Parametric Regression Discontinuity Analysis

Yelsh Gebreselassie

2022-09-21

Contents

```r
library(tidyverse)
library(broom)
library(rdrobust)
library(rddensity)
library(modelsummary)

tutoring <- read.csv("data/tutoring_program.csv")
attach(tutoring)
```

Step 1: Determine if the process of assigning treatment is rule based

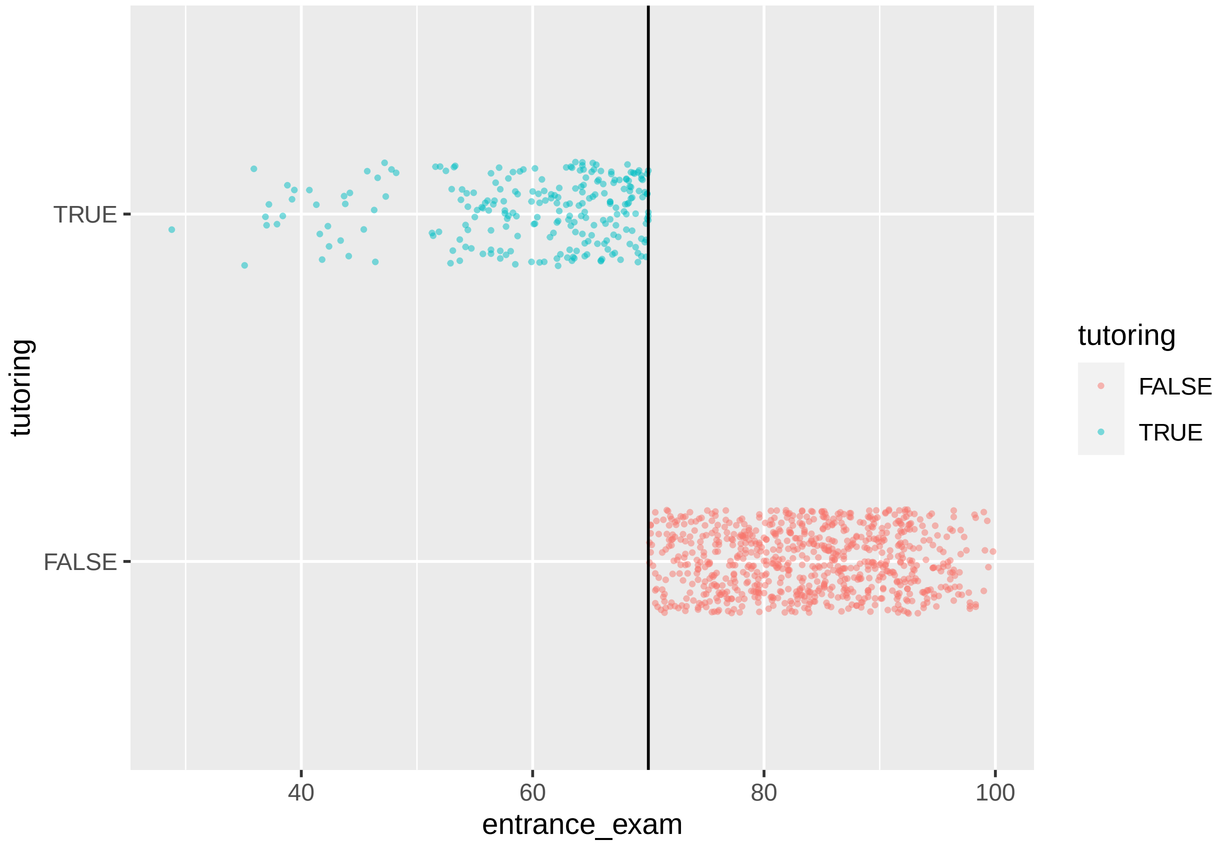
- We know if a student is assigned to a tutoring program based on certain rule if and only if we are familiar with the research process and the data. Looking at the dataset, we can see that only students who scored less that 70 are assigned to tutoring.

Step 2: Determine if the design in fuzzy or sharp

- This is to determine if, for example, a student who scored >70 is using tutoring or a student who scored <70 is not assigned to tutoring.

```
ggplot(data = tutoring,
       mapping = aes(x = entrance_exam, y = tutoring, color = tutoring)) +
  geom_point(size = 0.5, alpha = 0.5,
             position = position_jitter(width = 0, height = 0.15), seed = 1234) +
  geom_vline(xintercept = 70)
```

```
## Warning: Ignoring unknown parameters: seed
```



- We can see above that the line looks sharp. It doesn't look like anyone who scored above 70 is using tutoring or anyone who scored below 70 not using tutoring. We can also confirm this result numerically.

```
tutoring %>%
  group_by(tutoring, entrance_exam > 70) %>%
  summarize(count =n())
```

```
## `summarise()` has grouped output by 'tutoring'. You can override using the
## `.groups` argument.
```

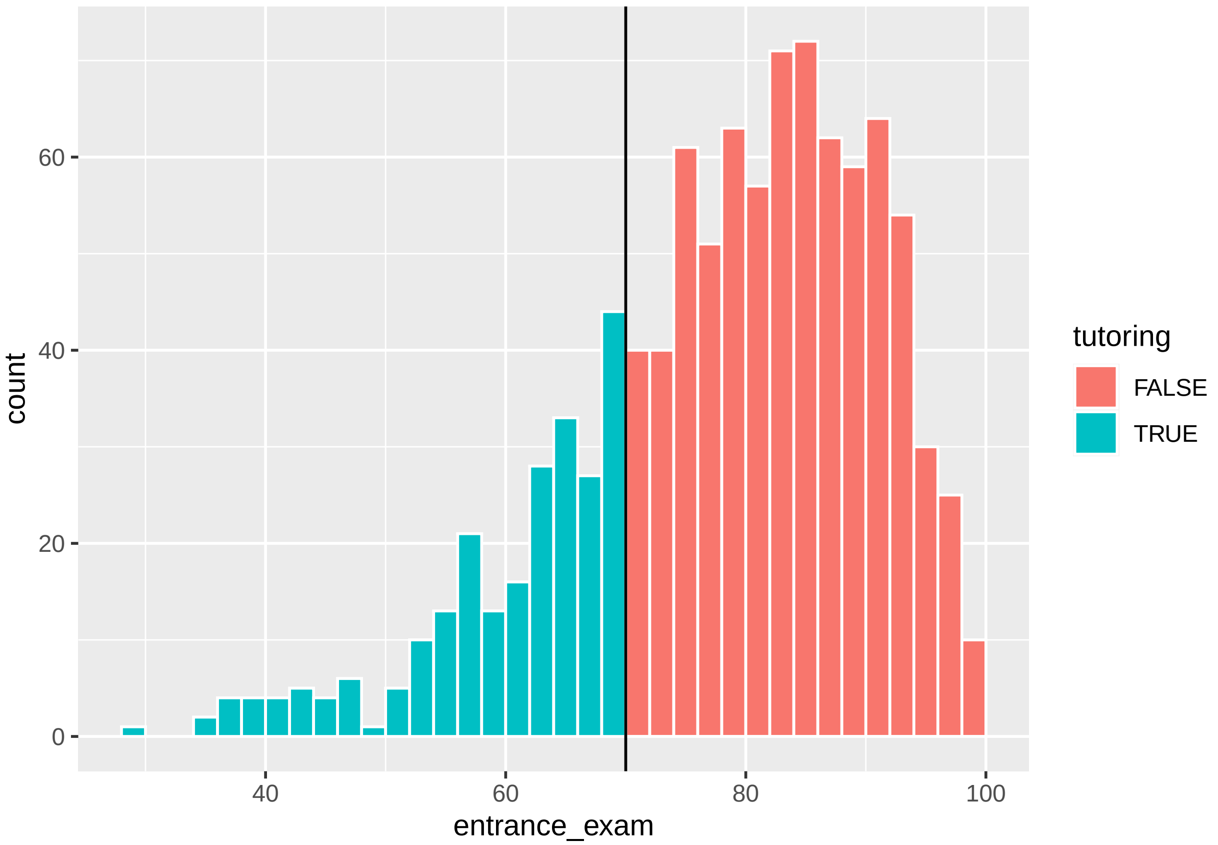
```
## # A tibble: 2 x 3
## # Groups:   tutoring [2]
##   tutoring `entrance_exam > 70` count
##   <lgl>    <lgl>                <int>
## 1 FALSE    TRUE                   759
## 2 TRUE     FALSE                  241
```

- We can see that all who have scored below 70 (241 students) have been assigned to tutoring and those who scored above 70 (759 students) have not been assigned to tutoring.

Step 3: Check for discontinuity in running variable around cutpoint

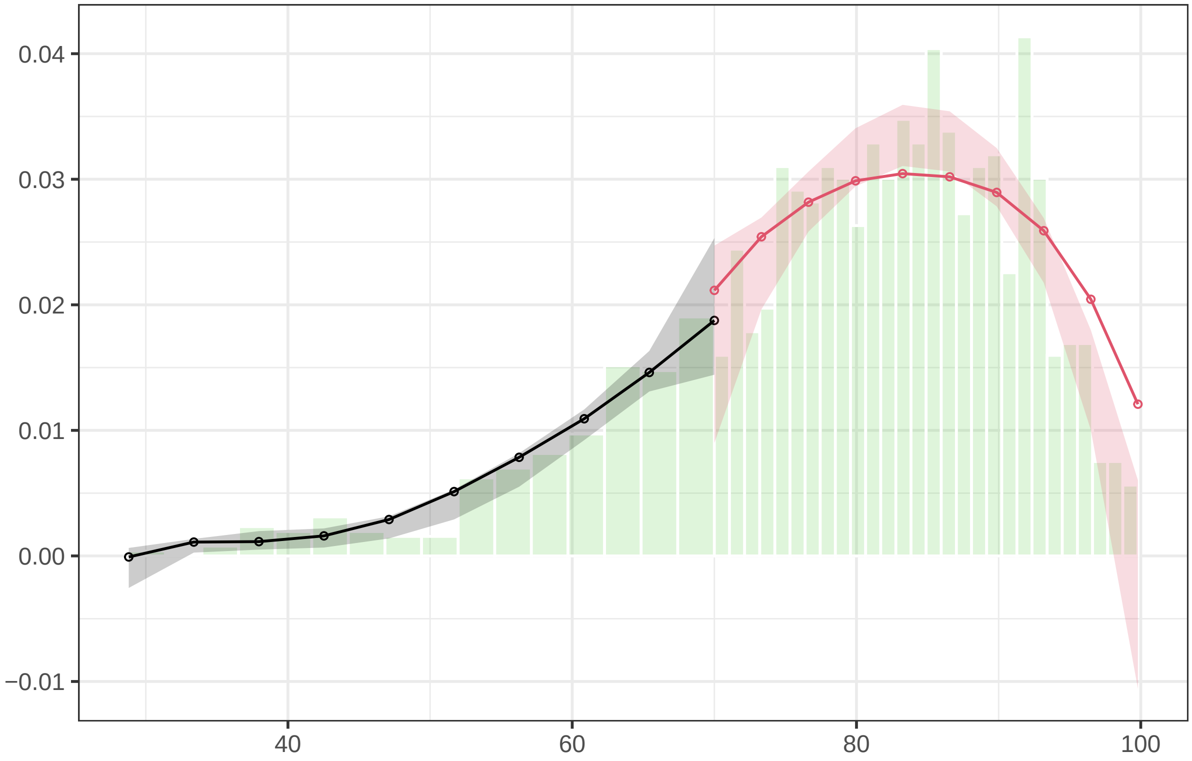
- This is to see if anyone is manipulating access to the entrance exam. For example, we need to make sure no one is purposely scoring a little over 70 to not be enrolled in the tutoring program or scoring just below 70 so they can be enrolled in tutoring.

```
ggplot(tutoring, aes(x = entrance_exam, fill = tutoring)) +
  geom_histogram(binwidth = 2, boundary = 70, color = "white") +
  geom_vline(xintercept = 70)
```



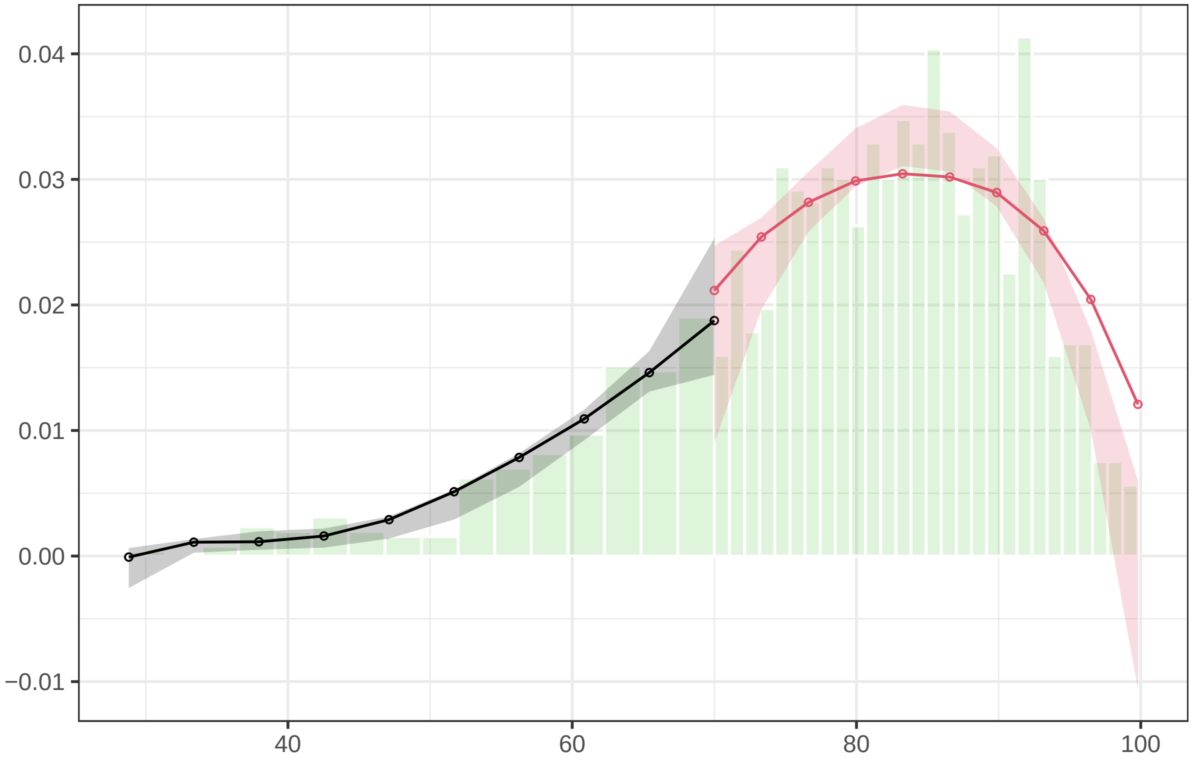
- There is a smooth transition at the cutoff point. We don't, for example, observe many students just to the left of the cutoff point and then a significant drop after. We can test this more officially using McCrary desnsity test.

```
rdplotdensity(rdd = rddensity(entrance_exam, c = 70),
              X = entrance_exam,
              type = "both")
```



```
## $Estl
## Call: lpdensity
##
## Sample size                                           241
## Polynomial order for point estimation    (p=)     2
## Order of derivative estimated            (v=)     1
## Polynomial order for confidence interval (q=)     3
## Kernel function                                       triangular
## Scaling factor                                        0.24024024024024
## Bandwidth method                                      user provided
##
## Use summary(...) to show estimates.
##
## $Estr
## Call: lpdensity
##
## Sample size                                           763
## Polynomial order for point estimation    (p=)     2
## Order of derivative estimated            (v=)     1
## Polynomial order for confidence interval (q=)     3
## Kernel function                                       triangular
## Scaling factor                                        0.762762762762763
## Bandwidth method                                      user provided
##
## Use summary(...) to show estimates.
##
```

```
## $Estplot
```

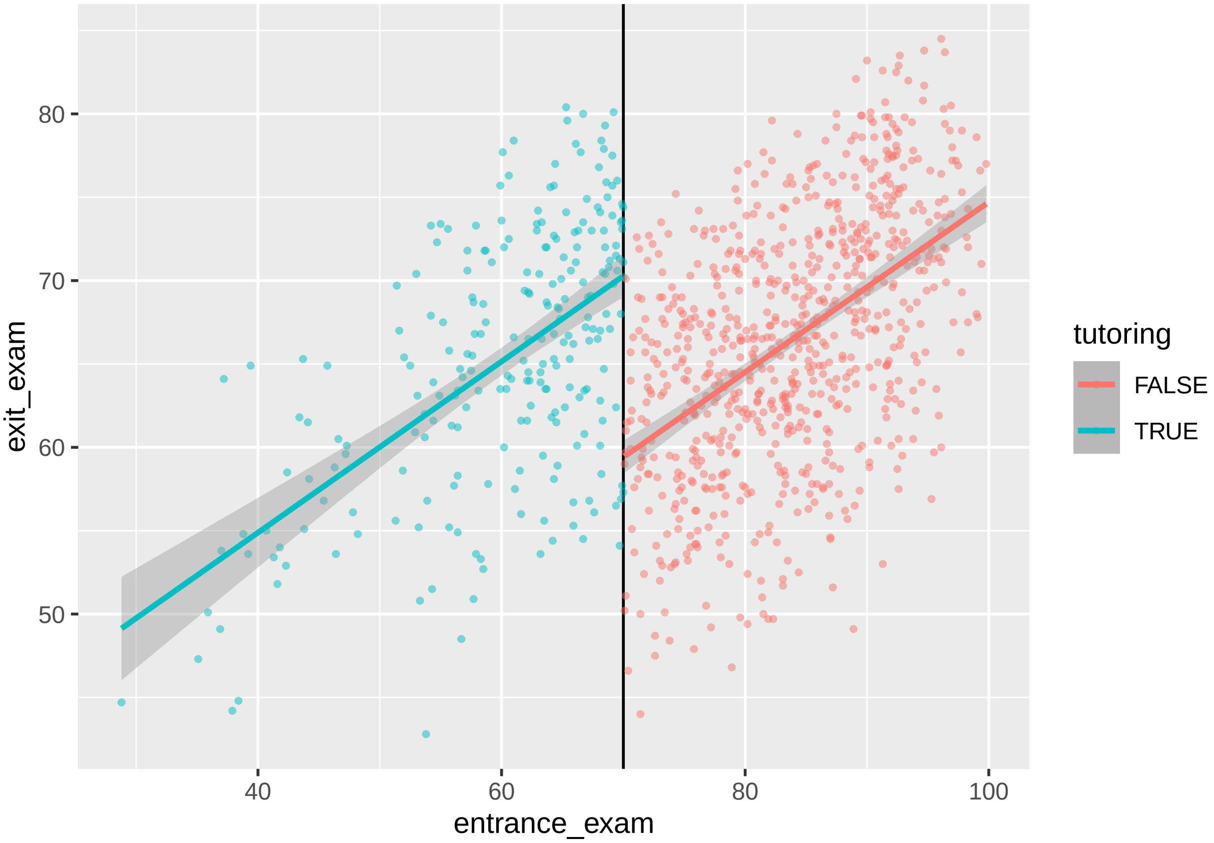


- We can see that there is some gap but it is within the confidence interval on both sides.

Step 4: Check for discontinuity in outcome across running variable.

```
ggplot(tutoring, aes(x = entrance_exam, y = exit_exam, color = tutoring)) +
  geom_vline(xintercept = 70) +
  geom_point(size = 0.75, alpha = 0.5) +
  geom_smooth(data = filter(tutoring, entrance_exam < 70), method = lm) +
  geom_smooth(data = filter(tutoring, entrance_exam > 70), method = lm)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Next we measure the gap.

Step 5: Measure the size of the effect

Parametric measurement

```
tutoring_centered <- tutoring %>%
  mutate(entrance_centered = entrance_exam - 70)
```

This will produce a new dataset called tutoring_centered with a new variable or column entrance_centered which shows how many points above or below a student scored.

Then we can run a regression model to measure the gap.

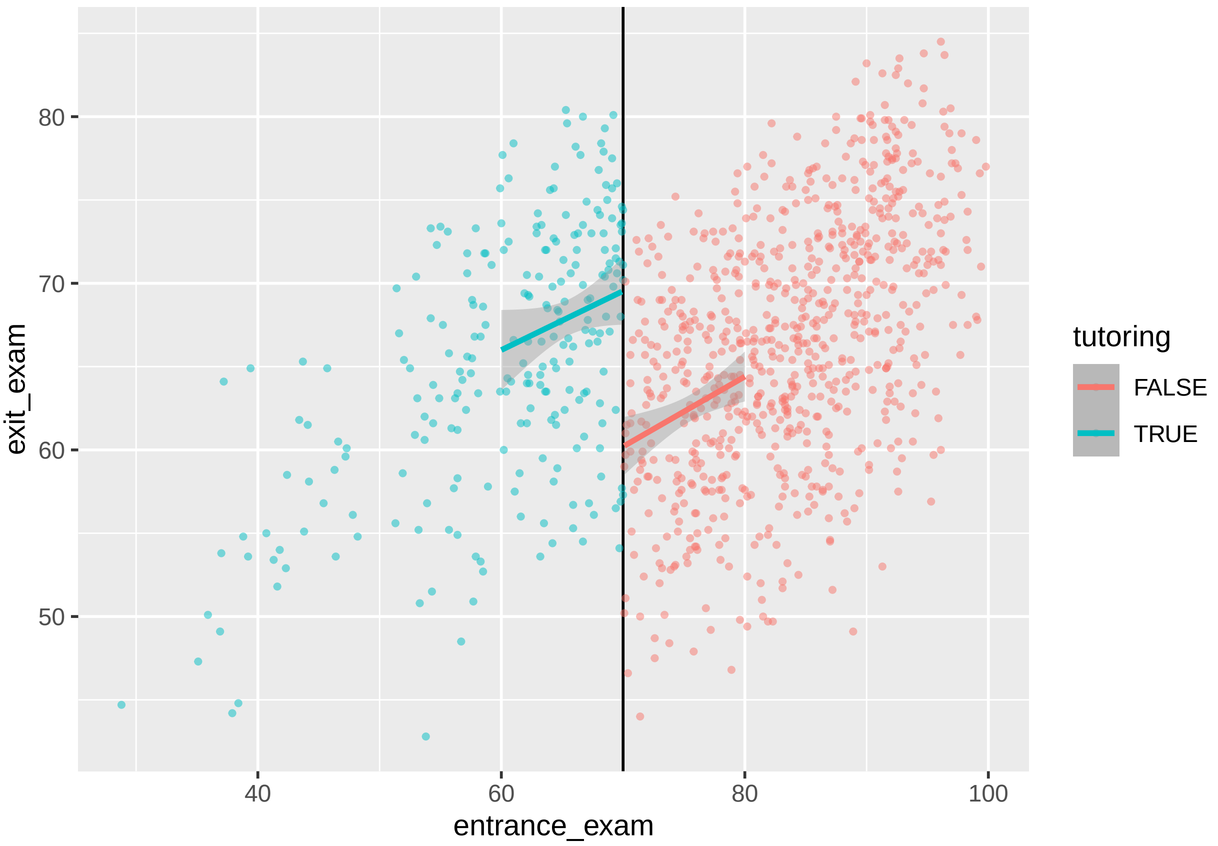
```
model_simple <- lm(exit_exam ~ entrance_centered + tutoring,
                   data = tutoring_centered)
tidy(model_simple)
```

```
## # A tibble: 3 x 5
##   term              estimate std.error statistic  p.value
##   <chr>                <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)         59.4      0.442      134.   0
## 2 entrance_centered    0.510    0.0269      18.9 1.40e-68
## 3 tutoringTRUE        10.8      0.800       13.5 3.12e-38
```

- We can interpret this as people who did not get tutoring and scored exactly 70 (meaning when entrance_centered = 0), their average exit exam score is 59. The entrance_centered coefficient which is 0.51 can be interpreted as, every time for a one point increase in score in the entrance exam, a student's exit exam score is increased by half a point. tutoringTRUE's coefficient which is 10.9 is saying that when tutored, s student's exit exam score increase by 10.9 on average.

```
ggplot(tutoring, aes(x = entrance_exam, y = exit_exam, color = tutoring)) +
  geom_vline(xintercept = 70) +
  geom_point(size = 0.75, alpha = 0.5) +
  geom_smooth(data = filter(tutoring,
                            entrance_exam < 70,
                            entrance_exam >=60),
              method = "lm") +
  geom_smooth(data = filter(tutoring,
                            entrance_exam > 70,
                            entrance_exam <=80),
              method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



```
model_bw10 <- lm(exit_exam ~ entrance_centered + tutoring,
                 data = filter(tutoring_centered,
                               entrance_centered <= 10,
                                 entrance_centered >= -10))
tidy(model_bw10)
```

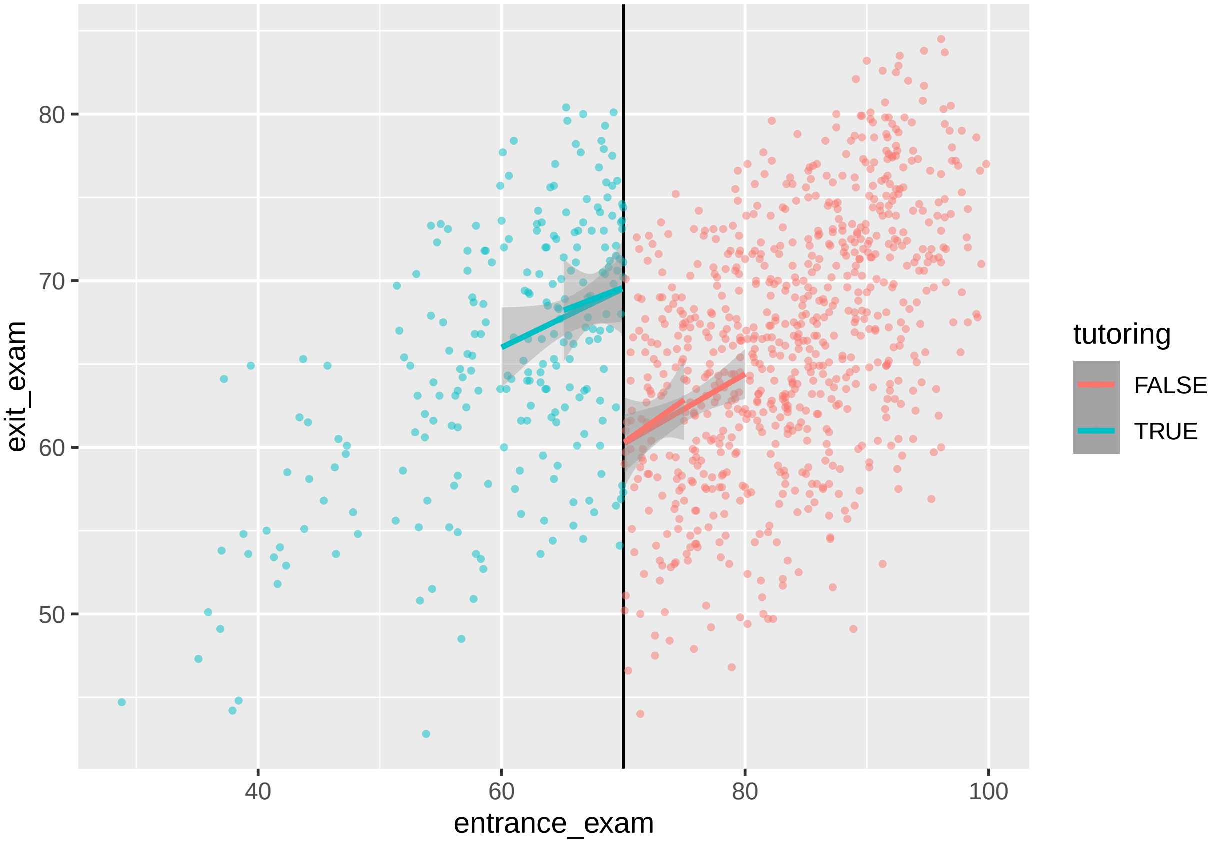
```
## # A tibble: 3 x 5
##   term              estimate std.error statistic   p.value
##   <chr>                <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)         60.4       0.752     80.3  2.99e-249
## 2 entrance_centered    0.388     0.114      3.40 7.45e-  4
## 3 tutoringTRUE         9.27      1.31       7.09 6.27e- 12
```

```
model_bw5 <- lm(exit_exam ~ entrance_centered + tutoring,
                   data = filter(tutoring_centered,
                                 entrance_centered <= 5,
                                 entrance_centered >= -5))
tidy(model_bw5)
```

```
## # A tibble: 3 x 5
##   term              estimate std.error statistic   p.value
##   <chr>                <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)         60.6       1.12      54.3  4.78e-118
## 2 entrance_centered    0.380     0.331      1.15 2.53e-  1
## 3 tutoringTRUE         9.12      1.91       4.77 3.66e-  6
```

```
ggplot(tutoring, aes(x = entrance_exam, y = exit_exam, color = tutoring)) +
  geom_vline(xintercept = 70) +
  geom_point(size = 0.75, alpha = 0.5) +
  geom_smooth(data = filter(tutoring,
                            entrance_exam < 70,
                            entrance_exam >=60),
              method = "lm") +
  geom_smooth(data = filter(tutoring,
                            entrance_exam > 70,
                            entrance_exam <=80),
              method = "lm") +
  geom_smooth(data = filter(tutoring,
                            entrance_exam < 70,
                            entrance_exam >=65),
              method = "lm") +
  geom_smooth(data = filter(tutoring,
                            entrance_exam > 70,
                            entrance_exam <=75),
              method = "lm")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
## ‘geom_smooth()‘ using formula ’y ~ x’
## ‘geom_smooth()‘ using formula ’y ~ x’
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Step 6: Compare all the effects

```
modelsummary(list(model_simple, model_bw10, model_bw5))
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'
```

	Model 1	Model 2	Model 3
(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
RMSE	6.52	6.47	6.81