RD Robustness Project

Exercise: dropping data away from cutoff

Maor Milgrom

2020-03-21

Introduction

The goal of this project is to test the robustness of the Regression-Discontinuity analysis to different extreme cases, via simulations, using the 'rdrobust' package.

This Document: Excercise #3

Testing whether the RD coefficient becomes biased when we gradually drop observations in different intervals around the cutoff.

Set Parameters

Here we set the main parameters for the excercise:

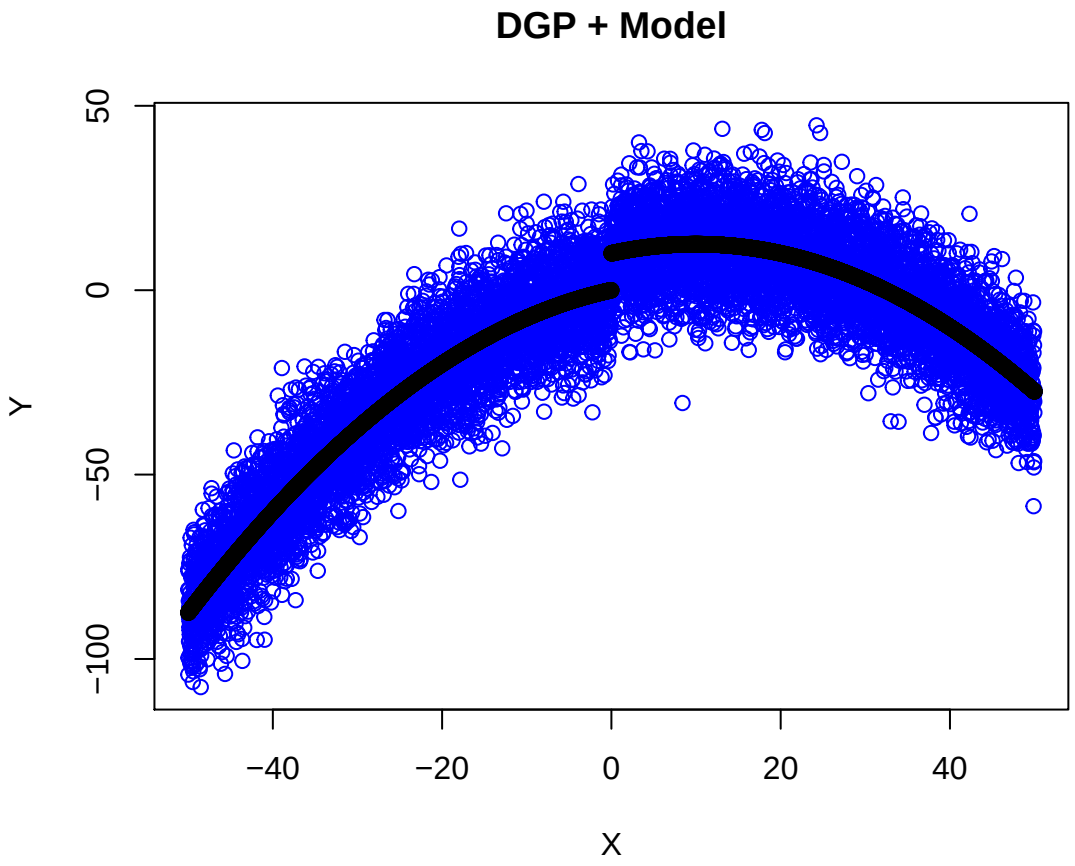
```
jump=10           # Size of jump at cutoff
loop=5
figs.iter.save=5
quadratic=T       # T - quadratic DGP, F - linear
symm_obw="mserd"  # mserd - symmetric OBW, msetwo - asymmetric OBW
normal.x=T        # T - normal draws of x around cutoff, F - uniform draws
dgp.sd=10         # sd of normal noise added to DGP
bc=F              # bias-corrected estimates or conventional
intervals=c(5,10,20,40)
var.list=c("coef.c","coef.bc","obw.c","obw.bc")
```

Constructing dataframes for simulation and for results

```
df <- as.data.frame(matrix(0, ncol = 0, nrow = length(seq(-100,100,0.01))))
df$x=round(seq(-100,100,0.01), digits=2)
df=subset(df,df$x!=0)
df$treated <- ifelse(df$x>0, 1, 0)
sample.x <- as.data.frame(matrix(0, ncol = 0, nrow = nrow(df)/10))

df.temp=as.data.frame(matrix(0, ncol = length(intervals), nrow = loop))
colnames(df.temp) = paste("int",intervals,sep="_")

for (df.name in var.list) {
  assign(df.name,df.temp)
}
```

Simulate DGP and plot

```
df$y.model<- 0.5*df$x  - 0.025*df$x^2*quadratic + jump*df$treated
df$y=df$y.model+rnorm(length(df$x),0,dgp.sd)

df %>%
  filter(x > -50 & x < 50) %T>%
  plot(y~x,., ylim = range(c(y,y.model)),
       col="blue", ylab = "Y", xlab = "X") %T>%
  par(new = T) %>%
  plot(y.model~x,., ylim = range(c(y,y.model)),
       axes = FALSE, xlab = "", ylab = "")
  title(main = "DGP + Model")
```



Iterations

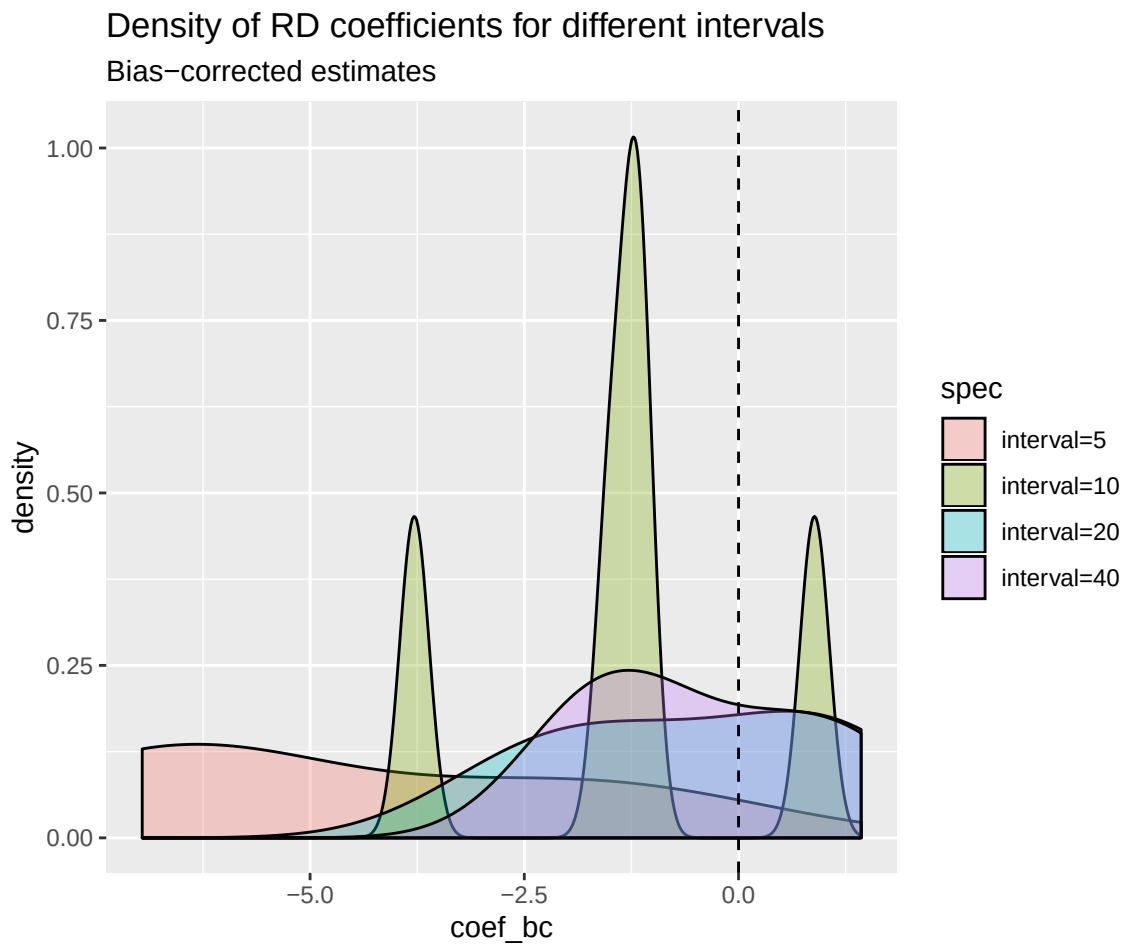
Running 5 iterations, for 4 intervals, and presenting figures from last iteration for illustration

```
for (i in 1:loop) {
  if (normal.x==T) {
    sample.x$x <- round(rnorm(nrow(df)/10, 0, 10),digits = 2)
  } else {
    sample.x$x <- round(runif(nrow(df)/10, -20,20),digits = 2)
  }
```

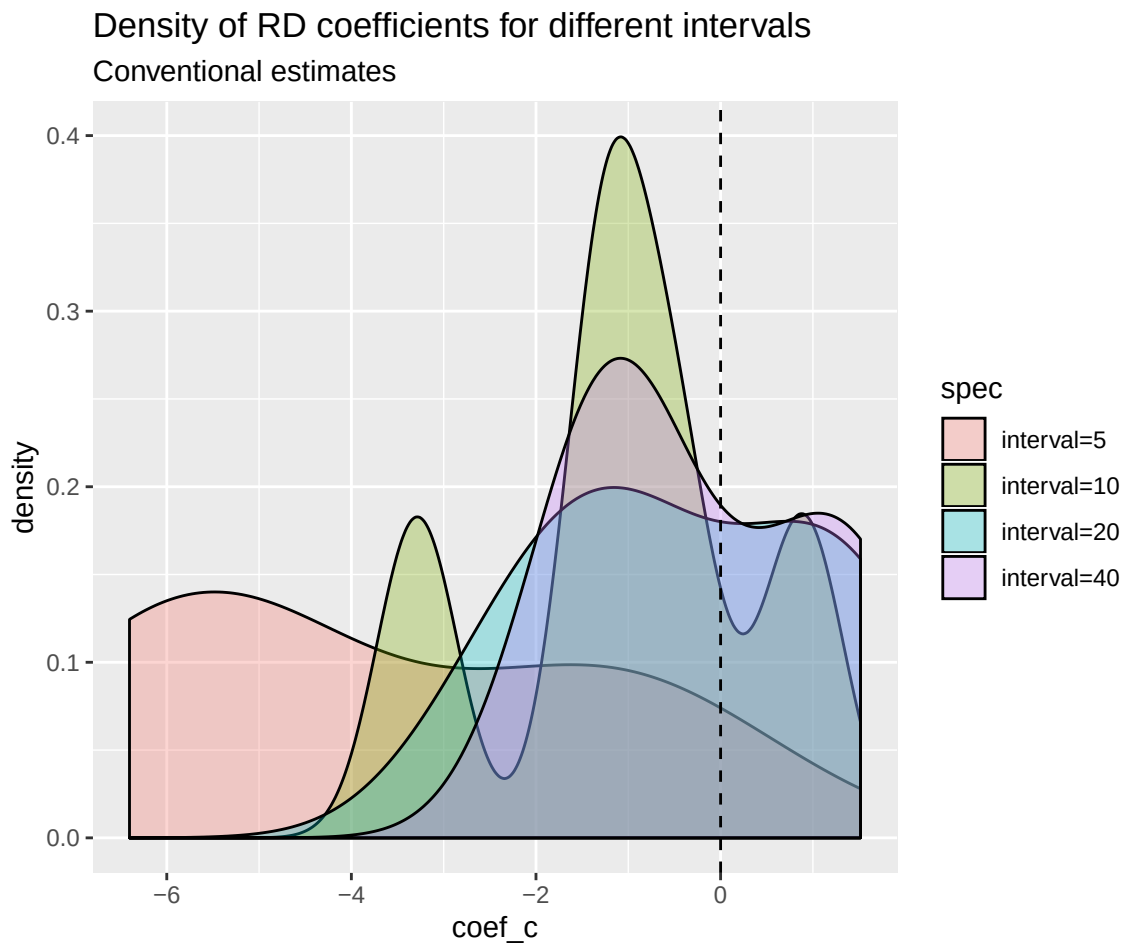
```
  sample.x=subset(sample.x, x>-100 & x<100)
  sample=as.data.frame(inner_join(df, sample.x, by="x"))

  for (j in intervals) {
    t=which(j==intervals)
    sample.int=subset(sample,x> -j & x< j)
     results.current=rdrobust(sample.int$y,sample.int$x, bwselect = symm_obw)
    coef.c[i,t]=results.current$coef[1]-jump
    coef.bc[i,t]=results.current$coef[2]-jump
    obw.c[i,t]=results.current$bws[1,1]
    obw.bc[i,t]=results.current$bws[2,1]

      if (i==1000) {
      sample.int %T>%
          plot(y~x,., ylim = range(c(y,y.model)),
             col="blue", ylab = "Y", xlab = "X") %T>%
        par(new = T) %>%
        plot(y.model~x,., ylim = range(c(y,y.model)),
             axes = FALSE, xlab = "", ylab = "")
      title(main = paste0("sample - interval of ",j))
    }
    }
  }
```

Figures summarizing iterations

```
fig.df <- data.frame(coef_bc=c(coef.bc$int_5,coef.bc$int_10,coef.bc$int_20,coef.bc$int_40), spec = rep(c
                                    each = loop))
fig.df$spec <- factor(fig.df$spec, levels =c("interval=5", "interval=10", "interval=20","interval=40"))
ggplot(fig.df, aes(x = coef_bc, fill = spec)) + geom_density(alpha = 0.3) +
  geom_vline(xintercept=0,  linetype="dashed") +
  labs(title = "Density of RD coefficients for different intervals",
       subtitle = "Bias-corrected estimates")
```




```
fig.df <- data.frame(coef_c=c(coef.c$int_5,coef.c$int_10,coef.c$int_20,coef.c$int_40),
                     spec = rep(c("interval=5", "interval=10", "interval=20","interval=40"),
fig.df$spec <- factor(fig.df$spec, levels =c("interval=5", "interval=10", "interval=20","interval=40"))
ggplot(fig.df, aes(x = coef_c, fill = spec)) + geom_density(alpha = 0.3) +
  geom_vline(xintercept=0,  linetype="dashed") +
  labs(title = "Density of RD coefficients for different intervals",
       subtitle = "Conventional estimates")
```



Summary results - TABLE

```
results_table=as.data.frame(matrix(0, ncol = 4, nrow = 0))
results_table[1,]=round(colMeans(coef.c),digits = 4)
results_table[2,]=round(colMeans(coef.bc),digits = 4)
results_table[3,]=round(colMeans(obw.c),digits = 4)
results_table[4,]=round(colMeans(obw.bc),digits = 4)
colnames(results_table) = paste("int",intervals,sep="_")
rownames(results_table) =var.list
kable(results_table, caption = "Summary Table")
```

Table 1: Summary Table

	int_5	int_10	int_20	int_40
coef.c	-3.8467	-1.0518	-0.3663	-0.1878
coef.bc	-4.5964	-1.3530	-0.5074	-0.4077
obw.c	1.1980	2.8772	4.6025	6.1277
obw.bc	2.0286	4.7999	7.5294	10.3830