OLS

575 C1 Team 3

2020/10/1

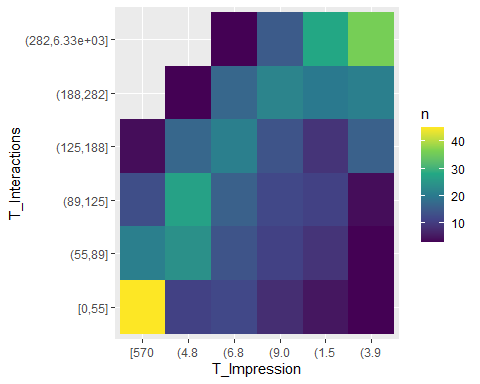
Team Members Dingjie Chen, Siwen He, Hanzi Yu, Jiaqi Yin, Runsheng Wang

In this deliverable, we are performing OLS on the facebook dataset. We first load the packages needed to perform the analysis and read in the delimited file. We modified the column names of the CSV file so that column names would not contain space, as space is not a valid name character in ggplot. We used the complete.cases() function to handle NA values. Also note that “Category” and “Paid” variables are being interpreted as a double by the col\_guess() function. To use these two variables as categorical, we could call the as.factor() function.

#loading packages and dataset  
suppressPackageStartupMessages(library(tidyverse))  
suppressPackageStartupMessages(library(modelr))  
fb <- read\_delim("dataset\_Facebook.csv", delim = ";")  
fb <- fb[complete.cases(fb), ]

To identify potentially meaningful relationships, we construct a heatmap for T\_Impression and T\_Interactions. Since facebook does a horrible job at explaining their metrics, it makes sense for us to first define clearly what each variable is accounting for. After doing some research online, we found that Facebook calculates

# construct heatmap  
fb %>% transmute(T\_Impression = cut\_number(T\_Impression, 6), T\_Interactions = cut\_number(T\_Interactions, 6)) %>% count(T\_Impression, T\_Interactions) %>% ggplot(aes(T\_Impression, T\_Interactions)) + geom\_tile(aes(fill = n)) + scale\_x\_discrete(labels = abbreviate) + scale\_fill\_viridis\_c()



On the heatmap, lighter colors correspond to higher number of posts that belongs to the bin with specific T\_Impression and T\_Interactions. We see that lighter colors are clustering along the diagonal of the heatmap, indicating a potentially positive correlation between the two variables. This heatmap motivate us to plot T\_Interactions against T\_Impression, as illustrated below.

First, we identify outliers and remove them.

# clean outliers  
fbmean = mean(fb$T\_Impression)  
fbsd = sd(fb$T\_Impression)  
fb.clean <- fb %>% filter(T\_Impression <= fbmean+3\*fbsd)  
# calculate percentage of datapoints considered  
removedt <- (1 - nrow(fb.clean)/nrow(fb))\*100  
percentage\_tibble <- tribble(~Variable, ~Percentage\_Removed, "Overall", removedt)  
(percentage\_tibble)

## # A tibble: 1 x 2  
## Variable Percentage\_Removed  
## <chr> <dbl>  
## 1 Overall 1.41

Notice that by including datapoints within 3 standard deviation from the mean removed 1.14% of the total data points.

Now, we proceed to calculate OLS

# calculate OLS  
m.ols <- lm(T\_Interactions ~ T\_Impression, data = fb.clean)  
summary(m.ols)

##   
## Call:  
## lm(formula = T\_Interactions ~ T\_Impression, data = fb.clean)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -835.1 -92.1 -43.1 39.7 1624.1   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.229e+02 1.211e+01 10.15 <2e-16 \*\*\*  
## T\_Impression 3.323e-03 2.879e-04 11.54 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 225.9 on 486 degrees of freedom  
## Multiple R-squared: 0.2152, Adjusted R-squared: 0.2136   
## F-statistic: 133.3 on 1 and 486 DF, p-value: < 2.2e-16

round(confint(m.ols,level=0.95),6)

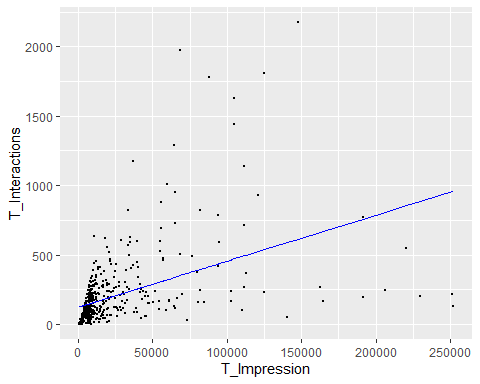
## 2.5 % 97.5 %  
## (Intercept) 99.149803 146.746087  
## T\_Impression 0.002758 0.003889

vcov(m.ols)

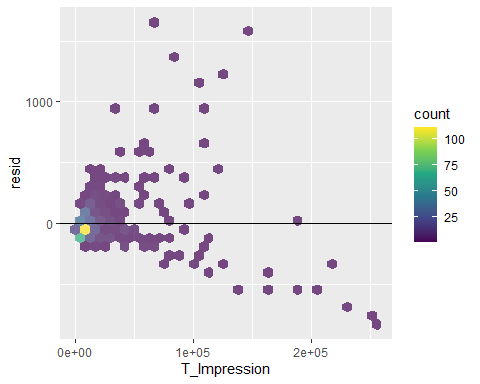
## (Intercept) T\_Impression  
## (Intercept) 146.697977566 -1.867812e-03  
## T\_Impression -0.001867812 8.287600e-08

As can be seen from the output, the fit is not quite good, with an adjusted R-square value of 0.1154. The intercept and slope seem to have a high t value and a low p value, indicating that T\_Impression is a significant predictor. However, the relationship under the linear model is not a strong one. The confidence interval for the intercept is very wide, whereas the confidence interval on slope is very narrow. This outcome agrees with our previous analysis that the relationship between T\_Impression and T\_Interactions might not be linear. We will plot the OLS model below with residuals, and offer an alternative solution to fitting this data later in this document.

#plot OLS with predictions  
ggplot(fb.clean %>% add\_predictions(m.ols), aes(T\_Impression, T\_Interactions)) + geom\_point(size = 0.1) + geom\_line(aes(y=pred), color = "blue")



# plot residual  
ggplot(fb.clean %>% add\_residuals(m.ols), aes(T\_Impression, resid))+geom\_hex(alpha = 0.7)+geom\_hline(aes(yintercept = 0))+scale\_fill\_viridis\_c()



As seen in the graphs above, the linear model doesn’t fit the data quite well. The residual plot also indicate that we have the assumption of equal variances do not hold in this case, as residuals tend to be greater for higher values of T\_Impression. Notice that data seem to cluster below T\_Impression = 30000. We perform OLS again on T\_Impression <= 30000

# clean outliers  
fb.clean1 <- fb %>% filter(T\_Impression <= 30000)  
# calculate percentage of datapoints considered  
removedt1 <- (1 - nrow(fb.clean1)/nrow(fb))\*100  
percentage\_tibble <- tribble(~Variable, ~Percentage\_Removed, "Overall", removedt1)  
(percentage\_tibble)

## # A tibble: 1 x 2  
## Variable Percentage\_Removed  
## <chr> <dbl>  
## 1 Overall 21.0

# calculate OLS  
m.ols1 <- lm(T\_Interactions ~ T\_Impression, data = fb.clean1)  
summary(m.ols1)

##   
## Call:  
## lm(formula = T\_Interactions ~ T\_Impression, data = fb.clean1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -227.86 -54.89 -15.76 43.92 482.86   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.187e+01 8.566e+00 6.055 3.3e-09 \*\*\*  
## T\_Impression 8.961e-03 7.695e-04 11.646 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 93.64 on 389 degrees of freedom  
## Multiple R-squared: 0.2585, Adjusted R-squared: 0.2566   
## F-statistic: 135.6 on 1 and 389 DF, p-value: < 2.2e-16

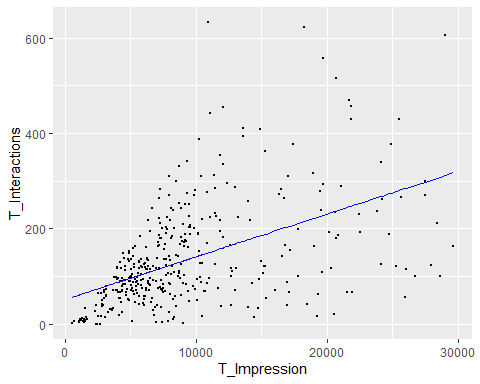
round(confint(m.ols1,level=0.95),6)

## 2.5 % 97.5 %  
## (Intercept) 35.029691 68.713121  
## T\_Impression 0.007448 0.010474

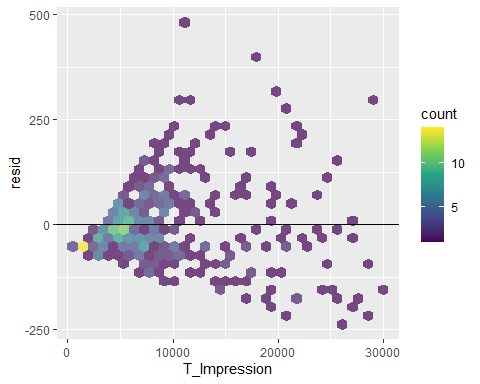
vcov(m.ols1)

## (Intercept) T\_Impression  
## (Intercept) 73.37866365 -5.492540e-03  
## T\_Impression -0.00549254 5.920583e-07

#plot OLS with predictions  
ggplot(fb.clean1 %>% add\_predictions(m.ols1), aes(T\_Impression, T\_Interactions)) + geom\_point(size = 0.1) + geom\_line(aes(y=pred), color = "blue")



# plot residual  
ggplot(fb.clean1 %>% add\_residuals(m.ols1), aes(T\_Impression, resid))+geom\_hex(alpha = 0.7)+geom\_hline(aes(yintercept = 0))+scale\_fill\_viridis\_c()



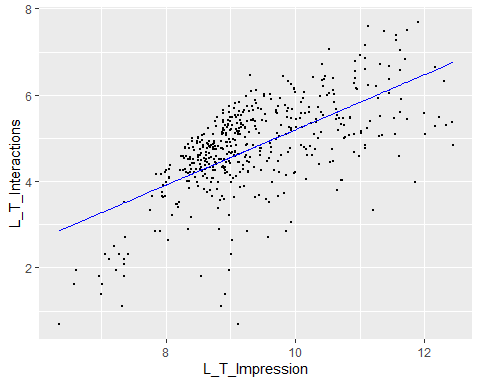
The output still has significant values for parameter estimation with slightly improved Adjusted R-Square. The output of this plot and the visual trend of the scatterplot motivates us to do a log transformation on the variable.

# log transform  
logfb <- fb.clean %>% transmute(L\_T\_Interactions = log(T\_Interactions), L\_T\_Impression = log(T\_Impression)) %>% filter(!is.infinite(L\_T\_Interactions))  
m.log <- lm(L\_T\_Interactions ~ L\_T\_Impression, data = logfb)  
summary(m.log)

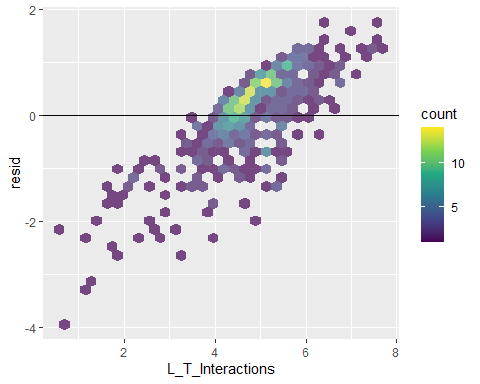
##   
## Call:  
## lm(formula = L\_T\_Interactions ~ L\_T\_Impression, data = logfb)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.9347 -0.4297 0.1383 0.5774 1.7011   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.19130 0.33136 -3.595 0.000358 \*\*\*  
## L\_T\_Impression 0.63860 0.03517 18.160 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.8358 on 481 degrees of freedom  
## Multiple R-squared: 0.4067, Adjusted R-squared: 0.4055   
## F-statistic: 329.8 on 1 and 481 DF, p-value: < 2.2e-16

This model is considerably better than the previous one, with very low p values for both parameters of interest. This output indicates that the both beta0 and beta1 are significant. Adjusted R-squared value is also high, at 0.3989, indicating that this fit is better compared to the linear fit. The graphs for the log-transformed variables are given below.

# plot log with predictions  
ggplot(logfb %>% add\_predictions(m.log), aes(L\_T\_Impression, L\_T\_Interactions)) + geom\_point(size = 0.1) + geom\_line(aes(y=pred), color = "blue")



# plot residual  
ggplot(logfb %>% add\_residuals(m.log), aes(L\_T\_Interactions, resid))+geom\_hex(alpha = 0.7)+geom\_hline(aes(yintercept = 0))+scale\_fill\_viridis\_c()



Notice that the log plot follows a roughly linear trend, with residuals roughly clustering around the y=0 line. There seems to be a linear trend in the residual, indicating that the linearity assumption could be violated. If log transformation yields a considerably good linear fit, the non-transformed data could exhibit an exponential relationship. We won’t expand on this concept for this project deliverable, but it will be considered when we construct our final project.