mensik_assignment_3

Table of contents

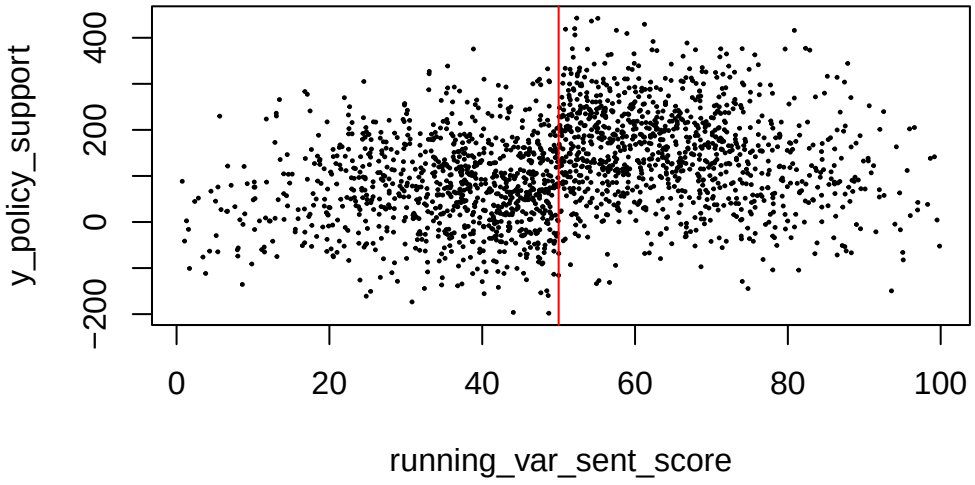
Note: Pretending I am interested in whether experiences with the the CJ system change how people think about policy/politics. I want to know if *being incarcerated* or crossing some threshold of exposure causes people to see the world differently. Pretending that there is such a thing as a 0-100 sentencing score that determines whether someone is sentenced to prison time or parole at 50 (at or greater means prison, less means parole), and that is my effectively arbitrary cutoff turned randomization device. (A bad example in a lot of ways)

So:

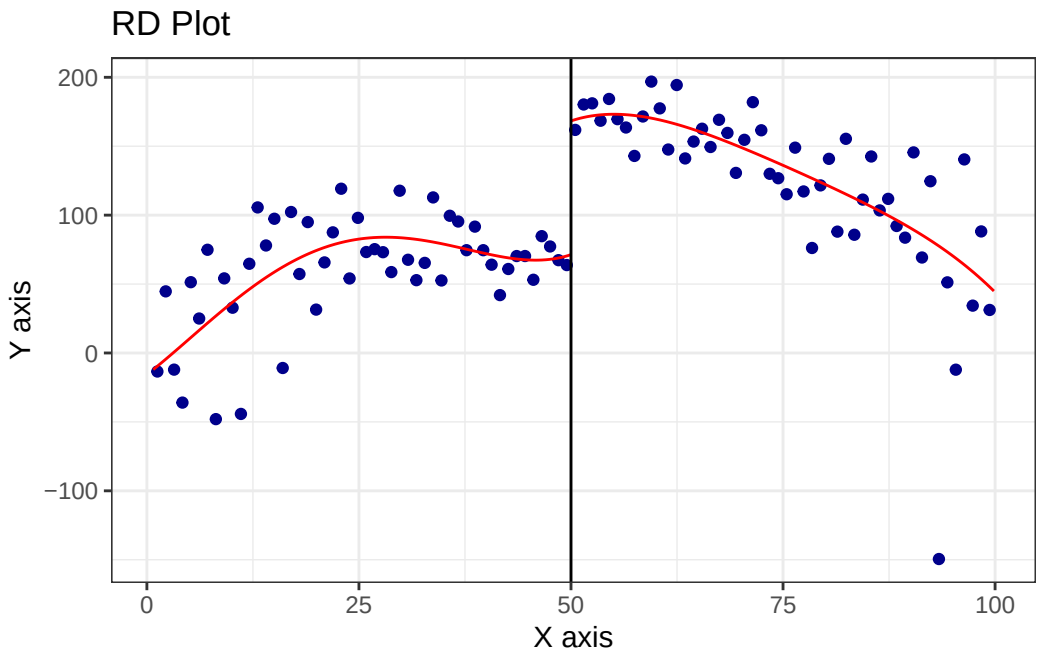
- running/forcing/assignment variable is sentencing score
- cutoff: where treatment rule switches is 50 points
- treatment assignment: defendants above 50 get treatment, below do not
- outcome: policy support
- key assumption: all traits (both observed and unobserved) other than sentencing score change smoothly with the score, except for treatment
- estimation: compare predicted outcome (policy support) on either side of 50 (the "jump") using local regressions

```
set.seed(15000)

# set up simulation for the running variable
running_var_sent_score <- rnorm(2000, 50, 20) #normally distributed with mean 50, sd 20
running_var_sent_score <- running_var_sent_score[running_var_sent_score >= 0 & running_var_se

# create the treatment assignment -- here is where "defendents whose sentencing scores is les
D_prison_time <- ifelse(running_var_sent_score > 50, 1, 0)
```

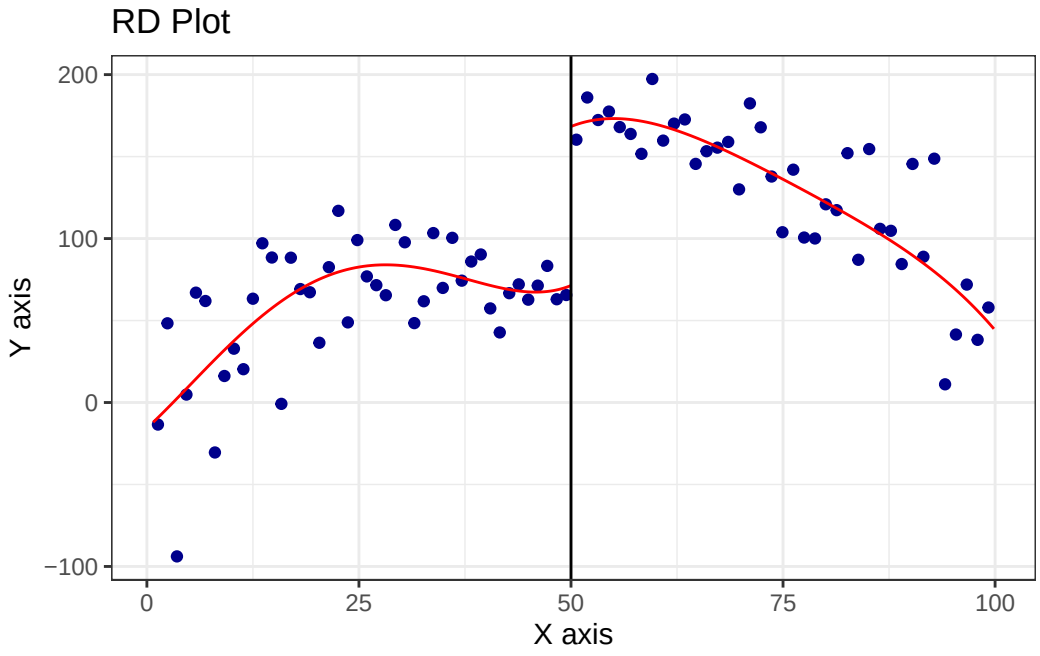
```r
# simulated outcome (observed y, via the 'true' GDP); added treatment effect of 100 for those
# baseline policy support is 45, being treated increases support by 100 units ("the jump")
# 0.022 * running_var means outcome increases by 2.2 points for every one unit increase in s
y_policy_support <- 45 + D_prison_time * 100 + 1.5 * running_var_sent_score - 0.022 * running

plot(running_var_sent_score, y_policy_support, pch = 19, cex = 0.2)
abline(v = 50, col = "red")
```



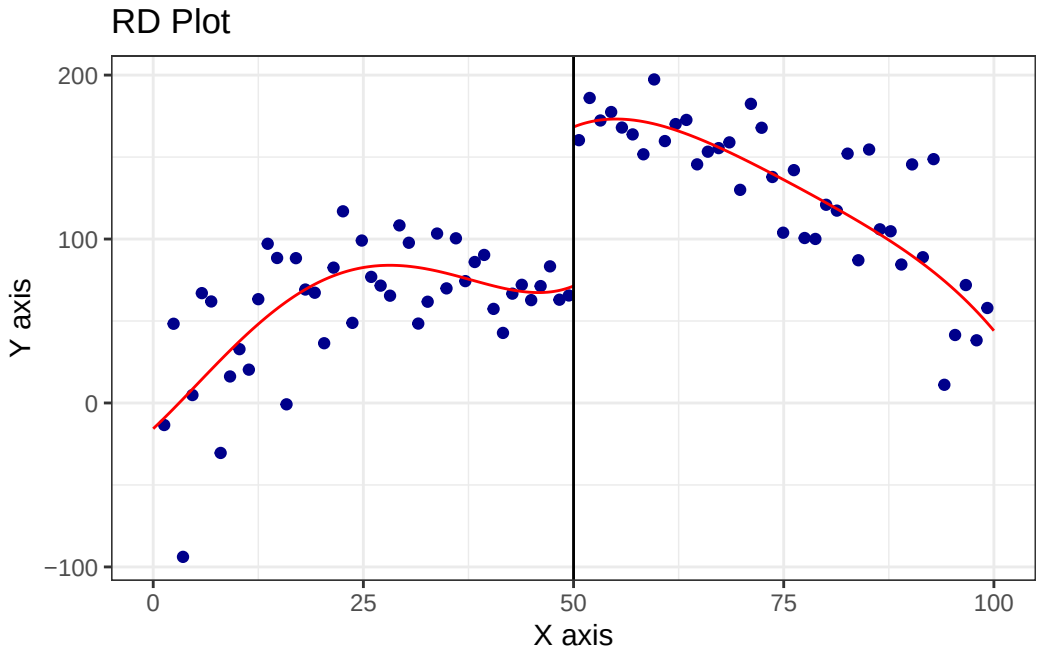

```r
# Bin the data and fit local polynomials... plot (using 50 bins left and right of c_0)
# with p=2 polynomial fitted below and above c_0
# (h = full span of data)
# rdplot bins data to left and right of cutoff sepearately and fits polynomia of order p
# uses 4th degree polynomia and 50 bins on each side (show binned means with fitted lines),
rdplot(y = y_policy_support, x = running_var_sent_score, c = 50, p = 4, nbins = c(50, 50))
```



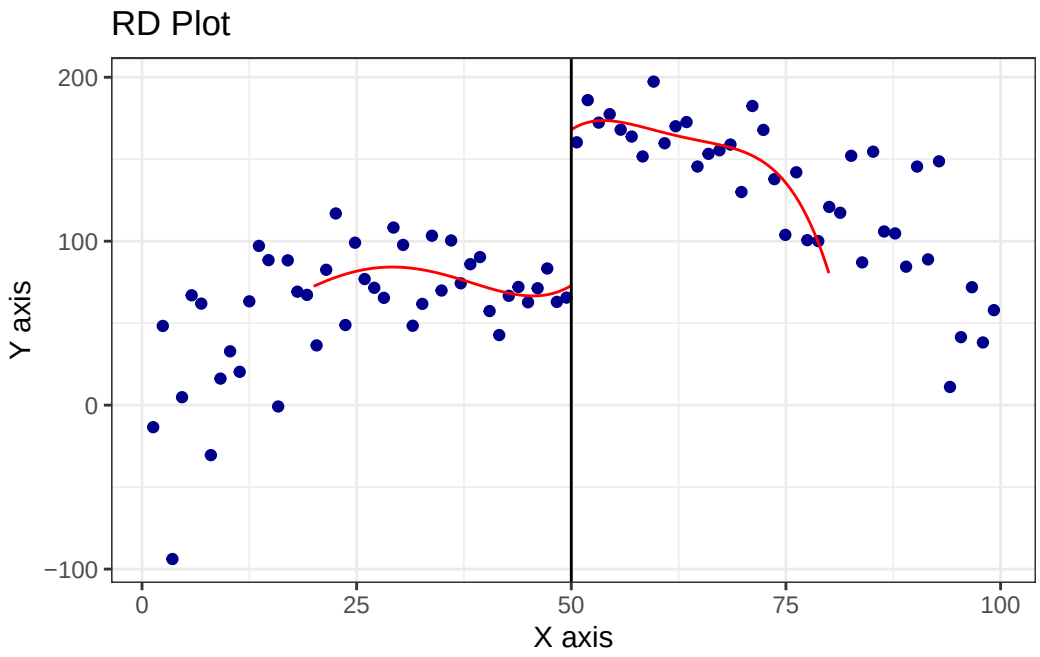

```
# automatic bin N selection
# (selects MSE optimal choice)
rdplot(y = y_policy_support, x = running_var_sent_score, c = 50, binselect = "esmv")
```



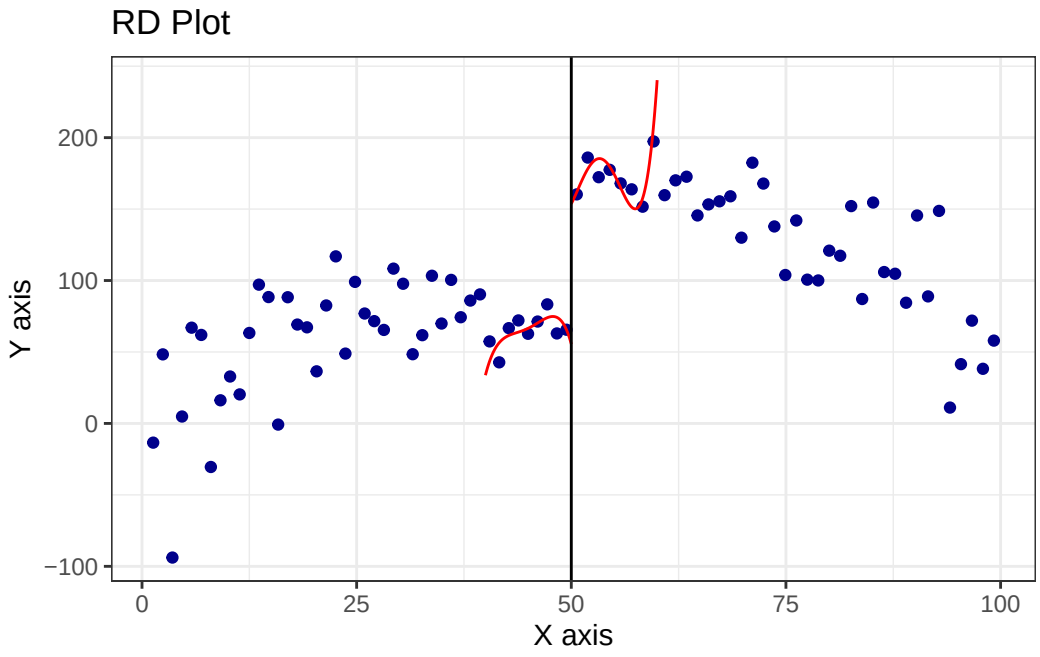
```r
# Illustration: the role of h (varying the bandwith to see the local nature)
rdplot(y = y_policy_support, x = running_var_sent_score, c = 50, binselect = "esmv", h = 50)
```




```r
rdplot(y = y_policy_support, x = running_var_sent_score, c = 50, binselect = "esmv", h = 30)
```




```
rdplot(y = y_policy_support, x = running_var_sent_score, c = 50, binselect = "esmv", h = 10)
```



```r
# BW selection
summary(rdbwselect(y = y_policy_support, x = running_var_sent_score, c = 50, bwselect = "mse
```

```
           Length Class  Mode
bws        4      -none- numeric
bwselect   1      -none- character
bw_list    1      -none- character
kernel     1      -none- character
p          1      -none- numeric
q          1      -none- numeric
c          1      -none- numeric
N          2      -none- numeric
N_h        2      -none- numeric
M          2      -none- numeric
vce        1      -none- character
masspoints 1      -none- character
call       5      -none- call
```

```r
summary(rdbwselect(y = y_policy_support, x = running_var_sent_score, c = 50, bwselect = "mse
```

```
           Length Class  Mode
bws        4      -none- numeric
bwselect   1      -none- character
bw_list    1      -none- character
kernel     1      -none- character
p          1      -none- numeric
q          1      -none- numeric
c          1      -none- numeric
N          2      -none- numeric
N_h        2      -none- numeric
M          2      -none- numeric
vce        1      -none- character
masspoints 1      -none- character
call       5      -none- call
```

```r
# estimate RDD treatment effect
# p = 1 (locally linear)
views_m1 <- rdrobust(y = y_policy_support, x = running_var_sent_score, c = 50, p = 1, bwselec
```

```r
# Simple RDD p = 2
views_m2 <- rdrobust(y = y_policy_support, x = running_var_sent_score, c = 50, p = 2, bwsele
summary(views_m1)
```

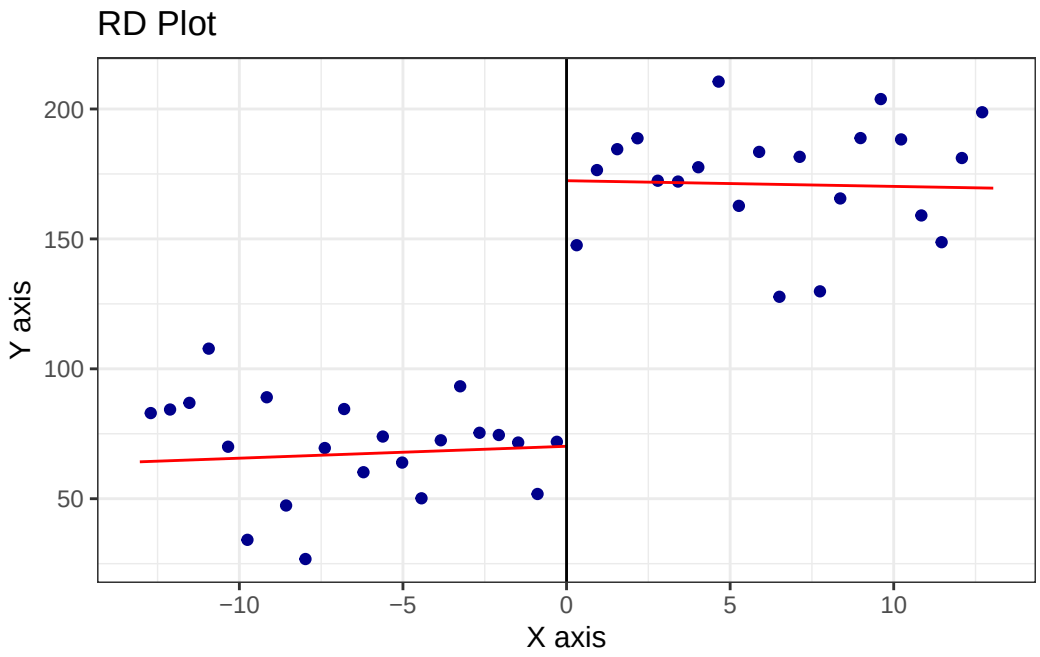
```
           Length Class  Mode
Estimate   4      -none- numeric
bws        4      -none- numeric
coef       3      -none- numeric
se         3      -none- numeric
z          3      -none- numeric
pv         3      -none- numeric
ci         6      -none- numeric
beta_Y_p_l 2      -none- numeric
beta_Y_p_r 2      -none- numeric
V_cl_l     4      -none- numeric
V_cl_r     4      -none- numeric
V_rb_l     4      -none- numeric
V_rb_r     4      -none- numeric
N          2      -none- numeric
N_h        2      -none- numeric
N_b        2      -none- numeric
M          2      -none- numeric
tau_cl     2      -none- numeric
tau_bc     2      -none- numeric
c          1      -none- numeric
p          1      -none- numeric
q          1      -none- numeric
bias       2      -none- numeric
kernel     1      -none- character
detail     0      -none- NULL
all        0      -none- NULL
vce        1      -none- character
bwselect   1      -none- character
level      1      -none- numeric
masspoints 1      -none- character
rdmodel    1      -none- character
beta_covs  0      -none- NULL
call       6      -none- call
```

```r
summary(views_m2)
```

```
           Length Class  Mode
```

```
Estimate   4      -none- numeric
bws        4      -none- numeric
coef       3      -none- numeric
se         3      -none- numeric
z          3      -none- numeric
pv         3      -none- numeric
ci         6      -none- numeric
beta_Y_p_l 3      -none- numeric
beta_Y_p_r 3      -none- numeric
V_cl_l     9      -none- numeric
V_cl_r     9      -none- numeric
V_rb_l     9      -none- numeric
V_rb_r     9      -none- numeric
N          2      -none- numeric
N_h        2      -none- numeric
N_b        2      -none- numeric
M          2      -none- numeric
tau_cl     2      -none- numeric
tau_bc     2      -none- numeric
c          1      -none- numeric
p          1      -none- numeric
q          1      -none- numeric
bias       2      -none- numeric
kernel     1      -none- character
detail     0      -none- NULL
all        0      -none- NULL
vce        1      -none- character
bwselect   1      -none- character
level      1      -none- numeric
masspoints 1      -none- character
rdmodel    1      -none- character
beta_covs  0      -none- NULL
call       6      -none- call
```

```r
# Construct plot (using obs within selected bw)
# xc = x - 50 centers the running variable so that the cutoff is at 0;
xc <- running_var_sent_score - 50
m <- rdrobust(y = y_policy_support, x = xc, p = 1)
rdplot(y = y_policy_support, x = xc,
    subset = -m$bws[1,1]<= xc & xc <= m$bws[1,2],
    binselect = "esmv", kernel = "triangular",
    h = c(m$bws[1,1], m$bws[1,2]), p = 1)
```



The RD effect of 93.706 means that crossing from a sentencing score of less than 50 to 50 increases policy support by 93.706 points, and if I repreated the whole study many times and built 95% confidence intervals each time, 95% of those intervals would contain the true effect of a jump between 56.815 and 124.840 points. The p score is 0 – the effect is significant.

Part 2

Adding a confounder Z that, in first scenario where continuity holds, would need to change gradually with running variable/not be discontinuous at the 50 juncture.

```
Z_noproblem <- running_var_sent_score*0.1 +
  rnorm(length(running_var_sent_score), 0, 500) # plus noise one draw per observation


y_confounded <- 45 + D_prison_time * 100 + 1.5 * running_var_sent_score +
                0.022 * Z_noproblem - 0.015 * running_var_sent_score^2 +
                rnorm(length(running_var_sent_score), 0, 100) #adjusted back


# local linear model

## y_confounded
m_a <- rdrobust(y = y_confounded, x = running_var_sent_score, c = 50, p = 1, bwselect = "mse
summary(m_a)
```

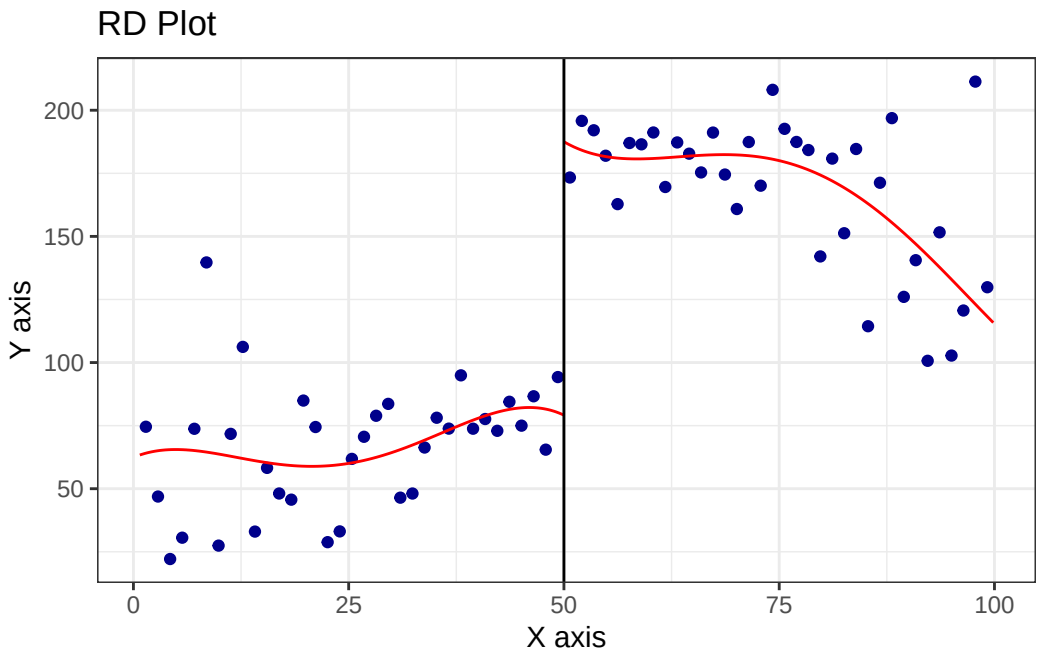
```
           Length Class  Mode
Estimate   4      -none- numeric
bws        4      -none- numeric
coef       3      -none- numeric
se         3      -none- numeric
z          3      -none- numeric
pv         3      -none- numeric
ci         6      -none- numeric
beta_Y_p_l 2      -none- numeric
beta_Y_p_r 2      -none- numeric
V_cl_l     4      -none- numeric
V_cl_r     4      -none- numeric
V_rb_l     4      -none- numeric
V_rb_r     4      -none- numeric
N          2      -none- numeric
N_h        2      -none- numeric
N_b        2      -none- numeric
M          2      -none- numeric
tau_cl     2      -none- numeric
tau_bc     2      -none- numeric
c          1      -none- numeric
p          1      -none- numeric
q          1      -none- numeric
bias       2      -none- numeric
kernel     1      -none- character
detail     0      -none- NULL
all        0      -none- NULL
vce        1      -none- character
bwselect   1      -none- character
level      1      -none- numeric
masspoints 1      -none- character
rdmodel    1      -none- character
beta_covs  0      -none- NULL
call       6      -none- call
```

```
# local quadratic model

## y_confounded
m_c <- rdrobust(y = y_confounded, x = running_var_sent_score, c = 50, p = 2, bwselect = "mse
summary(m_c)
```

```
           Length Class  Mode
```

```
Estimate   4      -none- numeric
bws        4      -none- numeric
coef       3      -none- numeric
se         3      -none- numeric
z          3      -none- numeric
pv         3      -none- numeric
ci         6      -none- numeric
beta_Y_p_l 3      -none- numeric
beta_Y_p_r 3      -none- numeric
V_cl_l     9      -none- numeric
V_cl_r     9      -none- numeric
V_rb_l     9      -none- numeric
V_rb_r     9      -none- numeric
N          2      -none- numeric
N_h        2      -none- numeric
N_b        2      -none- numeric
M          2      -none- numeric
tau_cl     2      -none- numeric
tau_bc     2      -none- numeric
c          1      -none- numeric
p          1      -none- numeric
q          1      -none- numeric
bias       2      -none- numeric
kernel     1      -none- character
detail     0      -none- NULL
all        0      -none- NULL
vce        1      -none- character
bwselect   1      -none- character
level      1      -none- numeric
masspoints 1      -none- character
rdmodel    1      -none- character
beta_covs  0      -none- NULL
call       6      -none- call
```

```r
## y_confounded plot
rdplot(y = y_confounded, x = running_var_sent_score, c = 50, binselect = "esmv")
```




```
### then

#with dummy to jump at 50, by 50
Z_problem <- 0.1*running_var_sent_score + 50 * ifelse(running_var_sent_score > 50, 1, 0) +
               rnorm(length(running_var_sent_score), 0, 10)

y_bad_confounded <- 45 + D_prison_time * 100 + 1.5 * running_var_sent_score +
                0.022 * Z_problem - 0.015 * running_var_sent_score^2 +
                rnorm(length(running_var_sent_score), 0, 100)



# local linear model

## y_bad_confounded
m_b <- rdrobust(y = y_bad_confounded, x = running_var_sent_score, c = 50, p = 1, bwselect = '
summary(m_b)
```

```
          Length Class  Mode
Estimate  4      -none- numeric
bws       4      -none- numeric
coef      3      -none- numeric
```

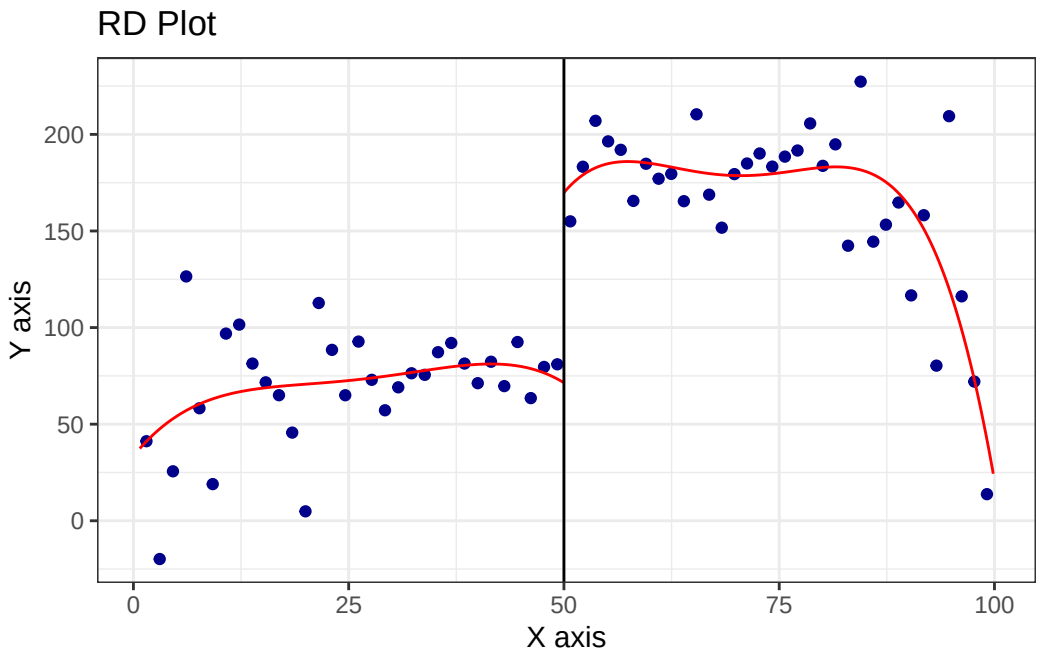
```
se         3      -none- numeric
z          3      -none- numeric
pv         3      -none- numeric
ci         6      -none- numeric
beta_Y_p_l 2      -none- numeric
beta_Y_p_r 2      -none- numeric
V_cl_l     4      -none- numeric
V_cl_r     4      -none- numeric
V_rb_l     4      -none- numeric
V_rb_r     4      -none- numeric
N          2      -none- numeric
N_h        2      -none- numeric
N_b        2      -none- numeric
M          2      -none- numeric
tau_cl     2      -none- numeric
tau_bc     2      -none- numeric
c          1      -none- numeric
p          1      -none- numeric
q          1      -none- numeric
bias       2      -none- numeric
kernel     1      -none- character
detail     0      -none- NULL
all        0      -none- NULL
vce        1      -none- character
bwselect   1      -none- character
level      1      -none- numeric
masspoints 1      -none- character
rdmodel    1      -none- character
beta_covs  0      -none- NULL
call       6      -none- call
```

```
# local quadratic model

## y_bad_confounded
m_d <- rdrobust(y = y_bad_confounded, x = running_var_sent_score, c = 50, p = 2, bwselect = '
summary(m_d)
```

```
          Length Class  Mode
Estimate  4      -none- numeric
bws       4      -none- numeric
coef      3      -none- numeric
se        3      -none- numeric
```

```
z          3      -none- numeric
pv         3      -none- numeric
ci         6      -none- numeric
beta_Y_p_l 3      -none- numeric
beta_Y_p_r 3      -none- numeric
V_cl_l     9      -none- numeric
V_cl_r     9      -none- numeric
V_rb_l     9      -none- numeric
V_rb_r     9      -none- numeric
N          2      -none- numeric
N_h        2      -none- numeric
N_b        2      -none- numeric
M          2      -none- numeric
tau_cl     2      -none- numeric
tau_bc     2      -none- numeric
c          1      -none- numeric
p          1      -none- numeric
q          1      -none- numeric
bias       2      -none- numeric
kernel     1      -none- character
detail     0      -none- NULL
all        0      -none- NULL
vce        1      -none- character
bwselect   1      -none- character
level      1      -none- numeric
masspoints 1      -none- character
rdmodel    1      -none- character
beta_covs  0      -none- NULL
call       6      -none- call
```

```r
# plot

## y_bad_confounded
rdplot(y = y_bad_confounded, x = running_var_sent_score, c = 50, binselect = "esmv")
```



Re-running both scenarios, the smooth confounder stays closer to the true effect but the problematic/discontinuous jumper exerts negative bias on the effect. The discontinuity contaminates the estimate because it reflects the treatment plus the jump.

Third – the characteristics of the RDD estimator (e.g., using linear vs higher-order local regression) when the data is noisy. What can you learn about false positive results?

Addint noise to the unproblematically confounded estimator increases the p value, unsurprisingly distorts point estimates.

(tried to set up MC but ran out of time)