Ad smart: A/B Testing

Taiwo Owoseni

Contents

Brief Introduction

In this Project, I will work with the **AdSmartABdata data** I found on Kaggle. I intend to run a case study where A/B testing is applied on the click through rate. The primary aim is to compare user interactions with the bio questionnaire to determine which interaction statistically improves CTR .

The Data Columns :

- **auction_id:** the unique id of the online user who has been presented the BIO questionnaire.
- **experiment:** which group the user belongs to - control or exposed.
  - **control:** users who have been shown a dummy ad
  - **exposed:** users who have been shown a creative, an online interactive ad, with the SmartAd brand.
- **date:** the date in YYYY-MM-DD format
- **hour:** the hour of the day in HH format.
- **device_make:** the name of the type of device the user has e.g. Samsung
- **platform_os:** the id of the OS the user has.
- **browser:** the name of the browser the user uses to see the BIO questionnaire.
- **yes:** 1 if the user chooses the "Yes" radio button for the BIO questionnaire.
- **no:** 1 if the user chooses the "No" radio button for the BIO questionnaire.

Questions

A/B testing Comparing CTR

1. Does the CTR of *exposed* perform better than *control* when
   - Users **click** on the BIO questionnaire?
   - Users **fill** the BIO questionnaire?

Causality - Blocking I will address these problems by blocking on a variable

2. Does clicking(answering) the bio questionnaire of the *smart ad* or *dummy ad* (yes or no = 1) cause an improvement in user engagement?
3. Does engaging (yes = 1) with the bio questionnaire of the s*mart ad(exposed)* or *dummy ad(control*) ad result in an improvement in user engagement?

Import packages for the analysis

```
ad.data<- read_csv("ad_data.csv")
```

Load data.

```
## Rows: 8077 Columns: 9-- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): auction_id, experiment, device_make, browser
## dbl  (4): hour, platform_os, yes, no
## date (1): date
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(ad.data)
```

```
## # A tibble: 6 x 9
##   auction_id   experiment date        hour device_make platform_os browser   yes
##   <chr>        <chr>      <date>     <dbl> <chr>             <dbl> <chr>   <dbl>
## 1 0008ef63-77~ exposed    2020-07-10     8 Generic Sm~           6 Chrome~     0
## 2 000eabc5-17~ exposed    2020-07-07    10 Generic Sm~           6 Chrome~     0
## 3 0016d14a-ae~ exposed    2020-07-05     2 E5823                 6 Chrome~     0
## 4 00187412-29~ control    2020-07-03    15 Samsung SM~           6 Facebo~     0
## 5 001a7785-d3~ control    2020-07-03    15 Generic Sm~           6 Chrome~     0
## 6 0027ce48-d3~ control    2020-07-03    15 Samsung SM~           6 Facebo~     0
## # ... with 1 more variable: no <dbl>
```

Add two new columns:

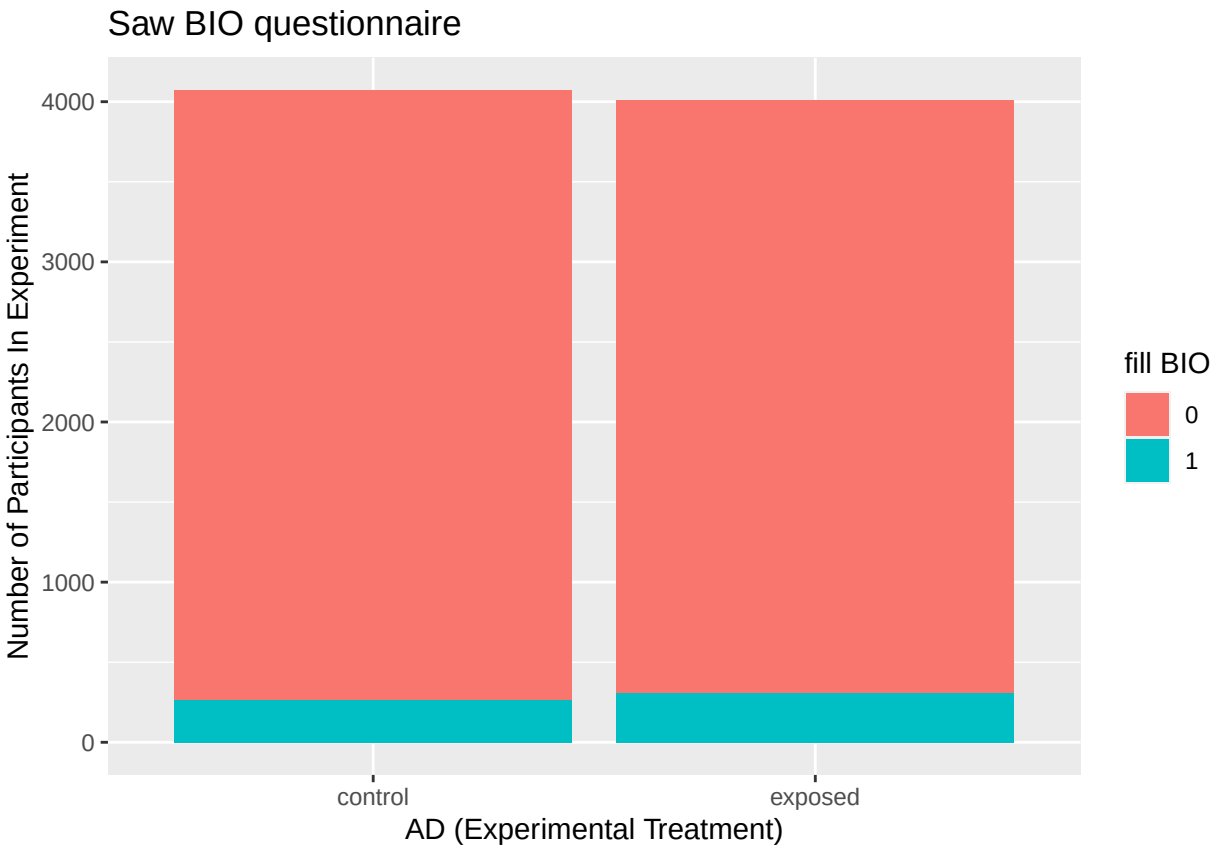
- **fill bio:** populate with 1 where the user responds to the bio questionnaire positively and 0 otherwise. That is; **yes = 1**
- **click bio:** populate with 1 where the user clicks the bio questionnaire (irrespective of their responses) and 0 otherwise. That is; **yes = 1 or no = 1.**

```
ad.data <- ad.data|>
          mutate(fill_bio = as.factor(case_when(yes == 1 ~ 1, yes == 0 ~ 0)),
          click_bio = as.factor(case_when(yes == 1 | no == 1 ~ 1, yes != 1 | no != 1 ~ 0 )))
```

```
fill_plot <-
    ggplot(ad.data) +
    ggtitle ("Saw BIO questionnaire") +
    geom_bar(mapping = aes(x = experiment, fill = fill_bio)) +
    xlab("AD (Experimental Treatment)") +
   ylab("Number of Participants In Experiment") +
   scale_fill_discrete(name = "fill BIO")

click_plot<-
   ggplot(ad.data) +
   ggtitle("Responded to BIO questionnaire") +
   geom_bar(mapping = aes(x = experiment, fill = click_bio)) +
   xlab("AD (Experimental Treatment)") +
   ylab("Number of Participants In Experiment") +
   scale_fill_discrete(name = "clicks BIO")

fill_plot
```


Saw BIO questionnaire
Number of Participants In Experiment
4000
3000
2000
1000
0
control
exposed
AD (Experimental Treatment)
fill BIO
0
1

```
click_plot
```

Using a chi-square to investigate the sample representation of the experiment.

```
# pvalue : 0.05
# chisquare to check the significance of the variation in the two experiemnts: control and
# exposure

chisq.test(ad.data$experiment, ad.data$fill_bio, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  ad.data$experiment and ad.data$fill_bio
## X-squared = 4.4449, df = 1, p-value = 0.03501
```

```
chisq.test(ad.data$experiment, ad.data$click_bio, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  ad.data$experiment and ad.data$click_bio
## X-squared = 6.2393, df = 1, p-value = 0.01249
```

From the Chi-Squared test result, we can conclude that then you know that the difference in the observed sample sizes due to randomness or chance.

```
#clicked bio_questionnaire regardless of response
# filter on yes and no
click_bio  <- ad.data|>filter(!(yes == 0 & no == 0))
fill_bio <- ad.data|>filter((yes == 1))
```

```r
ignore_bio <- ad.data|> filter((yes == 0 & no == 0))
```

```r
# how many users in the experiment?
# how many users in each experiment ignored the experiment?
# how many users clicked the bio in each experiment?
# how many users filled the bios in each experiment?


user_stat <- function(data, data_name){
            data |>
            group_by(experiment)|>
            summarize(users = n())|>
            mutate(type = data_name)
}

total_user_stat<- rbind(
      rbind(user_stat(ad.data, 'total'),
            user_stat(fill_bio, 'fill bio')),

      rbind(user_stat(ignore_bio, 'ignores bio'),
            user_stat(click_bio, 'clicks bio')))

total_user_stat
```

```
## # A tibble: 8 x 3
##   experiment users type
##   <chr>      <int> <chr>
## 1 control     4071 total
## 2 exposed     4006 total
## 3 control      264 fill bio
## 4 exposed      308 fill bio
## 5 control     3485 ignores bio
## 6 exposed     3349 ignores bio
## 7 control      586 clicks bio
## 8 exposed      657 clicks bio
```

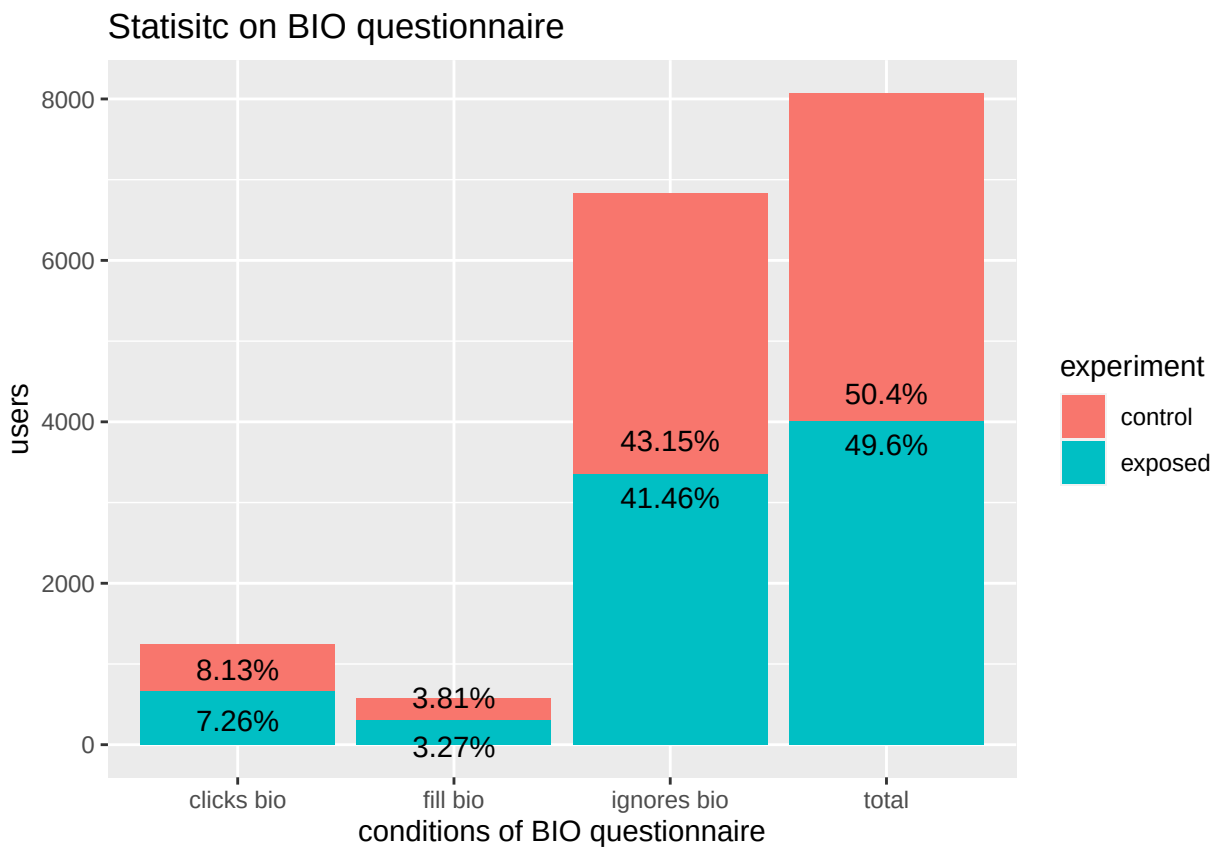
Visualizing User Statistics

- **Users who Click the BIO:** About **8.1%** of the total user who responded (clicked) to the bio are in the control group. **7.3%** of the total user who responded(clicked) to the bio are in the exposed group.
- **Users who Fill the BIO:** About **3.8%** of the total user who filled the bio are in the control group. and **3.7%** of the total user who filled the bio are in the exposed group.
- **Users who Ignore the BIO:** About **43.2%** of the total user who ignored the bio are in the control group. and **41.5%** of the total user who ignored the bio are in the exposed group.

Generally, It is observed that the representation of control and exposed in the three groups are NOT equally represented . It is still is a good representation because it has been statistically tested with chi-square test above.

```r
# Calculate y position, placing it in the middle
ggplot(total_user_stat, aes(x = type, y =users, fill = experiment)) +
  geom_col() +
  ggtitle('Statisitc on BIO questionnaire') +
  xlab('conditions of BIO questionnaire') +
  geom_text_repel(data = total_user_stat, size = 4,
```

```
                mapping = aes(x = type, y =users,
                label=paste0(round(users / nrow(ad.data) * 100, 2), "%")))
```



Comparing CTR

Question 1a. *Does the CTR of **exposed** perform better than **control** when users **click** on the BIO questionnaire?*

```
ctr_table <- function(col) {
  ad <- ad.data|>
          group_by(experiment)|>
         summarise(impressions =  n(),
                   clicks = sum(as.numeric({{col}} == 1)))|>
          mutate(ctr = clicks/impressions)

  ci <- binom.confint(ad$clicks, ad$impressions,
                      methods = "exact", conf.level = 0.9)
  ad$lower_ci <- ci$lower
  ad$upper_ci<- ci$upper

  ad$experiment <- fct_reorder(ad$experiment, desc(ad$ctr))
  ad

}

ctr_plot <- function(ad, label_y) {
```

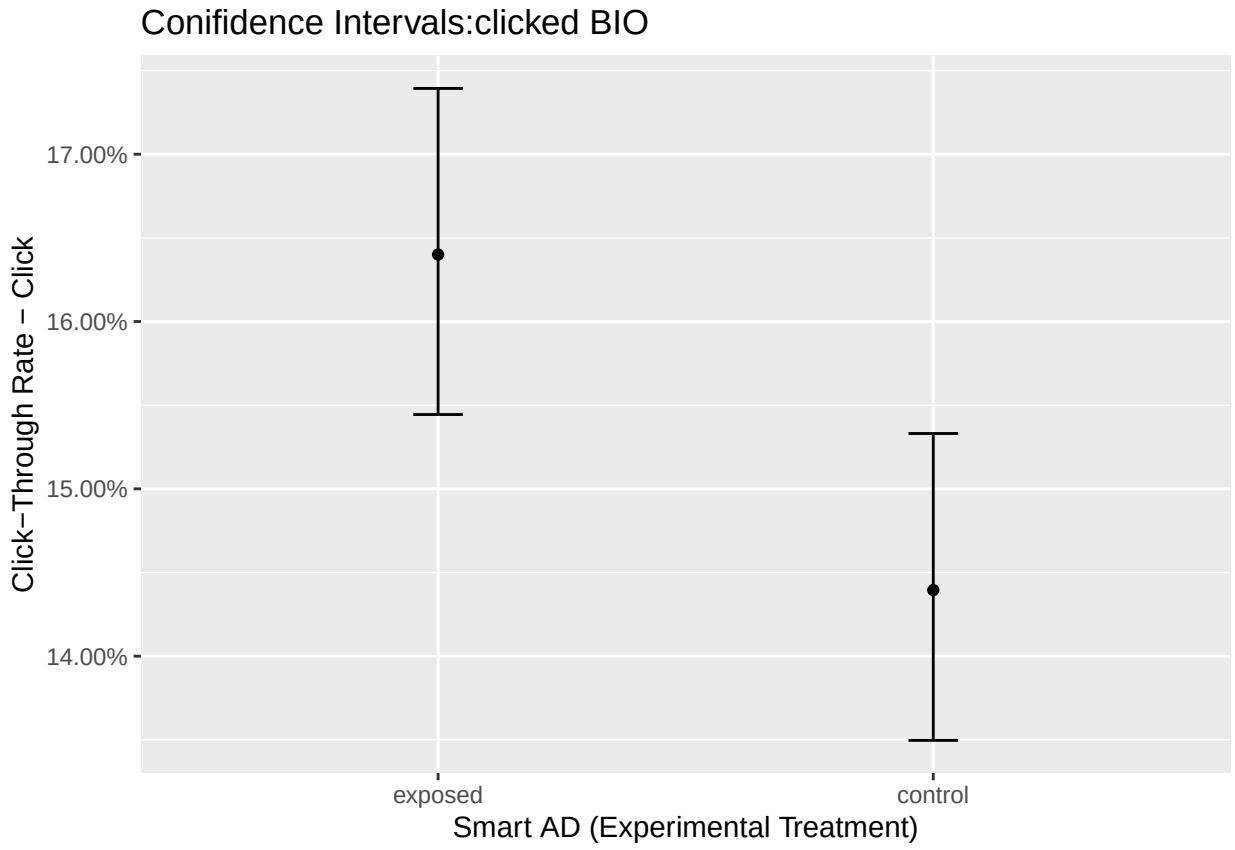
```
# Plotting as a point with 95% binomial exact confidence intervals.

  CI_plot <- ggplot(ad, aes(experiment, ctr)) +
    geom_point() +
    geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.1) +
    scale_y_continuous(labels = scales::percent) +
    xlab("Smart AD (Experimental Treatment)") +
    ylab(label_y)

  CI_plot
}
```

Visually, There **isn't an overlap** in the confidence interval of both the exposed and control. Exposure has the highest click through rate **approx. 16.5%** and control with **approx. 14.5%** . This isn't enough evidence that exposure was better than control. I will use a statistical test(prop test) to compare the treatments.

```
click_ad <- ctr_table(click_bio )
ctr_plot(click_ad, "Click-Through Rate - Click") +
  ggtitle('Conifidence Intervals:clicked BIO')
```



Stating the following Hypothesis:

Null Hypothesis:

$$H_o : CTR_{\text{exposure}} = CTR_{\text{control}}$$

Alternative Hypothesis:

$$H_a : CTR_{\text{exposure}} <> CTR_{\text{control}}$$

```r
successes <- click_ad$clicks
trials <- click_ad$impressions
names(successes) <- click_ad$experiment
names(trials) <- click_ad$experiment

click_single_comp_ad <- prop.test(successes, trials,
                      alternative = c("two.sided"),
                      correct = FALSE ) %>%
                      tidy()
click_single_comp_ad
```

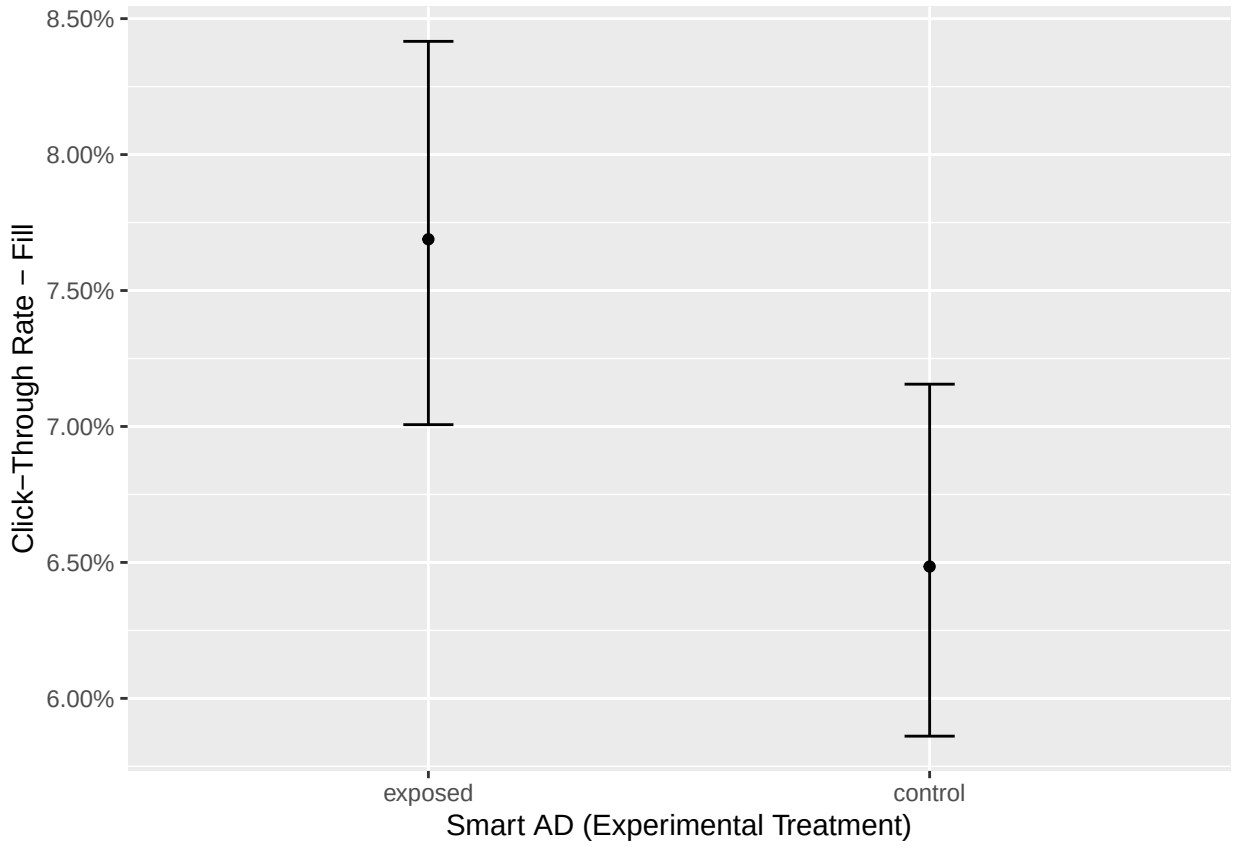
```
## # A tibble: 1 x 9
##   estimate1 estimate2 statistic p.value parameter conf.low conf.high method
##       <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>
## 1     0.144     0.164      6.24  0.0125         1  -0.0358  -0.00432 2-sample t~
## # ... with 1 more variable: alternative <chr>
```

Using alpha = 0.05, we have enough statistical evidence that Experiment exposure has a better click through rate than control. This is because the p-value is : 0.01.

Question 1b *Does the CTR of Exposure Perform better than Control when Users **fill** the BIO questionnaire?*

Visually, There **is an overlap** in the confidence interval of both the exposed and control experiments. Exposure has the highest click through rate **approx. 7.75%** and control with **approx. 6.50%** . This isn't enough evidence that exposure was better than control. I will use a statistical test(prop test) to compare the treatments.

```r
fill_ad <- ctr_table(fill_bio )
ctr_plot(fill_ad, "Click-Through Rate - Fill")
```



Stating the following Hypothesis:

Null Hypothesis:

$$H_o : CTR_{\text{exposure}} = CTR_{\text{control}}$$

Alternative Hypothesis:

$$H_a : CTR_{\text{exposure}} <> CTR_{\text{control}}$$

```
successes <- fill_ad$clicks
trials <- fill_ad$impressions
names(successes) <- fill_ad$experiment
names(trials) <-fill_ad$experiment

fill_single_comp_ad <- prop.test(successes, trials,
                       alternative = c("two.sided"),
                       correct = FALSE ) %>%
                         tidy()

fill_single_comp_ad
```

```
## # A tibble: 1 x 9
##   estimate1 estimate2 statistic p.value parameter conf.low conf.high method
##       <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl> <chr>
## 1    0.0648    0.0769      4.44  0.0350         1  -0.0232 -0.000843 2-sample t~
## # ... with 1 more variable: alternative <chr>
```

Using alpha = 0.05, we have enough statistical evidence to say that Experiment **exposure** has a better click through rate than **contro**l when the users fill the bio questionaire. This is because the p-value is : 0.03.

Blocking

Question 2a *Does clicking(answering) the bio questionnaire of the smart ad or dummy ad (yes or no = 1) cause an improvement in user engagement?*

Choosing a Blocking Variable/ feature/ Regressor

I am following this principle to select a blocking variable :

- It is included as a factor in the experiment.
- It is not of primary interest to the experimenter.
- It affects the dependent variable.
- It is unrelated to independent variables in the experiment.

Let's look at the number of unique factors in each of the possible blocking variables:

```
# possible groups for blocking
paste("Browser:", length(unique(ad.data$browser)))
```

```
## [1] "Browser: 15"
```

```
print('-----------')
```

```
## [1] "-----------"
```

```
paste("Device Make:", length(unique(ad.data$device_make)))
```

```
## [1] "Device Make: 269"
```

```
print('-----------')
```

```
## [1] "-----------"
```

```
paste("Platform OS:", length(unique(ad.data$platform_os)))
```

```
## [1] "Platform OS: 3"
```

```
print('-----------')
```

```
## [1] "-----------"
```

```
paste("Hours:", length(unique(ad.data$hour)))
```

```
## [1] "Hours: 24"
```

Observation

I will select the browser as a blocking variable. Then group the browsers into popular/general browser names. Other potential blocking variable would be to bin the **hours** or cluster the **device make** into popular blocks of popular device brand name.

```
unique(ad.data$browser)
```

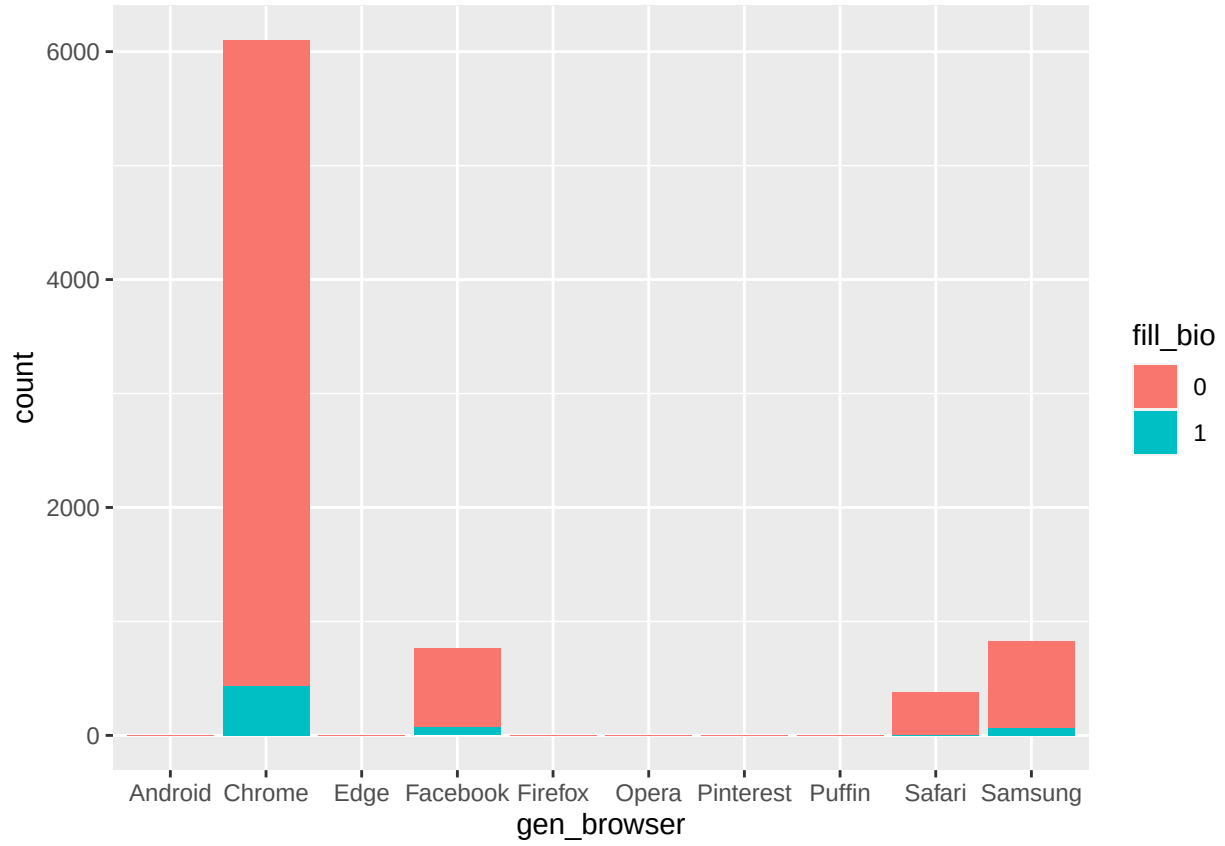
```
##  [1] "Chrome Mobile"              "Chrome Mobile WebView"
##  [3] "Facebook"                   "Mobile Safari"
##  [5] "Chrome Mobile iOS"          "Samsung Internet"
##  [7] "Mobile Safari UI/WKWebView" "Chrome"
##  [9] "Opera Mini"                 "Edge Mobile"
```

```
## [11] "Android"                    "Pinterest"
## [13] "Opera Mobile"               "Firefox Mobile"
## [15] "Puffin"
```

```
replace_browser <- function (data) {
    data %>%
     mutate(gen_browser = factor(case_when(
        str_detect( browser, 'Android') ~ "Android",
        str_detect( browser, 'Chrome') ~ "Chrome" ,
        str_detect( browser, 'Edge') ~ "Edge",
        str_detect(browser, 'Facebook') ~ "Facebook" ,
        str_detect( browser, 'Firefox') ~ "Firefox",
        str_detect( browser, 'Opera') ~ "Opera",
        str_detect( browser, 'Pinterest') ~ "Pinterest",
        str_detect( browser, 'Puffin') ~ "Puffin",
        str_detect( browser, 'Safari') ~ "Safari" ,
        str_detect( browser, 'Samsung') ~ "Samsung")))
}

click_bio<- replace_browser(click_bio)
ad.data<- replace_browser(ad.data)
fill_bio<- replace_browser(fill_bio)
```

```
ggplot(ad.data) +
  geom_bar(aes(x = gen_browser, fill = fill_bio))
```



Compute the click through rate - CTR per gen_browser

```
click_bio_ctr <-  click_bio %>%
  group_by(gen_browser) |>
   mutate(ctr_click =  n()/nrow(ad.data),
          n = n())
fill_bio_ctr <-  fill_bio %>%
  group_by(gen_browser) |>
   mutate(ctr_fill =  n()/nrow(ad.data),
          n = n())
```

Data Modelling

```
fill_lm <- lm(ctr_fill ~ gen_browser , data = fill_bio_ctr, )
tidy(fill_lm, conf.int = 0.95) %>% mutate_if(is.numeric, round, 3)
```

OLS

```
## # 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)            0.053         0   2.45e15       0    0.053     0.053
## 2 gen_browserFacebook   -0.045         0  -7.63e14       0   -0.045    -0.045
## 3 gen_browserSafari     -0.053         0  -2.32e14       0   -0.053    -0.053
## 4 gen_browserSamsung    -0.045         0  -7.58e14       0   -0.045    -0.045
```

```
click_lm <- lm(ctr_click ~ gen_browser , data = click_bio_ctr)
tidy(click_lm, conf.int = 0.95) %>% mutate_if(is.numeric, round, 3)
```

```
## # 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   2.85e15       0    0.115     0.115
## 2 gen_browserFacebook   -0.095         0  -8.99e14       0   -0.095    -0.095
## 3 gen_browserSafari     -0.112         0  -3.75e14       0   -0.112    -0.112
## 4 gen_browserSamsung    -0.097         0  -8.83e14       0   -0.097    -0.097
```

```
fill_lm.1 <- lm(ctr_fill ~ gen_browser + hour , data = fill_bio_ctr)
tidy(fill_lm.1, conf.int = 0.95) %>% mutate_if(is.numeric, round, 3)
```

```
## # A tibble: 5 x 7
##   term                estimate std.error statistic p.value conf.low conf.high
##   <chr>                  <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)            0.053         0   1.20e15   0        0.053     0.053
## 2 gen_browserFacebook   -0.045         0  -7.63e14   0       -0.045    -0.045
## 3 gen_browserSafari     -0.053         0  -2.31e14   0       -0.053    -0.053
## 4 gen_browserSamsung    -0.045         0  -7.58e14   0       -0.045    -0.045
## 5 hour                   0             0  -1.15e 0   0.252    0         0
```

```
click_lm.1 <- lm(ctr_click ~ gen_browser + hour , data = click_bio_ctr)
tidy(click_lm.1 , conf.int = 0.95) %>% mutate_if(is.numeric, round, 3)
```

```
## # A tibble: 5 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  1.44e+15   0        0.115     0.115
## 2 gen_browserFacebook   -0.095         0 -8.96e+14   0       -0.095    -0.095
## 3 gen_browserSafari     -0.112         0 -3.75e+14   0       -0.112    -0.112
```

```
## 4 gen_browserSamsung    -0.097         0 -8.83e+14   0       -0.097    -0.097
## 5 hour                   0             0  1.78e- 1   0.859    0         0
```

Interaction