Q1 (Matching):

In online retailing, Directed Search indicates when users search for a specific product name to purchase a product. On the other hand, undirected search indicates when a user stumbles upon a product to purchase. To examine if direct or undirected search increases sales, an analyst examines user activity from the website of an online retailer. The data set is developed from cookies that are saved on a user's computer along with data from the profile's users create:

```
library(dplyr)
library(ggplot2)
library(MatchIt)
search_data = read.csv("matching.csv")
```

(1) Run a naïve model where you regress sales (in $) on if the session was classified as a directed search or undirect search session. There are three metrics of the sales from a website. Examine the effect of directed sales on all three sales metrics.

```
promote_sales = summary(lm(PromotedSales ~ DirectedSearchUsage, data = search_data))
promote_sales
```

```
##
## Call:
## lm(formula = PromotedSales ~ DirectedSearchUsage, data = search_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1160.4  -454.8  -201.6   368.9  2895.2
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           510.172      6.561   77.76   <2e-16 ***
## DirectedSearchUsage  650.200     13.859   46.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 577.9 on 9998 degrees of freedom
## Multiple R-squared:  0.1804, Adjusted R-squared:  0.1803
## F-statistic:  2201 on 1 and 9998 DF,  p-value: < 2.2e-16
```

- PromotedSales=510.17+650.20×DirectedSearchUsage. The coefficent value of DirectedSearchUsage: 650.20, as DirectedSearchUsage increase by one unit, PromotedSales is estimated to increase by 650.20 units.

```
nonpromote_sales = summary(lm(NonpromotedSales ~ DirectedSearchUsage, data = search_data))
nonpromote_sales
```

```
##
## Call:
## lm(formula = NonpromotedSales ~ DirectedSearchUsage, data = search_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -152.20 -139.22  -85.96  101.72  929.67
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           152.197      1.974   77.12   <2e-16 ***
## DirectedSearchUsage  -66.232      4.169  -15.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 173.8 on 9998 degrees of freedom
## Multiple R-squared:  0.02462,    Adjusted R-squared:  0.02453
## F-statistic: 252.4 on 1 and 9998 DF,  p-value: < 2.2e-16
```

- NonpromotedSales = 152.19-66.23×DirectedSearchUsage. The coefficent value of DirectedSearchUsage: -66.23, as DirectedSearchUsage increase by one unit, NonpromotedSales is estimated to decrease by 66.23 units.

```r
overall_sales = summary(lm(OverallSales ~ DirectedSearchUsage, data = search_data))
overall_sales
```

```
##
## Call:
## lm(formula = OverallSales ~ DirectedSearchUsage, data = search_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1246.3  -591.5  -286.1   460.2  3824.8
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           662.370      8.463   78.27   <2e-16 ***
## DirectedSearchUsage  583.967     17.877   32.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 745.5 on 9998 degrees of freedom
## Multiple R-squared:  0.09643,    Adjusted R-squared:  0.09634
## F-statistic:  1067 on 1 and 9998 DF,  p-value: < 2.2e-16
```

- OverallSales = 662.37+583.97×DirectedSearchUsage. The coefficent value of DirectedSearchUsage: 583.97, as DirectedSearchUsage increase by one unit, OverallSales is estimated to increase by 583.97 units.

Users who indulge in directed search may be systematically different from those who perform undirect search. To overcome this issue, match users on the probability that they were performing directed search. Use propensity score matching to match users on log income, Education years, number of prior sessions, days since last purchase and historical total purchases. Regress the three sales metrics on the directed sales indicator for the matched set of users. Is there any difference in your findings after matching users?

```r
# Perform PScore
PScore = glm(DirectedSearchUsage ~ log(1+Income)+Education+NumSessions
             +DaysSinceLastPurchase+HistoricalTotalPurchases,
             data = search_data, family = "binomial")$fitted.values
search_data$PScore = PScore
```

```r
# Perform Matching. Use the matchIT command to generate propensity scores and match
# replace=FALSE will not take into match for next round
match_output = matchit(DirectedSearchUsage ~ log(1+Income)+Education+NumSessions
                       +DaysSinceLastPurchase+HistoricalTotalPurchases,
                       data= search_data,
                       method = "nearest",
                       distance = "logit",
                       caliper = .001, #how close treatment and control need to be
                       #if increase this value we will get more matched household
                       replace = FALSE, #match without replacement
                       ratio = 1) #one on one matching
summary(match_output)
```

```
##
## Call:
## matchit(formula = DirectedSearchUsage ~ log(1 + Income) + Education +
##     NumSessions + DaysSinceLastPurchase + HistoricalTotalPurchases,
##     data = search_data, method = "nearest", distance = "logit",
##     replace = FALSE, caliper = 0.001, ratio = 1)
##
## Summary of Balance for All Data:
##                          Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance                        0.6027        0.1148          1.7531     2.4928
## log(1 + Income)                 3.8669        1.7926          2.0008     0.8432
## Education                      15.4641       14.8536          0.3022     1.0197
## NumSessions                     5.0156        5.0447         -0.0132     0.9675
## DaysSinceLastPurchase           4.5695        4.4481          0.0206     0.9895
## HistoricalTotalPurchases        4.4289        4.4666         -0.0065     1.0011
##                          eCDF Mean eCDF Max
## distance                    0.4158   0.6767
## log(1 + Income)             0.4229   0.6622
## Education                   0.0360   0.1273
## NumSessions                 0.0038   0.0110
## DaysSinceLastPurchase       0.0095   0.0239
## HistoricalTotalPurchases    0.0058   0.0183
##
## Summary of Balance for Matched Data:
##                          Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance                        0.3549        0.3549          0.0002     1.0003
## log(1 + Income)                 3.0683        3.0673          0.0010     1.5004
## Education                      15.3364       15.3398         -0.0017     1.1173
## NumSessions                     5.0670        5.0409          0.0118     1.1020
## DaysSinceLastPurchase           4.5189        4.7481         -0.0388     0.7827
## HistoricalTotalPurchases        4.6232        4.2327          0.0674     1.7362
##                          eCDF Mean eCDF Max Std. Pair Dist.
## distance                    0.0001   0.0034          0.0004
## log(1 + Income)             0.0391   0.1148          0.4332
## Education                   0.0081   0.0386          1.1457
## NumSessions                 0.0085   0.0273          1.1033
## DaysSinceLastPurchase       0.0093   0.0284          0.8116
## HistoricalTotalPurchases    0.0087   0.0261          0.7833
##
## Sample Sizes:
```

```
##           Control Treated
## All          7759    2241
## Matched       880     880
## Unmatched    6879    1361
## Discarded       0       0
```

- Before matching, there were imbalances in covariates, as evidenced by non-zero standardized mean differences.
- After matching, the standardized mean differences are closer to 0, indicating improved balance.

```r
match_data = match.data(match_output)
```

```r
promote_sales = summary(lm(PromotedSales ~ DirectedSearchUsage, data = match_data))
promote_sales
```

```
##
## Call:
## lm(formula = PromotedSales ~ DirectedSearchUsage, data = match_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1035.2  -476.7  -187.5   402.1  2429.3
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           585.05      21.01   27.85   <2e-16 ***
## DirectedSearchUsage   450.15      29.71   15.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 623.1 on 1758 degrees of freedom
## Multiple R-squared:  0.1155, Adjusted R-squared:  0.115
## F-statistic: 229.6 on 1 and 1758 DF,  p-value: < 2.2e-16
```

- $\text{PromotedSales} = 585.05 + 450.15 \times \text{DirectedSearchUsage}$. The coefficent value of DirectedSearchUsage: 450.15, as DirectedSearchUsage increase by one unit, PromotedSales is estimated to increase by 450.15 units.
- The influence of directed search on sales appears to be lower in the matched data compared to the non-matched data.

```r
nonpromote_sales = summary(lm(NonpromotedSales ~ DirectedSearchUsage, data = match_data))
nonpromote_sales
```

```
##
## Call:
## lm(formula = NonpromotedSales ~ DirectedSearchUsage, data = match_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -148.83 -111.80  -91.07   89.81  751.44
##
## Coefficients:
```

```
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           148.832      5.671  26.246  < 2e-16 ***
## DirectedSearchUsage  -57.760      8.020  -7.202 8.74e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 168.2 on 1758 degrees of freedom
## Multiple R-squared:  0.02866,    Adjusted R-squared:  0.02811
## F-statistic: 51.87 on 1 and 1758 DF,  p-value: 8.736e-13
```

- NonpromotedSales = 148.83-57.76$\times$DirectedSearchUsage. The coefficent value of DirectedSearchUsage: -57.76, as DirectedSearchUsage increase by one unit, NonpromotedSales is estimated to decrease by 57.76 units.
- The influence of directed search on sales appears to be lower in the matched data compared to the non-matched data.

```r
overall_sales = summary(lm(OverallSales ~ DirectedSearchUsage, data = match_data))
overall_sales
```

```
##
## Call:
## lm(formula = OverallSales ~ DirectedSearchUsage, data = match_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1126.3  -595.9  -274.2   489.1  3147.6
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            733.89      26.36   27.84   <2e-16 ***
## DirectedSearchUsage    392.39      37.27   10.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 781.9 on 1758 degrees of freedom
## Multiple R-squared:  0.0593, Adjusted R-squared:  0.05876
## F-statistic: 110.8 on 1 and 1758 DF,  p-value: < 2.2e-16
```

- OverallSales = 733.89+392.39$\times$DirectedSearchUsage. The coefficent value of DirectedSearchUsage: 392.39, as DirectedSearchUsage increase by one unit, OverallSales is estimated to increase by 392.39 units.
- The influence of directed search on sales appears to be lower in the matched data compared to the non-matched data.

Q2 (Synthetic Control):

One of the first contexts where the synthetic control approach was developed was to understand if raising taxes on cigarette sales in California decreased sales. In this question, you are required to import data on cigarette taxes and sales. Here is what you need to do

```r
library(tidyr)
library(glmnet)
library(janitor)
library(Synth)
library(ggthemes)
library(patchwork)
```

```r
smoke_data = read.csv("smoking.csv")
smoke_data <- as.data.frame(smoke_data)
```

(1) Import the smoking.csv data.

```r
smoke_data$state_id <- as.numeric(factor(smoke_data$state))
smoke_data$year <- as.numeric(smoke_data$year)
smoke_data$treat <- smoke_data$state_id == 3
```

(2) Create unique numeric state IDs

(3) Use the following variables as predictors: Inincome, retprice, age15to25,beer, lagged cigertte sales from 1975, 1980, 1988

```r
treatment_unit = 3
Sys.setenv(LANGUAGE = "en")
# Now we can use Synth's data preparation package.
dataprep.out=
  dataprep(foo = smoke_data,
           dependent = "cigsale",
           unit.variable = "state_id",
           time.variable = "year",

           predictors = c("lnincome", "retprice", "age15to24", "beer"),
           predictors.op = "mean",
           special.predictors = list(list("cigsale", 1975, "mean"), list("cigsale", 1980, "mean"), list
           treatment.identifier = 3,

           #which panels are we using to construct the synthetic control?
           # Controls here will be every other district.!!
           controls.identifier = setdiff(unique(smoke_data$state_id), treatment_unit),

           time.predictors.prior = c(1970:1988),
           time.optimize.ssr = c(1970:1988),

           unit.names.variable = "state",
           time.plot = 1970:2000)
```

(4) Use data between 1970 and 1988 to create synthetic control

```
##
##  Missing data- treated unit; predictor: lnincome ; for period: 1970
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: lnincome ; for period: 1971
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1970
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1971
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1972
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1973
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1974
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1975
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1976
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1977
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1978
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1979
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1980
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1981
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1982
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1983
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: lnincome ; for period: 1970
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
```

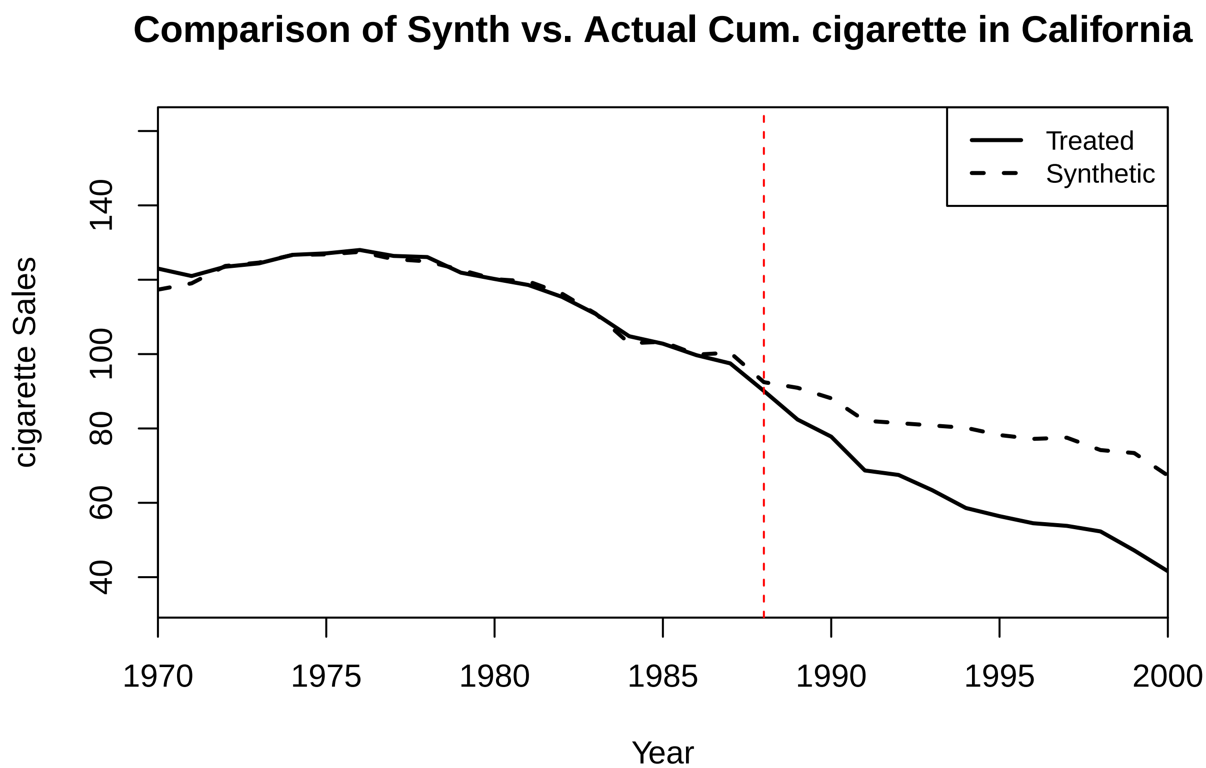
```
##  Missing data - control unit: 1 ; predictor: lnincome ; for period: 1971
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: beer ; for period: 1970
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: beer ; for period: 1971
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: beer ; for period: 1972
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: beer ; for period: 1973
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
```

```
synth.out = synth(dataprep.out)
```

```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 3.069261
##
## solution.v:
##  0.0010901 0.009765174 0.001041528 0.01119902 0.5811205 0.3235615 0.07222212
##
## solution.w:
##  3.7609e-06 3.0555e-06 0.09383158 0.1106215 4.3302e-06 3.8177e-06 1.3774e-06 1.53224e-05 3.0171e-06 8
```

```
path.plot(dataprep.res = dataprep.out, synth.res = synth.out,Xlab="Year",Ylab="cigarette Sales",Main="C
abline(v=1988,lty=2,col="red")
```

(5) Using ggplot, plot the line for the actual sales in California and the synthetic control



Q3 (Regression Discontinuity): Does the position of an online advertisement impact the number of clicks it obtains? This question can be analyzed using a regression discontinuity model.

```
library(rddtools)
library(rdrobust)
library(rdd)
```

```
rd_data = read.csv("rd.csv")
```

```
rd_data <- rd_data %>%
  group_by(auction_id) %>%
  arrange(desc(bid)) %>%
  mutate(rank = row_number())
```

(1) Assign a rank to each of the bids. These ranks will be the within auction rank of the different bids. In auction id = 1, the bid with a value of 4.23 will have rank 1 (as it is the bid in the auction with the highest value), the bid with a value of 4.15 will have a value of 2, and so on.

```r
rd_data <- rd_data %>% filter(rank %in% c(1, 2))
```

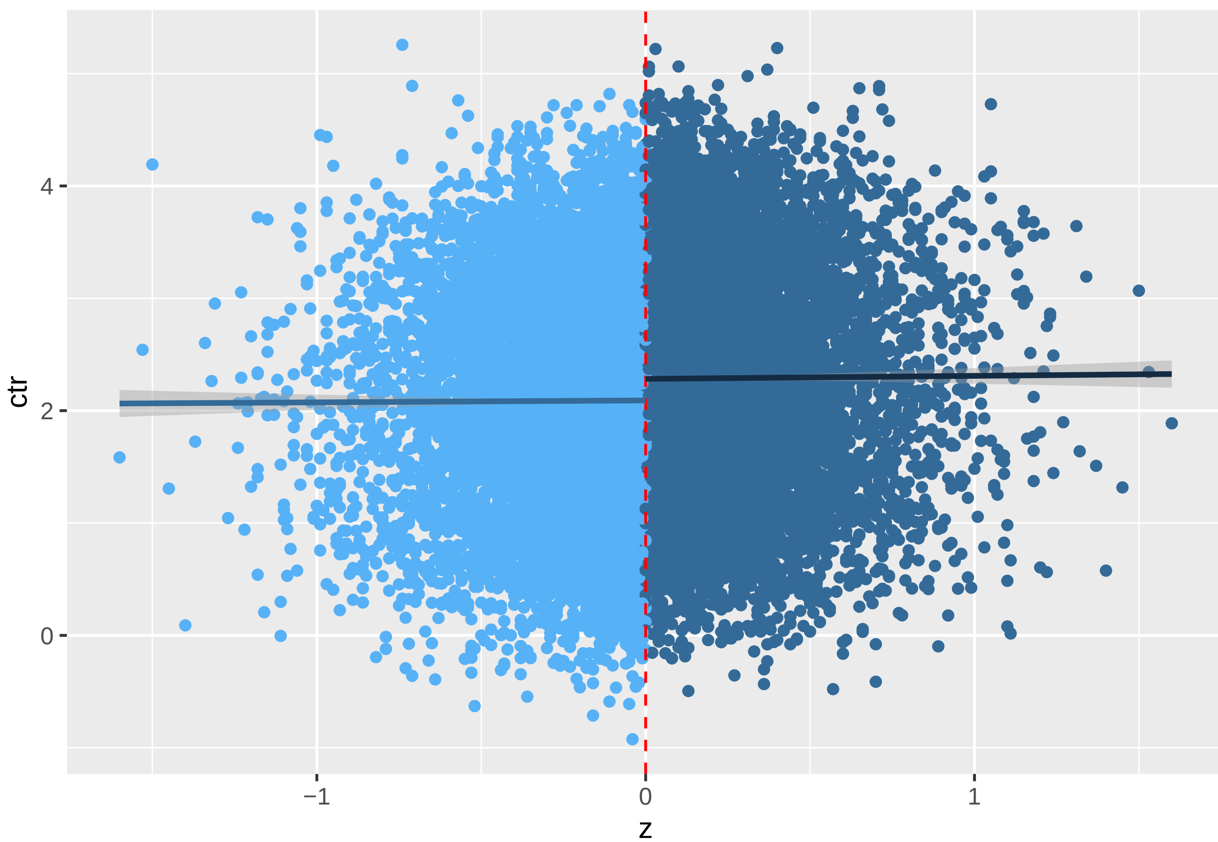
(2) Limit your analysis to advertisements that have a rank of 1 or 2.

```r
rd_data <- rd_data %>%
  arrange(auction_id, rank) %>%
  group_by(auction_id) %>%
  mutate(z = ifelse(rank == 1, -diff(bid), diff(bid)))
rd_data <- rd_data %>%
  mutate(treat = ifelse(rank==1,1,0))
```

```r
ggplot(rd_data, aes(y=ctr,x=z)) +
  geom_point(aes(color = rank+1),show.legend = FALSE) +
  geom_vline(xintercept = 0, linetype="dashed", color="red") +
  geom_smooth(aes(group = factor(rank), color = rank), method = "lm", show.legend=FALSE)
```

(3) Within each auction id, compute the difference in the bid between advertisements with rank 1 and rank 2. For example, the difference in the bid amount between advertisements with rank 1 and 2 is 0.08. The positive value of this will serve as the forcing value for the bid with rank 1 and the negative value of this will serve as the forcing value for the bid with rank 2.

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




```r
# naive rdd without bandwidth
summary(lm(ctr ~ treat+z, data=rd_data))
```

Use this dataset to examine how click through rates differ for advertisements for rank 1 and 2 using a regression discontinuity framework. The value of the cutoff will be 0 in this case.

```
##
## Call:
## lm(formula = ctr ~ treat + z, data = rd_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0156 -0.7584 -0.0026  0.7613  3.1801
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.09221    0.01268 164.997   <2e-16 ***
## treat        0.19181    0.02110   9.089   <2e-16 ***
## z            0.02203    0.03178   0.693    0.488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9946 on 19997 degrees of freedom
```

```
## Multiple R-squared:  0.0103, Adjusted R-squared:  0.01021
## F-statistic: 104.1 on 2 and 19997 DF,  p-value: < 2.2e-16
```

- Use naive linear regression, including the confound variable z, ranking in the first place increase average 19% in click through rate.

```r
## Use rdrobust to find out the bandwidth
rd_robust_rdd = rdrobust(rd_data$ctr, rd_data$z, c=0)
```

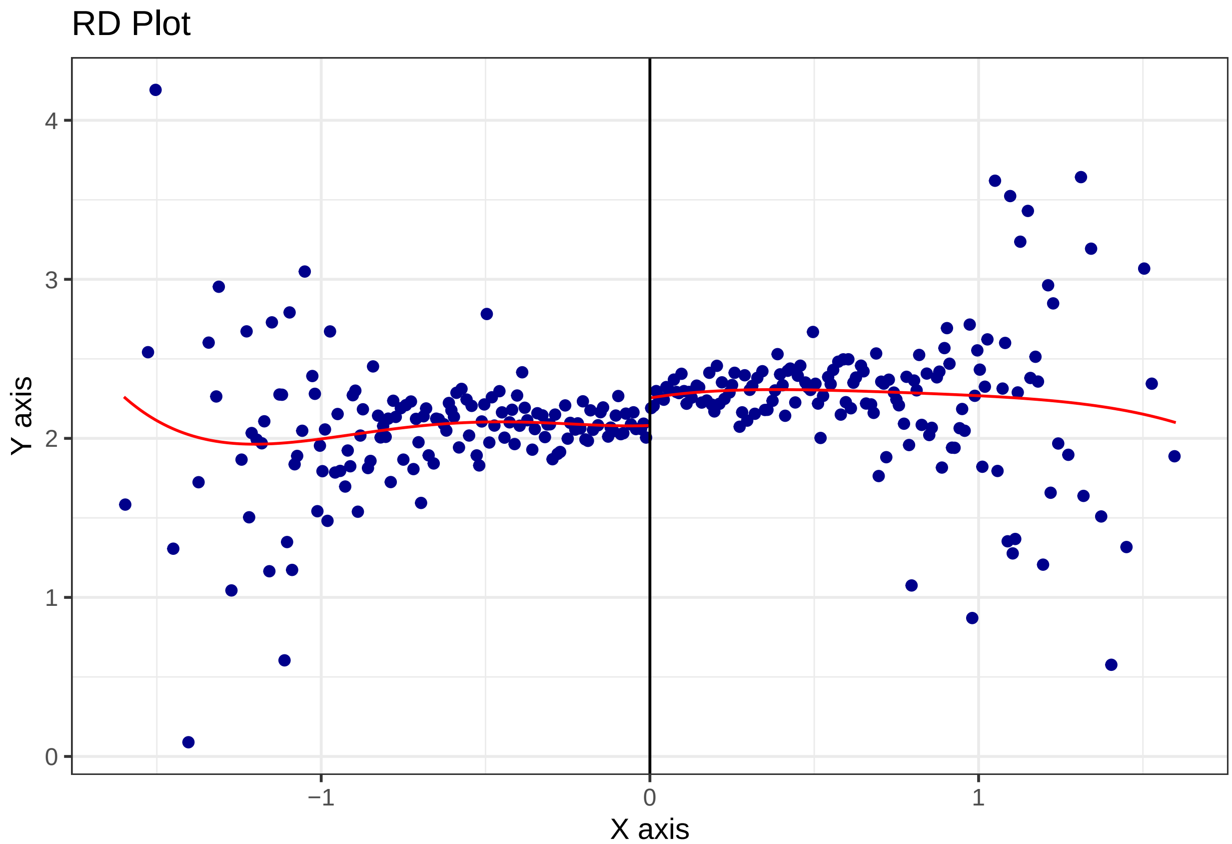
```
## Warning in rdrobust(rd_data$ctr, rd_data$z, c = 0): Mass points detected in the
## running variable.
```

```r
summary(rd_robust_rdd)
```

```
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                 20000
## BW type                        mserd
## Kernel                    Triangular
## VCE method                        NN
##
## Number of Obs.                  9837        10163
## Eff. Number of Obs.             7118         7444
## Order est. (p)                     1            1
## Order bias  (q)                    2            2
## BW est. (h)                    0.339        0.339
## BW bias (b)                    0.602        0.602
## rho (h/b)                      0.564        0.564
## Unique Obs.                      457          458
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##   Conventional     0.182     0.031     5.807     0.000     [0.121 , 0.244]
##         Robust         -         -     4.926     0.000     [0.108 , 0.250]
## =============================================================================
```

```r
rdplot(rd_data$ctr, rd_data$z, c=0)
```

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

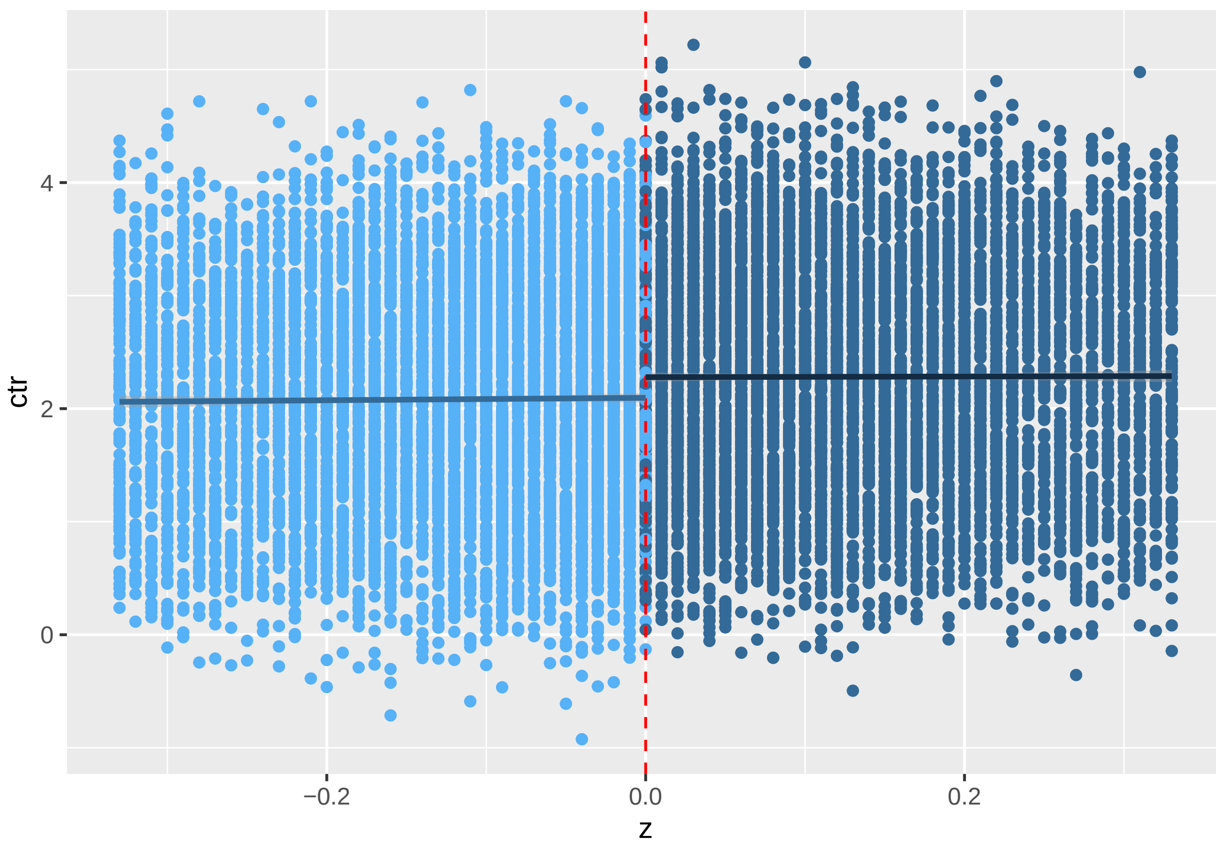


- Use built-in package to find out the bandwidth and coefficient. Here, ranking in the first place increase average 18% in click through rate.

```
rd_2 = rd_data %>% filter(z >= -0.339 & z <= 0.339)

ggplot(rd_2, aes(y=ctr,x=z)) +
  geom_point(aes(color = rank+1), show.legend = FALSE) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "red") +
  geom_smooth(aes(group = factor(rank), color = rank), method = "lm", show.legend=FALSE)
```

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




```r
# try considering polynomial trend
summary(lm(ctr ~ treat+z+z_sq, data=rd_2 %>% mutate(z_sq = z*z)))
```

```
##
## Call:
## lm(formula = ctr ~ treat + z + z_sq, data = rd_2 %>% mutate(z_sq = z *
##     z))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.01848 -0.75841 -0.01005  0.76177  2.93697
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.09734    0.01831 114.520  < 2e-16 ***
## treat        0.18267    0.02913   6.270 3.71e-10 ***
## z            0.06827    0.08836   0.773    0.440
## z_sq        -0.24891    0.27465  -0.906    0.365
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9915 on 14558 degrees of freedom
## Multiple R-squared:  0.0103, Adjusted R-squared:  0.01009
## F-statistic: 50.48 on 3 and 14558 DF,  p-value: < 2.2e-16
```

```r
# try considering different slopes below vs. above the threshold
summary(lm(ctr ~ treat*z, data=rd_2 %>% mutate(z_sq = z*z)))
```

```
##
## Call:
## lm(formula = ctr ~ treat * z, data = rd_2 %>% mutate(z_sq = z *
##     z))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.01595 -0.75861 -0.01014  0.76202  2.93928
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.09600    0.02060 101.743  < 2e-16 ***
## treat        0.18267    0.02913   6.270 3.71e-10 ***
## z            0.10815    0.12497   0.865    0.387
## treat:z     -0.07975    0.17673  -0.451    0.652
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9915 on 14558 degrees of freedom
## Multiple R-squared:  0.01025,    Adjusted R-squared:  0.01005
## F-statistic: 50.27 on 3 and 14558 DF,  p-value: < 2.2e-16
```

Conclusion: The effect of the rank on click through rate. Compared to rank in the second place, rank in the first place increase about 18% click through rate.