be followed to run a RD analysis. The FHS is a longitudinal cohort study which began in 1948 and has since helped to uncover some of the determinants of cardiovascular disease.[4] In our application of RD to the teaching FHS dataset, we seek to estimate the causal effect of antihypertensive medications on cardiovascular disease incidence among individuals in this cohort. In our example, the cutoff (C) for treatment will be a systolic blood pressure of 140 mmHg, and the treatment will be the use of an anti-hypertensive medication. The following is a brief data dictionary of relevant variables and their purpose in our walk-through.

Variable	Description	Type of variable
CVD	Cardiovascular Disease in follow-up period (0 = No, 1 = Yes)	Outcome variable (Y)
SYBP	Systolic Blood Pressure (mmHg)	Running variable (X)
BPMEDS	Use of Anti-hypertensive medication at exam (0 = No, 1 = Yes)	Treatment status variable (D)
AGE	Age at exam (years)	Variable at baseline used to assess covariate balance

Next, we walk through the 7 steps of an RD analysis summarized in Section 3 of the commentary.

We first set up our R environment by attaching the necessary packages, and by loading and summarizing our dataset. In **Step 1**, we define the variables we would like to use for our RD analysis, comprised of the outcome value (Y), score/running variable (X), and the cutoff value (C).

```
rm(list=ls())

### Load R packages
library(rdrobust)
library(rddensity)
```

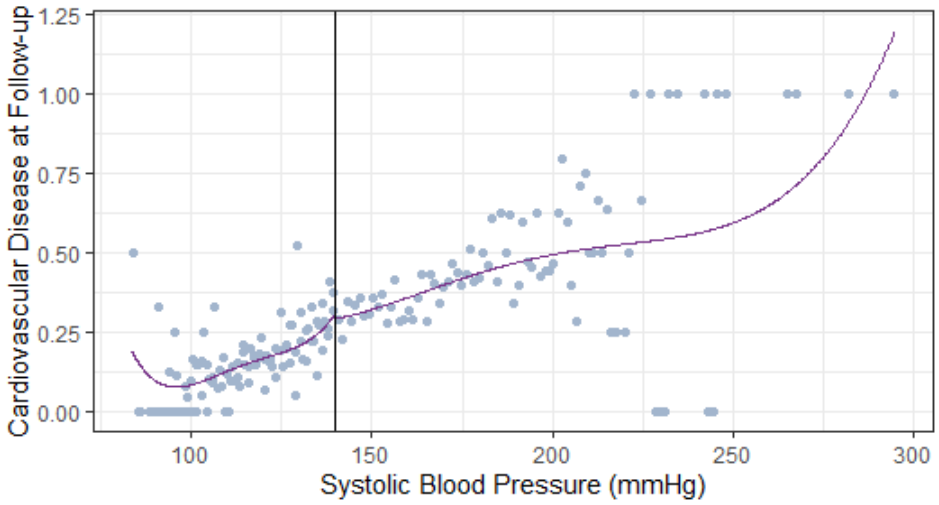
```
### Load dataset
setwd("Insert Directory Here") # Put location of your working directory here
frm <-read.csv("frmgham2.csv")

### For illustrative purposes, and to keep samples consistent between Sharp R
D and Fuzzy RD models, we will limit our sample to individuals with a non-mis
sing value of the outcome variable, running variable, and actual treatment ta
ke-up indicator

frm <-frm[complete.cases(frm[c("CVD", "SYSBP", "BPMEDS")]), ]



################################################################################
##### Step 1. Identify main components of the RD model ####################
################################################################################

Y = frm$CVD    # outcome variable
X = frm$SYSBP  # running variable
D = frm$BPMEDS # actual treatment take-up indicator
C = 140        # cutoff value
T = 1*(X>=C)   # treatment assignment indicator

summary(Y)
summary(X)
summary(D)
summary(T)
```

```
> summary(Y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.2494  0.0000  1.0000
> summary(X)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   83.5   120.0   132.0   136.1   149.0   295.0
> summary(D)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.08555 0.00000 1.00000
> summary(T)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.3786  1.0000  1.0000
```

In **Step 2**, we generate an exploratory plot, which enables us to visualize the global relationship between the running variable and our outcome, separately at each side of the cutoff.

```
################################################################################
      Step 2. RD Plots: Exploratory plot to illustrate the nature
                              of the relationship
################################################################################

rdplot1 = rdplot(y = Y, x = X, c = C,
                 x.label = "Systolic Blood Pressure (mmHg)",
                 y.label = "Cardiovascular Disease at Follow-up",
                 col.dots="lightsteelblue3",col.lines="mediumorchid4")
```

```
summary(rdplot1)
```



In **Step 3**, we set up the local polynomial model by specifying a polynomial order (i.e., P = 1, corresponding to a linear fit), a weighting function (i.e., kernel), and a selection procedure for computing the bandwidth (i.e., MSE optimal).[5]

```
###########################################################################
##### Step 3. Set up the local polynomial model ###################
###########################################################################

P  = 1             # local linear approximation
K  = 'triangular'  # triangular kernel
BW = 'mserd'       # MSE optimal bandwidth
```

In **Step 4,** we estimate the RD effect by running ***rdrobust*** using our choices for the tuning parameters.[1] From the function output, we can recover the number of observations on each side of the cutoff, the bandwidth (h), and the number of observations within the bandwidth (i.e., Effective Number of Observations). In addition, the coefficient value on the row labeled "Conventional" corresponds to the RD causal estimate, the causal effect of the treatment on the outcome among subjects within the selected bandwidth. In this case, we obtain a reduction on the probability of a CVD event of 5.1 percentage points.

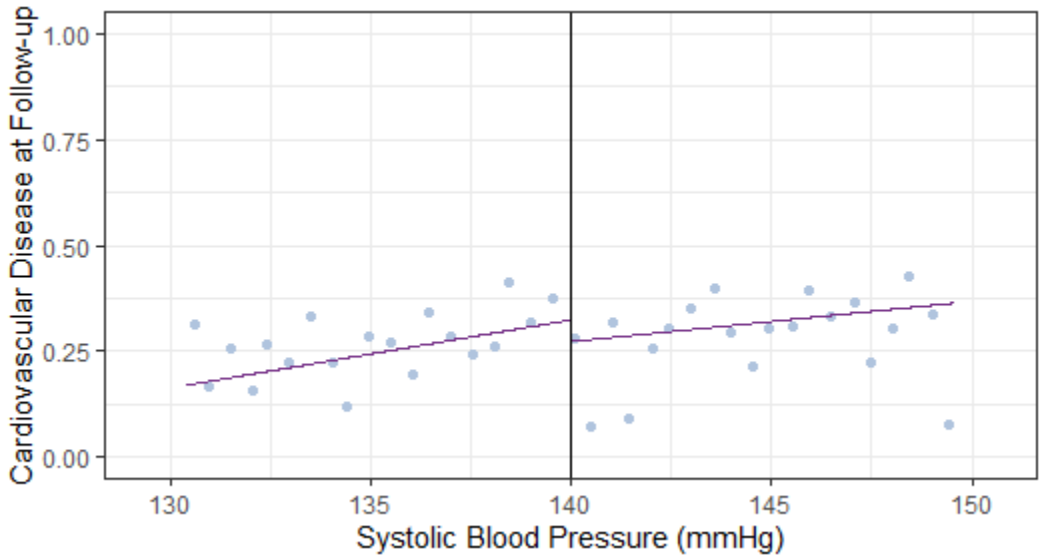
In addition, in **Step 5** we compute the 95% CI, corresponding to a plausible range of RD estimates consistent with our data, and the associated p-value. Here, the appropriate 95% CI can be recovered from the row labeled "Robust", [-xxx, xxx], which accounts for the nonparametric features of our estimation approach and should be preferred over the "Conventional" one. The p-value associated with this 95% CI is 0.072.[6]

```
#########################################################################
##### Steps 4-5. RD Estimation and Inference ############################
#########################################################################

rd_manual = rdrobust(y = Y, x = X, c = C,
                     kernel = K, p = P, bwselect = BW)
summary(rd_manual)

# You can also run RD using default choices, as shown here:
rd_default = rdrobust(y = Y, x = X, c = C)
summary(rd_default)
```

```
Call: rdrobust

Number of Obs.                11034
BW type                       mserd
Kernel                   Triangular
VCE method                       NN

Number of Obs.                 6857         4177
Eff. Number of Obs.            1675         1492
Order est. (p)                    1            1
Order bias  (q)                   2            2
BW est. (h)                   9.571        9.571
BW bias (b)                  17.892       17.892
rho (h/b)                     0.535        0.535
Unique Obs.                     104          155

=============================================================================
        Method     Coef. Std. Err.          z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional    -0.051     0.036    -1.417     0.157    [-0.122 , 0.020]
        Robust         -         -    -1.610     0.107    [-0.147 , 0.014]
=============================================================================
```

We can also plot our estimated RD model using the ***rdplot*** function, similar to the one in Figure 1, but now reflecting our choices for the estimation of the RD model instead of a global fit.

```
# Plotting the RD model

bw = rdrobust(y = Y, x = X, c = C,
              kernel = 'triangular', p = 1, bwselect = 'mserd')$bws[1]

RD_plot = rdplot(y = Y, x = X, c = C,
                 p = 1, kernel = 'triangular', h = bw,
                 x.label = "Systolic Blood Pressure (mmHg)",
                 y.label = "Cardiovascular Disease at Follow-up",
                 subset = (X >= (C - bw) & X <= (C + bw)), x.lim = c(C - bw –
                 1,+ C + bw + 1), y.lim = c(0,+ 1), col.dots="lightsteelblue,
                 col.lines="mediumorchid4")
```



In the above figure, we can see a discontinuity at the cutoff. Individuals just above the cutoff (eligible for antihypertensive treatment) observed a decreased incidence in cardiovascular disease, as compared to individuals just below the cutoff. The distance between the lines at the cutoff gives you the ATE, which corresponds to the parameter estimate provided in the summary function (-0.051).

In **Step 6**, we can run a set of robustness checks and validation tests to assess the validity of our RD model. First, we explore the robustness of our RD effect estimates to the implementation choices of our local polynomial estimator. In particular, we look at alternative kernels, polynomial orders, and bandwidth choices.

```
#########################################################################
##### Step 6. Robustness checks and validation ##################
#########################################################################

# Change to uniform kernel
rd_2 <- rdrobust(y = Y, x = X, c = C,
                 kernel = 'uniform', p = 1, bwselect = 'mserd')
summary(rd_2)

# Change to local quadratic regression
rd_3 <- rdrobust(y = Y, x = X, c = C,
                 kernel = 'triangular', p = 2, bwselect = 'mserd')
summary(rd_3)

# Change to different bandwidths on each side:
rd_4 <- rdrobust(y = Y, x = X, c = C,
                 kernel = 'triangular', p = 1, bwselect = 'msetwo')
summary(rd_4)

# Other bandwidth choices:
rd_bws <- rdbwselect(y = Y, x = X, c = C, all = TRUE)
```

```
summary(rd_bws)
```

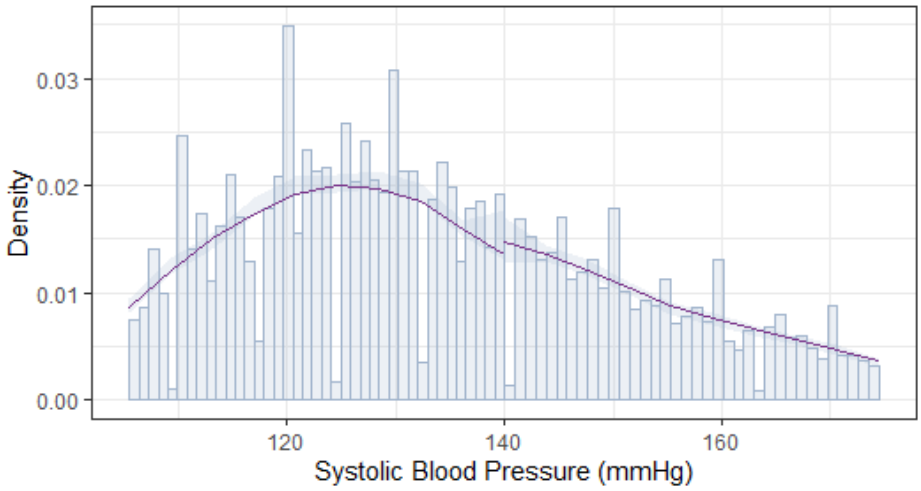
```
> summary(rd_2)
Call: rdrobust

Number of Obs.                11034
BW type                       mserd
Kernel                      Uniform
VCE method                       NN

Number of Obs.                 6857         4177
Eff. Number of Obs.            1257         1243
Order est. (p)                    1            1
Order bias  (q)                   2            2
BW est. (h)                   7.594        7.594
BW bias (b)                  16.335       16.335
rho (h/b)                     0.465        0.465
Unique Obs.                     104          155

=============================================================================
        Method     Coef. Std. Err.          z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional    -0.042     0.037    -1.130     0.258    [-0.116 , 0.031]
        Robust         -         -    -1.348     0.178    [-0.137 , 0.025]
=============================================================================

> summary(rd_3)
Call: rdrobust

Number of Obs.                11034
BW type                       mserd
Kernel                   Triangular
VCE method                       NN

Number of Obs.                 6857         4177
Eff. Number of Obs.            3050         2188
Order est. (p)                    2            2
Order bias  (q)                   3            3
BW est. (h)                  15.413       15.413
BW bias (b)                  23.186       23.186
rho (h/b)                     0.665        0.665
Unique Obs.                     104          155

=============================================================================
        Method     Coef. Std. Err.          z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional    -0.069     0.042    -1.632     0.103    [-0.152 , 0.014]
        Robust         -         -    -1.713     0.087    [-0.174 , 0.012]
=============================================================================
```

```
> summary(rd_4)
Call: rdrobust

Number of Obs.                11034
BW type                      msetwo
Kernel                    Triangular
VCE method                        NN

Number of Obs.                 6857         4177
Eff. Number of Obs.            1659         2911
Order est. (p)                    1            1
Order bias  (q)                   2            2
BW est. (h)                   9.270       24.862
BW bias (b)                  17.971       37.880
rho (h/b)                     0.516        0.656
Unique Obs.                     104          155

=============================================================================
        Method     Coef. Std. Err.          z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional    -0.032     0.033    -0.968     0.333    [-0.098 , 0.033]
        Robust         -         -    -1.179     0.238    [-0.122 , 0.030]
=============================================================================

> summary(rd_bws)
Call: rdbwselect

Number of Obs.                11034
BW type                         All
Kernel                    Triangular
VCE method                        NN

Number of Obs.                 6857         4177
Order est. (p)                    1            1
Order bias  (q)                   2            2
Unique Obs.                     104          155

=====================================================
                  BW est. (h)     BW bias (b)
             Left of c Right of c  Left of c Right of c
=====================================================
     mserd     9.571      9.571     17.892     17.892
    msetwo     9.270     24.862     17.971     37.880
    msesum    13.741     13.741     21.248     21.248
  msecomb1     9.571      9.571     17.892     17.892
  msecomb2     9.571     13.741     17.971     21.248
     cerrd     6.009      6.009     17.892     17.892
    certwo     5.820     15.610     17.971     37.880
    cersum     8.627      8.627     21.248     21.248
  cercomb1     6.009      6.009     17.892     17.892
  cercomb2     6.009      8.627     17.971     21.248
=====================================================
```

Overall, we see that the main results remain (i.e., a protective effect is observed between the indicated anti-hypertensive medication and cardiovascular disease). Next, we conduct a manipulation test, looking for potential violation of the assumed continuity of observations around the threshold. If this assumption holds, we should expect to see a relatively consistent frequency of observations at each side of the cutoff. This test can be easily computed using the ***rddensity*** package, which provides the results of a statistical test together with a density plot illustrating the findings.[3]

```
# Test for manipulation of the running variable
rdd <- rddensity(X = X, c = C)
summary(rdd)

# Figure
plot <- rdplotdensity(rdd, X, xlabel = "Systolic Blood Pressure (mmHg)", ylab
el = "Density", lcol = "mediumorchid4",pcol = "lightsteelblue3", histLineCol
= "lightsteelblue3", histFillCol = "lightsteelblue3", CIcol = "lightsteelblue
3")
```

```
> summary(rdd)

Manipulation testing using local polynomial density estimation.

Number of obs =       11034
Model =               unrestricted
Kernel =              triangular
BW method =           estimated
VCE method =          jackknife

c = 140               Left of c           Right of c
Number of obs         6857                4177
Eff. Number of obs    2182                1788
Order est. (p)        2                   2
Order bias (q)        3                   3
BW est. (h)           11.5                11.5

Method                T                   P > |T|
Robust                1.0241              0.3058


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

Window Length / 2          <c     >=c    P>|T|
0.000                       0     179    0.0000
```

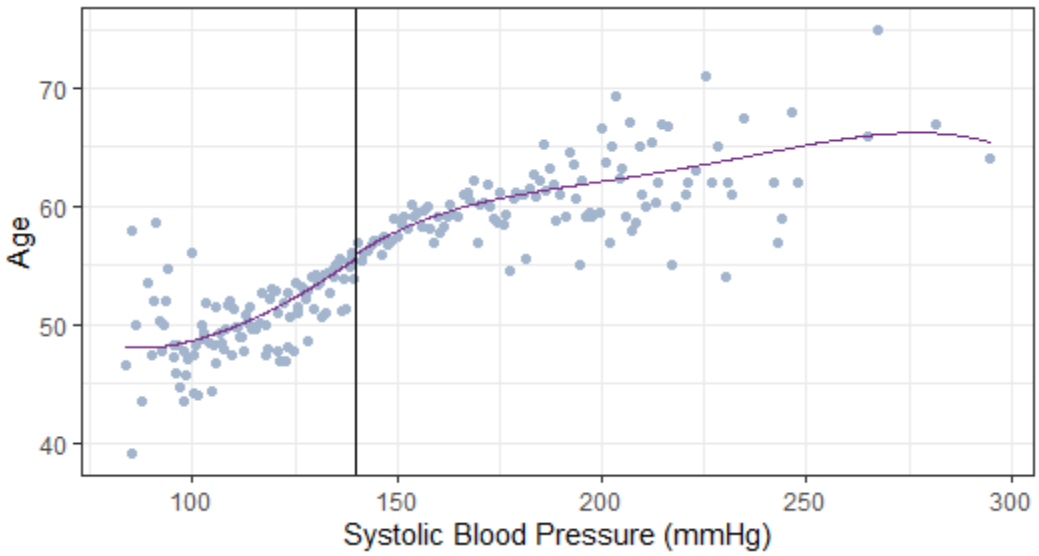


From both the plot and the results of the test, we can conclude that there is no evidence of a jump in the density of the running variable at the cutoff, which is consistent with a lack of manipulation.

We also assess exchangeability around the cutoff by looking at covariate balance; this involves taking a given covariate (e.g., age) and treating it as an outcome in an RD model. One can subsequently view the RD effect and visualize a plot to see whether there is a significant difference in the level of the covariate of interest, comparing observations below and above the cutoff. If the exchangeability assumption holds, then we should expect the levels of this covariate to be comparable below and above the cutoff (i.e., the RD effect here should be close to 0).

```
## Covariate balance assessment (Age)

rd_covs = rdrobust(y = frm$AGE, x = X, c = C)
summary(rd_covs)

rdplot(y = frm$AGE, x = X, c = C,
  title = "Covariate Balance - RD Plot",
  x.label = "Systolic Blood Pressure (mmHg)",
  y.label = "Age", col.dots="lightsteelblue3",col.lines="mediumorchid4")
```

```
> summary(rd_covs)
Call: rdrobust

Number of Obs.                11034
BW type                       mserd
Kernel                   Triangular
VCE method                       NN

Number of Obs.                 6857         4177
Eff. Number of Obs.            2812         2080
Order est. (p)                    1            1
Order bias  (q)                   2            2
BW est. (h)                  14.601       14.601
BW bias (b)                  22.187       22.187
rho (h/b)                     0.658        0.658
Unique Obs.                     104          155

=============================================================================
        Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional     0.596     0.565     1.055     0.292     [-0.512 , 1.704]
        Robust         -         -     0.927     0.354     [-0.708 , 1.978]
=============================================================================
```



The RD model output and plot above show that there is no significant difference in age comparing individuals below and above the cutoff; this provides some evidence for the exchangeability assumption required for RD design.

Next, in **Step 7**, we account for imperfect compliance with treatment assignment by running a fuzzy RD model.

```
#####################################################################
#### Step 7: Fuzzy RD: average effect for compliers #########
#####################################################################

# Fuzzy RD model:

RD_fuzzy = rdrobust(y = Y, x = X, c = C, fuzzy = D)
summary(RD_fuzzy)
RD_fuzzy$Estimate[1] # Fuzzy RDD effect = -0.2301855
```

```
> summary(RD_fuzzy)
Call: rdrobust

Number of Obs.                11034
BW type                       mserd
Kernel                   Triangular
VCE method                       NN

Number of Obs.                 6857         4177
Eff. Number of Obs.            3713         2448
Order est. (p)                    1            1
Order bias  (q)                   2            2
BW est. (h)                  18.679       18.679
BW bias (b)                  30.694       30.694
rho (h/b)                     0.609        0.609
Unique Obs.                     104          155

=============================================================================
        Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional    -0.230     1.060    -0.217     0.828    [-2.307 , 1.847]
        Robust         -         -    -0.581     0.561    [-3.184 , 1.727]
=============================================================================
```

In this output, the coefficient value (-0.230) corresponds to the per-protocol treatment effect of anti-hypertensive medication on cardiovascular disease; that is, this is the complier-specific treatment effect, computed by dividing the sharp RD effect by the compliance rate (the percentage of participants who received the treatment that they were assigned to).

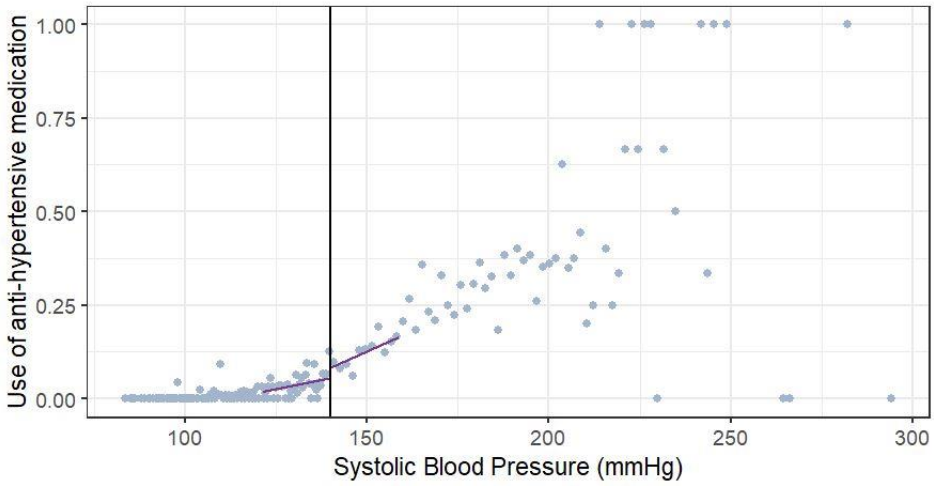
We can also manually calculate this fuzzy RDD effect (-0.230) by running two separate steps: first, we can calculate the sharp RD effect (note that we are using the same sample size and optimal bandwidth from the fuzzy RD model). As shown below, this sharp RD effect is (-0.006).

```
# First, assess sharp RDD effect within bandwidth from the above fuzzy model
RD_sharp = rdrobust(y = Y, x = X, c = C, h = RD_fuzzy$bws[1,1], b = RD_fuzzy$
bws[2,1], subset=!is.na(D))
summary(RD_sharp)
RD_sharp$Estimate[1] # Sharp RDD effect = -0.00558034
```

Next, we calculate the specific amount of non-compliance in our application. We do this by running an RDD where we use actual treatment take-up (BPMEDS) as the outcome, and the same running variable (here we use the same sample size and optimal bandwidth from the fuzzy RD model as well). Our estimated effect (the amount of compliance at the cutoff) is 0.024, which is the difference between treatment take-up in the treatment group (8.06%) and treatment take-up in the control group (5.64%).

```
# Next, assess compliance by computing RD effect using D as the outcome

RD_comp = rdrobust(y = D, x = X, c = C, h = RD_fuzzy$bws[1], b = RD_fuzzy$bws
[2,1])
summary(RD_comp)
RD_comp$Estimate[1] # Compliance RDD effect = 0.02424286
RD_comp$beta_p_r[1] # Treatment Take-up for Treatment Group = 0.08062456
RD_comp$beta_p_l[1] # Treatment Take-up for Control Group = 0.0563817
RD_comp_plot = rdplot(y = D, x = X, c = C,
                      x.label = "Systolic Blood Pressure (mmHg)",
                      y.label = "Use of anti-hypertensive medication", col.do
ts="lightsteelblue3",col.lines="mediumorchid4")
```

```
> summary(RD_comp)
Call: rdrobust

Number of Obs.                11034
BW type                      Manual
Kernel                   Triangular
VCE method                       NN

Number of Obs.                6857         4177
Eff. Number of Obs.           3713         2448
Order est. (p)                   1            1
Order bias  (q)                  2            2
BW est. (h)                 18.679       18.679
BW bias (b)                 30.694       30.694
rho (h/b)                    0.609        0.609
Unique Obs.                    104          155

=============================================================================
        Method     Coef. Std. Err.          z     P>|z|      [ 95% C.I. ]
=============================================================================
  Conventional     0.024     0.014      1.677     0.094    [-0.004 , 0.053]
        Robust         -         -      1.599     0.110    [-0.006 , 0.061]
=============================================================================
```



Finally, we can recover the fuzzy RDD estimate (-0.2301855) by dividing the sharp RDD estimate (-0.005580354) by the compliance rate (0.02424286).

```
# Final calculation
RD_sharp$Estimate[1] / RD_comp$Estimate[1] # -0.2301855
```

A2. Supplemental R Code for RDD Application

Box A2: R code

Sample statistical codes (in R) used in RDD analysis

Required R packages include ***rdrobust***[1,2] and ***rddensity***[3] (https://rdpackages.github.io/rdrobust/). The statistical codes presented below are based on the RDD application investigating the causal effect of blood pressure medication (BPMEDS) on CVD.

Step 1: Identify main components of the RD model

```
Y = frm$CVD    # outcome variable
X = frm$SYSBP  # running variable
D = frm$BPMEDS # actual treatment take-up indicator
C = 140        # cutoff value
T = 1*(X>=C)   # treatment assignment indicator
```

Step 2: Plot the RD data (global fit)

```
rdplot1 = rdplot(y = Y, x = X, c = C,
                 x.label = "Systolic Blood Pressure (mmHg)",
                 y.label = "Cardiovascular Disease at Follow-up")
summary(rdplot1)
```

Step 3: Set up the local polynomial model

```
P  = 1             # local linear approximation
K  = 'triangular'  # triangular kernel
BW = 'mserd'       # MSE optimal bandwidth
```

Steps 4-5: RD estimation and inference

```
rd_manual = rdrobust(y = Y, x = X, c = C,
                     kernel = K, p = P, bwselect = BW)
summary(rd_manual)
```

```
We can re-plot our RD data (as in Step 2) but now using our choices for the estima
tion of the RD model instead of a global fit.
```

```
bw = rd_manual $bws[1]

RD_plot = rdplot(y = Y, x = X, c = C,
                 p = 1, kernel = 'triangular', h = bw,
                 x.label = "Systolic Blood Pressure (mmHg)",
                 y.label = "Cardiovascular Disease at Follow-up",
                 subset = (X >= (C - bw) & X <= (C + bw)), x.lim = c(C - bw –
                 1,+ C + bw + 1), y.lim = c(0,+ 1))
```

Box A2 (continued)

Sample statistical codes (in R) used in RDD analysis – continued

Step 6: Robustness checks and validation (a comprehensive list can be found in the Appendix)

```
# Change to uniform kernel
rd_2 <- rdrobust(y = Y, x = X, c = C,
                 kernel = 'uniform', p = 1, bwselect = 'mserd')
summary(rd_2)

# Change to different bandwidths on each side:
rd_4 <- rdrobust(y = Y, x = X, c = C,
                 kernel = 'triangular', p = 1, bwselect = 'msetwo')
summary(rd_4)

# Test for manipulation of the running variable
rdd <- rddensity(X = X, c = C)
summary(rdd)
plot <- rdplotdensity(rdd, X)

## Covariate balance assessment (Age is the only covariate in this example)
rd_covs = rdrobust(y = frm$AGE, x = X, c = C)
summary(rd_covs)

rdplot(y = frm$AGE, x = X, c = C,
  title = "Covariate Balance - RD Plot",
  x.label = "Systolic Blood Pressure (mmHg)",
  y.label = "Age")
```

Step 7: Account for imperfect compliance

```
# Fuzzy RD model:
RD_fuzzy = rdrobust(y = Y, x = X, c = C, fuzzy = D)
summary(RD_fuzzy)

# Next, manually calculate this fuzzy RD effect by running the below 2 steps:
# First, assess sharp RDD effect within bandwidth from the above fuzzy model
RD_sharp = rdrobust(y = Y, x = X, c = C, h = RD_fuzzy$bws[1,1], b = RD_fuzzy$bws[2
,1], subset=!is.na(D))
summary(RD_sharp)
RD_sharp$Estimate[1] # Sharp RDD effect = -0.005580354

# Next, assess compliance by computing RD effect using D as the outcome
RD_comp = rdrobust(y = D, x = X, c = C, h = RD_fuzzy$bws[1], b = RD_fuzzy$bws[2,1]
)
summary(RD_comp)
RD_comp$Estimate[1] # Compliance RDD effect = 0.02424286
RD_comp$beta_p_r[1] # Treatment Take-up for Treatment Group = 0.08062456
RD_comp$beta_p_l[1] # Treatment Take-up for Control Group = 0.0563817
RD_comp_plot = rdplot(y = D, x = X, c = C, h = RD_fuzzy$bws[1,1], p=1,
                      x.label = "Systolic Blood Pressure (mmHg)",
```

A3. Supplemental Stata Code for RDD Application

Box A3: Stata code

Sample statistical codes (in Stata) used in RDD analysis

Required Stata packages include ***rdrobust***[1,2] and ***rddensity***[3] (https://github.com/rdpackages). The statistical codes presented below are based on the RDD application investigating the causal effect of blood pressure medication (BPMEDS) on CVD.

Step 1: Identify main components of the RD model

```
global Y cvd    /* outcome variable */
global X sysbp  /* running variable  */
global D bpmeds /* actual treatment take-up indicator */
global C 140        /* cutoff value */
gen T = 1*($X>=$C)        /* treatment assignment indicator */
```

Step 2: Plot the RD data (global fit)

```
rdplot $Y $X, c($C) ///
graph_options(title("") ///
      ytitle(Cardiovascular Disease at Follow-up) ///
      xtitle(Systolic Blood Pressure (mmHg)) ///
      graphregion(color(white)) legend(off))
```

Step 3: Set up the local polynomial model

```
local P  = 1             /* local linear approximation */
local K  = "triangular"  /* triangular kernel  */
local BW = "mserd"       /* MSE optimal bandwidth*/
```

Steps 4-5: RD estimation and inference

```
rdrobust $Y $X, c($C) kernel(`K') p(`P') bwselect(`BW')

* You can also run RD using default choices, as shown here:
rdrobust $Y $X, c($C)

We can re-plot our RD data (as in Step 2) but now using our choices for the estima
tion of the RD model instead of a global fit.

rdplot $Y $X if -e(h_l)<= $X-$C & $X-$C <= e(h_r), c($C) kernel(`K') p(`P') bwsele
ct(`BW') ///
      h(`e(h_l)' `e(h_r)')  ///
      graph_options(title("") ///
      ytitle(Cardiovascular Disease at Follow-up) ///
      xtitle(Systolic Blood Pressure (mmHg)) ///
      graphregion(color(white)) legend(off))
```

Box A3 (continued)

Sample statistical codes (in Stata) used in RDD analysis – continued

Step 6: Robustness checks and validation (a comprehensive list can be found in the Appendix)

```
* Change to uniform kernel
rdrobust $Y $X, c($C) kernel("uniform") p(1) bwselect("mserd")

* Change to different bandwidths on each side:
rdrobust $Y $X, c($C) kernel("triangular") p(1) bwselect("msetwo")

* Test for manipulation of the running variable: rddensity test
rddensity $X, c($C) plot ///
                      graph_opt(title("") ///
                     ytitle(Cardiovascular Disease at Follow-up) ///
                     xtitle(Systolic Blood Pressure (mmHg)) ///
                     graphregion(color(white)) legend(off))

** Covariate balance assessment (Age)
rdrobust age $X, c($C)

rdplot age $X, c($C) ///
                    graph_options(title("Covariate Balance - RD Plot") ///
                     ytitle(Age) ///
                     xtitle(Systolic Blood Pressure (mmHg)) ///
                     graphregion(color(white)) legend(off))
```

Step 7: Account for imperfect compliance

```
# First, assess compliance by computing RD effect using D as the outcome
rdrobust $D $X, c($C)

disp "Treatment take-up for the treatment group: " e(tau_cl_r)
disp "Treatment take-up for the control group: " e(tau_cl_l)
disp "Compliance rate: " e(tau_cl_r)-e(tau_cl_l)

rdplot $D $X, c($C) p(1) ///
                      graph_options(title("First Stage") ///
                     ytitle(Use of anti-hypertensive medication) ///
                     xtitle(Systolic Blood Pressure (mmHg)) ///
                     graphregion(color(white)) legend(off))

# Run Fuzzy RD model:
rdrobust $Y $X, c($C) fuzzy($D)
```

A4. List of Statistical Software for RD Designs

- **rdrobust**: RD plots, estimation, robust inference, bandwidth selection and other related features for RD designs employing local polynomial and partitioning methods. It is described in more detail in Calonico et al. (2017). Stata and R packages: https://rdpackages.github.io/rdrobust/

- **frdboot**: iterative bootstrap from Bartalotti et al. (2017) and He and Bartalotti (2020). R package: github.com/yhe0802/FRD-bootstrap/

- **rdexo**: implements external validity tests of Bertanha and Imbens (2020). Stata and Matlab packages: https://sites.google.com/site/mbertanha/home/code-bertanha-imbens.

- **rdlocrand**: statistical inference and graphical procedures for RD designs employing local randomization methods. It provides point estimators, confidence intervals estimators, windows selectors, automatic plots, sensitivity analysis and other related features. Stata and R packages: https://rdpackages.github.io/rdlocrand/

- **optrdd**: optimized inference in RD designs, as proposed by Imbens and Wager (2019). R package: https://github.com/swager/optrdd

- **rdhonest**: honest confidence intervals in fuzzy and sharp RD designs using procedures from Armstrong and Kolesár (2020). R package: https://github.com/kolesarm/RDHonest

- **rdmulti**: RD plots, estimation, inference and extrapolation methods for RD designs with multiple cutoffs and multiple scores. Stata and R packages: https://rdpackages.github.io/rdmulti/

- **rddensity**: manipulation tests employing local polynomial density estimation methods. It provides hypothesis tests and bandwidth selectors for manipulation testing. Stata and R packages: https://rdpackages.github.io/rddensity/

- **rdcont**: continuity test for the density of the running variable based on g-order statistics developed in Bugni and Canay (2020). Stata package: https://bitbucket.org/iacanay/rdcont-stata/overview

References

1. Calonico S, Cattaneo MD, Titiunik R. rdrobust: An R Package for Robust Nonparametric Inference in Regression-Discontinuity Designs. *R Journal.* 2015;7:38-51.
2. Calonico S, Cattaneo MD, Farrell MH, Titiunik R. Rdrobust: Software for Regression-discontinuity Designs. *The Stata Journal.* 2017;17(2):372-404.
3. Cattaneo MD, Jansson M, Ma X. Manipulation Testing Based on Density Discontinuity. *Stata Journal.* 2018;18:234-261.
4. Tsao CW, Vasan RS. Cohort Profile: The Framingham Heart Study (FHS): overview of milestones in cardiovascular epidemiology. *International journal of epidemiology.* 2015;44(6):1800-1813.
5. Calonico S, Cattaneo MD, Farrell M. Optimal bandwidth choice for robust bias-corrected inference in regression discontinuity designs. *The Econometrics Journal.* 2019.
6. Calonico S, Cattaneo M, Titiunik R. Robust Data-Driven Inference in the Regression-Discontinuity Design. *Stata J.* 2014;14.