

A/B Testing Time Series Analysis

Zac Rodeheffer

```
rm(list = ls()) # Clear the workspace
```

import the dataset and name it as dat

```
dat <- read.csv("RatingData.csv")
summary(dat)
```

```
##      id      month      rating      treat
## Min.   : 75   Min.   : 1.00   Min.   :1.000   Min.   :0.0000
## 1st Qu.:4224   1st Qu.: 7.00   1st Qu.:3.000   1st Qu.:0.0000
## Median :16821  Median :14.00   Median :4.000   Median :0.0000
## Mean   :29071  Mean   :13.39   Mean   :3.742   Mean   :0.2341
## 3rd Qu.:47383  3rd Qu.:20.00   3rd Qu.:5.000   3rd Qu.:0.0000
## Max.   :104668 Max.   :24.00   Max.   :5.000   Max.   :1.0000
```

Plot the Average Rating with CIs for All Months for both treatment and control

```
library(dplyr)
```

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

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

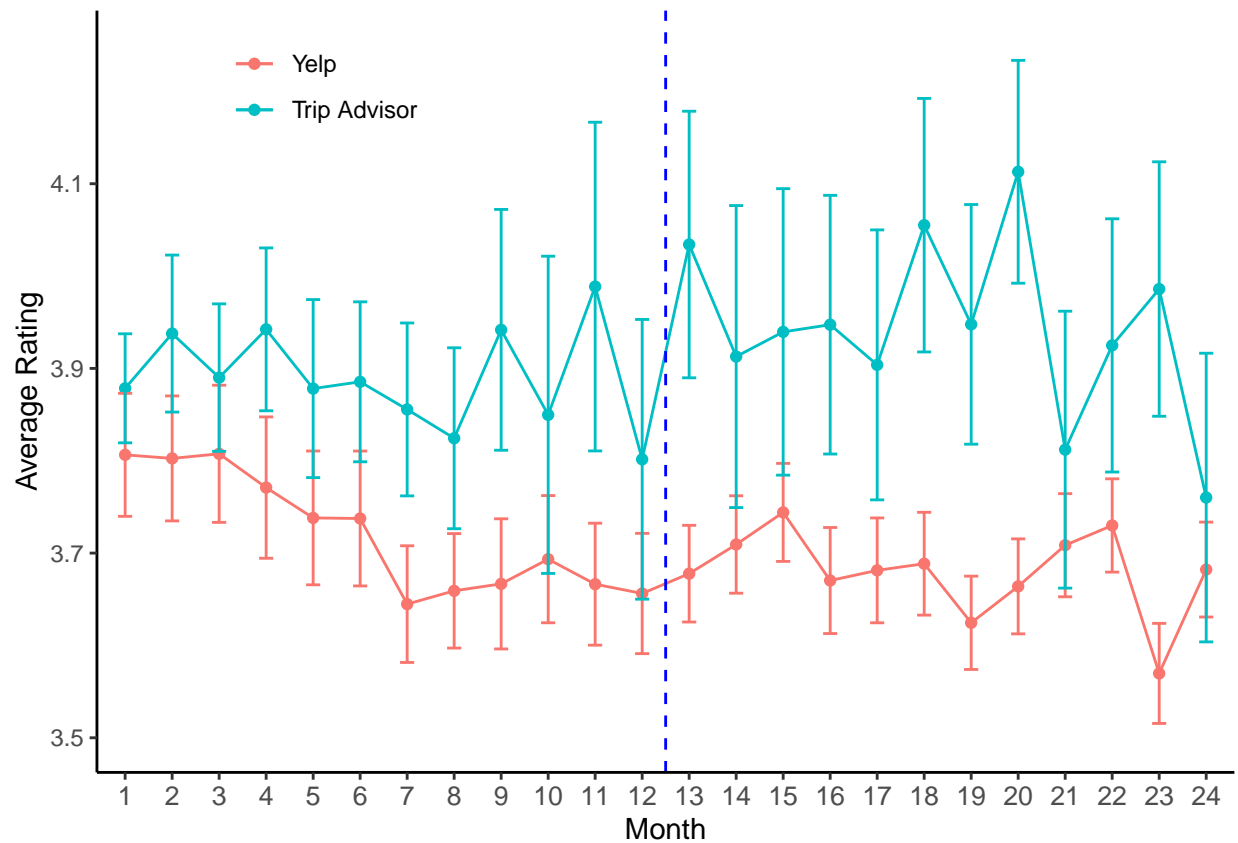
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
summary <- dat %>%
  mutate(treat = as.factor(treat)) %>%
  mutate(month = as.factor(month)) %>%
  group_by(treat, month) %>% #group the data by treat and month
  summarise (n = length(id),
             m.rating= mean(rating),
             error = sd(rating)/sqrt(n),
             lci = m.rating - 1.96*error,
             uci = m.rating + 1.96*error)
```

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

```
library(ggplot2)
summary %>%
  ggplot(aes(x=month, y = m.rating, group = treat, color = treat)) +
  # here we declare what's the horizontal and vertical vars for the plot
  # and what variable we want to group by
  geom_point()+ #add the dots to the plot
  geom_line()+ # connect the dots
  geom_vline(xintercept=12.5, linetype="dashed", color = "blue")+
  # add a dash line to indicate the change time.
  geom_errorbar(aes(ymin = lci, ymax = uci,
                    color=treat), width= 0.3)+
  scale_y_continuous(limits = c(3.5,4.25)) + #define the range of y-axis
  ylab("Average Rating") +
  xlab("Month") +
  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(0.2,0.9),
        plot.title=element_text(hjust=.5) ) +
  scale_color_discrete(name=NULL, labels = c("Yelp", "Trip Advisor"))
```

```
## Warning: A numeric 'legend.position' argument in 'theme()' was deprecated in ggplot2
## 3.5.0.
## i Please use the 'legend.position.inside' argument of 'theme()' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



Regression Analysis

```
#generate a dummy variable named post to indicate
#if this observation is recorded after the change month or not
dat$post = ifelse(dat$month>12,1,0)
```

```
#generate a dummy variable named post to indicate
#if this observation is in the treatment region AND recorded after the change month
dat$post_treat = dat$post*dat$treat
```

```
library(jtools)
fit.rating = lm(rating~treat+post+post_treat, data = dat)
summ(fit.rating, robust = "HC1", confint = TRUE, digits = 3)
```

```
## MODEL INFO:
## Observations: 38004
## Dependent Variable: rating
## Type: OLS linear regression
##
## MODEL FIT:
## F(3,38000) = 97.734, p = 0.000
## R2 = 0.008
## Adj. R2 = 0.008
```

```
##
## Standard errors: Robust, type = HC1
## -----
##           Est.      2.5%    97.5%    t val.      p
## -----
## (Intercept)      3.714     3.694     3.734    367.218    0.000
## treat             0.173     0.139     0.207     9.977    0.000
## post            -0.036    -0.061    -0.011    -2.832    0.005
## post_treat       0.095     0.039     0.151     3.344    0.001
## -----
```