Regression Discontinuity

EC 607 Metrics, Tutorial 9

Philip Economides
Spring 2021

Today

- Packages
- Examples
- Drunk Driving

Packages

We'll work through a walkthrough by Andrew Heiss

```
p_load(rdrobust,rddensity,modelsummary, data.table)
```

`rdrobust` package employs local polynomial and partitioning methods. Provides point estimators, confidence intervals estimators, bandwidth selectors, automatic RD plots, and many other features.

As of Winter 2020:

- Discrete running variable checks and adjustments
- Bandwidth selection adjustments for too few mass points in and/or overshooting of the support of the running variable
- RD Plots with additional covariates plotted at their mean

Packages

`rdrobust::rdbwselect()`:
implements bandwidth selectors for local polynomial Regression Discontinuity (RD) point estimators and inference procedures developed in Calonico, Cattaneo and Titiunik (2014).

rdbwselect(dep_var, run_var, rd_cutoff)

Additionally;

p: order of the local-polynomial for point-est construction.
kernel: triangular (default), epanechnikov, uniform.
bwselect: bandwidth selection procedure (see choices)

Packages

`rddensity::rddensity(run_var, c = cutoff)`
implements manipulation testing procedures using the local polynomial density estimators proposed in Cattaneo, Jansson and Ma (2020), and implements graphical procedures with valid confidence bands using the results in Cattaneo, Jansson and Ma (2021).

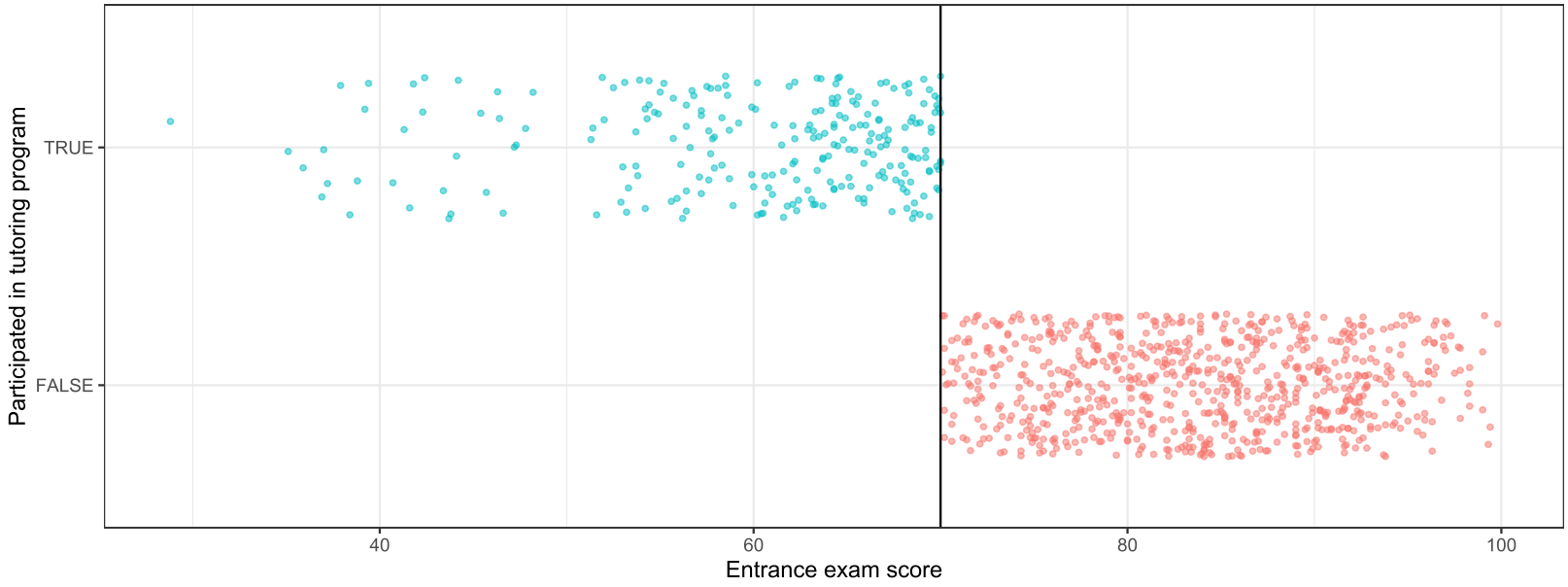
`rddensity::rdplotdensity(rdd, X, type, histBreaks)`
constructs density plots. It is based on the local polynomial density estimator proposed in Cattaneo, Jansson and Ma (2020, 2021). A companion Stata package is described in Cattaneo, Jansson and Ma (2018).

rdd: `rddensity` object
X: running variable
type: how point estimates plotted ("line", "points", "both")
histBreaks: giving the breakpoints between histogram cells

Examples

Dataframe describes student outcomes, where students who score higher than 70 are not eligible for a tutoring program.

How do we gauge slip-through?



Examples

Is there discontinuity in running variable around cutpoint? We can check to see if that jump is statistically significant with a McCrary density test.

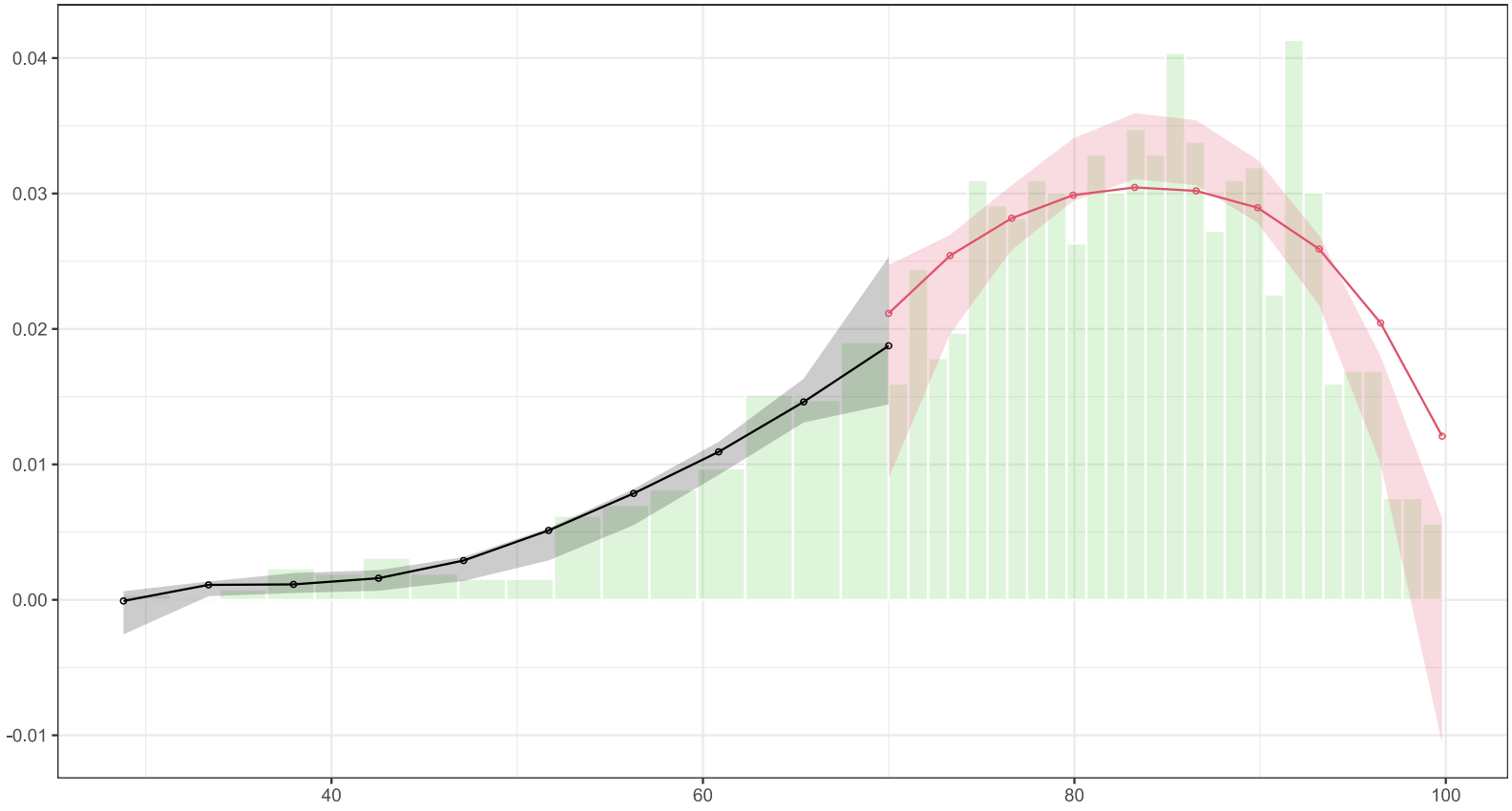
```
test_density ← rddensity(tutoring$entrance_exam, c = 70)
# To see proper set of results
#summary(test_density)
test_density[["test"]][["p_jk"]]
```

```
#> [1] 0.5808685
```

The p-value for the size of that overlap is 0.5809, which is a lot larger than 0.05, so we don't have good evidence that there's a significant difference between the two lines.

Examples

```
plot_density_test ← rdplotdensity(rdd = test_density,type = "both",
                                  X = tutoring$entrance_exam)
```



Examples

```
tutoring_centered ← tutoring %>%
  mutate(entrance_centered = entrance_exam - 70)
lm(exit_exam ~ entrance_centered + tutoring,
                   data = tutoring_centered) %>% huxreg()
```

	(1)
(Intercept)	59.411 ***
	(0.442)
entrance_centered	0.510 ***
	(0.027)
tutoringTRUE	10.800 ***
	(0.800)
N	1000
R2	0.268

Examples

```
lm(exit_exam ~ entrance_centered + tutoring,
  data = filter(tutoring_centered,
      entrance_centered ⩾ -5 &  entrance_centered ⩽ 5)) %>% huxreg()
```

	(1)
(Intercept)	60.631 ***
	(1.117)
entrance_centered	0.380
	(0.331)
tutoringTRUE	9.122 ***
	(1.912)
N	194
R2	0.222

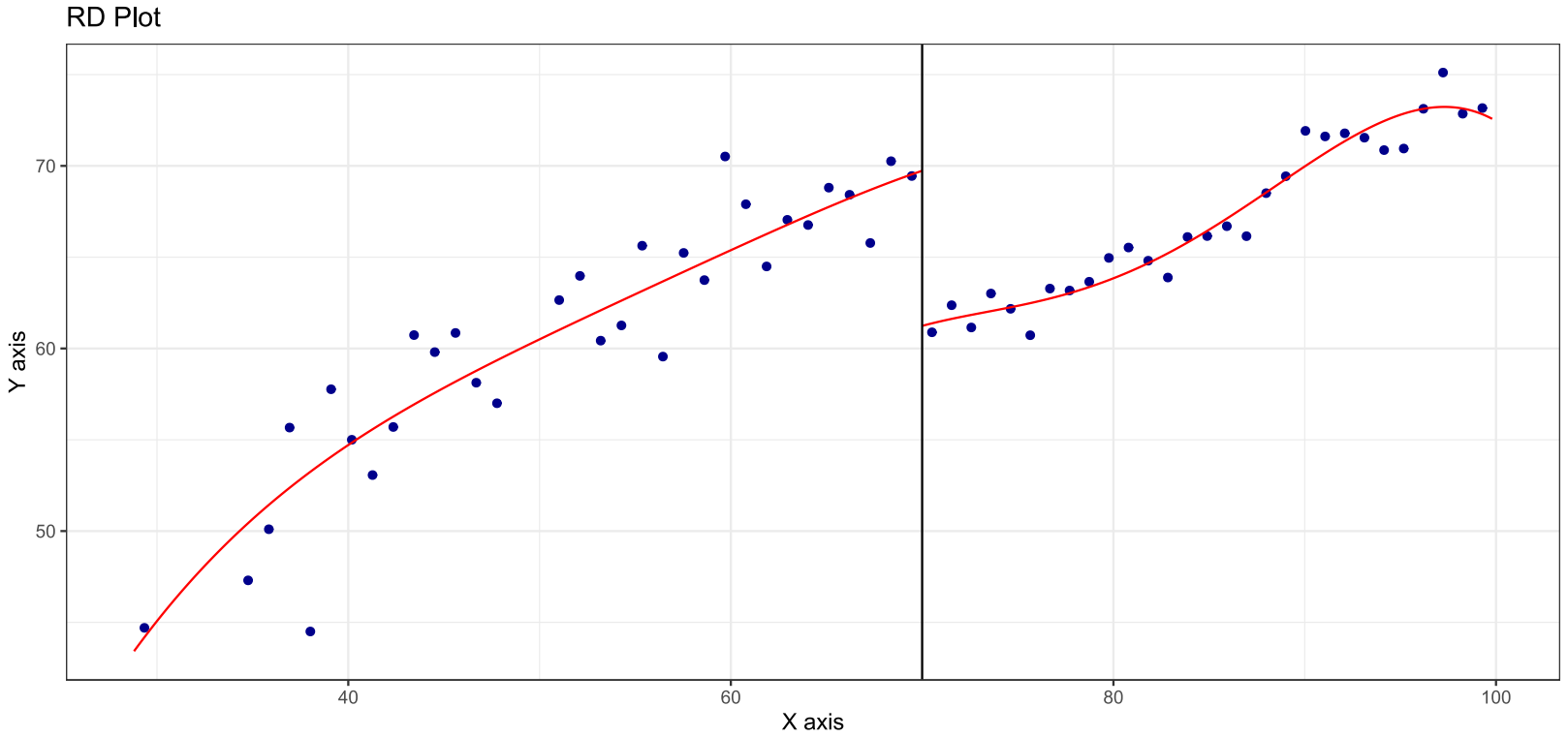
Examples

	Full data	Bandwidth = 10	Bandwidth = 5
(Intercept)	59.411	60.377	60.631
	(0.442)	(0.752)	(1.117)
entrance_centered	0.510	0.388	0.380
	(0.027)	(0.114)	(0.331)
tutoringTRUE	10.800	9.273	9.122
	(0.800)	(1.309)	(1.912)
Num.Obs.	1000	404	194
R2	0.268	0.162	0.222
R2 Adj.	0.267	0.158	0.214
AIC	6595.3	2663.5	1303.1
BIC	6615.0	2679.5	1316.2
Log.Lik.	-3293.663	-1327.755	-647.567

Examples

```
rdplot(y = tutoring$exit_exam, x = tutoring$entrance_exam, c = 70)
```

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



Drunk Driving

Replicating **Hansen (2015)** Punishment and Deterrence: Evidence from Drunk Driving, AER

```
# Load blood alcohol content data
rd_df ← fread("bac.csv")

glimpse(rd_df)
```

```
#> Rows: 214,558
#> Columns: 10
#> $ Date       <chr> "23 Jun 01", "15 Jun 02", "31 Jan 03", "11 Apr 99", "06 ...
#> $ Alcohol1   <int> 0, 142, 83, 198, 194, 62, 175, 85, 87, 198, 144, 185, 20...
#> $ Alcohol2   <int> 0, 139, 85, 207, 196, 66, 180, 89, 88, 195, 143, 178, 20...
#> $ bac        <dbl> 0.000, 0.139, 0.083, 0.198, 0.194, 0.062, 0.175, 0.085, ...
#> $ male       <int> 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
#> $ white      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,...
#> $ recidivism <int> 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
#> $ acc        <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,...
#> $ aged       <int> 25, 27, 54, 32, 30, 21, 33, 44, 34, 32, 43, 24, 47, 26, ...
#> $ year       <int> 2001, 2002, 2003, 1999, 1999, 1999, 1999, 1999, 1999, 19...
```

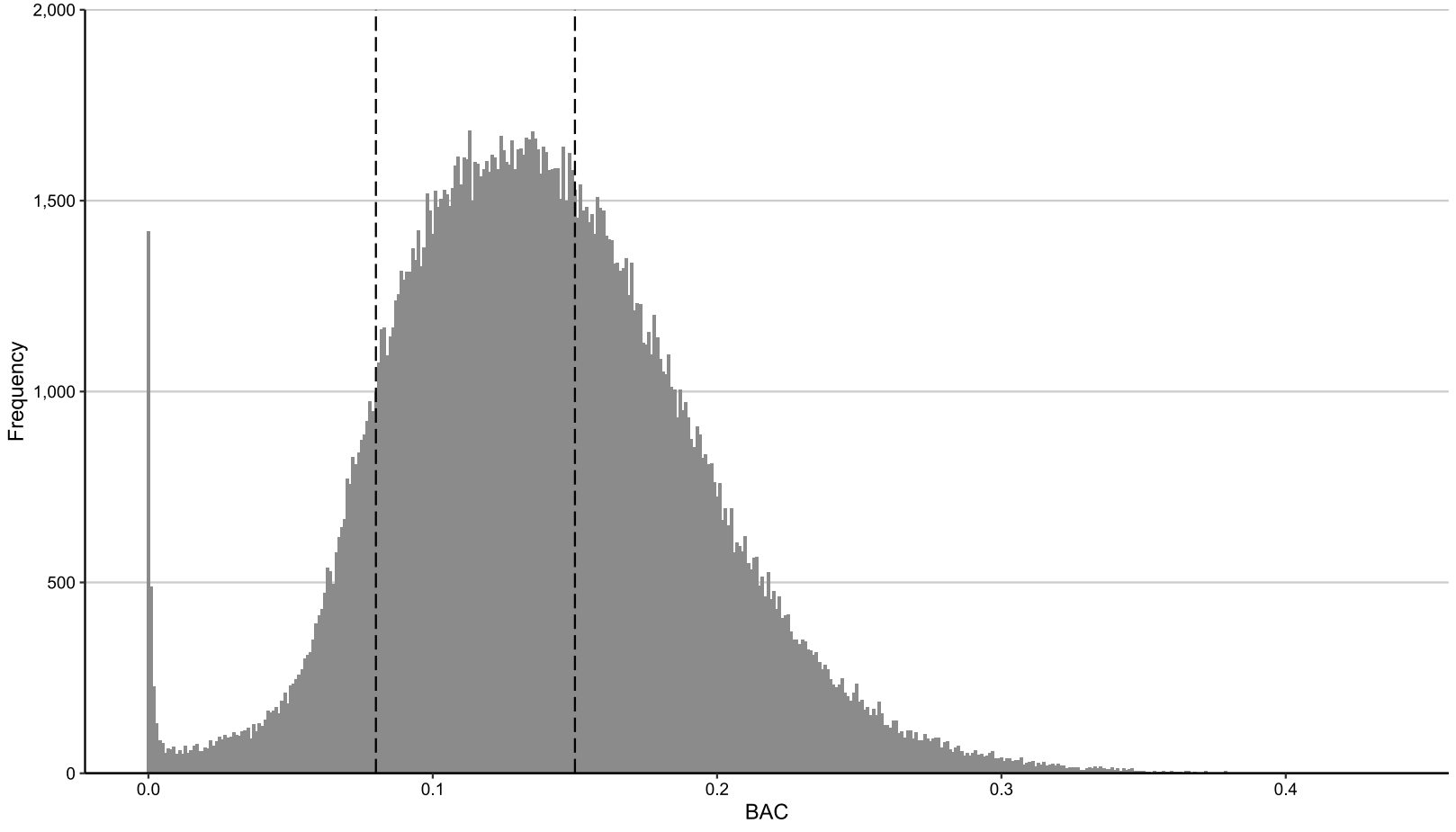
Drunk Driving

Background:

- Exploits discrete thresholds that determine both the current as well as potential future punishments for drunk drivers.
- Specifically, in WA BAC measured above 0.08 is considered a DUI
- BAC above 0.15 is considered an aggravated DUI, or a DUI that results in higher fines, increased jail time, and a longer license suspension period.
- Do individuals essentially just as drunk on either side of the threshold exhibit differences in recidivism rates?

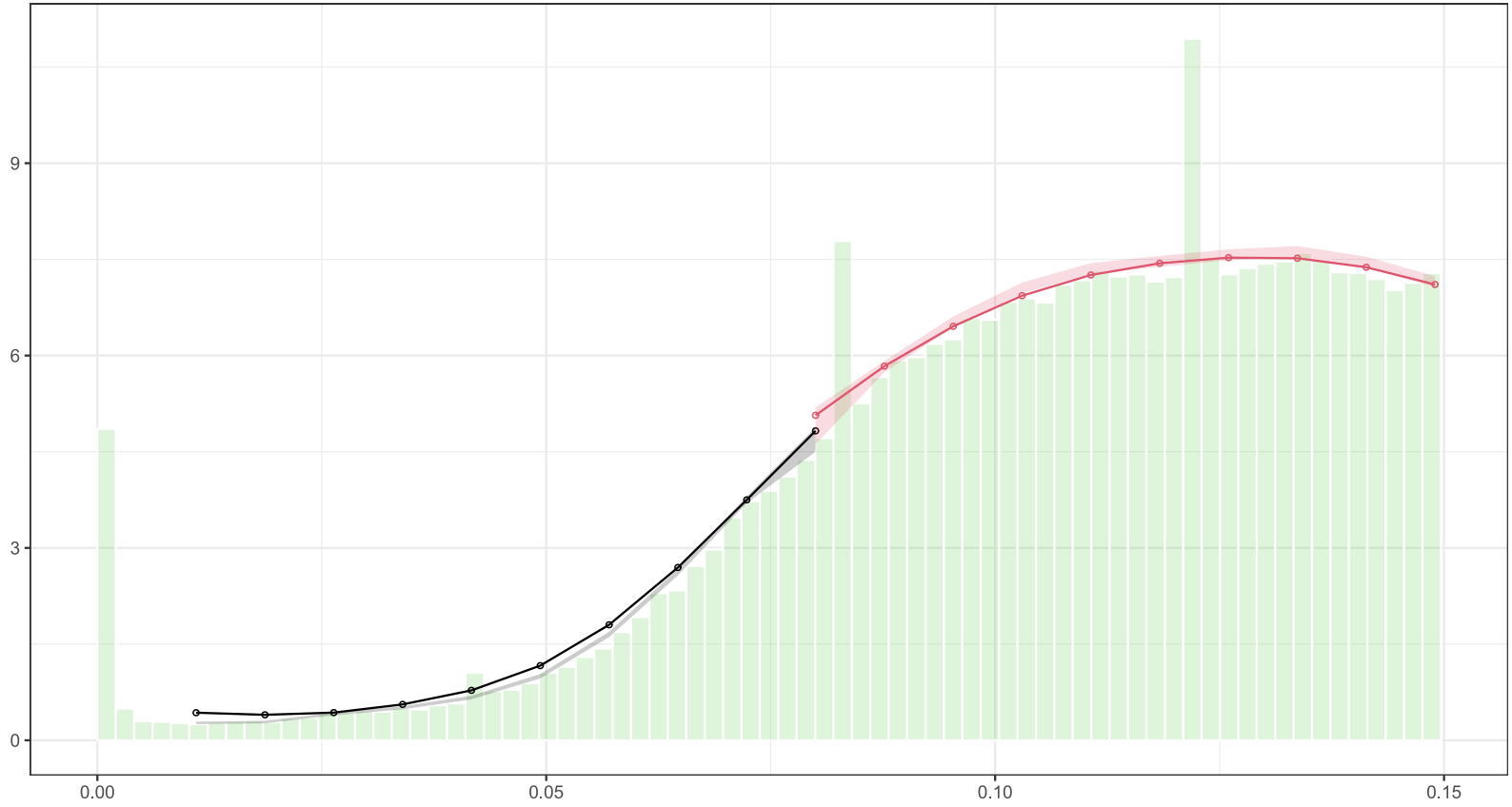
Drunk Driving

Is there any sorting at these thresholds? Histogram:



Drunk Driving

Is there evidence of sorting?



Drunk Driving

Regression discontinuity model

```
# Create a dummy variable for a BAC over .08.
rd_dfc ← rd_df %>% mutate(over08 = ifelse(bac ⩾ 0.08, 1, 0),
                          resbac = bac - 0.08) %>%
          filter(bac ⩽ 0.13 & bac ⩾ 0.03)

smooth_age = lm_robust(data=rd_dfc, aged  ~ over08 + resbac + over08:resbac)
smooth_gen = lm_robust(data=rd_dfc, male  ~ over08 + resbac + over08:resbac)
smooth_acc = lm_robust(data=rd_dfc, acc   ~ over08 + resbac + over08:resbac)
smooth_rac = lm_robust(data=rd_dfc, white ~ over08 + resbac + over08:resbac)

export_summs(smooth_age, smooth_gen, smooth_acc, smooth_rac,
             model.names=c("Age", "Male", "Accident","White"),
            statistics = c(N = 'nobs'))
```

Drunk Driving

Regression discontinuity model

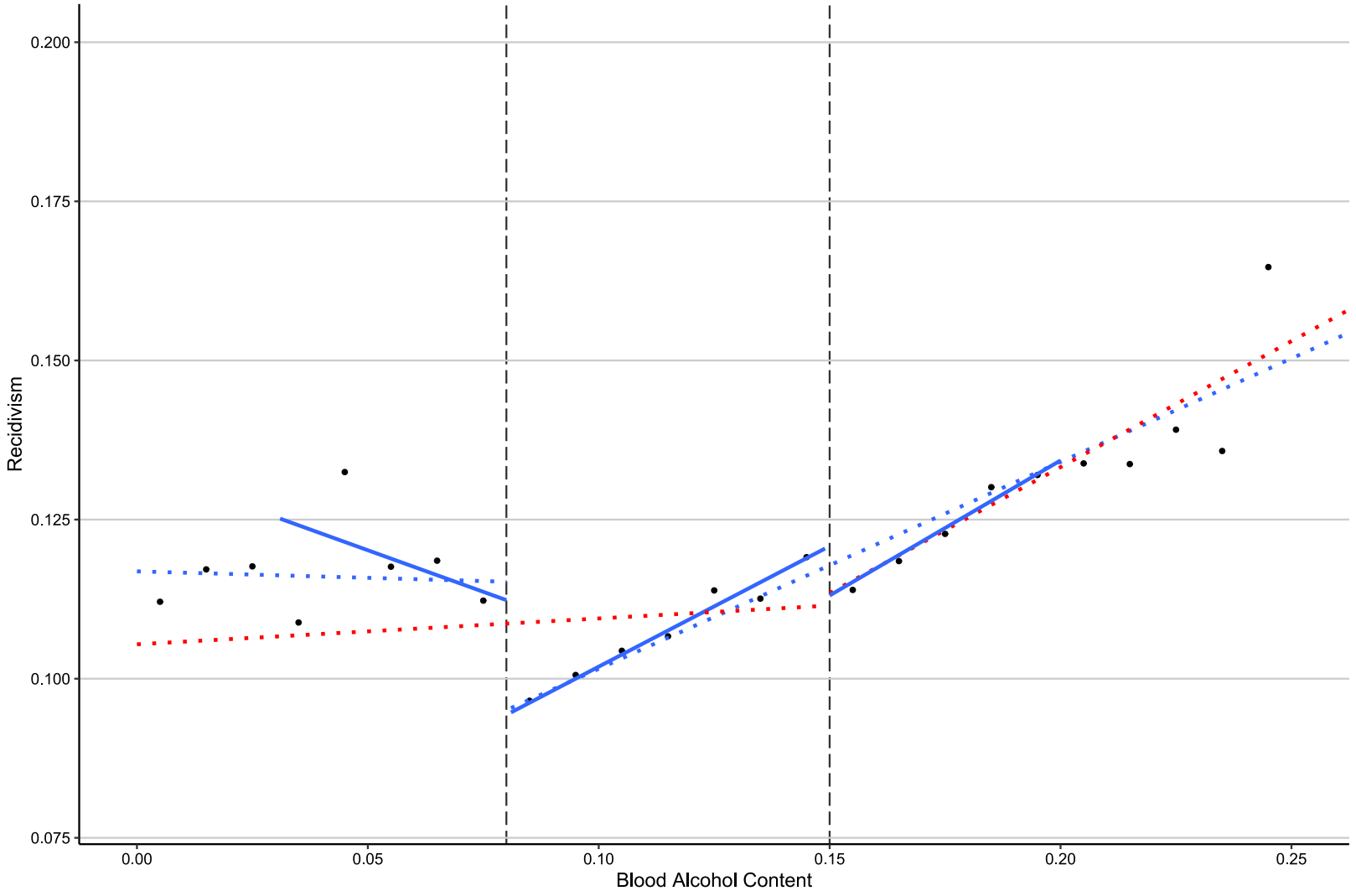
	Age	Male	Accident	White
(Intercept)	33.92 ***	0.79 ***	0.08 ***	0.85 ***
	(0.13)	(0.00)	(0.00)	(0.00)
over08	-0.17	0.01	-0.00	0.00
	(0.16)	(0.01)	(0.00)	(0.00)
resbac	-66.52 ***	-0.18	-1.04 ***	0.06
	(7.01)	(0.23)	(0.18)	(0.21)
over08:resbac	74.29 ***	0.27	1.92 ***	0.04
	(7.63)	(0.26)	(0.19)	(0.23)
N	93792	93792	93792	93792

Drunk Driving

```
lm_robust(data=rd_dfc,
     recidivism   ~ over08 + resbac + over08:resbac)%>% huxreg()
```

	(1)
(Intercept)	0.112 ***
	(0.003)
over08	-0.019 ***
	(0.004)
resbac	-0.261
	(0.182)
over08:resbac	0.684 ***
	(0.199)
N	93792

Drunk Driving



Drunk Driving

Lets walk through the code for this. Can adjust for multiple RD's, adjust the smoothing function using `geom_smooth` in future use.

```
bin_range = seq(0,0.4,by=0.01)
bac_x = c()
rec_y = c()

for(i in 1:(length(bin_range)-1)){
  df_bin ← rd_df %>%
    filter(bac ⩾ bin_range[i] & bac ⩽ bin_range[i+1])

  rec_y[i] = mean(df_bin$recidivism)
  bac_x[i] = (bin_range[i] + bin_range[i+1])/2
}
```

Drunk Driving

```
# Bind these two vectors into a dataframe to map
avg_recid = bind_cols(bac_x,rec_y) %>% rename(bac_x = ...1, rec_y= ...2)
avg_recid_cond = avg_recid[26:40,]
# List of results
high_bac = list()

for(j in 1:5){
  #avg_recid[26:40,]
 df_loop = avg_recid[(26+(3*(j-1))):(28+(3*(j-1))), ]
high_bac[[j]] = df_loop %>% summarize(bac_x=mean(bac_x),
                        rec_y=mean(rec_y))

}

avg_recid_adj = bind_rows(avg_recid[1:25,],bind_rows(high_bac))
```

Drunk Driving

```
# Prepare horizontal greylines
minors←seq(0.075,0.2,by=0.025)
p = ggplot(rd_df, aes(x = bac, y = recidivism)) +
  geom_vline(xintercept=c(0.08, 0.15), linetype = "longdash", alpha=0.8)+
  geom_hline(mapping=NULL, yintercept=minors,colour='grey80')+
  geom_point(data=avg_recid_adj, aes(x= bac_x, y=rec_y), size=1, fill="white")+
  # Add lines for the full model at 0.08 (model_simple)
  geom_smooth(data = filter(rd_df, bac ⩽ 0.08),
              method = "lm", se = FALSE, linetype = "dotted", size = 1) +
  geom_smooth(data = filter(rd_df, bac > 0.08),
              method = "lm", se = FALSE, linetype = "dotted", size = 1) +
  # Add lines for bandwidth (first and second)
  geom_smooth(data = filter(rd_df, bac ⩽ 0.08, bac ⩾ 0.03),
              method = "lm", se = FALSE, size = 1) +
  geom_smooth(data = filter(rd_df, bac > 0.08, bac ⩽ 0.15),
              method = "lm", se = FALSE, size = 1) +
  # Add lines for the full model at 0.15 (model_simple)
  geom_smooth(data = filter(rd_df, bac ⩽ 0.15), color="red",
              method = "lm", se = FALSE, linetype = "dotted", size = 1) +
  geom_smooth(data = filter(rd_df, bac > 0.15), color="red",
              method = "lm", se = FALSE, linetype = "dotted", size = 1) +
  # Add lines for bandwidth (third)
  geom_smooth(data = filter(rd_df, bac > 0.15, bac ⩽ 0.2),
```

Drunk Driving

Underlying resource that I've tuned up for plotting: Heiss' plots in section 5