rdd_Apr1_DM.R

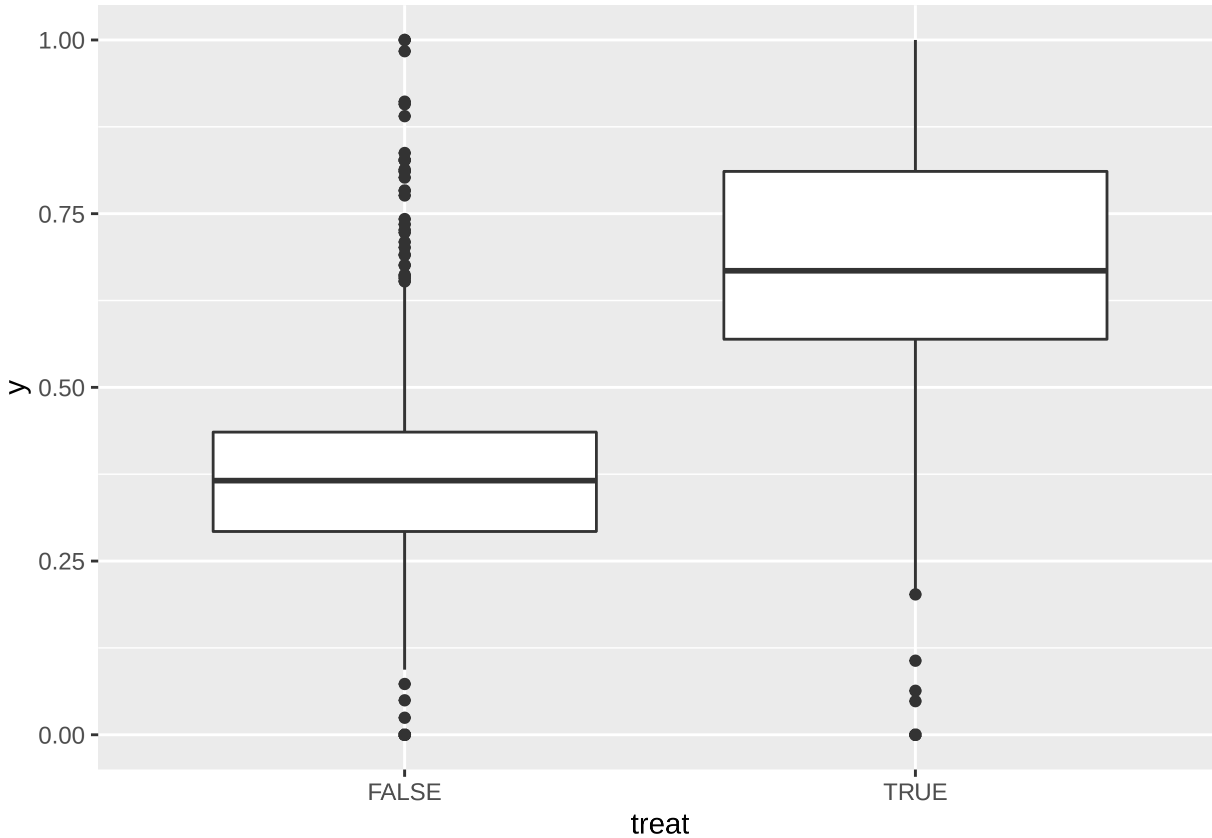
danny

2020-03-31

```r
# Author: Gordon Burtch and Gautam Ray
# Course: MSBA 6440
# Session: Regression Discontinuity
# Topic: RDD Example
# Lecture 8

suppressWarnings(suppressPackageStartupMessages({
library(rdrobust)
library(rdd)
library(ggplot2)
}))

# Dataset used by Lee (2008), which is a paper that talks about the "incumbency advantage" in US politi
# If I hold a congressional seat right now, to what degree does that increase my party's chances of win
# The discontinuity design here is based on vote share in the *last* election. Essentially, I am "assig
# If I won the last election, and not if not. Whether I win an election is based on a simple majority t
# Usually it means I passed 50% of the vote. So, we're going to use that 50% cutoff in the last electio
# winning last time, on winning this time (versus losing)

HouseData <-read.csv("house.csv")

# Let's define our treatment variable (0 = equal proportion of vote in last election)
HouseData$treat <- (HouseData$x>0)

# Let's run the endogenous regression first... says 35% increase in vote share due to winning last elec
# Of course we know that's wrong...
ols <- lm(data=HouseData,y~treat)
summary(ols)
```

```
##
## Call:
## lm(formula = y ~ treat, data = HouseData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69788 -0.10061 -0.00360  0.09631  0.65348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.346522   0.003201  108.25   <2e-16 ***
## treatTRUE   0.351358   0.004195   83.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1676 on 6556 degrees of freedom
## Multiple R-squared:  0.5169, Adjusted R-squared:  0.5168
## F-statistic:  7014 on 1 and 6556 DF,  p-value: < 2.2e-16
```

```r
# What is this OLS actually estimating? It's a t-test comparing vote outcomes in current election betwe
ggplot(data=HouseData) + geom_boxplot(aes(y=y,x=treat))
```

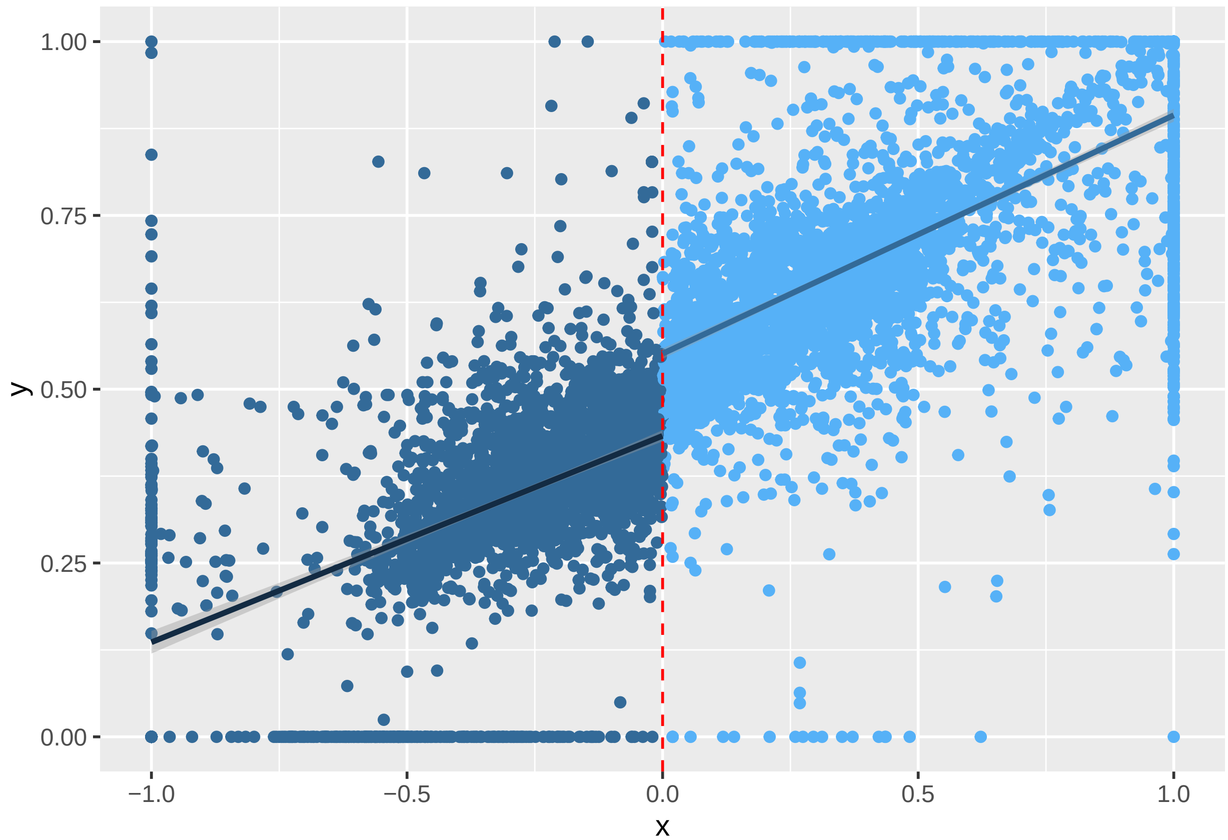



```r
# Let's try RDD now, where we condition on the relationship between y and x, to get at the effect right
# By using "all" the data we are implicitly using a maximum bandwidth (use all of the range of x around
# This says the local treatment effect is 0.11 (11% increase in vote outcome due to the discontinuity).
ols <- lm(data=HouseData,y~treat + x)
summary(ols)
```

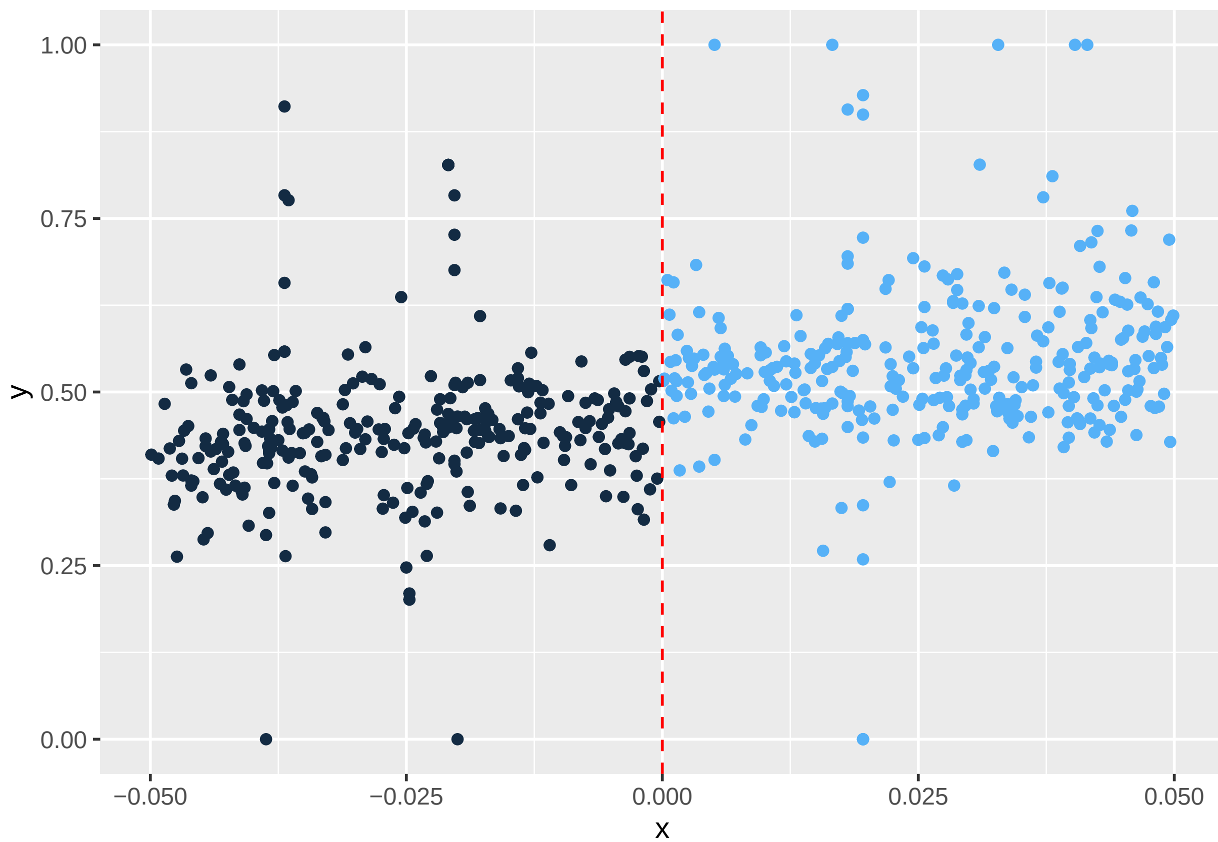
```
##
## Call:
## lm(formula = y ~ treat + x, data = HouseData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88700 -0.06324  0.00027  0.07082  0.88780
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.442736   0.003168  139.75   <2e-16 ***
## treatTRUE   0.113728   0.005528   20.57   <2e-16 ***
## x           0.330533   0.005989   55.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1385 on 6555 degrees of freedom
## Multiple R-squared:  0.6701, Adjusted R-squared:   0.67
## F-statistic:  6658 on 2 and 6555 DF,  p-value: < 2.2e-16
```

```
# Here's a plot of what we are estimating by running this regression.
ggplot(HouseData, aes(y=y,x=x)) + geom_point(aes(col=treat+1),show.legend = FALSE) + geom_vline(xinterc
  geom_smooth(aes(group=treat,col=as.numeric(treat)),method = "lm",show.legend=FALSE)
```



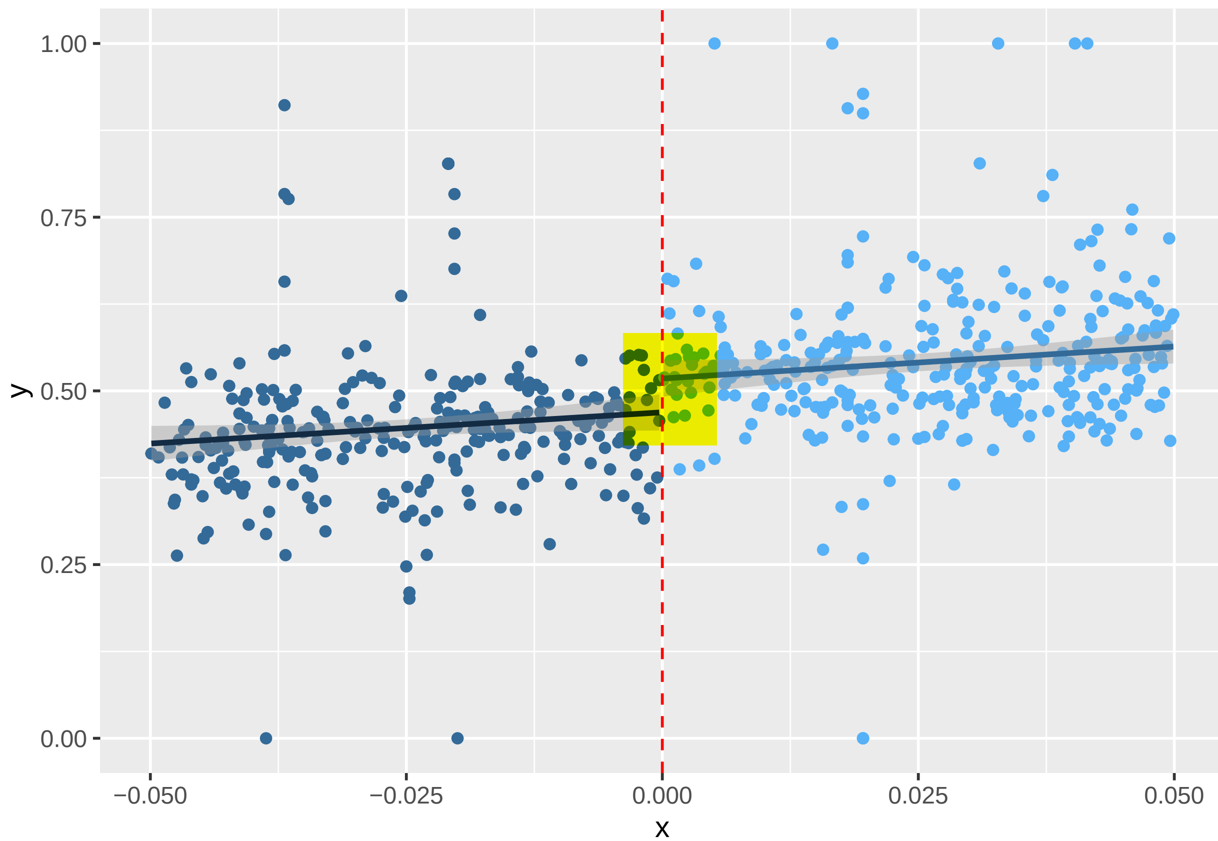

```
# But we don't really believe that incumbents and prior losers are comparable "generally", so we don't
# for the same reason we don't trust the OLS more generally...
# We *might* believe the comparison is fair right around the election win threshold, however.
# Here, we are zooming in to a 5% differential on either side of the cutoff, h = 0.05.
Pared_House <- HouseData[HouseData$x >= -0.05 & HouseData$x <= 0.05,]
ggplot(Pared_House, aes(y=y,x=x,col=as.numeric(treat))) + geom_point(show.legend = FALSE) + geom_vline(
```




```r
# Looks better...

# Okay let's run our RDD regression now. Our estimate falls to about 5% with this tighter bandwidth.
house_rdd <- lm(data=Pared_House,y ~ treat + x)
summary(house_rdd)
```

```
##
## Call:
## lm(formula = y ~ treat + x, data = Pared_House)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53590 -0.05496 -0.00756  0.03574  0.47729
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.46942    0.01046  44.872  < 2e-16 ***
## treatTRUE    0.04865    0.01881   2.586  0.00995 **
## x            0.90988    0.31979   2.845  0.00459 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1113 on 607 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1625
## F-statistic: 60.09 on 2 and 607 DF,  p-value: < 2.2e-16
```

```r
ggplot(Pared_House, aes(y=y,x=x)) + geom_point(aes(col=treat+1),show.legend = FALSE) + geom_vline(xinte
  geom_smooth(aes(group=treat,col=as.numeric(treat)),method = "lm",show.legend=FALSE)
```




```r
# Lets try an interaction to see if the slopes are different
house_rdd_int <- lm(data=Pared_House,y ~ treat*x)
summary(house_rdd_int)
```

```
##
## Call:
## lm(formula = y ~ treat * x, data = Pared_House)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53585 -0.05515 -0.00759  0.03566  0.47747
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.46914    0.01387  33.824   <2e-16 ***
## treatTRUE    0.04870    0.01890   2.577   0.0102 *
## x            0.89896    0.47951   1.875   0.0613 .
## treatTRUE:x  0.01970    0.64394   0.031   0.9756
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1114 on 606 degrees of freedom
```

```
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1611
## F-statistic:    40 on 3 and 606 DF,  p-value: < 2.2e-16
```

```r
# Lets try a square term to see if there is any curvilinearity

Pared_House$x_sq <- Pared_House$x*Pared_House$x
house_rdd_sq <- lm(data=Pared_House,y ~ treat + x + x_sq)
summary(house_rdd_sq)
```

```
##
## Call:
## lm(formula = y ~ treat + x + x_sq, data = Pared_House)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53600 -0.05504 -0.00757  0.03580  0.47714
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.46962    0.01216  38.631  < 2e-16 ***
## treatTRUE    0.04860    0.01890   2.571  0.01037 *
## x            0.91112    0.32233   2.827  0.00486 **
## x_sq        -0.20117    6.21873  -0.032  0.97420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1114 on 606 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1611
## F-statistic:    40 on 3 and 606 DF,  p-value: < 2.2e-16
```

```r
# These days, we don't implement it all manually.
# We use packages that implement algorithms that choose bandwidth, specification and other things for u
# We probably want to use a weighting function, for example (further away from cutoff, we down-weight y
# in tandem with the optimally chosen band-width, etc.
# rdrobust() chooses everything for you, based on some cross-validation, etc.
# This says that we are still over-doing it! A more accurate estimate of the effect is actually about f
House_Robust_RDD <- rdrobust(HouseData$y,HouseData$x,c=0)
```

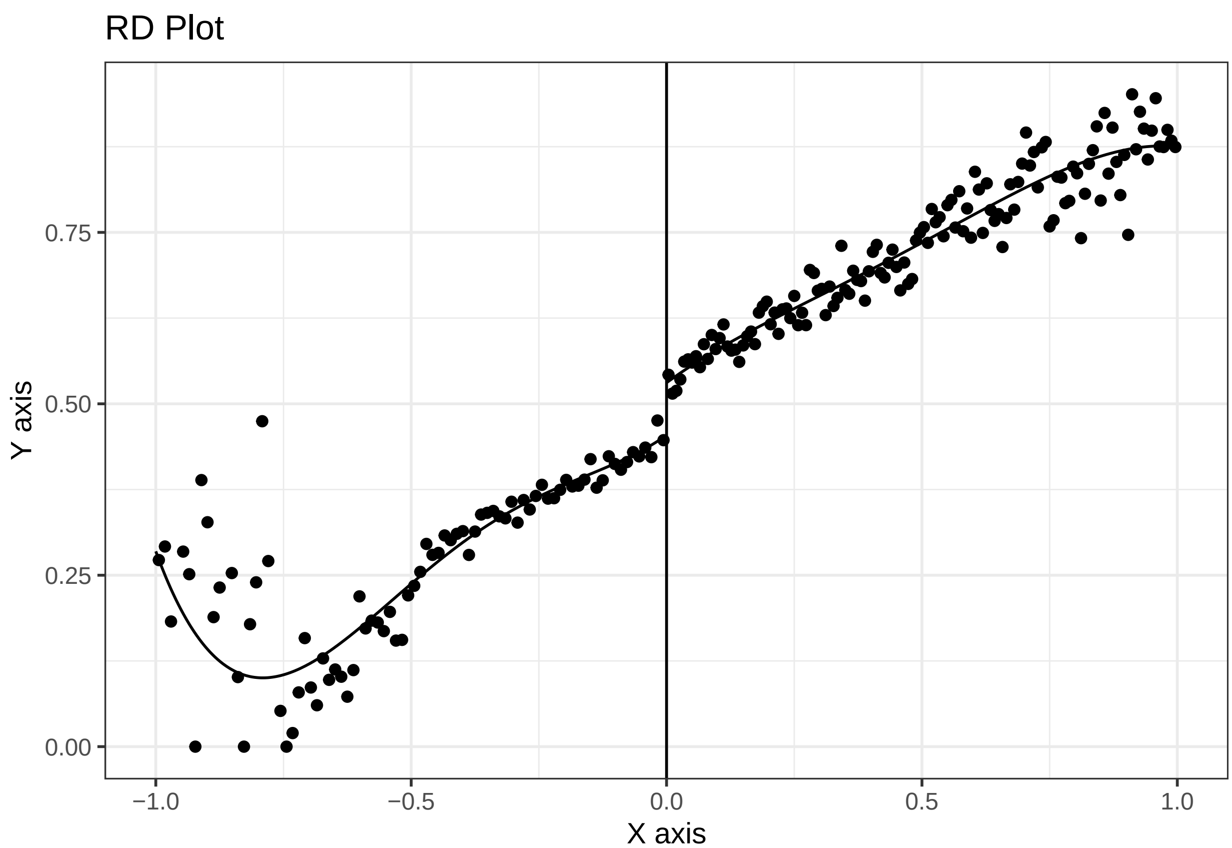
```
## [1] "Mass points detected in the running variable."
```

```r
summary(House_Robust_RDD)
```

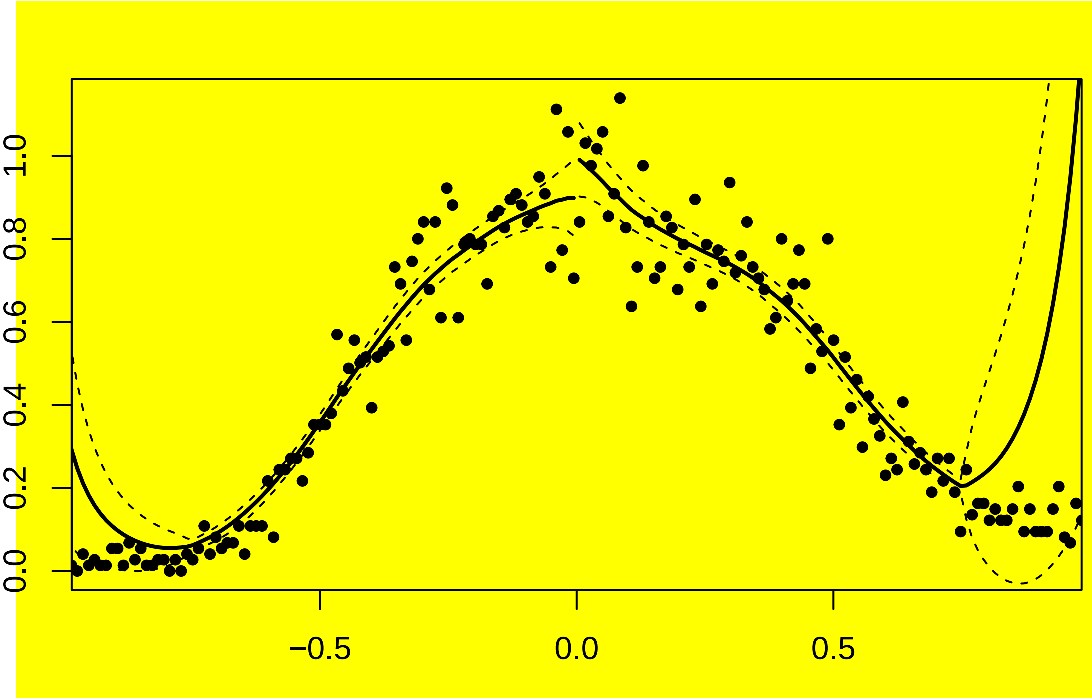
```
## Call: rdrobust
##
## Number of Obs.                 6558
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                 2740         3818
## Eff. Number of Obs.             786          816
## Order est. (p)                    1            1
```

```
## Order bias  (q)                  2           2
## BW est. (h)                  0.135       0.135
## BW bias (b)                  0.240       0.240
## rho (h/b)                    0.564       0.564
## Unique Obs.                   2108        2581
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##   Conventional     0.064     0.011     5.806     0.000     [0.042 , 0.085]
##         Robust         -         -     4.731     0.000     [0.035 , 0.084]
## =============================================================================
```

```r
rdplot(HouseData$y,HouseData$x)
```




```r
# It can't "fix" self-selection, however, so let's again run a density check around the cut-point to ev
# the number it spits out is the p-value associated with the non-parametric test of density differences
# In this case, the p-value is ~0.19, which is fairly far away from being a problem (no evidence of sor
DCdensity(HouseData$x,0,plot=TRUE)
```




```
## [1] 0.1952357
```