RocketFuel Case

9084959957

2022-10-31

Load the Data

We will load all the necessary libraries in order to work on our code. Then we will read the file into R where all the data is store.

```
rm(list = ls())

library(readxl)
library(knitr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(dplyr)

rocketfuel <- read.csv("rocketfuel_deciles.csv", header=TRUE)
summary(rocketfuel)# gives the summary of our dataset
```

```
##     user_id             test        converted         tot_impr
##  Min.   : 900000   Min.   :0.00   Min.   :0.00000   Min.   :   1.00
##  1st Qu.:1143190   1st Qu.:1.00   1st Qu.:0.00000   1st Qu.:   4.00
##  Median :1313725   Median :1.00   Median :0.00000   Median :  13.00
##  Mean   :1310692   Mean   :0.96   Mean   :0.02524   Mean   :  24.82
##  3rd Qu.:1484088   3rd Qu.:1.00   3rd Qu.:0.00000   3rd Qu.:  27.00
##  Max.   :1654483   Max.   :1.00   Max.   :1.00000   Max.   :2065.00
##  mode_impr_day   mode_impr_hour  tot_impr_decile
##  Min.   :1.000   Min.   : 0.00   Min.   : 1.000
##  1st Qu.:2.000   1st Qu.:11.00   1st Qu.: 3.000
##  Median :4.000   Median :14.00   Median : 5.000
##  Mean   :4.026   Mean   :14.47   Mean   : 5.448
##  3rd Qu.:6.000   3rd Qu.:18.00   3rd Qu.: 8.000
##  Max.   :7.000   Max.   :23.00   Max.   :10.000
```

```r
head(rocketfuel)#shows the how the data looks
```

```
##   user_id test converted tot_impr mode_impr_day mode_impr_hour tot_impr_decile
## 1 1391842    1         0        2             3             19               2
## 2 1215269    1         0        1             4             12               1
## 3 1604030    1         0        2             6             11               2
## 4 1278452    1         0        1             7             18               1
## 5 1363432    1         0        1             6             13               1
## 6  909876    0         0        1             3             12               1
```

```r
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
psych::describe(rocketfuel) # this describes the data we have
```

```
##                 vars      n       mean        sd  median    trimmed       mad
## user_id            1 588101 1310692.22 202225.98 1313725 1313693.47 252713.62
## test               2 588101       0.96      0.20       1       1.00      0.00
## converted          3 588101       0.03      0.16       0       0.00      0.00
## tot_impr           4 588101      24.82     43.72      13      16.27     14.83
## mode_impr_day      5 588101       4.03      2.00       4       4.03      2.97
## mode_impr_hour     6 588101      14.47      4.83      14      14.59      4.45
## tot_impr_decile    7 588101       5.45      2.86       5       5.43      2.97
##                   min     max  range  skew kurtosis     se
## user_id         9e+05 1654483 754483 -0.10    -1.04 263.70
## test            0e+00       1      1 -4.69    20.04   0.00
## converted       0e+00       1      1  6.05    34.65   0.00
## tot_impr        1e+00    2065   2064  7.43   109.92   0.06
## mode_impr_day   1e+00       7      6 -0.04    -1.24   0.00
## mode_impr_hour  0e+00      23     23 -0.34     0.10   0.01
## tot_impr_decile 1e+00      10      9  0.03    -1.21   0.00
```

After looking at the code we can understand how our data looks. In this dataset we have
user id: Unique identifier of the user

test: Whether the user was exposed to advertising or was in the control group. 1 if the user was exposed to the real ad, 0 if the user was in the control group and was shown a PSA.

converted: Whether the user converted. 1 if the user bought the handbag during the campaign, 0 if not.

tot_impr: The total number of ad impressions the user encountered. For users in the control group this counts the number of times they encountered the PSA.For exposed users it counts the number of times they were shown the ad.

mode_impr_day: Shows the day of the week on which the user encountered the most number of impressions. 1 means Monday, 7 means Sunday. For example if a given user encountered 2 impressions on Mondays, 3 on

Tuesdays, 7 on Wednesdays, 0 on Thursdays and, Fridays, 9 on Saturdays and 2 on Sundays, this column takes the value of 6 (Saturday).

tot_impr_decile: A column labeling people into deciles (i.e., 10% groups) by tot_impr.

We can also look at the mean, max, median of all the vairables described above.

Tabulating the treatment groups to see the shares Treatment vs. Control

In this we will look at the count and share of the control group and treatment group. Here our control group are the people to whom ads were not shown and our treatment group are the people to whom the ads were shown to.

```
attach(rocketfuel)
tb_treatment_full <- matrix(NA, nrow = 2, ncol = 2)
tb_treatment_full[1,] <- format(table(test), digits = 1)
tb_treatment_full[2,] <- format(prop.table(table(test)),digits = 2)
rownames(tb_treatment_full) <- c("Frequency", "Proportion" )
colnames(tb_treatment_full) <- c("Control Group",  "Treatment Group")
kable(tb_treatment_full)
```

	Control Group	Treatment Group
Frequency	23524	564577
Proportion	0.04	0.96

```
detach(rocketfuel)
```

We see that there is significant difference between the control group and the treatment group which is correct as we know that TaskaBella did not want a huge control group as they wanted maximum people to see the ad in order to increase their sells.

Check for balance in the variables

We'll first look at the mean and standard deviation of all the oter variables

```
sumtb<- matrix(NA,nrow = 2, ncol = 7)
rownames(sumtb) <- c("mean", "sd")
colnames(sumtb) <- colnames(rocketfuel)
sumtb[1,] <- round(apply(rocketfuel,2,mean),2)
sumtb[2,] <- round(apply(rocketfuel,2,sd),2)
sumtb <- sumtb[,4:6]
kable(sumtb)
```

	tot_impr	mode_impr_day	mode_impr_hour
mean	24.82	4.03	14.47
sd	43.72	2.00	4.83

Then we'll look at the mean of control group and treatment group of other variables seen above in order to check the balance in variables

```r
attach(rocketfuel)

preexp <- rocketfuel %>%
  select(tot_impr, mode_impr_day , mode_impr_hour)

tb_preexp <- matrix(NA, nrow = 3, ncol = 2)
colnames(tb_preexp) <- c( "Mean Control Group", "Mean treatment group")
rownames(tb_preexp) <- colnames(preexp)

m<-as.matrix(round(aggregate(.~test,preexp,mean),2))

tb_preexp[,1:2] <-t(m)[2:4,]

kable(tb_preexp)
```

	Mean Control Group	Mean treatment group
tot_impr	24.76	24.82
mode_impr_day	3.95	4.03
mode_impr_hour	14.30	14.48

```r
detach(rocketfuel)
```

We can see that the difference between the mean of control and treatment is very less.

Similarly, we can see the standard deviation for both control and treatment group

```r
attach(rocketfuel)

preexp <- rocketfuel %>%
  select(tot_impr, mode_impr_day , mode_impr_hour)

tb_preexp <- matrix(NA, nrow = 3, ncol = 2)
colnames(tb_preexp) <- c( "Sd Control group", "Sd treatment group")
rownames(tb_preexp) <- colnames(preexp)

msd<-as.matrix(round(aggregate(.~test,preexp,sd),3))

tb_preexp[,1:2] <-t(msd)[2:4,]

kable(tb_preexp)
```

	Sd Control group	Sd treatment group
tot_impr	42.861	43.750
mode_impr_day	1.949	2.006
mode_impr_hour	4.656	4.842

```r
detach(rocketfuel)
```

Here too the difference is very less that suggests that the varibles are balanced.

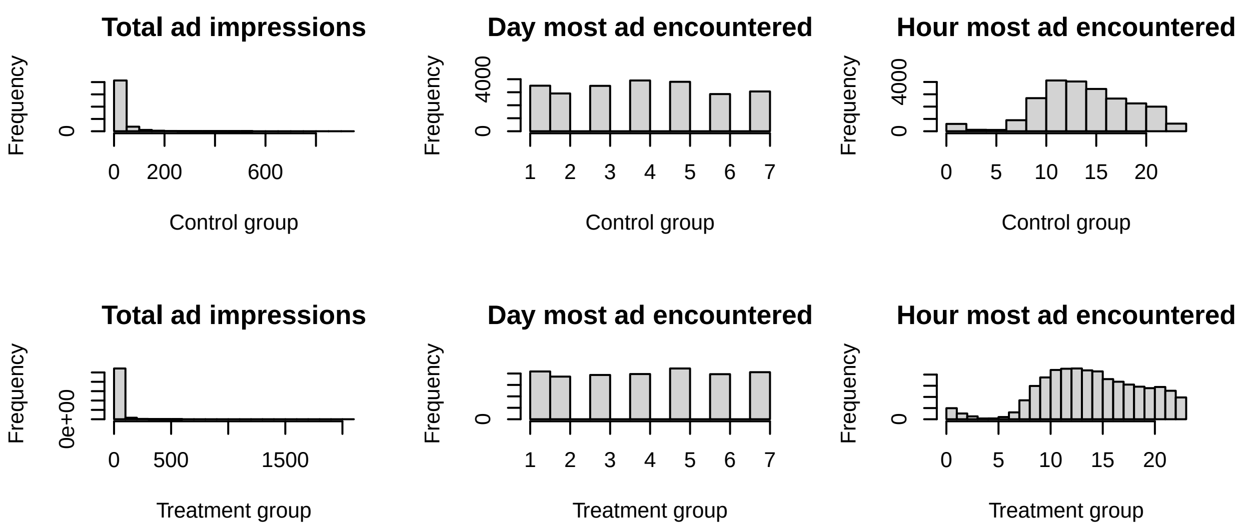
Histogram

```r
attach(rocketfuel)
par(mfrow=c(3,3)) # output multiple subfigures into one figure, with 2 subfigures each row and 3 rows (

hist(tot_impr[test==0], main = paste("Total ad impressions"), xlab = "Control group")
#plot the histogram of numdoctors for control group
hist(mode_impr_day[test==0], main = paste("Day most ad encountered"), xlab = "Control group")#plot the
hist(mode_impr_hour[test==0], main = paste("Hour most ad encountered"), xlab = "Control group")

hist(tot_impr[test==1], main = paste("Total ad impressions"), xlab = "Treatment group")
#plot the histogram of numdoctors for control group
hist(mode_impr_day[test==1], main = paste("Day most ad encountered"), xlab = "Treatment group")#plot th
hist(mode_impr_hour[test==1], main = paste("Hour most ad encountered"), xlab = "Treatment group")

detach(rocketfuel)
```



Even though the frequencies varies for both the plots we see that there is not much of a difference in the histograms.

Summary Table

Here we will look at the CI and mean of the outcome variables

```r
attach(rocketfuel)
# Create a summary table
summary <- rocketfuel %>%  #create a table called summary that will hold the info that starts with the
```

```
  mutate(test = as.factor(test)) %>%        #and then tell R that treatment is a factor variable taking
  group_by(test) %>%        # and then create groups by treatment
  summarise(n = length(user_id),            # create a new table with summary measures
            mean.converted = round(mean(converted),2),          # get the mean for each group (the ,2 is
            error.converted = round(sd(converted)/sqrt(n),3),  # calculate the standard error on the me
            LCI.converted = round(mean.converted - 1.96*error.converted,3),    # calulate confidence in
            UCI.converted = round(mean.converted + 1.96*error.converted,3))

kable(summary) # this code outputs the table in a readable format
```

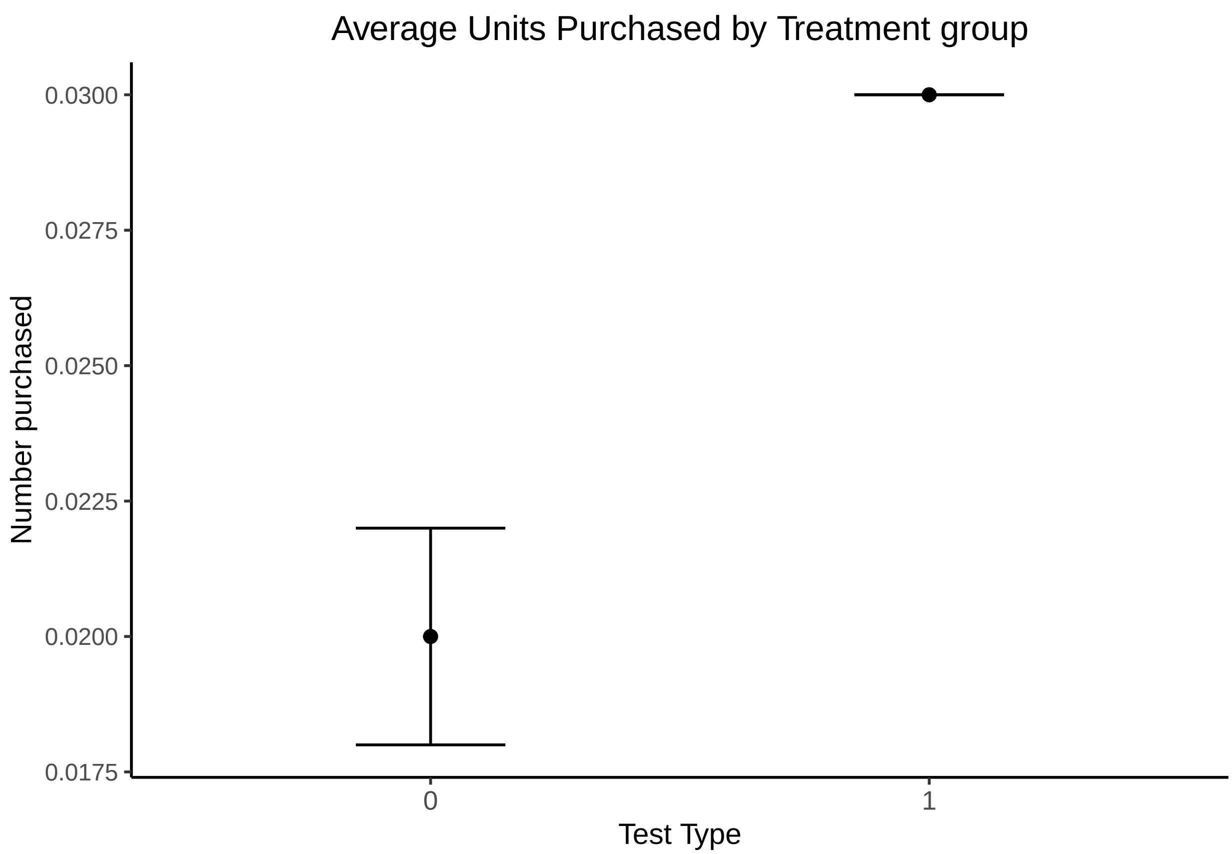
test	n	mean.converted	error.converted	LCI.converted	UCI.converted
0	23524	0.02	0.001	0.018	0.022
1	564577	0.03	0.000	0.030	0.030

```
detach(rocketfuel)
```

Here we see that the mean converted of treatment is higher than control and also the CI difference for control group is higher compared to the treatment and that is because the control sample was only 4% as mentioned in the case.

For better understanding we can see the plot below in order to understand the difference of CI between both the groups.

```
summary %>%
  ggplot(aes(x=test)) +
  geom_point(aes(y = mean.converted), size = 2) +
  scale_shape_manual(values=c(15, 16)) +
  ggtitle("Average Units Purchased by Treatment group") +
  ylab("Number purchased") + xlab("Test Type") +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(),axis.line = element_line(colour = "black"),
        axis.text.x= element_text(size = 10), legend.position=c(.5,.5),
        plot.title=element_text(hjust=.5))+
  geom_errorbar(aes(ymin = LCI.converted,
                    ymax = UCI.converted), width = .3)+
  scale_color_manual(values=c("darkgrey","black"))
```



Average Treatment effect

```
attach(rocketfuel)
ATE <-matrix(NA, nrow = 1, ncol = 4)  # create an empty matrix to store the results
colnames(ATE) <- c("Control Mean","Treatment ATE", "Treatment LCI", "Treatment UCI")  # name the column
rownames(ATE) <- c("Conversions") # name the rows
mean.control <- t(summary[1,3])  # call the means from summary table
mean.treat1 <- t(summary[2,3])
ATE[,1] <- round(mean.control,4)
ATE[,2] <- effect.treat1 <- round(mean.treat1-mean.control,4)  # calculate ATE for treatment1
#now calculate sd to construct CI for each outcome variables
#first, we make the s.d. of outcomes as a vector in each treatment condition
sd.control <- t(summary[1,4])
sd.treat1 <- t(summary[2,4])

#then construct the s.d. for computing CI based on the s.d. vector we just created
error.treat1 <- sqrt(sd.control^2+sd.treat1^2)

#computing CI
ATE[,3]<-LCI.treat1 <- round(effect.treat1 -1.96*error.treat1,4)
ATE[,4]<-UCI.treat1 <- round(effect.treat1 +1.96*error.treat1,4)

kable(ATE)
```

	Control Mean	Treatment ATE	Treatment LCI	Treatment UCI
Conversions	0.02	0.01	0.008	0.012

```
detach(rocketfuel)
```

ATE (regression approach)

```
#Start by creating two "dummy variables" in our dataframe to indicate the two treatments
rocketfuel$treat <- as.numeric(rocketfuel$test == 1)

# We need to estimate standard errors that allow for heteroskedasticy (i.e., different standard errors
library("lmtest")
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library("sandwich")

# Let's do the regression on pageviews first
fit.converted <- lm(converted~treat + tot_impr + mode_impr_day + mode_impr_hour , data = rocketfuel)  #
# Now we report the point estimates and standard errors for each parameter by coeftest()
coeftest(fit.converted, vcov = vcovHC(fit.converted, type = "HC3"))
```

```
##
## t test of coefficients:
##
##                  Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)    -5.1015e-03  1.1085e-03  -4.6022 4.181e-06 ***
## treat           7.6495e-03  8.7490e-04   8.7432 < 2.2e-16 ***
## tot_impr        7.8119e-04  1.2867e-05  60.7104 < 2.2e-16 ***
## mode_impr_day  -1.7020e-03  1.0356e-04 -16.4352 < 2.2e-16 ***
## mode_impr_hour  7.2282e-04  3.8514e-05  18.7679 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefci(fit.converted, vcov = vcovHC(fit.converted)) #get the according CIs by coefci()
```

```
##                       2.5 %        97.5 %
## (Intercept)    -0.0072741007 -0.0029288894
## treat           0.0059347034  0.0093642628
## tot_impr        0.0007559702  0.0008064100
## mode_impr_day  -0.0019049959 -0.0014990495
## mode_impr_hour  0.0006473380  0.0007983093
```

Summary Table of over 10 deciles of total impressions

```
summary_list <- data.frame(matrix(ncol = 9, nrow = 0))
for (x in 1:10) {
impr_data <- rocketfuel[ which(rocketfuel$tot_impr_decile == x),]
summary_chain = impr_data %>%
  mutate(test = as.factor(test)) %>%
  group_by(test) %>%
  summarise(n = length(user_id),
            mean.converted = round(mean(converted),2),
            mean.tot_impr = round(mean(tot_impr),2),
            mean.mode_impr_day = round(mean(mode_impr_day),2),
            mean.mode_impr_hour = round(mean(mode_impr_hour),2),
            sd.converted =  round(sd(converted),2),
            sd.tot_impr =  round(sd(tot_impr),2),
            sd.mode_impr_day =  round(sd(mode_impr_day),2),
            sd.mode_impr_hour =  round(sd(mode_impr_hour),2)
            )
summary_chain$tot_impr_decile <- c(x,x)
summary_chain <- t(summary_chain)
print(kable(summary_chain))
summary_list <- append(summary_list,summary_chain)
}
```

```
##
##
## |                   |      |      |
## |:------------------|:-----|:-----|
## |test               |0     |1     |
## |n                  |2308  |54298 |
## |mean.converted     |0     |0     |
## |mean.tot_impr      |1     |1     |
## |mean.mode_impr_day |3.81  |4.09  |
## |mean.mode_impr_hour |14.23 |14.42 |
## |sd.converted       |0.04  |0.04  |
## |sd.tot_impr        |0     |0     |
## |sd.mode_impr_day   |1.92  |1.96  |
## |sd.mode_impr_hour  |4.68  |5.13  |
## |tot_impr_decile    |1     |1     |
##
##
## |                   |      |      |
## |:------------------|:-----|:-----|
## |test               |0     |1     |
## |n                  |3249  |65239 |
## |mean.converted     |0     |0     |
## |mean.tot_impr      |2.41  |2.42  |
## |mean.mode_impr_day |3.75  |3.92  |
## |mean.mode_impr_hour |14.23 |14.42 |
## |sd.converted       |0.05  |0.05  |
## |sd.tot_impr        |0.49  |0.49  |
## |sd.mode_impr_day   |1.99  |2.01  |
## |sd.mode_impr_hour  |4.47  |4.90  |
```

```
## |tot_impr_decile     |2     |2     |
##
##
## |                    |      |      |
## |:-------------------|:-----|:-----|
## |test                |0     |1     |
## |n                   |2304  |50425 |
## |mean.converted      |0.01  |0.00  |
## |mean.tot_impr       |4.56  |4.56  |
## |mean.mode_impr_day  |3.84  |3.95  |
## |mean.mode_impr_hour |14.46 |14.55 |
## |sd.converted        |0.07  |0.06  |
## |sd.tot_impr         |0.5   |0.5   |
## |sd.mode_impr_day    |1.94  |1.99  |
## |sd.mode_impr_hour   |4.56  |4.79  |
## |tot_impr_decile     |3     |3     |
##
##
## |                    |      |      |
## |:-------------------|:-----|:-----|
## |test                |0     |1     |
## |n                   |2490  |56051 |
## |mean.converted      |0.01  |0.00  |
## |mean.tot_impr       |6.85  |6.88  |
## |mean.mode_impr_day  |3.98  |3.96  |
## |mean.mode_impr_hour |14.45 |14.60 |
## |sd.converted        |0.07  |0.06  |
## |sd.tot_impr         |0.81  |0.81  |
## |sd.mode_impr_day    |1.93  |1.99  |
## |sd.mode_impr_hour   |4.60  |4.74  |
## |tot_impr_decile     |4     |4     |
##
##
## |                    |      |      |
## |:-------------------|:-----|:-----|
## |test                |0     |1     |
## |n                   |2250  |60882 |
## |mean.converted      |0.01  |0.01  |
## |mean.tot_impr       |10.95 |11.05 |
## |mean.mode_impr_day  |3.92  |3.97  |
## |mean.mode_impr_hour |14.45 |14.54 |
## |sd.converted        |0.08  |0.08  |
## |sd.tot_impr         |1.42  |1.41  |
## |sd.mode_impr_day    |1.93  |2.01  |
## |sd.mode_impr_hour   |4.72  |4.77  |
## |tot_impr_decile     |5     |5     |
##
##
## |                    |      |      |
## |:-------------------|:-----|:-----|
## |test                |0     |1     |
## |n                   |1734  |56368 |
## |mean.converted      |0.01  |0.01  |
## |mean.tot_impr       |15.60 |15.54 |
```

```
## |mean.mode_impr_day  |4.10  |4.07  |
## |mean.mode_impr_hour |14.39 |14.50 |
## |sd.converted        |0.09  |0.09  |
## |sd.tot_impr         |1.10  |1.06  |
## |sd.mode_impr_day    |1.95  |2.02  |
## |sd.mode_impr_hour   |4.75  |4.74  |
## |tot_impr_decile     |6     |6     |
##
##
## |                    |      |      |
## |:-------------------|:-----|:-----|
## |test                |0     |1     |
## |n                   |2662  |61873 |
## |mean.converted      |0.01  |0.01  |
## |mean.tot_impr       |21.23 |20.83 |
## |mean.mode_impr_day  |4.01  |4.09  |
## |mean.mode_impr_hour |14.25 |14.45 |
## |sd.converted        |0.12  |0.11  |
## |sd.tot_impr         |2.19  |2.10  |
## |sd.mode_impr_day    |1.98  |2.02  |
## |sd.mode_impr_hour   |4.60  |4.79  |
## |tot_impr_decile     |7     |7     |
##
##
## |                    |      |      |
## |:-------------------|:-----|:-----|
## |test                |0     |1     |
## |n                   |1868  |47226 |
## |mean.converted      |0.02  |0.02  |
## |mean.tot_impr       |28.24 |28.59 |
## |mean.mode_impr_day  |4.02  |4.12  |
## |mean.mode_impr_hour |14.17 |14.37 |
## |sd.converted        |0.14  |0.15  |
## |sd.tot_impr         |2.69  |2.64  |
## |sd.mode_impr_day    |1.94  |2.02  |
## |sd.mode_impr_hour   |4.87  |4.84  |
## |tot_impr_decile     |8     |8     |
##
##
## |                    |      |      |
## |:-------------------|:-----|:-----|
## |test                |0     |1     |
## |n                   |2234  |57194 |
## |mean.converted      |0.03  |0.05  |
## |mean.tot_impr       |43.52 |43.38 |
## |mean.mode_impr_day  |4.06  |4.10  |
## |mean.mode_impr_hour |14.26 |14.47 |
## |sd.converted        |0.18  |0.22  |
## |sd.tot_impr         |6.77  |6.83  |
## |sd.mode_impr_day    |1.91  |2.02  |
## |sd.mode_impr_hour   |4.73  |4.85  |
## |tot_impr_decile     |9     |9     |
##
##
```

```
## |                    |       |       |
## |:-------------------|:------|:------|
## |test                |0      |1      |
## |n                   |2425   |55021  |
## |mean.converted      |0.08   |0.15   |
## |mean.tot_impr       |118.19 |118.48 |
## |mean.mode_impr_day  |4.16   |4.02   |
## |mean.mode_impr_hour |14.19  |14.42  |
## |sd.converted        |0.28   |0.35   |
## |sd.tot_impr         |80.80  |90.88  |
## |sd.mode_impr_day    |1.94   |2.01   |
## |sd.mode_impr_hour   |4.73   |4.85   |
## |tot_impr_decile     |10     |10     |
```

Here we can see the summary of the mean and the standard deviation of variables in the data set for both treatment and control group over the 10 deciles of total impressions.

Creating a table of count and proportion of treatment and control over the 10 deciles of total impressions.

```
attach(rocketfuel)
tb_treatment_sub <- matrix(NA, nrow = 20, ncol = 2)
tb_treatment_sub[c(1,3,5,7,9,11,13,15,17,19),] <- format(t(table(test,tot_impr_decile)), digits = 1)

tb_treatment_sub[c(2,4,6,8,10,12,14,16,18,20),] <- format(t(prop.table(table(test,tot_impr_decile))),di

rownames(tb_treatment_sub) <- c("Frequency in Group 1", "Proportion in Group 1" ,"Frequency in Group 2"
colnames(tb_treatment_sub) <- c("Control Group",  "Treatment Group")  # name the columns
kable(tb_treatment_sub)
```

	Control Group	Treatment Group
Frequency in Group 1	2308	54298
Proportion in Group 1	0.00392	0.09233
Frequency in Group 2	3249	65239
Proportion in Group 2	0.00552	0.11093
Frequency in Group 3	2304	50425
Proportion in Group 3	0.00392	0.08574
Frequency in Group 4	2490	56051
Proportion in Group 4	0.00423	0.09531
Frequency in Group 5	2250	60882
Proportion in Group 5	0.00383	0.10352
Frequency in Group 6	1734	56368
Proportion in Group 6	0.00295	0.09585
Frequency in Group 7	2662	61873
Proportion in Group 7	0.00453	0.10521
Frequency in Group 8	1868	47226
Proportion in Group 8	0.00318	0.08030
Frequency in Group 9	2234	57194
Proportion in Group 9	0.00380	0.09725
Frequency in Group 10	2425	55021
Proportion in Group 10	0.00412	0.09356

```
detach(rocketfuel)
```

Here we see the mean and 95% CI on "converted" separately for treatment and control over the 10 deciles of total impressions.

```
summary2 = rocketfuel %>%  #create a new data frame that I am calling summary2 here from our casedata
  mutate(test = as.factor(test)) %>%  #denote treatment is a factor variable
  mutate(deciles = as.factor(tot_impr_decile)) %>%
  group_by(test,deciles) %>%
  summarise(n = length(user_id),
            m.converted = mean(converted),         # get the mean for each group
            e.converted = sd(converted)/sqrt(n),
            Lci.converted = m.converted - 1.96*e.converted,    # calculate confidence interval boundari
            Uci.converted = m.converted + 1.96*e.converted,)
```

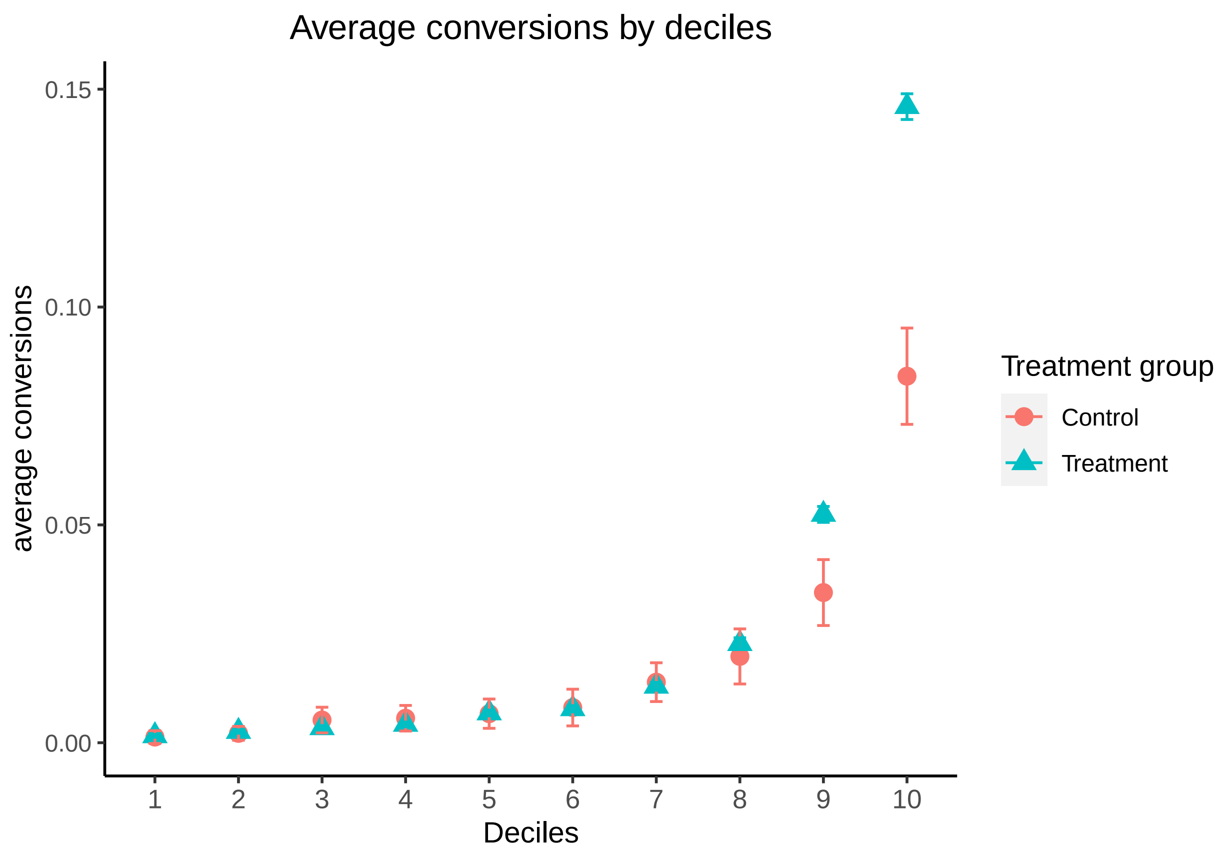
```
## 'summarise()' has grouped output by 'test'. You can override using the
## '.groups' argument.
```

```
summary2
```

```
## # A tibble: 20 x 7
## # Groups:   test [2]
##    test  deciles     n m.converted e.converted Lci.converted Uci.converted
##    <fct> <fct>   <int>       <dbl>       <dbl>         <dbl>         <dbl>
##  1 0     1        2308     0.00130    0.000750     -0.000170       0.00277
##  2 0     2        3249     0.00215    0.000814      0.000560       0.00375
##  3 0     3        2304     0.00521    0.00150       0.00227        0.00815
##  4 0     4        2490     0.00562    0.00150       0.00268        0.00856
##  5 0     5        2250     0.00667    0.00172       0.00330        0.0100
##  6 0     6        1734     0.00807    0.00215       0.00386        0.0123
##  7 0     7        2662     0.0139     0.00227       0.00945        0.0183
##  8 0     8        1868     0.0198     0.00322       0.0135         0.0261
##  9 0     9        2234     0.0345     0.00386       0.0269         0.0420
## 10 0     10       2425     0.0841     0.00564       0.0731         0.0952
## 11 1     1       54298     0.00158    0.000171      0.00125        0.00192
## 12 1     2       65239     0.00258    0.000198      0.00219        0.00296
## 13 1     3       50425     0.00343    0.000260      0.00292        0.00394
## 14 1     4       56051     0.00421    0.000274      0.00367        0.00475
## 15 1     5       60882     0.00683    0.000334      0.00618        0.00749
## 16 1     6       56368     0.00779    0.000370      0.00706        0.00851
## 17 1     7       61873     0.0129     0.000454      0.0121         0.0138
## 18 1     8       47226     0.0227     0.000686      0.0214         0.0241
## 19 1     9       57194     0.0524     0.000932      0.0506         0.0542
## 20 1     10      55021     0.146      0.00151       0.143          0.149
```

```
summary2 %>%
  ggplot(aes(deciles)) +
  geom_point(aes(y = m.converted, shape = test, color = test), size = 3) +  #plot the averages and give
  geom_errorbar(aes(ymin = Lci.converted,
                    ymax = Uci.converted, color=test), width = .15)+  #Give it error bars with the same
  ggtitle("Average conversions by deciles") +
  ylab("average conversions") + xlab("Deciles") +
```

```r
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(),axis.line = element_line(colour = "black"),
        axis.text.x= element_text(size = 10),
        plot.title=element_text(hjust=.5) ) +
#Working on Legends
  scale_shape_discrete(name = "Treatment group", labels = c("Control", "Treatment"))+
  scale_color_discrete(name = "Treatment group", labels = c("Control", "Treatment"))
```



Conculsion

After looking at the graph and CI of the 10 deciles we see that 9th and 10th decile treatment has the best conversions than the others. Also after looking at the difference between the control and treatment group of 9 & 10th decile, we can conclude that ads shown has a better conversion rate.