Data_replication_1_code

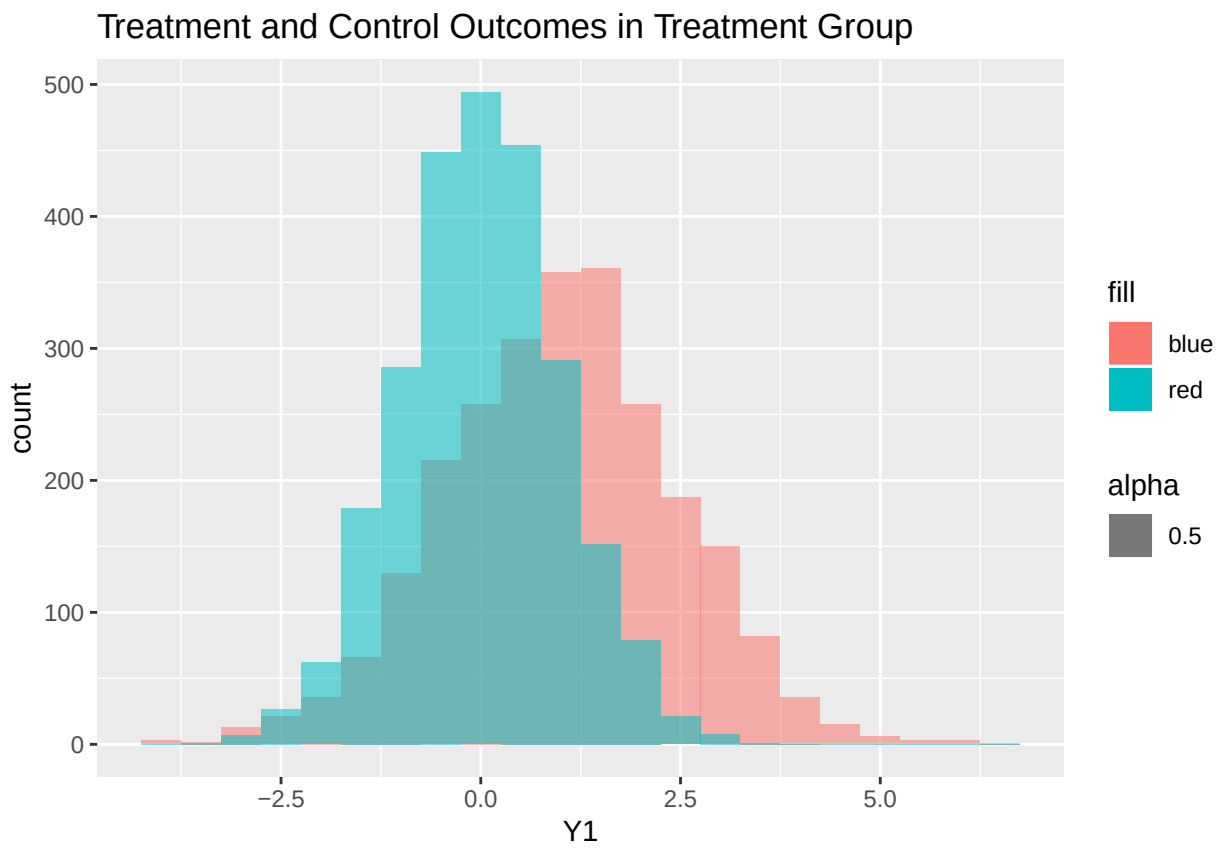
shan zhang

1/31/2020

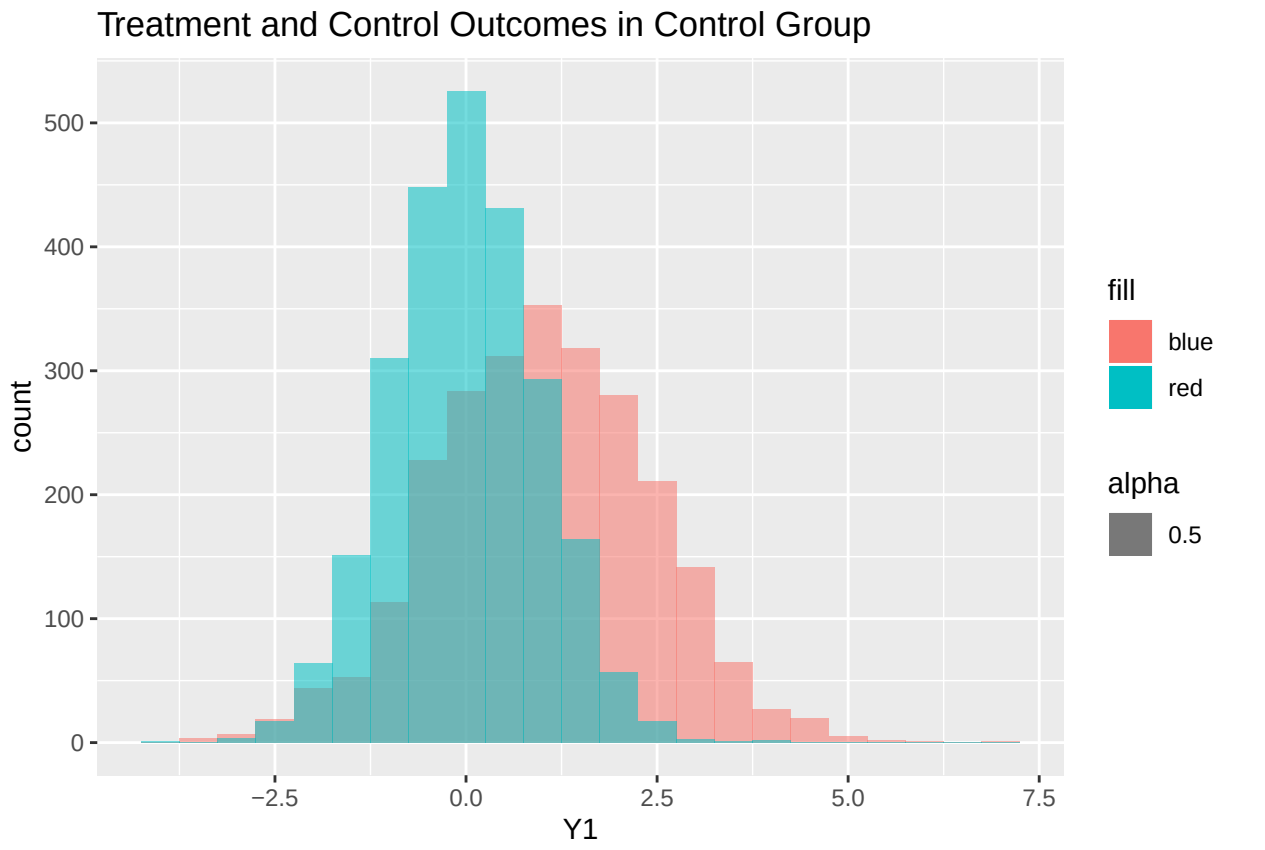
##Q1

a. There is an outcome Y1 which is the treatment outcomes for everyone. There is an outcome Y0 which is the control outcomes for everyone. There is an treatment variable D which indicates whether individuals are in the treatment group (or control group). Make a histogram comparing the treatment and control outcomes for the treatment group, and then comparing the treatment and control outcomes for the control group.

```
Q1 = read.csv(here::here("/data/Better_Late_Than_Never.csv"))

Q_t = Q1 %>%
  filter(D==1)
  ggplot(Q_t) +
geom_histogram(aes(Y1, fill = "blue", alpha=0.5), binwidth = 0.5, ) +
  geom_histogram(aes(Y0, fill = "red", alpha = 0.5), binwidth = 0.5) +
    ggtitle("Treatment and Control Outcomes in Treatment Group ")
```

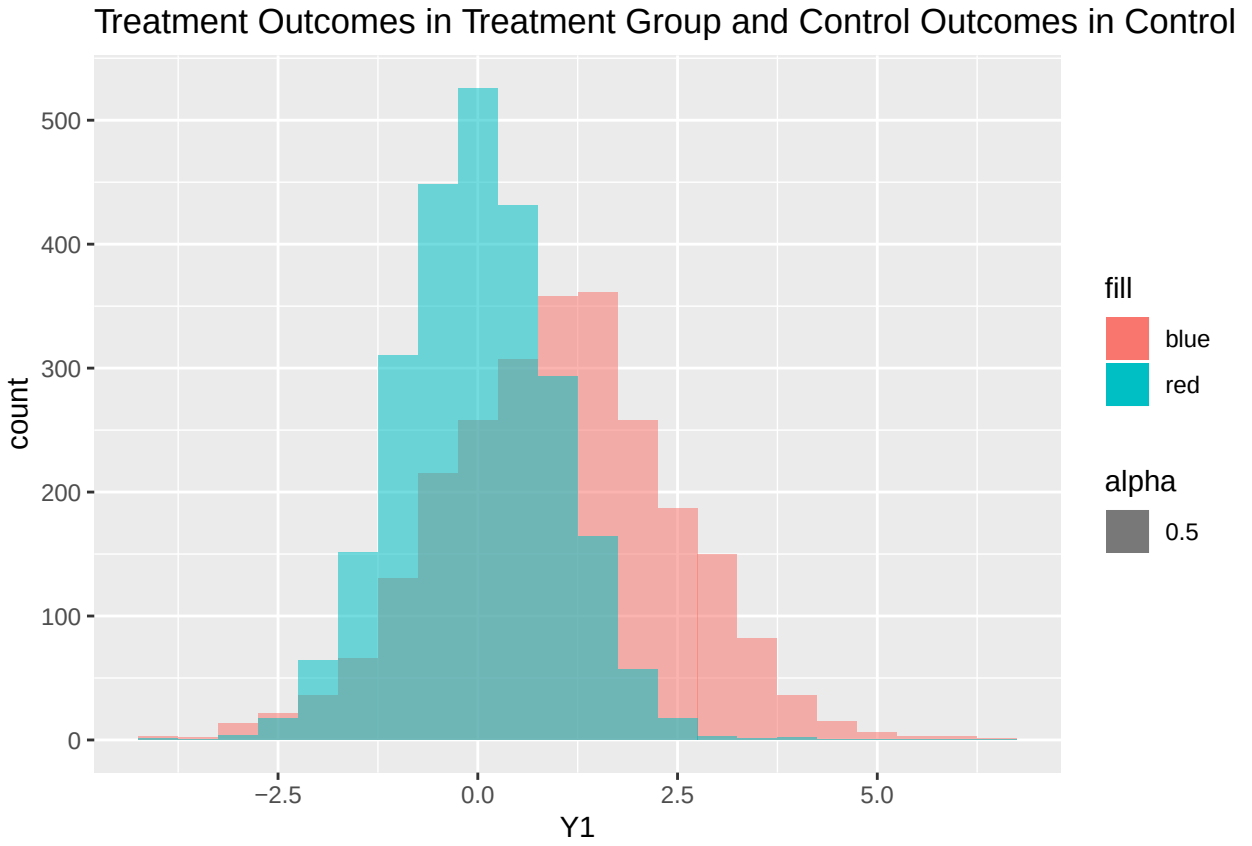


```
Q_c = Q1 %>%
  filter(D == 0)
ggplot(Q_c)+
geom_histogram(aes(Y1, fill="blue", alpha=0.5), binwidth = 0.5, ) +
  geom_histogram(aes(Y0, fill = "red", alpha = 0.5), binwidth = 0.5) +
  ggtitle("Treatment and Control Outcomes in Control Group")
```



b. Now make a histogram comparing the treatment outcomes for the treatment group, and the control outcomes for the control group. How does the compare to your histograms for part a? Why?

```
ggplot() +
    geom_histogram(data = Q_t,aes(Y1,fill = "blue", alpha = 0.5),binwidth = 0.5) +
    geom_histogram(data = Q_c,aes(Y0,fill = "red", alpha = 0.5),binwidth = 0.5)  +
  ggtitle("Treatment Outcomes in Treatment Group and Control Outcomes in Control Group")
```



This histogram is similar to the histograms in part a. The mean of treatment group is a little above 1 and the mean of control group is around 0.From this result we know that this experiment has been randomly assigned.

c. Finally, calculate the actual ATE by finding the average difference between Y1 and Y0 for the entire population.

```
avg_treat = Q_t %>% summarise(mean(Y1))
avg_control = Q_c %>% summarise(mean(Y0))
ATE = avg_treat - avg_control
ATE
```

```
##   mean(Y1)
## 1 1.016916
```

d. How does this compare to the coefficient from a linear regression where you only observe the Y1 outcome for treatment, Y0 for control, and a D variable for whether you are in treatment. What does this tell you about the importance of random assignment?

```
Q_t$ Y0 = 0
Q_c$ Y1 = 0
Q2 =  bind_rows(Q_t,Q_c)
lm2 = lm(Q1$y ~ Q1$D)
tidy(lm2)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) -0.00720    0.0247    -0.291 7.71e-  1
## 2 Q1$D         1.02       0.0348    29.2   5.35e-173
```

When we have random assignment, the coefficient(estimate) of treatment effect is the actual ATE and there is no selection bias.

##Q2 Suppose you are thinking about running an experiment. You hope to study whether assignment to Ben Hansen's metrics increases the odds of finding a job over taking Glen Waddell's class. The odd's of finding a job coming out of Glen's class is 70 percent.

a If you want a minimal detectable effect of increasing the odds of finding a job by 5 percent, how big would the entire sample need to be (assume the odds of ending up in either class if 50/50)?

$\beta_{MDE} = (t_{\frac{\alpha}{2}} + t_{1-k})\sqrt{(\frac{V(u)}{\overline{D}(1-\overline{D})N})}$ **Since** $(t_{\frac{\alpha}{2}} + t_{1-k}) = 2.8$ **And** $\beta_{MDE} = 0.05$**, u follows a bernoulli distribution,** $V(u) = 0.7 * 0.3 = 0.21$ $1 - \overline{D} = 0.5$ **Plug into the equation and solve for N, we get N=2635.**

b What is your minimal detectable effect if you have a sample size of 1000?

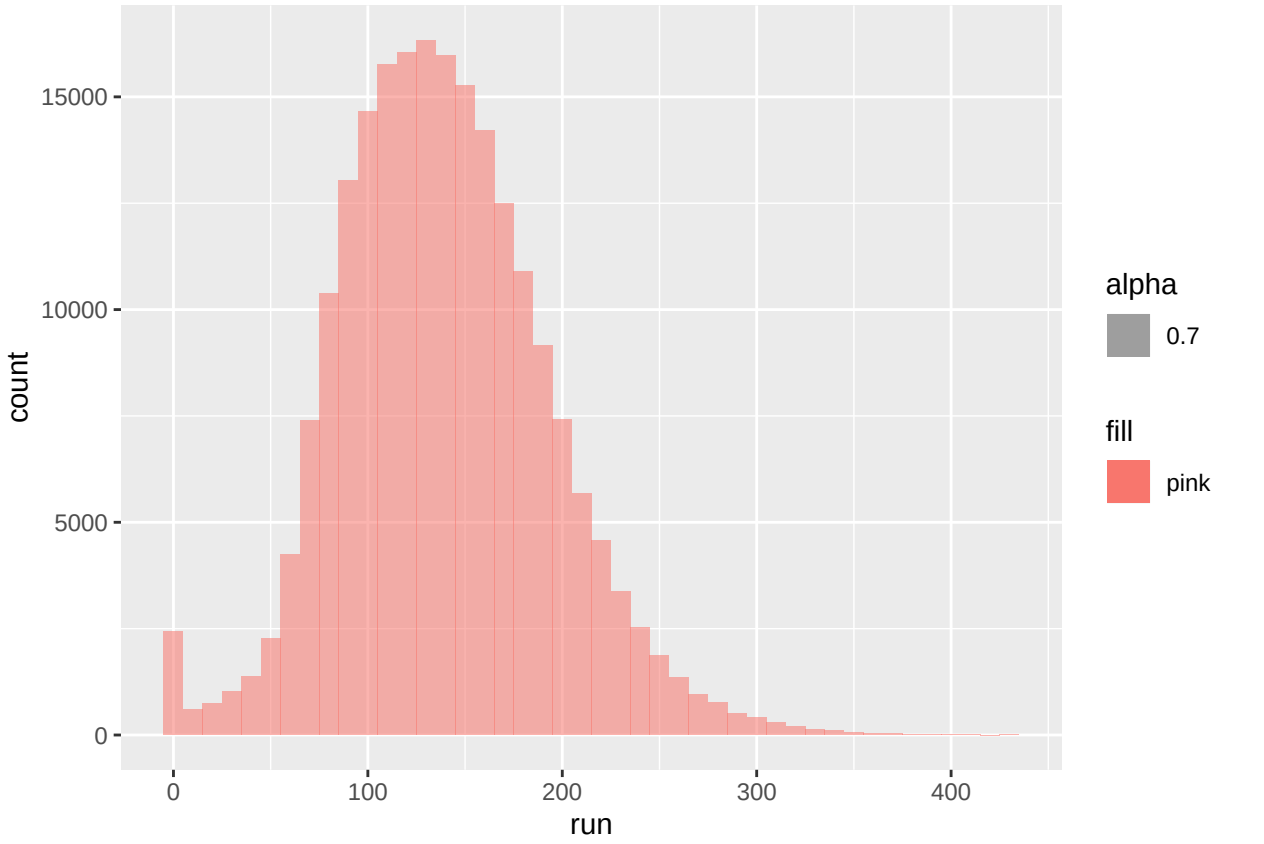
$\beta_{MDE} = (t_{\frac{\alpha}{2}} + t_{1-k})\sqrt{(\frac{V(u)}{\overline{D}(1-\overline{D})N})}$ **Since** $(t_{\frac{\alpha}{2}} + t_{1-k}) = 2.8$ **And** $N = 1000$**, u follows a bernoulli distribution,** $V(u) = 0.7 * 0.3 = 0.21$ $1 - \overline{D} = 0.5$ **Plug into the equation and solve for** β_{MDE}**, we get** $\beta_{MDE} = 0.081$.

Q3

```
BAC = read.csv(here::here("/data/BAC_deiden.csv"))
```

a. Create a histogram of the running variable, BAC. Make sure you do it allowing for discrete bins. Is there evidence of clear sorting at the threshold?

```
ggplot(BAC)+
  geom_histogram(aes(run,alpha=0.7,fill = "pink"), binwidth = 10)
```



From the histogram, I see some distinct jumps after 60 to 100.But I don't see evidence for clear shorting at threshold.

b. Next run a regression discontinuity model. To do so, create a dummy variable for a BAC over .08. Include that dummy variable, and the rescaled BAC (BAC-.08) as a control, and also include an interaction between that dummy variable and the running variable in model. First use age, gender, accident at the scene and race as outcomes. Do those factors shift at .08?

```
## # A tibble: 4 x 5
##   term          estimate std.error statistic p.value
##   <chr>            <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)  0.849      0.00410    207.      0
## 2 run_d        0.00169    0.00493      0.343   0.732
## 3 run          0.000162   0.000206     0.786   0.432
## 4 run_inter   -0.0000485  0.000225    -0.215   0.830
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  33.9      0.134     254.    0.
## 2 run_d        -0.147    0.161      -0.913 3.61e- 1
## 3 run          -0.0651   0.00673    -9.68  3.89e-22
## 4 run_inter     0.0722   0.00735     9.82  9.83e-23
```

```
## # A tibble: 4 x 5
##   term          estimate std.error statistic p.value
##   <chr>            <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)  0.785      0.00469    167.      0
## 2 run_d        0.00607    0.00564      1.07    0.282
## 3 run         -0.000152   0.000236    -0.644   0.520
## 4 run_inter    0.000251   0.000258     0.972   0.331
```

```
## # A tibble: 4 x 5
##   term          estimate std.error statistic   p.value
##   <chr>            <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)  0.0811     0.00351     23.1   5.91e-118
## 2 run_d       -0.000849   0.00422     -0.201 8.41e-  1
## 3 run         -0.00117    0.000176    -6.64  3.24e- 11
## 4 run_inter    0.00204    0.000193    10.6   3.86e- 26
```

Those factors do not shift at 0.08 since the coefficient estimates on both dummy and the interaction term are insignificant.

c. Now run a regression of recidivism on the same regression discontinuity design. What is your estimated effect using a bandwidth of .05, and a rectangular kernel (no weighting). Create a visualization of this by graphing the mean recidivism rate against the running variable. Show this for the whole BAC distribution, and the range from .03 to .13. Please include a fitted line.

```
lm_rd = lm(recidivism ~ run + run_inter + run_d, data = BAC_local, bandwidth=5, kernel="rectangular" )
```

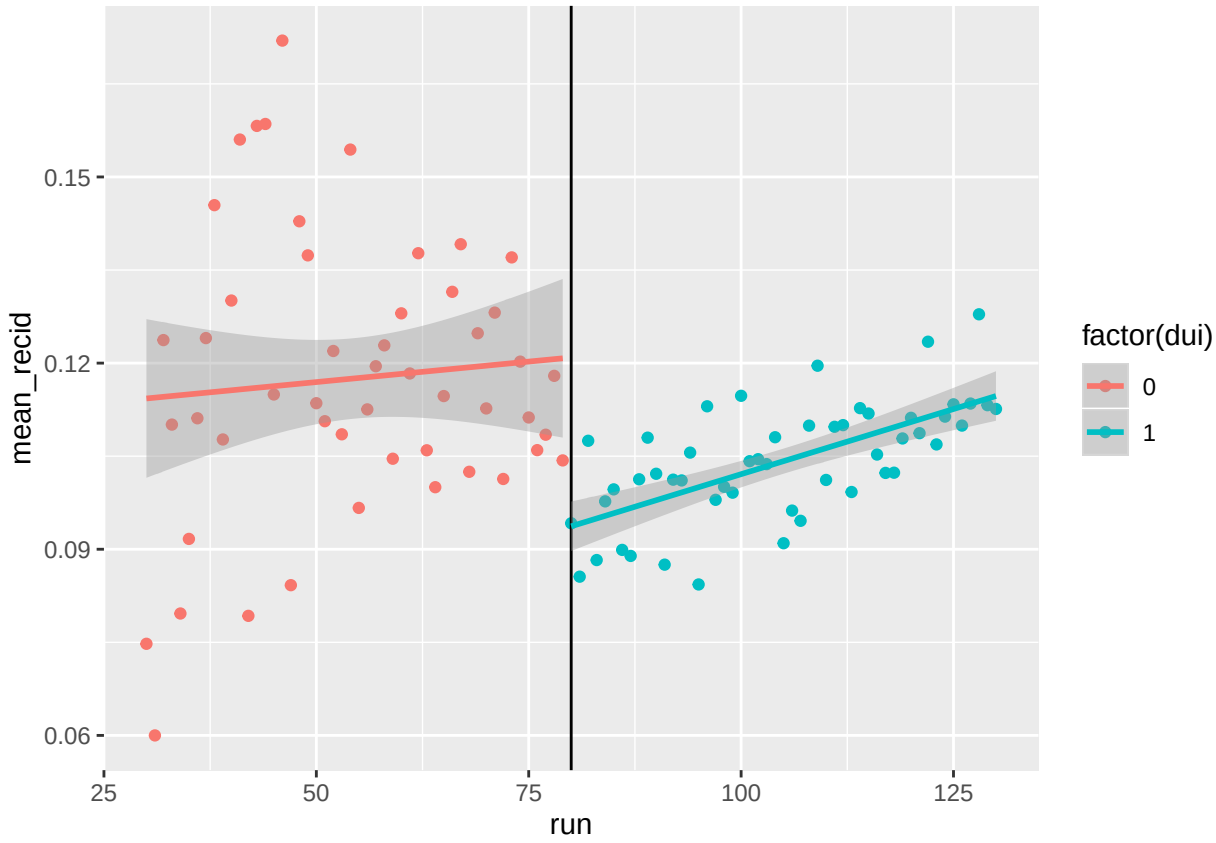
```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra arguments 'bandwidth', 'kernel' will be disregarded
```

```
tidy(lm_rd, conf.int = TRUE)
```

```
## # A tibble: 4 x 7
##   term          estimate std.error statistic   p.value  conf.low conf.high
##   <chr>            <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)  0.115      0.00358     32.2   8.00e-227  0.108     0.123
```

```
## 2 run          -0.000102  0.000180    -0.566 5.71e-  1 -0.000455  0.000251
## 3 run_inter     0.000524  0.000197     2.66  7.76e-  3  0.000138  0.000910
## 4 run_d        -0.0218    0.00431     -5.07  4.01e-  7 -0.0303   -0.0134
```

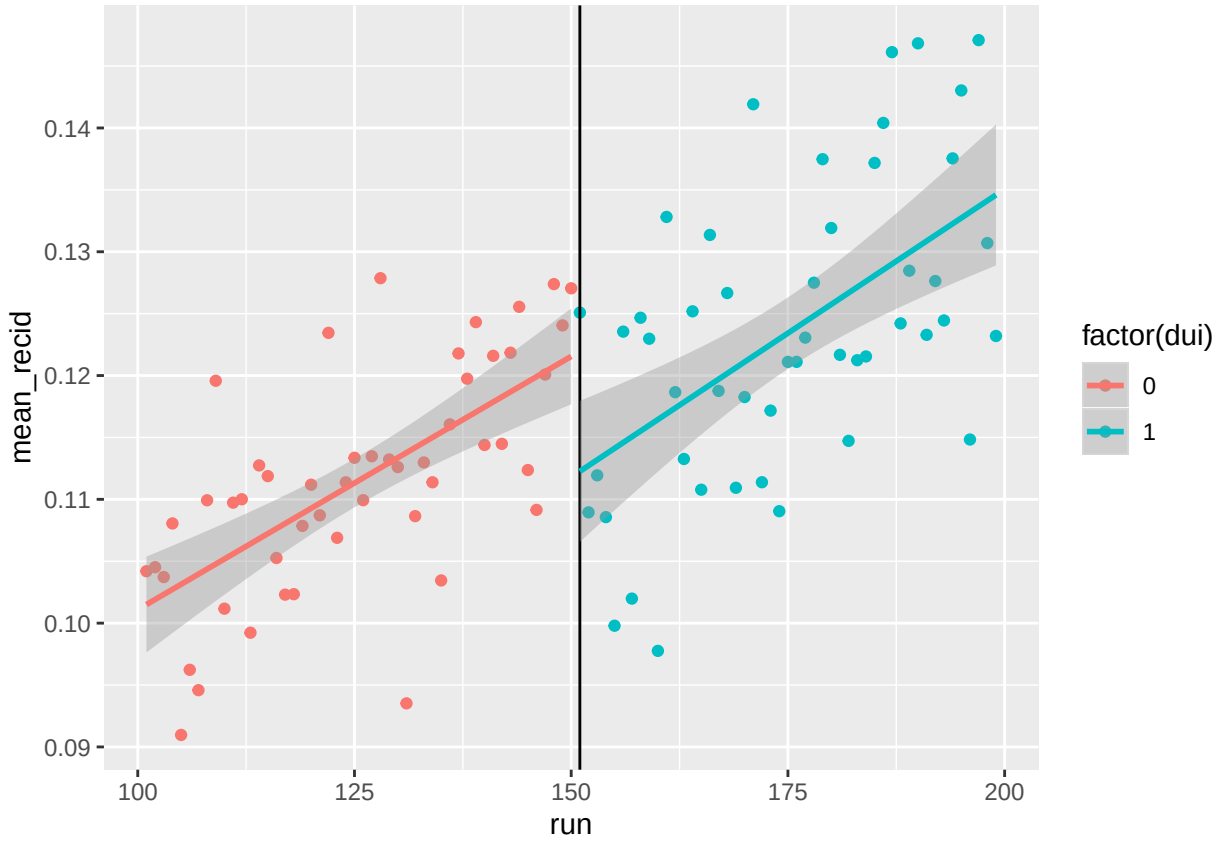
```
duimean <- BAC_local %>%
  group_by(run) %>%
   summarize(mean_recid = mean(recidivism, na.rm= TRUE))

duimean = duimean %>%
  mutate(run=run + 80) %>%
  mutate (dui = ifelse(run>=80,1,0))

ggplot(data=duimean, aes(x = run, y = mean_recid, colour=factor(dui)))+
  geom_point()+
  geom_vline(xintercept = 80) +
  stat_smooth( method = "lm", formula = y ~ x )
```



d.Do the same thing as part D but for the aggravated threshold of .151.

```
BAC_local_151 = BAC %>%
  subset(run > 100 & run < 200) %>%
  mutate(run_d = ifelse(run>=151,1,0)) %>%
  mutate(run = run -151) %>%
  mutate(run_inter = run * run_d)

duimean_151 <- BAC_local_151 %>%
  group_by(run) %>%
   summarize(mean_recid = mean(recidivism, na.rm= TRUE))
```

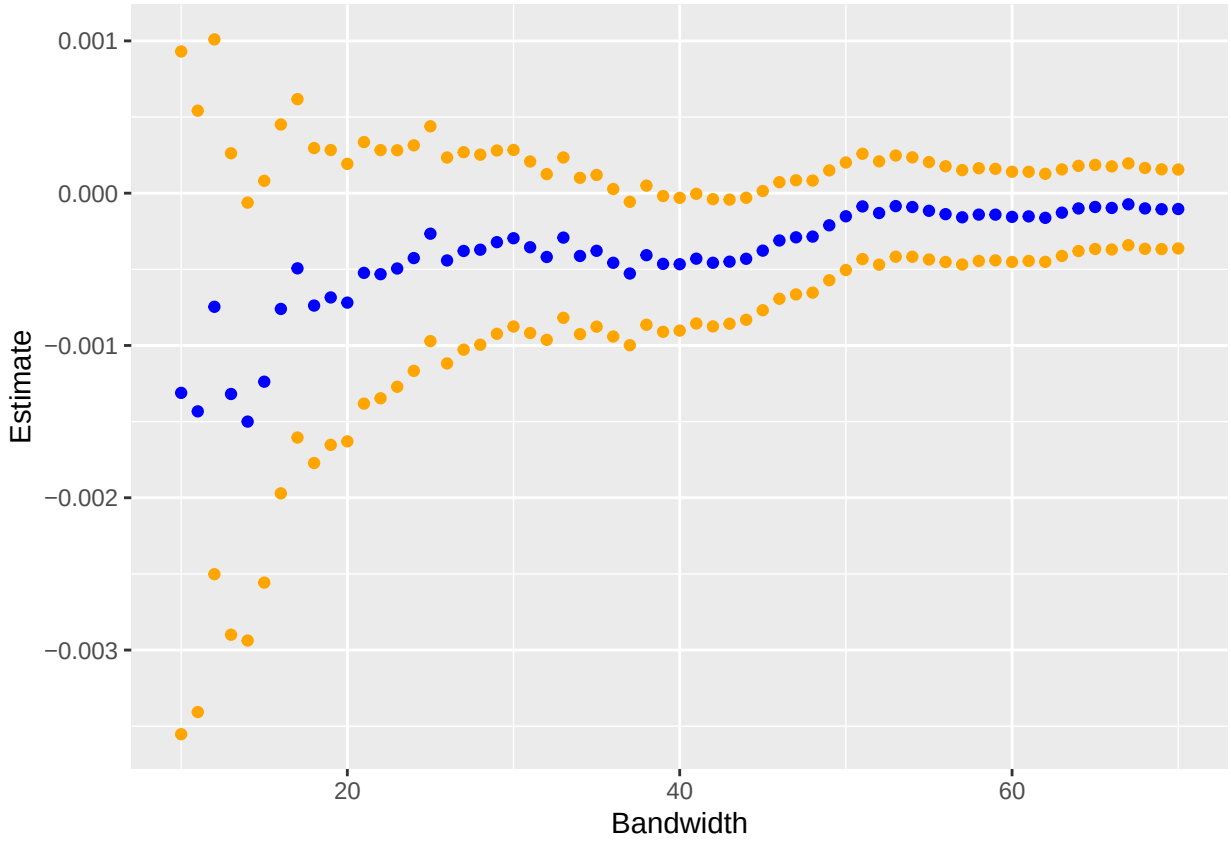
```
duimean_151 = duimean_151 %>%
  mutate(run=run + 151) %>%
  mutate (dui = ifelse(run>=151,1,0))

ggplot(data=duimean_151, aes(x = run, y = mean_recid, colour=factor(dui)))+
  geom_point()+
  geom_vline(xintercept = 151) +
  stat_smooth( method = "lm", formula = y ~ x )
```



e.Now run this model for every possible bandwidth between .01 and .07. Store both the point estimates and lower and upper confidence intervals. Create a scatter plot of the confidence interval and the point estimates. Are the estimates robust? Create a visualization of this.

```
BAC_loop = BAC %>%
   mutate(run_d = ifelse(run>=80,1,0)) %>%
  mutate(run_r = run - 80) %>%
  mutate(run_inter = run_r * run_d)




Data_sum = data.frame(
  var = numeric(0),
  estimate = numeric(0),
  std_error = numeric(0),
  stat = numeric(0),
  p_value = numeric(0),
```

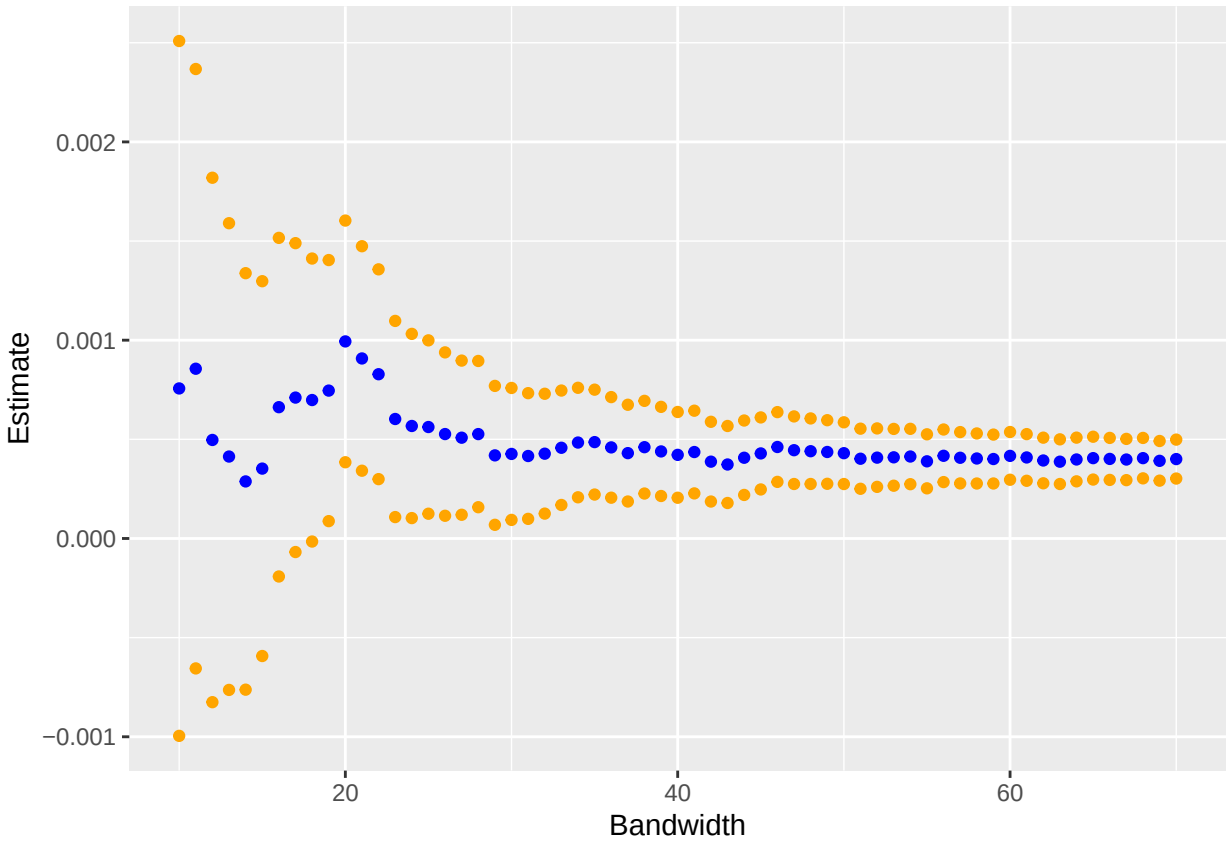
```
  conf_low = numeric(0),
  conf_high = numeric(0)
)

for (i in seq(10,70,1)){
  data = subset(BAC_loop, run > 79 - i & run < 81 +i)
  b = lm(recidivism ~ run_r + run_d + run_inter + aged + male +acc + white, data,kernel= "Rectangular")
  tidy(conf.int = TRUE)
  b=b[2,1:7]
  Data_sum = rbind(Data_sum, b)

}
Data_sum$term =10:70

ggplot(Data_sum,aes(x = term, y = estimate)) +
  geom_point(aes(x = term, y = estimate),color = "blue") +
  geom_point(aes(x = term, y = conf.high), color = "orange") +
  geom_point(aes(x = term, y = conf.low), color = "orange") +
  labs(x = "Bandwidth", y = "Estimate")
```



**

```
BAC_loop_151 = BAC %>%
   mutate(run_d = ifelse(run>=151,1,0)) %>%
  mutate(run_r = run - 151) %>%
  mutate(run_inter = run * run_d)
```

```r
Data_sum_151 = data.frame(
  var = numeric(0),
  estimate = numeric(0),
  std_error = numeric(0),
  stat = numeric(0),
  p_value = numeric(0),
  conf_low = numeric(0),
  conf_high = numeric(0)
)




for (i in seq(10,70,1)){
  data = subset(BAC_loop_151, run > 150 - i & run < 152 +i)
  b = lm(recidivism ~ run_r + run_d + run_inter + aged + male +acc + white, data) %>%
  tidy(conf.int = TRUE)
  b=b[2,1:7]
  Data_sum_151 = rbind(Data_sum_151, b)
}


Data_sum_151$term =10:70

ggplot(Data_sum_151,aes(x = term, y = estimate)) +
    geom_point(aes(x = term, y = estimate),color = "blue") +
  geom_point(aes(x = term, y = conf.high), color = "orange") +
  geom_point(aes(x = term, y = conf.low), color = "orange") +
  labs(x = "Bandwidth", y = "Estimate")
```



The orange points are upper and lower bounds of estimates and the blue points are estimates. From the scatter plots we can see, after bandwidth = 40, the estimates are more and more consistant(robust). Intuitively, when we have a very small bandwidth(sample), our variance is higher than the variance with a wider bandwidth.