Problem Set 5

Prasanna Parasurama

Due: 4/12/19

The packages used this week are

- ggplot2
- xtable (build tables quickly)
- data.table (data tables are computationally efficient and IMHO easier to work with)
- rdd (package for regression discontinuity designs)

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(reticulate)
library(gridExtra)
library(rdd)
use_python("/Users/parasu/anaconda3/bin/python")
```

Problem 1 (Coding Exercise)

The dataset for this exercise comes from a paper by Benjamin Olken entitled "Monitoring Corruption: Evidence from a Field Experiment in Indoneisa". The paper evaluates an attempt to reduce corruption in road building in Indonesia. The treatment we focus on was "accountability meetings". These meetings were held at a village level, and project officials were probed to account for how they spent project funds. Before construction began, residents in the treated villages were encouraged to attend these meetings. The dataset is called "olken.csv".

The outcome we care about is **pct.missing**, the difference between what officials claim they spent on road construction and an independent measure of expenditures. Treatment is given by **treat.invite** such that:

$$\text{treat.invite} = \begin{cases} 1 & \text{if village received intervention} \\ 0 & \text{if village was control} \end{cases}$$

We have the following four pre-treatment covariates:

- head.edu : the education of the village head
- mosques : mosques per 1000 residents
- pct.poor : the percentage of households below the poverty line
- total.budget : the budget for each project

We now have the following questions:

a. Create a balance table. For each pretreatment covariate, include comparisons for treated and untreated units in terms of the mean and standard deviation. Report a test, for each covariance, of the hypothesis that the difference in means between treatment conditions is zero.

```
import pandas as pd
df = pd.read_csv("data/olken.csv")
```

```
pre_treat_cov = ["head.edu", "mosques", "pct.poor", "total.budget"]
df.groupby("treat.invite")[pre_treat_cov].agg({"mean": "mean", "sd": "std"}).transpose()
```

```
## treat.invite               0          1
## mean head.edu      11.583851  11.546624
##      mosques        1.472887   1.419518
##      pct.poor       0.400366   0.410624
##      total.budget  83.354210  83.164344
## sd   head.edu       2.721578   2.719745
##      mosques        0.825695   0.837757
##      pct.poor       0.212467   0.213046
##      total.budget  42.908959  60.410979
##
## /Users/parasu/anaconda3/lib/python3.6/site-packages/pandas/core/groupby/generic.py:1315: FutureWarni
##   return super(DataFrameGroupBy, self).aggregate(arg, *args, **kwargs)
```

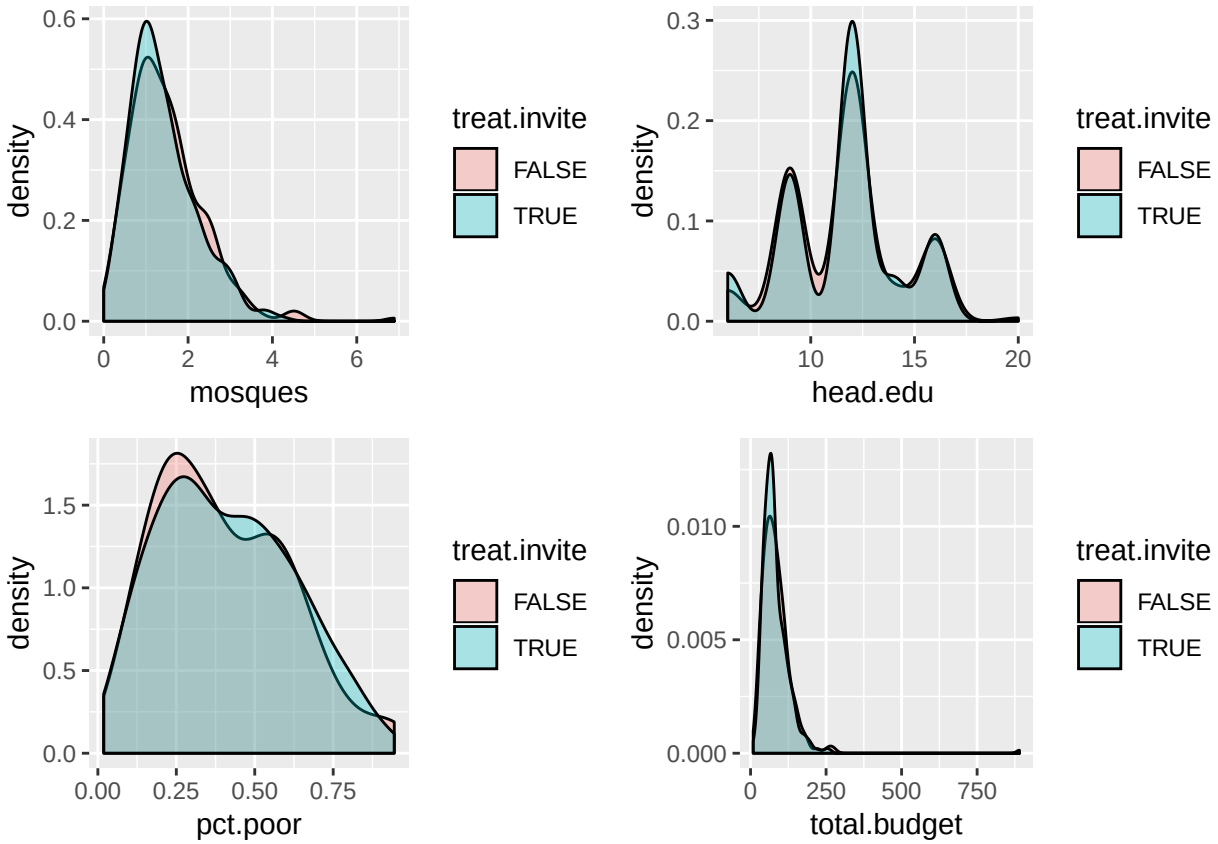
```
df = read.csv("data/olken.csv")
```

```
# t-test
pre_treatment_variables = c("head.edu", "mosques", "pct.poor", "total.budget")

for(v in pre_treatment_variables){
  print(v)
  print(t.test(as.formula(sprintf("%s ~ treat.invite", v)), data=df)$p.value)
}
```

```
## [1] "head.edu"
## [1] 0.8880239
## [1] "mosques"
## [1] 0.5081709
## [1] "pct.poor"
## [1] 0.6196482
## [1] "total.budget"
## [1] 0.9685551
```

b. For each covariate, plot its distribution under treatment and control (either side-by-side using facet_grid or overlap).

```
df = df %>% mutate(treat.invite = as.logical(treat.invite))


p1 = ggplot(df, aes(x=mosques, fill=treat.invite)) + geom_density(alpha=0.3)
p2 = ggplot(df, aes(x=head.edu, fill=treat.invite)) + geom_density(alpha=0.3)
p3 = ggplot(df, aes(x=pct.poor, fill=treat.invite)) + geom_density(alpha=0.3)
p4 = ggplot(df, aes(x=total.budget, fill=treat.invite)) + geom_density(alpha=0.3)

grid.arrange(p1,p2,p3,p4, nrow=2, ncol=2)
```



c. Given your answers to part a and b, do the villagers seem similar in their pre-treatment covariates?

Yes, the distributions are very similar, and t-tests fail to reject null the means between the groups are the same.

d. Regress the treatment on the pre-treatment covariates. What do you conclude?

```
selection_model = glm(treat.invite ~ mosques+ head.edu+ pct.poor+ total.budget, data=df)

summary(selection_model)
```

```
##
## Call:
## glm(formula = treat.invite ~ mosques + head.edu + pct.poor +
##     total.budget, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7075  -0.6474   0.3263   0.3449   0.4480
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.767e-01  1.172e-01   5.772 1.43e-08 ***
## mosques      -1.871e-02  2.646e-02  -0.707    0.480
## head.edu     -8.972e-04  8.099e-03  -0.111    0.912
## pct.poor      5.744e-02  1.044e-01   0.550    0.582
## total.budget -4.698e-05  4.013e-04  -0.117    0.907
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for gaussian family taken to be 0.2267874)
##
##     Null deviance: 106.08  on 471  degrees of freedom
## Residual deviance: 105.91  on 467  degrees of freedom
## AIC: 646.12
##
## Number of Fisher Scoring iterations: 2
```

None of the covariates are significantly correlated with treatment, so treatment seems to be exogenous.

e. Using the difference-in-means estimator, estimate the ATE and its standard error.

```
y_treated = (df %>% filter(treat.invite == TRUE))$pct.missing
y_untreated = (df %>% filter(treat.invite == FALSE))$pct.missing

# ATE
print(mean(y_treated) - mean(y_untreated))
```

```
## [1] -0.02494953
```

```
sqrt(var(y_treated)/(length(y_treated)) + var(y_untreated)/(length(y_untreated)))
```

```
## [1] 0.03310019
```

$$\hat{ATE}_{DM} = -0.0249$$

$$\hat{\sigma}_{\hat{ATE}_{DM}} = 0.033$$

f. Using a simple regression of outcomes on treatment, estimate the ATE and its standard error. Compare your answer in (f) to (e). Make adjustments to your regression strategy to make (f) and (e) match exactly.

```
df = df %>% mutate(treat.invite = as.logical(treat.invite))
summary(lm(pct.missing ~ treat.invite, data=df))
```

```
##
## Call:
## lm(formula = pct.missing ~ treat.invite, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33064 -0.21249 -0.01284  0.18281  1.42154
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.25277    0.02716   9.306   <2e-16 ***
## treat.inviteTRUE -0.02495    0.03346  -0.746    0.456
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3447 on 470 degrees of freedom
## Multiple R-squared:  0.001181,   Adjusted R-squared:  -0.0009438
## F-statistic: 0.5559 on 1 and 470 DF,  p-value: 0.4563
```

g. Using the same regression from part (f), include pre-treatment covariates in your regression equation (additively and linearly). Report estimates of treatment effects and its standard error. Do you expect (g) to differ from (f) and (e)? Explain your answer.

If the pre-treatment covariates are not correlated with treatment and the outcome (i.e treatment randomization was successful) the ATE estimates will be the same.

The standard errors of the ATE in (g) will be lesser than equal to (f) and (e).

```
summary(lm(pct.missing ~ treat.invite + mosques + head.edu + pct.poor + total.budget, data=df))
```

```
##
## Call:
## lm(formula = pct.missing ~ treat.invite + mosques + head.edu +
##     pct.poor + total.budget, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28605 -0.21411 -0.01291  0.18932  1.42530
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.3904455  0.0869471   4.491 8.96e-06 ***
## treat.inviteTRUE -0.0264183  0.0331558  -0.797   0.4260
## mosques          -0.0481914  0.0189702  -2.540   0.0114 *
## head.edu         -0.0055082  0.0058032  -0.949   0.3430
## pct.poor         -0.1177125  0.0747921  -1.574   0.1162
## total.budget      0.0005307  0.0002875   1.846   0.0655 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3412 on 466 degrees of freedom
## Multiple R-squared:  0.0294, Adjusted R-squared:  0.01898
## F-statistic: 2.823 on 5 and 466 DF,  p-value: 0.01594
```

Problem 2 (Coding Exercise)

We will be using a dataset that was simulated from real data. Oftentimes due to privacy concerns researchers will provide simulated data from the distribution of real data. The dataset you will be using are from a tutoring program focused on math for 7th graders. The dataset is called "tests_Rd.csv". The tutoring is the treatment variable, **treat**. Tutoring was given to students based on a pretest score, **pretest** thus the pretest score is the forcing variable. Students that received less than 215 were given a tutor. Our outcome of interest is the test score after tutoring, **posttest**.

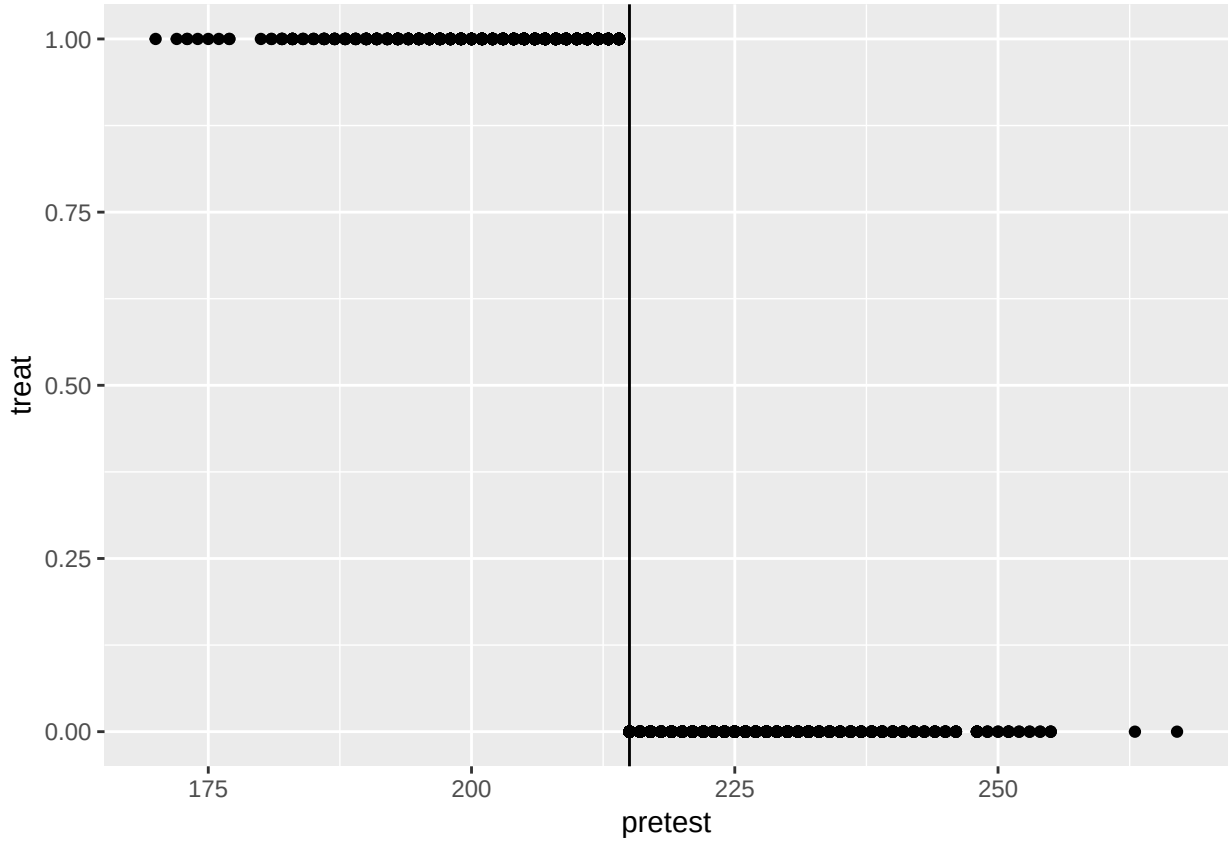
We also have a series of control variables:

- age : age of student as of September 2010
- gender : 1 if student's gender is male
- frlunch : 1 if student is eligible a free lunch
- esol : 1 if student has english as a second language
- white : 1 if student's race/ethnicity is white
- asian : 1 if student's race/ethnicity is asian
- black : 1 if student's race/ethnicity is black
- hispanic: 1 if student's race/ethnicity is hispanic

We ask you to answer the following questions:

a. We want you to first plot the graph that justifies a sharp RD design. Plot the treatment as a function of the forcing variable. What do you see?

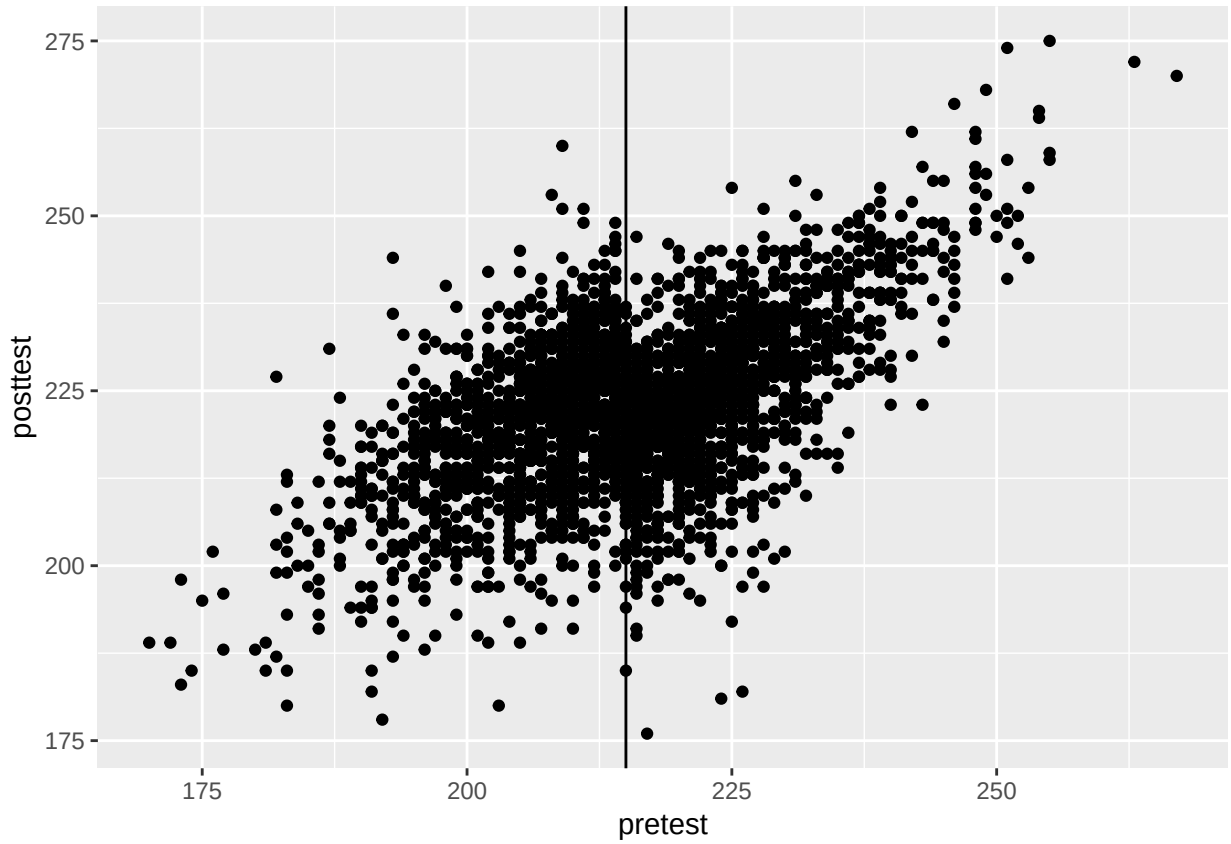
```
df2 = read.csv("data/tests_Rd.csv")

ggplot(df2, aes(x=pretest, y=treat)) + geom_point() + geom_vline(xintercept = 215)
```



Note the sharp discontinuity at 215.

b. We now want you to plot the graph that justifies our forcing variable. Plot the outcome as a function of the forcing variable. What do you see?

```
ggplot(df2, aes(x=pretest, y=posttest)) + geom_point() + geom_vline(xintercept = 215)
```



Note the disconinuity in the trend at 215.

c. Estimate the local average treatment effect (LATE) at the threshold using a linear model with common slopes for treated and control units (with no control variables). What are the additional assumptions required for this estimation strategy? Provide a plot of the post test scores (y-axis) and forcing variable (x-axis) in which you show the fitted curves and the underlying scatterplot of the data. Interpret your resulting estimate.

```
m1 = lm(posttest~treat+pretest, data=df2)
summary(m1)
```

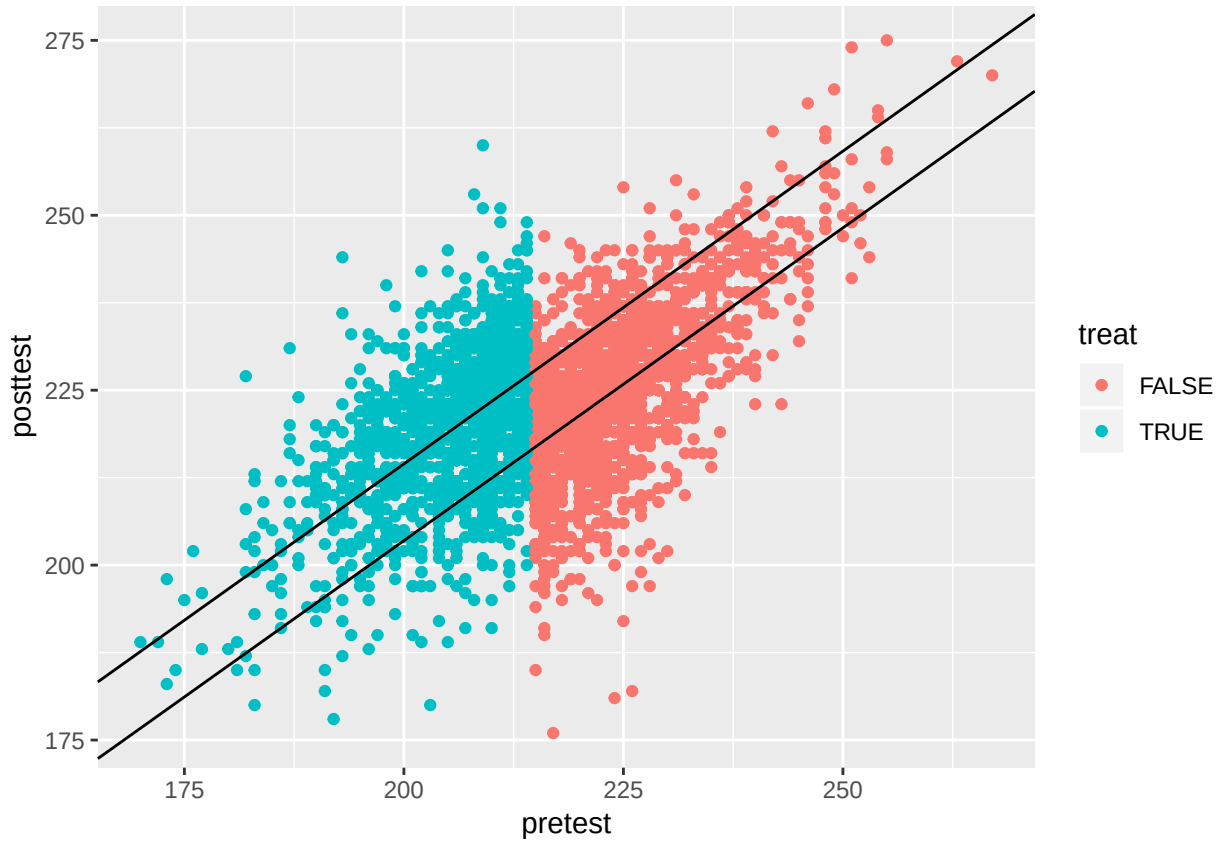
```
##
## Call:
## lm(formula = posttest ~ treat + pretest, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.929  -5.666   0.493   6.123  37.312
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.74034    5.12749   4.825 1.48e-06 ***
## treat       10.96764    0.58835  18.641  < 2e-16 ***
## pretest      0.89464    0.02278  39.281  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.552 on 2764 degrees of freedom
```

```
## Multiple R-squared:  0.4116, Adjusted R-squared:  0.4112
## F-statistic: 966.7 on 2 and 2764 DF,  p-value: < 2.2e-16
```

The main underlying assumption is that `pretest` is linear with the outcome `posttest`.

There is 10.9 point increase associated with tutoring atleast for those who around 215 cutoff threshold.

```
df2 = df2 %>% mutate(treat = as.logical(treat))
ggplot(df2, aes(pretest, posttest, color=treat)) +
  geom_point() +
  geom_abline(slope = 0.894, intercept = 24.7) +
  geom_abline(slope = 0.894, intercept = 24.7+10.967)
```




```
  geom_vline(xintercept = 215)
```

```
## mapping: xintercept = ~xintercept
## geom_vline: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

d. Re-do c., but use the control variables that are provided in the dataset. Interpret any differences you see.

```
m2 = lm(posttest~treat+pretest+age+gender+frlunch+esol+white+asian+black+hispanic, data=df2)
summary(m2)
```

```
##
## Call:
## lm(formula = posttest ~ treat + pretest + age + gender + frlunch +
##     esol + white + asian + black + hispanic, data = df2)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.813  -5.564   0.602   6.030  35.526
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 44.70578    7.02889   6.360 2.36e-10 ***
## treatTRUE   10.97281    0.59250  18.519  < 2e-16 ***
## pretest      0.87338    0.02352  37.141  < 2e-16 ***
## age         -1.19797    0.31496  -3.804 0.000146 ***
## gender      -0.53058    0.36969  -1.435 0.151350
## frlunch     -0.84770    0.41954  -2.021 0.043425 *
## esol         0.30282    0.64703   0.468 0.639809
## white        1.28124    1.16819   1.097 0.272837
## asian        5.34090    1.71424   3.116 0.001855 **
## black       -0.32084    1.16855  -0.275 0.783672
## hispanic     0.46652    1.20797   0.386 0.699377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.449 on 2671 degrees of freedom
##   (85 observations deleted due to missingness)
## Multiple R-squared:  0.4302, Adjusted R-squared:  0.4281
## F-statistic: 201.7 on 10 and 2671 DF,  p-value: < 2.2e-16
```

The LATE estimate is pretty much the same as (c).

e. Use the rdd package in R to estimate the LATE at the threshold using a local linear regression with a triangular kernel. Note that the function RDestimate automatically uses the Imbens-Kalyanamaran optimal bandwidth calculation. Report your estimate for the LATE and an estimate of uncertainty.

```
rde = RDestimate(posttest~pretest|age+gender+frlunch+esol+white+asian+black+hispanic,
                 data=df2,
                 cutpoint = 215)

summary(rde)
```

```
##
## Call:
## RDestimate(formula = posttest ~ pretest | age + gender + frlunch +
##     esol + white + asian + black + hispanic, data = df2, cutpoint = 215)
##
## Type:
## sharp
##
## Estimates:
##            Bandwidth  Observations  Estimate  Std. Error  z value
## LATE        6.009     1061          -10.77    1.3936       -7.732
## Half-BW     3.004      603          -11.09    2.4802       -4.471
## Double-BW  12.017     1838          -10.05    0.9382      -10.713
##            Pr(>|z|)
## LATE       1.060e-14  ***
## Half-BW    7.781e-06  ***
## Double-BW  8.798e-27  ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## F-statistics:
##             F      Num. DoF  Denom. DoF  p
## LATE       18.06  11         1049        0
## Half-BW    11.88  11          591        0
## Double-BW  21.99  11         1826        0
```

f. How do the estimates of the LATE at the threshold differ based on your results from parts (b) to (e)? In other words, how robust are the results to different specifications of the regression? What other types of robustness checks might be appropriate?

The results are fairly robust to different specifications. All the LATE estimates are within 2% of each other.

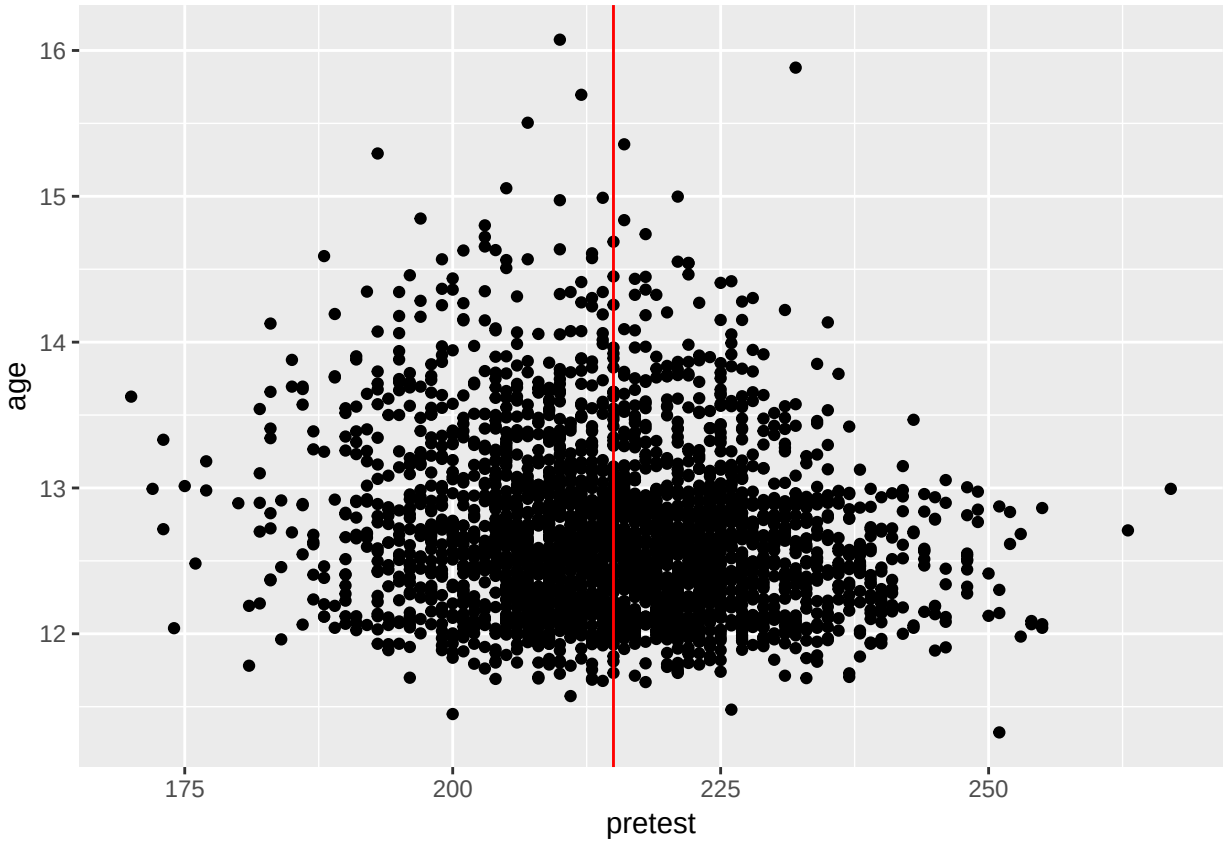
Some strategies for robustness checks may include: - Verify there is no manipulability in terms of who gets tutoring (i.e look at dsitribution of pretest scores. There should be no discontinuity) - Verify balance between treatment and control around the threshold.

We are now going to do a series of robustness checks:

h. Plot the age variable as a function of the forcing variable. What should this graph look like for our RDD to be a valid design? What do you see? How does this relate to the covariate balance exercise we did in Problem 1?

```
ggplot(df2, aes(x=pretest, y=age)) + geom_point() + geom_vline(xintercept = 215, color="red")
```

```
## Warning: Removed 44 rows containing missing values (geom_point).
```



We should expect age to be balanced atleast around the threshold. It looks the age around the threshold may be a little higher for students that received tutoring.

i. One type of placebo test is to pick arbitrary cutoffs of your forcing variable and estimate LATE's for those cutoffs. Pick 10 cutoffs and report the average LATE across those cutoffs. Defining $\hat{\tau}$ as your average LATEs, what should the null hypothesis on the population counterpart of this estimator be for our design to be valid. Feel free to use which specification you want for estimating LATE, but please specify it.

We can use RDD design with triangular kernel (RDEstimate) to estimate LATE. Our null hypothesis is that $\hat{\tau} = 0$. Another null hypothesis could be that all placebo LATE estimates (except at the actual cutoff) are jointly 0.

```
cutoffs = seq(min(df2$pretest)+10, max(df2$pretest)-10, 10)
placebo_lates = tibble(cutoff=double(), late=double())


for(c in cutoffs){

  placebo = RDestimate(posttest~pretest|age+gender+frlunch+esol+white+asian+black+hispanic,
           data=df2,
           cutpoint = c)
  placebo_lates = add_row(placebo_lates,
                          cutoff=c,
                          late=summary(placebo)$coefficients[1, "Estimate"])
}
```

```
print(placebo_lates)
```

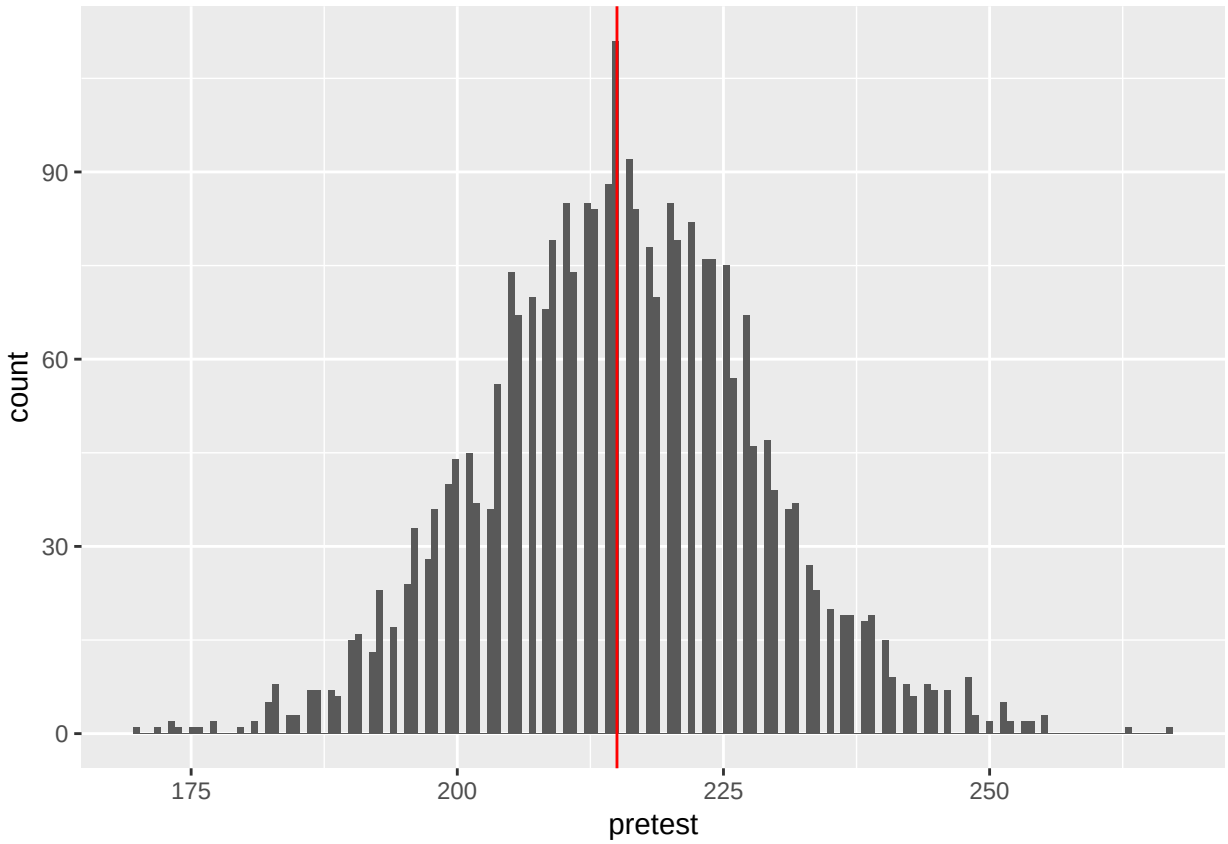
```
## # A tibble: 8 x 2
##   cutoff     late
##    <dbl>    <dbl>
## 1    180 -10.6
## 2    190  -5.30
## 3    200  -0.0164
## 4    210  -1.88
## 5    220   0.0403
## 6    230   0.153
## 7    240  -3.18
## 8    250  -4.45
```

```
mean(placebo_lates$late)
```

```
## [1] -3.154116
```

k. An issue with RD designs is manipulation, or sorting around the cutoff point. To assess this, plot a histogram of the forcing variable, drawing a line at the cutoff point. What would sorting around the cutoff point look like? What do you see?

```
ggplot(df2, aes(x=pretest)) + geom_histogram(bins=150) + geom_vline(xintercept = 215, color="red")
```



There is a spike right below the cutoff. There is some evidence for manipulability to get students into tutoring program.