Chi-square Test on User Conversion Rates

Aaron Stearns

In this analysis I will be taking a look at the conversion rates of a treatment and control group who were shown two different versions of a landing page for a website. I will manually calculate the chi-square statistic for the conversion and non-conversion rates for each set of users, and then use R's built in `chisq.test()` function to compare my results.

Data taken from "A/B testing" dataset on kaggle https://www.kaggle.com/zhangluyuan/ab-testing

I'll start by importing the data and taking a look at the first few rows:

```
library(dplyr)

data <- read.csv("~/Downloads/ab_data.csv", stringsAsFactors = FALSE)

head(data)
```

```
##   user_id                   timestamp     group landing_page converted
## 1  851104 2017-01-21 22:11:48.556739   control     old_page         0
## 2  804228 2017-01-12 08:01:45.159739   control     old_page         0
## 3  661590 2017-01-11 16:55:06.154213 treatment     new_page         0
## 4  853541 2017-01-08 18:28:03.143765 treatment     new_page         0
## 5  864975 2017-01-21 01:52:26.210827   control     old_page         1
## 6  936923 2017-01-10 15:20:49.083499   control     old_page         0
```

```
# Checking for NA values
table(is.na(data))
```

```
##
##   FALSE
## 1472390
```

The data is supposed to be be split 50/50, but we'll see if that's by the "landing page" or "group" column

```
table(data$group)
```

```
##
##   control treatment
##    147202    147276
```

```
table(data$landing_page)
```

```
##
## new_page old_page
##   147239   147239
```

```
table(data$group, data$landing_page)
```

```
##
##             new_page old_page
##   control       1928   145274
##   treatment   145311     1965
```

It looks like although all of the control group should have been shown the old page, and all of the treatment group should have been shown the new page, there were users in each group that were shown the wrong page.

I'll filter out the observations that were shown the incorrect pages and check to make sure that they each had unique user ids.

```
control <- data %>%
            filter(group == "control" & landing_page == "old_page")
treatment <- data %>%
              filter(group == "treatment" & landing_page == "new_page")

print(
  paste(
    "Unique users in control group:",
    length(unique(control$user_id)),
    "Total rows in control group:",
    nrow(control)
  )
)
```

```
## [1] "Unique users in control group: 145274 Total rows in control group: 145274"
```

```
print(
  paste(
    "Unique users in treatment group:",
    length(unique(treatment$user_id)),
    "Total rows in treatment group:",
    nrow(treatment)
  )
)
```

```
## [1] "Unique users in treatment group: 145310 Total rows in treatment group: 145311"
```

It looks like one user has two observations in the treatment group. Let's see which one:

```
treatment %>%
  group_by(user_id) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>% head(3)
```

```
## # A tibble: 3 x 2
##   user_id     n
##     <int> <int>
## 1  773192     2
## 2  630000     1
## 3  630001     1
```

So user 773192 appears in two observations.

```
treatment %>%
  filter(user_id == 773192)
```

```
##   user_id                   timestamp     group landing_page converted
## 1  773192 2017-01-09 05:37:58.781806 treatment     new_page         0
## 2  773192 2017-01-14 02:55:59.590927 treatment     new_page         0
```

Since these observations are both in the treatment group and were both shown the new landing page, and neither one of them converted, we can simply remove one of them. I'll check to see if I can filter based on timestamp:

```
treatment %>%
  filter(timestamp == "2017-01-14 02:55:59.590927")
```

```
##   user_id                   timestamp     group landing_page converted
## 1  773192 2017-01-14 02:55:59.590927 treatment     new_page         0
```

```
treatment <- treatment %>%
             filter(!timestamp == "2017-01-14 02:55:59.590927")

controlConverted <- control %>%
                    group_by(converted) %>%
                    summarise(observed = n())

treatmentConverted <- treatment %>%
                      group_by(converted) %>%
                      summarise(observed = n())

controlConverted
```

```
## # A tibble: 2 x 2
##   converted observed
##       <int>    <int>
## 1         0   127785
## 2         1    17489
```

```
treatmentConverted
```

```
## # A tibble: 2 x 2
##   converted observed
##       <int>    <int>
## 1         0   128046
## 2         1    17264
```

```
# Add together the converts from the treatment and control groups
converts <- sum(controlConverted$observed[2],
                treatmentConverted$observed[2])

# Divide the number of converts by the total rows in each group
pHat <- converts / sum(nrow(treatment), nrow(control))

# Creating "observed" and "expected" columns for each group
# and then combining them into one data frame
controlConverted$expected <- nrow(control) - round(nrow(control) * pHat)
controlConverted$expected[2] <- round(nrow(control) * pHat)
controlConverted$group <- "control"

treatmentConverted$expected <- nrow(treatment) - round(nrow(treatment) * pHat)
treatmentConverted$expected[2] <- round(nrow(treatment) * pHat)
treatmentConverted$group <- "treatment"

df <- rbind(controlConverted, treatmentConverted)

df <- df %>%
        mutate(
          toAdd = ((observed - expected)^2)/expected)
df
```

```
## # A tibble: 4 x 5
##   converted observed expected group     toAdd
##       <int>    <int>    <dbl> <chr>     <dbl>
```

```
## 1         0   127785   127900 control   0.103
## 2         1    17489    17374 control   0.761
## 3         0   128046   127931 treatment 0.103
## 4         1    17264    17379 treatment 0.761
```

Now I'll add the values of the "toAdd" column for the chi-square statistic:

```
chiSq <- round(sum(df$toAdd), digits = 1)

chiSq
```

```
## [1] 1.7
```

And now I'll use R's built-in function with the data formatted properly for it to check if I arrived at the same chi-square value as the built-in function. Th summary of the function will show the chi-square value, the degrees of freedom (which in this case will be 1), and the p-value.

```
ddf <- as.table(rbind(unlist(controlConverted[2]), unlist(treatmentConverted[2])))

dimnames(ddf) <- list(group = c("control", "treatment"),
                      convert = c("noConvert", "convert"))

xsf <- chisq.test(ddf, correct = F)

xsf
```

```
##
##  Pearson's Chi-squared test
##
## data:  ddf
## X-squared = 1.7185, df = 1, p-value = 0.1899
```

It looks like with a p-value of 0.1899 we can safely say that there is no significant statistical difference.