RocketFuel Case Study

Ming Chen

10/30/2021

1. Load the data and take a first look

Load the libraries we want and the RocketFuel Case Data

```
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(jtools)
library(rcompanion)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
casedata <- read.csv("rocketfuel_deciles.csv")
```

2. Divide subgroups

For the analysis we are going to break down results by sub-groups depending on the number of times the individual was targeted for ads (tot_impr). The variable called 'rocketfuel_deciles' has been created for this purpose. This simply takes the original rocketfuel data and adds a column labeling people into deciles (i.e., 10% groups) by tot_impr.

3. Summarize the data

```
summary(casedata)
```

```
##     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
```

There are 7 variables in the data. 'user_id' is just a unique identifier of the user. 'test' is used to identify whether the user is in treatment group. converted', 'tot_impr', 'mode_impr_day', 'mode_impr_hour' is our outcome metrics. 'tot_impr_decile' is generated from breaking down 'tot_impr' variable results by 10 sub-groups depending on the number of times the individual was targeted for ads.

4. Check for the numbers and shares of individuals who were in the treatment vs. control group

```
attach(casedata)

tb_treatment_full <- matrix(NA, nrow = 2, ncol = 2)
tb_treatment_full[1,] <- format(table(test), digits = 0)
tb_treatment_full[2,] <- format(prop.table(table(test)),digits = 3)

rownames(tb_treatment_full) <- c("Frequency", "Proportion" )
colnames(tb_treatment_full) <- c("Control",  "Exposure to Ads")
kable(tb_treatment_full)
```

	Control	Exposure to Ads
Frequency	23524	564577
Proportion	0.04	0.96

```
detach(casedata)
```

As we can see, there are only 4% of users are assigned to control group. This is because TaskaBella wanted to avoid having too large of a control group to save money on buying PSA ads, which might lead to many problems related to the effectiveness of the experiment.We'll see what's going to happen later on.

5. Check for balance on pre-treatment variables and graph histogram

Now we want to check whether the pre-treatment variables are balanced across treatment and control groups. For a successful experiment we are hoping to have these look similar between the groups, since they can't be affected by the treatment.The following code will create a table where there is a row for each of these variables. The columns will be the averages in each of the treatment groups.

```
attach(casedata)
preexp <- casedata %>%
  dplyr::select(tot_impr,mode_impr_day, mode_impr_hour)
# Summarize the means of those variables by treatment
```

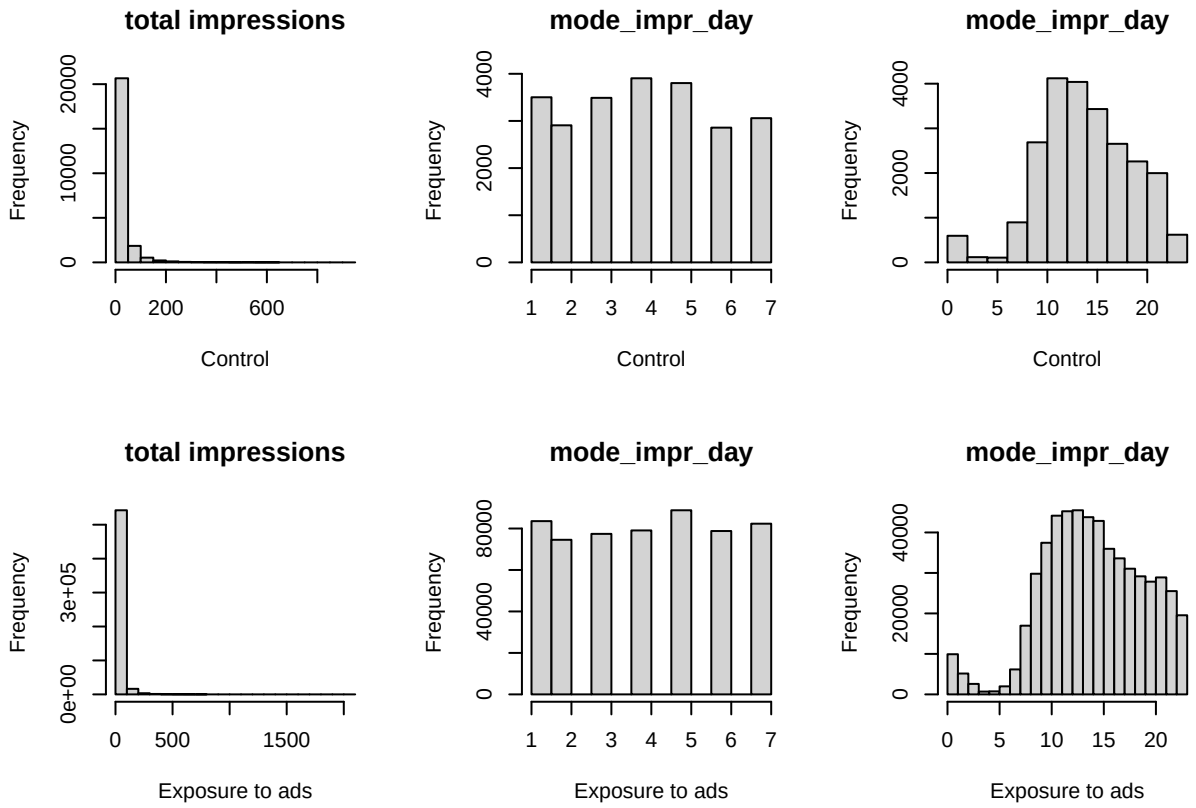
```
tb_preexp <- matrix(NA, nrow = 3, ncol = 2)
colnames(tb_preexp) <- c( "Mean Control", "Mean Exposure to ads")
rownames(tb_preexp) <- colnames(preexp)
m<-as.matrix(round(aggregate(.~test,preexp,mean),2))
tb_preexp[,1:2] <-t(m)[2:4,]
kable(tb_preexp)
```

	Mean Control	Mean Exposure to ads
tot_impr	24.76	24.82
mode_impr_day	3.95	4.03
mode_impr_hour	14.30	14.48

```
detach(casedata)
```

We can see that everything looks very balanced here. The averages for each of these three variables look really similar across each of the groups. This is exactly what we would expect. But sometimes averages can hide something important (i.e. different distributions). So let's be thorough here and plot the histogram for our three continuous pre-experiment variables.

```
attach(casedata)
par(mfrow=c(2,3))
#plot the histogram of variables for control group
hist(tot_impr[test==0], main = paste("total impressions"), xlab = "Control")
hist(mode_impr_day[test==0], main = paste("mode_impr_day"), xlab = "Control")
hist(mode_impr_hour[test==0], main = paste("mode_impr_day"), xlab = "Control")
#plot the histogram of variables for "Exposure to ads" group
hist(tot_impr[test==1], main = paste("total impressions"), xlab = "Exposure to ads")
hist(mode_impr_day[test==1], main = paste("mode_impr_day"), xlab = "Exposure to ads")
hist(mode_impr_hour[test==1], main = paste("mode_impr_day"), xlab = "Exposure to ads")
```




```
detach(casedata)
```

From the distribution of each variable of each groups, we can see that the frequency distributions are well balanced across 2 different groups, which is exactly what we expected for randomized experiment.

6. Plot the means and CI of the main outcome "converted" by control and treatment.

```
attach(casedata)
summary <- casedata %>%
  mutate(test = as.factor(test)) %>%
  group_by(test) %>%
  summarise(n = length(user_id),
            avgconverted = round(mean(converted),4),
             error = round(sd(converted)/sqrt(n),3),
             lowerCI = round(avgconverted - 1.96*error,3),
             upperCI = round(avgconverted + 1.96*error,3))
kable(summary)
```

test	n	avgconverted	error	lowerCI	upperCI
0	23524	0.0179	0.001	0.016	0.020
1	564577	0.0255	0.000	0.025	0.025

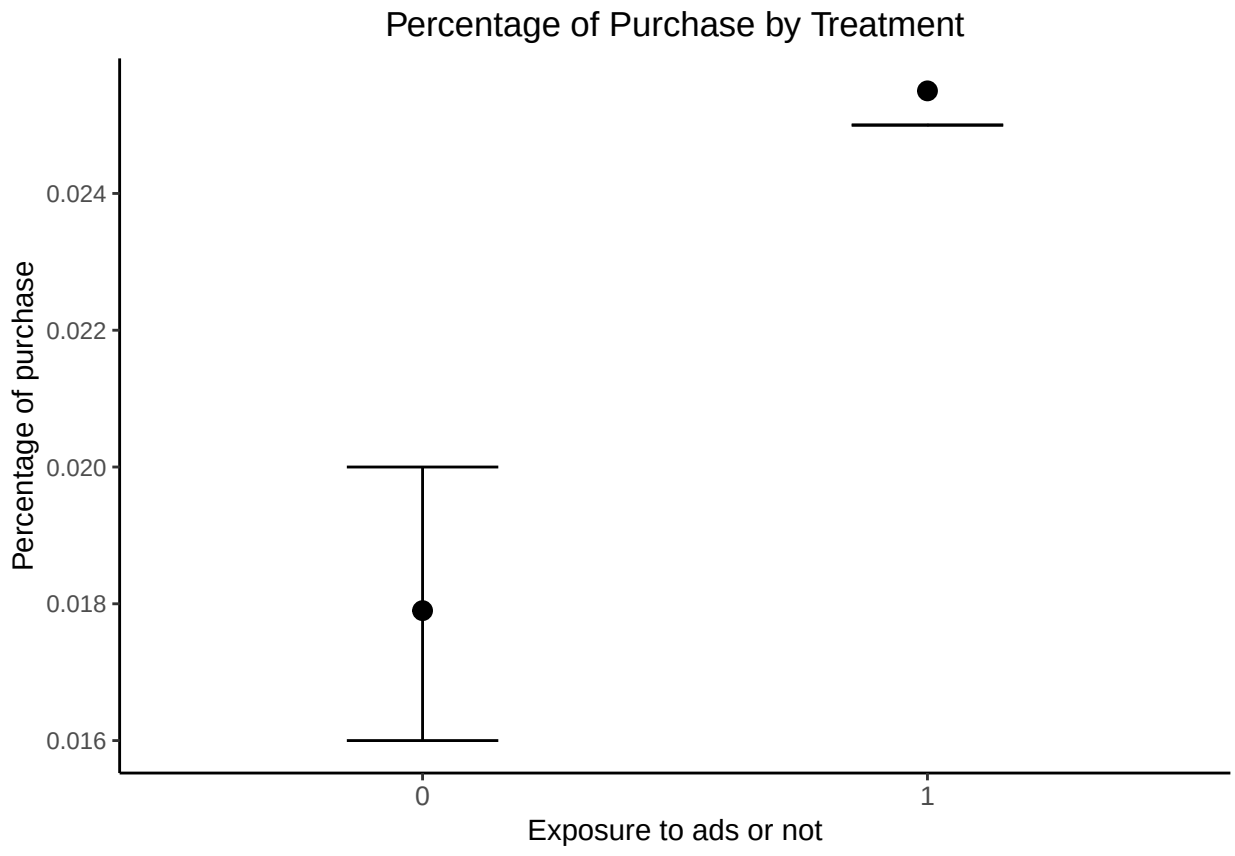
```
detach(casedata)
```

As we can see, there are 2% of user in control group purchased handbags, with standard deviation of 0.001

and the confidence interval is exactly the same as the mean. For users in treatment group, the conversion rate is 1% higher than in the control group. Also, 0 standard deviation and the tight confidence interval suggest that we are pretty confident to say that there are 3% users who are exposed to ads will buy Taskabella's handbag.

For better understanding of the results, let's produce a visual of the results by creating a graph of the means with their confidence intervals.

```
# Plot the information from that summary table
summary %>%
  ggplot(aes(x=test)) +
  geom_point(aes(y = avgconverted), size = 3) +
  scale_shape_manual(values=c(15, 16)) +
  ggtitle("Percentage of Purchase by Treatment") +
  ylab("Percentage of purchase") + xlab("Exposure to ads or not") +
  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 = lowerCI,
                    ymax = upperCI), width = .3)+
  scale_color_manual(values=c("darkgrey","black"))
```



7. Calculate estimate of ATE of the ads

We start by creating a summary table that has the means and confidence intervals for the outcome for all of the treatments. Importing the data and computing the group-wise mean, variance, and confidence intervals

```
attach(casedata)
# Create a summary table
summary = casedata %>%
  mutate(test = as.factor(test)) %>%
  group_by(test) %>%
  summarise(n = length(user_id),
            mean.tot_impr = round(mean(tot_impr),3),
            mean.mode_impr_day = round(mean(mode_impr_day),3),
            mean.mode_impr_hour = round(mean(mode_impr_hour),3),
            mean.converted = round(mean(converted),3),
            error.tot_impr = round(sd(tot_impr)/sqrt(n),3),
            error.mode_impr_day = round(sd(mode_impr_day)/sqrt(n),3),
            error.mode_impr_hour = round(sd(mode_impr_hour)/sqrt(n),3),
            error.converted = round(sd(converted)/sqrt(n),3),
            LCI.tot_impr = round(mean.tot_impr - 1.96*error.tot_impr,3),
            LCI.mode_impr_day = round(mean.mode_impr_day - 1.96*error.mode_impr_day,3),
            LCI.mode_impr_hour = round(mean.mode_impr_hour - 1.96*error.mode_impr_hour,3),
            LCI.converted = round(mean.converted - 1.96*error.converted,3),
            UCI.tot_impr = round(mean.tot_impr + 1.96*error.tot_impr,3),
            UCI.mode_impr_day = round(mean.mode_impr_day + 1.96*error.mode_impr_day,3),
            UCI.mode_impr_hour = round(mean.mode_impr_hour + 1.96*error.mode_impr_hour,3),
            UCI.converted = round(mean.converted + 1.96*error.converted,3))
kable(summary)
```

test	n	mean.tot_impr	mean.mode_impr_day	mean.mode_impr_hour	mean.converted	error.tot_impr	error.mode_impr_day	error.mode_impr_hour	error.converted	LCI.tot_impr	LCI.mode_impr_day	LCI.mode_impr_hour	LCI.converted	UCI.tot_impr	UCI.mode_impr_day	UCI.mode_impr_hour	UCI.converted
0	23524	24.761	3.953	14.305	0.018	0.279	0.013	0.030	0.001	24.214	3.928	14.246	0.016	25.308	3.978	14.364	0.020
1	564	24.823	4.029	14.476	0.026	0.058	0.003	0.006	0.000	24.709	4.023	14.464	0.026	24.937	4.035	14.488	0.026

```
detach(casedata)
```

This is pretty hard to read. Now let's calculate our table of average treatment effects with CIs

```
attach(casedata)
ATE <-matrix(NA, nrow = 4, ncol = 4)
colnames(ATE) <- c("Control Mean", "Exposure to ads ATE", "LCI Exposure to ads", "UCI Exposure to ads")
rownames(ATE) <- c("tot_impr", "mode_impr_day", "mode_impr_hour","converted")

mean.control <- t(summary[1,3:6])
mean.treat1 <- t(summary[2,3:6])
ATE[,1] <- mean.control
ATE[,2] <- effect.treat1 <- mean.treat1-mean.control
sd.control <- t(summary[1,7:10])
sd.treat1 <- t(summary[2,7:10])
error.treat1 <- sqrt(sd.control^2+sd.treat1^2)
ATE[,3]<-LCI.treat1 <- round(effect.treat1 -1.96*error.treat1,3)
ATE[,4]<-UCI.treat1 <- round(effect.treat1 +1.96*error.treat1,3)

kable(ATE)
```

	Control Mean	Exposure to ads ATE	LCI Exposure to ads	UCI Exposure to ads
tot_impr	24.761	0.062	-0.497	0.621
mode_impr_day	3.953	0.076	0.050	0.102

	Control Mean	Exposure to ads ATE	LCI Exposure to ads	UCI Exposure to ads
mode_impr_hour	14.305	0.171	0.111	0.231
converted	0.018	0.008	0.006	0.010

```
detach(casedata)
```

As we can see from the ATE summary table, the total number of impression increased by 0.06 with confidence interval including zero, which indicates that we are quite uncertain about whether exposure to ads can increase the number of impression by 0.06. However, we are pretty sure that the average treatment effect on conversion rate, which has a pretty tight confidence interval.The largest ATE here occurred to 'mode_impr_hour' variable, which has an ATE of 0.18. It means the hour of the day in which user encountered the ads increased about 15 minutes, which doesn't seem to have actual meaning in this case.

8.Calculate the ATE again using the regression approach

```
#Start by creating a "dummy variable" in our dataframe to indicate the treatment
casedata$treat1 <- as.numeric(casedata$test == 1)

# We need to estimate standard errors that allow for heteroskedasticty
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 tot_impr first
fit.tot_impr <- lm(tot_impr~treat1, data = casedata)  #Simple linear regression
coeftest(fit.tot_impr, vcov = vcovHC(fit.tot_impr, type = "HC3"))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.761138   0.279456  88.605   <2e-16 ***
## treat1       0.062228   0.285457   0.218   0.8274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##                  2.5 %     97.5 %
## (Intercept) 24.2134132 25.3088620
## treat1      -0.4972596  0.6217147
```

```
fit.mode_impr_day <- lm(mode_impr_day~treat1, data = casedata)  #Simple linear regression
coeftest(fit.mode_impr_day, vcov = vcovHC(fit.mode_impr_day)) #point estimates
```

```
##
## t test of coefficients:
```

```
##
##             Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) 3.952644   0.012707 311.0534 < 2.2e-16 ***
## treat1      0.075926   0.012985   5.8473 4.999e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefci(fit.mode_impr_day, vcov = vcovHC(fit.mode_impr_day)) #confidence intervals
```

```
##                  2.5 %    97.5 %
## (Intercept) 3.92773824 3.9775500
## treat1      0.05047622 0.1013757
```

```
fit.mode_impr_hour <- lm(mode_impr_hour~treat1, data = casedata)  #Simple linear regression
coeftest(fit.mode_impr_hour, vcov = vcovHC(fit.mode_impr_hour))  #point estimates
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) 14.304923   0.030359 471.1906 < 2.2e-16 ***
## treat1       0.170977   0.031035   5.5091 3.608e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefci(fit.mode_impr_hour, vcov = vcovHC(fit.mode_impr_hour))   #confidence intervals
```

```
##                  2.5 %     97.5 %
## (Intercept) 14.2454198 14.3644255
## treat1       0.1101486  0.2318055
```

These results match our ATE table above. Specifically, the p-value for tot_impr is 0.83, which is a lot higher than 0.05, indicating the uncertainty of the estimated mean.

9. Create a summary table over the 10 deciles of total impressions

We're going to create a summary table showing the sample size, 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.

```
attach(casedata)
summary_subgroups = casedata %>%
  mutate(test = as.factor(test)) %>%
  mutate(tot_impr_decile = as.factor(tot_impr_decile)) %>%
  group_by(test, tot_impr_decile) %>%
  summarise(n=length(user_id),
            mean.converted =round(mean(converted),4),
            mean.tot_impr =round(mean(tot_impr),4),
            mean.mode_impr_day =round(mean(mode_impr_day),4),
            mean.mode_impr_hour =round(mean(mode_impr_hour),4),
            error.converted = round(sd(converted)/sqrt(n),4),
            error.tot_impr = round(sd(tot_impr)/sqrt(n),4),
            error.mode_impr_day = round(sd(mode_impr_day)/sqrt(n),4),
            error.mode_impr_hour = round(sd(mode_impr_hour)/sqrt(n),4),
            LCI.converted = round(mean.converted - 1.96*error.converted,2),
            LCI.tot_impr = round(mean.tot_impr - 1.96*error.tot_impr,2),
            LCI.mode_impr_day = round(mean.mode_impr_day - 1.96*error.mode_impr_day,2),
            LCI.mode_impr_hour = round(mean.mode_impr_hour - 1.96*error.mode_impr_hour,2),
```

```
            UCI.converted = round(mean.converted + 1.96*error.converted,2),
            UCI.tot_impr = round(mean.tot_impr + 1.96*error.tot_impr,2),
            UCI.mode_impr_day = round(mean.mode_impr_day + 1.96*error.mode_impr_day,2),
            UCI.mode_impr_hour = round(mean.mode_impr_hour + 1.96*error.mode_impr_hour,2)
            )
```

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

```
summary_subgroups
```

```
## # A tibble: 20 x 19
## # Groups:   test [2]
##    test  tot_impr_decile     n mean.converted mean.tot_impr mean.mode_impr_day
##    <fct> <fct>           <int>          <dbl>         <dbl>              <dbl>
##  1 0     1                2308         0.0013          1                  3.81
##  2 0     2                3249         0.0022          2.41               3.75
##  3 0     3                2304         0.0052          4.56               3.84
##  4 0     4                2490         0.0056          6.85               3.98
##  5 0     5                2250         0.0067         11.0                3.92
##  6 0     6                1734         0.0081         15.6                4.10
##  7 0     7                2662         0.0139         21.2                4.01
##  8 0     8                1868         0.0198         28.2                4.02
##  9 0     9                2234         0.0345         43.5                4.06
## 10 0     10               2425         0.0841        118.                 4.16
## 11 1     1               54298         0.0016          1                  4.09
## 12 1     2               65239         0.0026          2.42               3.92
## 13 1     3               50425         0.0034          4.56               3.95
## 14 1     4               56051         0.0042          6.88               3.96
## 15 1     5               60882         0.0068         11.0                3.97
## 16 1     6               56368         0.0078         15.5                4.07
## 17 1     7               61873         0.0129         20.8                4.09
## 18 1     8               47226         0.0227         28.6                4.12
## 19 1     9               57194         0.0524         43.4                4.10
## 20 1     10              55021         0.146         118.                 4.02
## # ... with 13 more variables: mean.mode_impr_hour <dbl>, error.converted <dbl>,
## #   error.tot_impr <dbl>, error.mode_impr_day <dbl>,
## #   error.mode_impr_hour <dbl>, LCI.converted <dbl>, LCI.tot_impr <dbl>,
## #   LCI.mode_impr_day <dbl>, LCI.mode_impr_hour <dbl>, UCI.converted <dbl>,
## #   UCI.tot_impr <dbl>, UCI.mode_impr_day <dbl>, UCI.mode_impr_hour <dbl>
```

By dividing users into 10 subgroups by the total number of impression, we can clearly see the difference in the mean/CI of four variables between these 10 subgroups. For example, the 10th percentile has a large number of total impression relative to other groups, so the conversion rate is a lot higher than other subgroups.In addition, from the mean of total impression, we can infer the distribution of it for the whole sample: the higher the percentile in total impression, the more volatile the total number impression is for different users within the subgroup, thus result in relatively higher confidence interval, meaning we are more uncertain about the estimates of the mean. Also, we should note that the proportion of users are not well-balanced between these subgroups. We can see that group 6, 8 has relative small sample size, which could lead to relatively wider range of confidence interval.

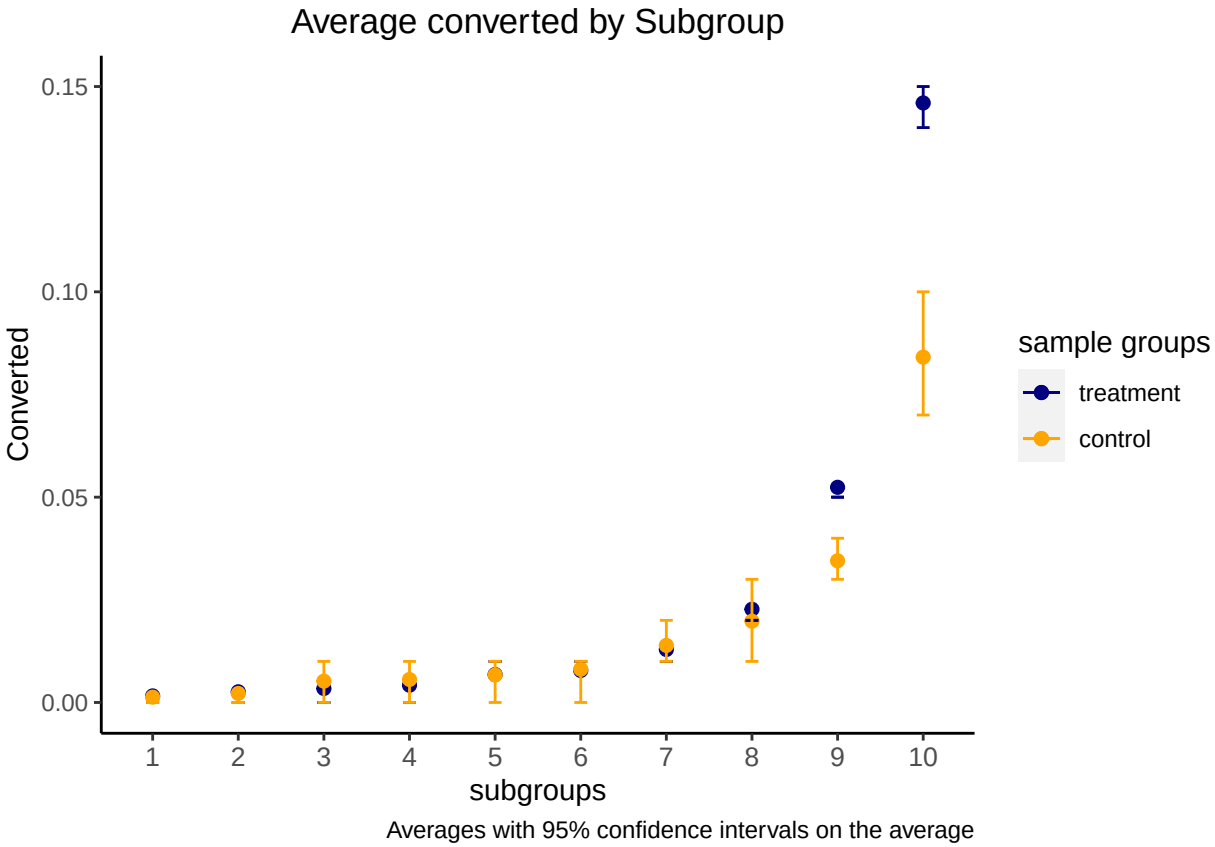
10. Plot the "converted" by treatment and 10 subgroups

Next create a graph that shows the mean and 95% CI on "converted" separately for treatment and control plotted over the 10 deciles of total impressions.

```
attach(casedata)
```

```
## The following objects are masked from casedata (pos = 3):
##
##     converted, mode_impr_day, mode_impr_hour, test, tot_impr,
##     tot_impr_decile, treat1, user_id
```

```
color <- c("treatment"="navy","control"="orange")
ggplot(NULL,aes(x=tot_impr_decile)) +
  geom_point(data=summary_subgroups[ which(summary_subgroups$test == "1"),],
             aes(y = mean.converted, color = "treatment"), size = 2) +
  geom_point(data=summary_subgroups[ which(summary_subgroups$test == "0"),],
             aes(y = mean.converted, color = "control"), size = 2) +
  scale_shape_manual(values=c(15, 16)) +
  labs(
    title = "Average converted by Subgroup",
    caption = "Averages with 95% confidence intervals on the average"
  )+
  ylab("Converted") +
  xlab('subgroups')+
  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=,
        plot.title=element_text(hjust=.5))+
  geom_errorbar(data=summary_subgroups[ which(summary_subgroups$test == "1"),],
                aes(ymin = LCI.converted,
                ymax = UCI.converted, color = "treatment"), width = .15)+
  geom_errorbar(data=summary_subgroups[ which(summary_subgroups$test == "0"),],
                aes(ymin = LCI.converted,
                ymax = UCI.converted, color = "control"), width = .15)+
  scale_color_manual(name="sample groups",values=color)
```



The graph clearly shows the effect of the treatment for different subgroups. We can see that, overall, the confidence interval of treatment groups is narrower than that of control groups, which means that we are more certain that user are more or less likely to buy Teskabella's handbag. In terms of the treatment effect for each subgroups, the conversion rate in subgroup 3, 4, 6, 7 are slightly lower after the treatment. However, for subgroups of 9th and 10 percentile, the treatment effect is a lot higher, showing a distinct positive relationship between the total number impression and conversion rate.