575 C1 Team 3

2020/10/26

Multiple Linear Regression

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

In this deliverable, we are performing Multiple Linear Regression 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
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(modelr))
suppressPackageStartupMessages(library(hrbrthemes))
suppressPackageStartupMessages(library(GGally))
suppressPackageStartupMessages(library(gridExtra))
suppressPackageStartupMessages(library(plotly))

# loading datasets
fb <- read_delim("dataset_Facebook.csv", delim = ";")
fb <- fb[complete.cases(fb), ]

# center titles for ggplot
theme_update(plot.title = element_text(hjust = 0.5))
```

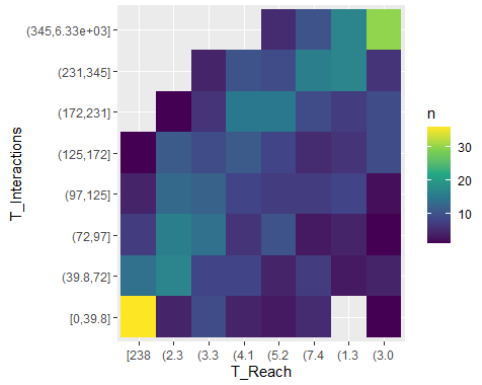
Identifying Potential Covariates

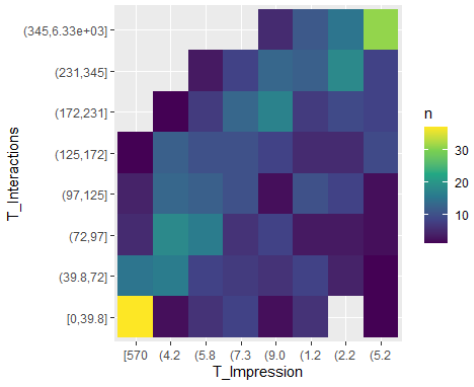
Using T_Interactions as the response variable, we construct heatmaps for our potential covariates to identify meaningful relationships.

Intuitively, we picked the following covariates to construct heatmaps: T_Reach, T_Impression, Engaged_Users, Consumers, Consumption, Category, Paid

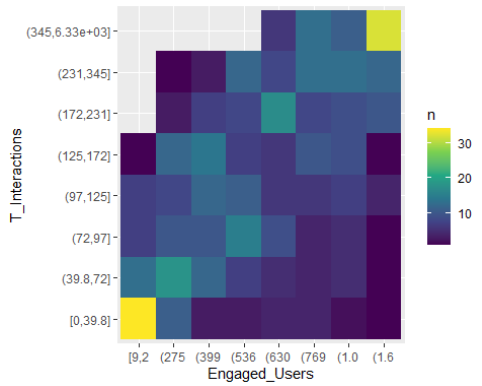
Since T_Interactions includes like, comment, and shares, some metrics such as "LP_Engage_With_Post" does not provide useful predictions for our response variable because they are composed of element from T_Interactions. Other variables such as "Comment" is simply a component of T_Interactions. Thus, we don't consider these either. In other words, these variables are simply a function of T_Interactions.

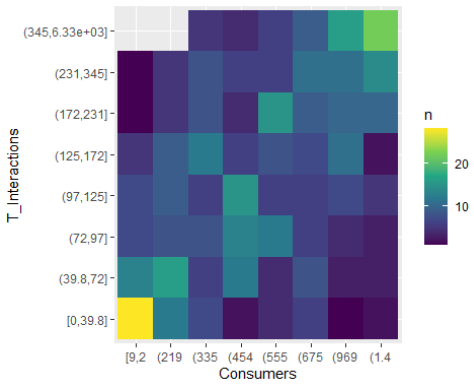
```r
# T_Reach heatmap
fb %>% transmute(T_Reach = cut_number(T_Reach, 8), T_Interactions =
cut_number(T_Interactions, 8)) %>% count(T_Reach, T_Interactions) %>%
ggplot(aes(T_Reach, T_Interactions)) + geom_tile(aes(fill = n)) +
scale_x_discrete(labels = abbreviate) + scale_fill_viridis_c()

# T_Impression heatmap
fb %>% transmute(T_Impression = cut_number(T_Impression, 8),
T_Interactions = cut_number(T_Interactions, 8)) %>% 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()
```



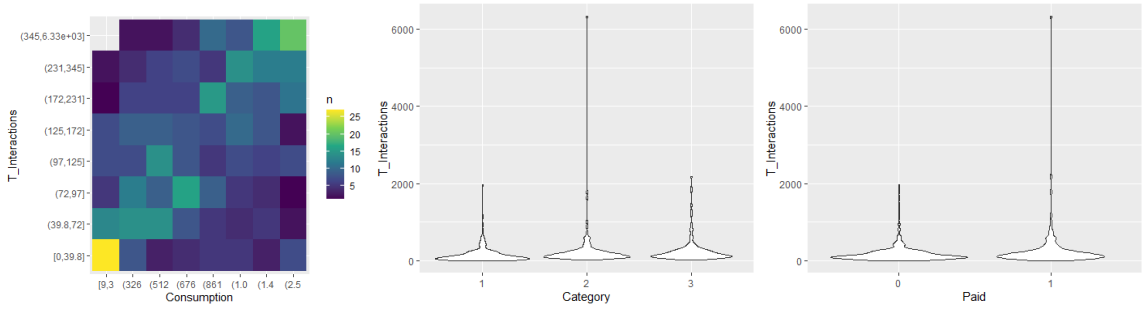



```r
# Engaged_Users heatmap
fb %>% transmute(Engaged_Users = cut_number(Engaged_Users, 8),
T_Interactions = cut_number(T_Interactions, 8)) %>%
count(Engaged_Users, T_Interactions) %>% ggplot(aes(Engaged_Users,
T_Interactions)) + geom_tile(aes(fill = n)) + scale_x_discrete(labels =
abbreviate) + scale_fill_viridis_c()

# Consumers heatmap
fb %>% transmute(Consumers = cut_number(Consumers, 8), T_Interactions =
cut_number(T_Interactions, 8)) %>% count(Consumers, T_Interactions) %>%
ggplot(aes(Consumers, T_Interactions)) + geom_tile(aes(fill = n)) +
scale_x_discrete(labels = abbreviate) + scale_fill_viridis_c()
```





```
# Consumption heatmap
fb %>% transmute(Consumption = cut_number(Consumption, 8),
T_Interactions = cut_number(T_Interactions, 8)) %>% count(Consumption,
T_Interactions) %>% ggplot(aes(Consumption, T_Interactions)) +
geom_tile(aes(fill = n)) + scale_x_discrete(labels = abbreviate) +
scale_fill_viridis_c()

# Category plot
fb %>% ggplot(aes(as.factor(Category), T_Interactions)) + geom_violin()
+ scale_x_discrete(labels = abbreviate) + xlab("Category")

# Paid plot
fb %>% ggplot(aes(as.factor(Paid), T_Interactions)) + geom_violin() +
scale_x_discrete(labels = abbreviate) + xlab("Paid")
```



On the heatmaps, lighter colors correspond to higher number of posts that belongs to the bin with specific predictor variables and T_Interactions. If lighter colors are clustering along the diagonal of the heatmap, then there is potentially a correlation between the two variables. The heatmap for T_Reach, T_Impression, Engaged_Users, and Consumption all show such property. Therefore, they should be considered as covariates. On the other hand, the heatmap for Consumers does not seem to exhibit an obvious linear trend. It should be noted that Consumers and Consumption are very similar metrics. Thus, including both variables might not improve our prediction by a lot. Including more covariates also comes at the cost of less degrees of freedom and potentially lower Adjusted R-Square values. Thus, we remove Consumers from the covariates. Category and Paid are not continuous variables, so we don't construct a heatmap for them. Instead, we make a violin plot to see the density of their distribution. T_Impression doesn't seem to differ much for difference Category and for Paid vs Unpaid posts. However, the tails on each violin plots are very long. We keep these variables for now until scatter matrix.

Removing Outliers

Before proceeding with our analysis, we need to first remove outliers from our data

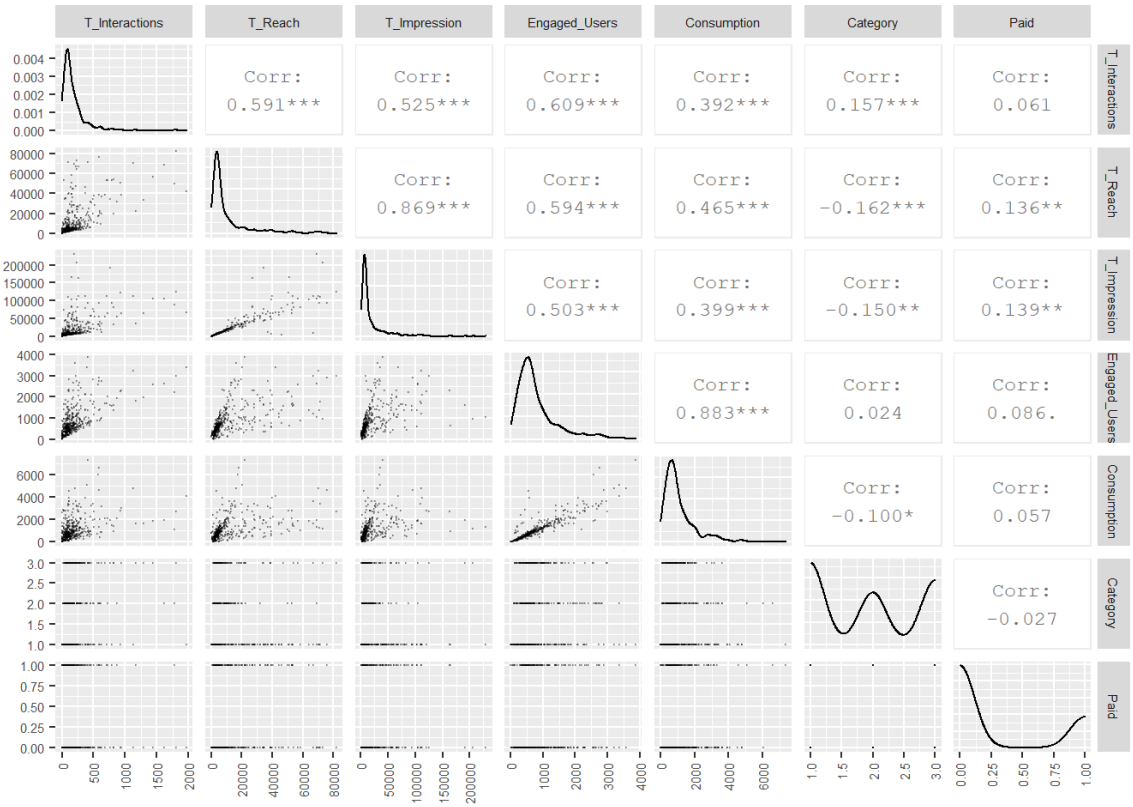
```
# computer mean and sd
reach_mean = mean(fb$T_Reach)
reach_sd = sd(fb$T_Reach)
impress_mean = mean(fb$T_Impression)
impress_sd = sd(fb$T_Impression)
cons_mean = mean(fb$Consumption)
cons_sd = sd(fb$Consumption)
eu_mean = mean(fb$Engaged_Users)
eu_sd = sd(fb$Engaged_Users)

# compute table without outliers beyond 3 standard deviation
fb.clean <- fb %>% filter(T_Reach <= reach_mean+3*reach_sd,
T_Impression <= impress_mean+3*impress_sd, Consumption <=
cons_mean+3*cons_sd, Engaged_Users <= eu_mean+3*eu_sd)

# 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                5.45
```

By including datapoints within 3 standard deviation from the mean for all four variables, we removed 5.45% of the total data points.

Now, we proceed to calculate MLS

```
# scatter matrix
g1 <- fb.clean %>% select(T_Interactions, T_Reach, T_Impression,
Engaged_Users, Consumption, Category, Paid) %>% ggpairs(lower =
list(continuous = wrap("points", alpha = 0.3, size=0.1)), upper =
list(continuous = wrap("cor", size = 3))) + theme(axis.text.x =
element_text(angle = 90, vjust = 0.5, hjust=1))
print(g1 + theme(strip.placement = "outside", text = element_text(size
= 7)))
```



From the scatter matrix, we noticed that there are two pairs of variables who have a very high correlation between them. Namely, T_Reach and T_Impression has a correlation factor of 0.869; Engaged_Users and Consumption has a correlation factor of 0.883. These can also be seen from the linear trend that's present in the scatter plots for these two pairs of variable. Based on the scatter matrix, we decided to remove one variable from each pair of strongly correlated variable. We choose to remove the one that has a lower correlation to our response variable T_Interactions. Since T_Reach and Engaged_Users have a higher correlation to T_Impression in each respective pairs, we remove T_Impression and Consumption from the MLR. Notice that Paid is a categorical variable that could affect the intercept. However, it's correlation is extremely low. Thus, we remove Paid as well. We choose to keep Category because it has a significant correlation with T_Interactions (from the *** next to the correlation coefficient). Our model would be better after removing these variable because they do not provide much extra information, and we would suffer from losing degrees of freedom as well as a potential for lower R-Square values.

Performing MLS

```
m.mls <- lm(T_Interactions ~ T_Reach + Engaged_Users +
as.factor(Category), data = fb.clean)
summary(m.mls)

##
## Call:
## lm(formula = T_Interactions ~ T_Reach + Engaged_Users +
as.factor(Category),
##     data = fb.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -683.95  -73.43   -1.08   51.57 1305.52
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -5.004e+01  1.559e+01  -3.210  0.00142 **
## T_Reach               6.939e-03  7.023e-04   9.879  < 2e-16 ***
## Engaged_Users         1.268e-01  1.562e-02   8.116 4.38e-15 ***
## as.factor(Category)2  8.704e+01  2.077e+01   4.190 3.34e-05 ***
## as.factor(Category)3  1.202e+02  1.890e+01   6.362 4.78e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 170.3 on 463 degrees of freedom
## Multiple R-squared:  0.4984, Adjusted R-squared:  0.494
## F-statistic:   115 on 4 and 463 DF,  p-value: < 2.2e-16

coef(m.mls)

##          (Intercept)              T_Reach        Engaged_Users
##        -50.044717045          0.006938575          0.126770682
## as.factor(Category)2 as.factor(Category)3
##         87.035268104        120.223631544
```

Based on the output of MLS, our model predicts the data moderately well with Adjusted R-Squared of 0.494. It is not a high value, but we do see significance on all coefficients. We are least confident about the intercept, which represent "Category 1" since Category is a factor. Its t values is -3.210 (with associated p values of 0.00142), which allows us to reject the null hypothesis of Intercept = 0 at 99.9% confidence level. Notice that the estimate for Intercept is a negative value, which does not make sense because a post cannot have negative total interactions. Thus, we can say that our model does not approximate the intercept very well.

Both T_Reach and Engaged_Users have high t-values (9.879 and 8.116 respectively), allowing us to reject the null at 99.99% confidence level. Their p values are order of more than one magnitude smaller than 1, we could even reject at 99.9999%. The standard errors and estimates for these two variables differ by an order of one

magnitude, which indicates we have small errors on these variables relative to their actual values.

We are also very confident with the other two categorical predictors. Notice that the null hypothesis for both of them is to test whether they differ from the intercept of category 1. It is not a test for whether the intercepts are zero or not. As we can see from the relative high t values (4.190 and 6.362) and the very low p values (3.34e-05 and 4.78e-10), we can safely say at 99.99% confidence that category differences do affect T_Interactions.

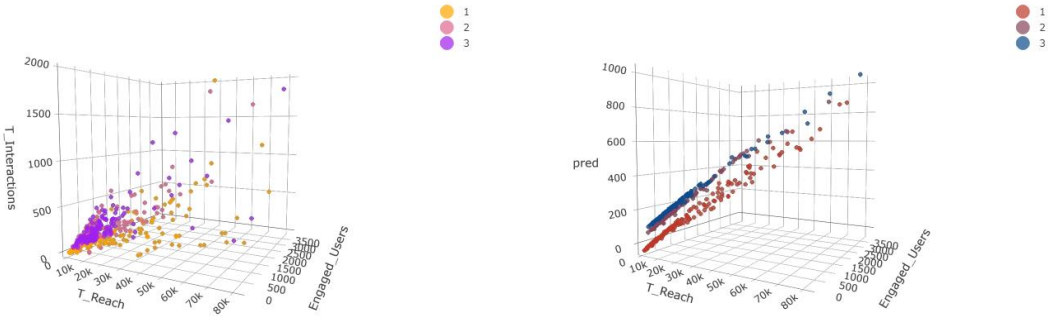
Based on the t values and p values, we can say that these predictors are definitely useful for inferring values of T_Interactions. However, we could still improve on our model to increase the R-Square value. The equation of the mls prediction is

T_Interactions = -50.044717045 + 0.006938575(T_Reach) + 0.126770682(Engaged_Users) + 87.035268104(Category2) + 120.223631544 (Category3)

3D Model

We created graphs to visualize the actual data against our predicted data.

```
# this portion constructs the two 3D plots.
fb.clean <- fb.clean %>% add_predictions(m.mls)
fb.clean %>% plot_ly(x = ~T_Reach, y = ~Engaged_Users, z =
~T_Interactions, color = ~as.factor(Category), colors = c('Orange',
'Purple')) %>% add_markers(size = 5)
fb.clean %>% plot_ly(x = ~T_Reach, y = ~Engaged_Users, z = ~pred, color
= ~as.factor(Category), colors = c('#BF382A', '#0C4B8E')) %>%
add_markers(size = 5)
```



Here are the two 3D plots for our datasets. The one on the left is from actual values of T_Interactions, whereas the one on the right has predicted values for T_Interactions based on MLS. Visually, our model overestimates for higher (x,y) value pairs. On the predicted model, we could see different layers of surface corresponding to different intercepts. Currently, the surface appear to be linear (since we performed multiple linear regression). Based on the scatter points, a quadratic or higher approximation could produce a better fit for the model.

Interaction Terms

One way to improve our model from regular mls is to account for interaction terms.

From our scatter matrix previously, we see that T_Reach and Engaged_Users are somewhat correlated (with correlation coefficient 0.594). We attempt to take into account potential interactions between these two terms (since they might not be completely independent). We want to use SSR from the "Engaged_Users ~ T_Reach" model to predict the SSR of the mls model.

```
m.mls <- lm(T_Interactions ~ T_Reach + Engaged_Users +
as.factor(Category) + T_Reach:Engaged_Users, data = fb.clean)
summary(m.mls)
```

```
##
## Call:
## lm(formula = T_Interactions ~ T_Reach + Engaged_Users +
as.factor(Category) +
##     T_Reach:Engaged_Users, data = fb.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1040.18   -70.08   -18.14    40.04  1096.82
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.723e+01  1.724e+01   1.579   0.1149
## T_Reach                -3.144e-04  1.088e-03  -0.289   0.7728
## Engaged_Users           3.774e-02  1.806e-02   2.090   0.0372 *
## as.factor(Category)2    8.874e+01  1.938e+01   4.578 6.04e-06 ***
## as.factor(Category)3    1.165e+02  1.764e+01   6.605 1.09e-10 ***
## T_Reach:Engaged_Users   5.202e-06  6.230e-07   8.349 8.07e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 158.9 on 462 degrees of freedom
## Multiple R-squared:  0.5641, Adjusted R-squared:  0.5594
## F-statistic: 119.6 on 5 and 462 DF,  p-value: < 2.2e-16
```

```
coef(m.mls)
```

```
##           (Intercept)               T_Reach         Engaged_Users
##          2.723438e+01         -3.143857e-04          3.774444e-02
##  as.factor(Category)2  as.factor(Category)3 T_Reach:Engaged_Users
##          8.874017e+01          1.165077e+02          5.201554e-06
```

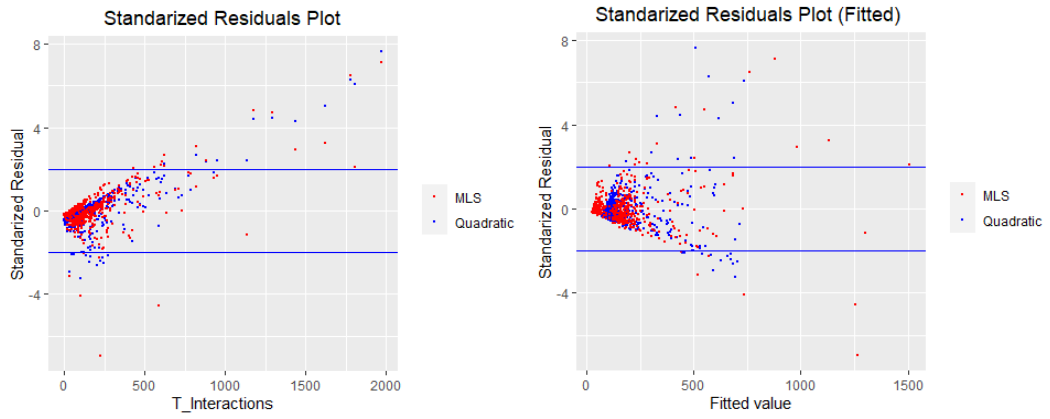
From this output, we see that we can no longer reject the null hypothesis for both the Intercept and T_Reach. In fact, the p value for T_Reach is extremely large in this model, and we lost confidence on Engaged_Users as well. We do have confidence in the interaction term as well as a higher R-Square value, but this model is not useful

due to its lack of confidence for T_Reach and Engaged_Users as predictors. Interpretations on this model coefficients are similar to that of the non-interaction scenario.

Residuals

Now we construct plots for residuals on mls

```
# first build residual from quadratic model to compare against mls
m.qls <- fb.clean %>% lm(T_Interactions ~ T_Reach + I(T_Reach^2), .)
StanResQLS <- rstandard(m.qls)

# standardize for mls and plotting
StanResMLS <- rstandard(m.mls)
fb.clean %>% ggplot() + geom_point(aes(T_Interactions, StanResQLS,
color = "Quadratic"), size = 0.1) +
geom_point(aes(T_Interactions, StanResMLS, color = "MLS"), size = 0.1)
+
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2,
color='blue') +
scale_color_manual(name = element_blank(), labels =
c("MLS","Quadratic"), values = c("red","blue")) +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals
Plot")
```

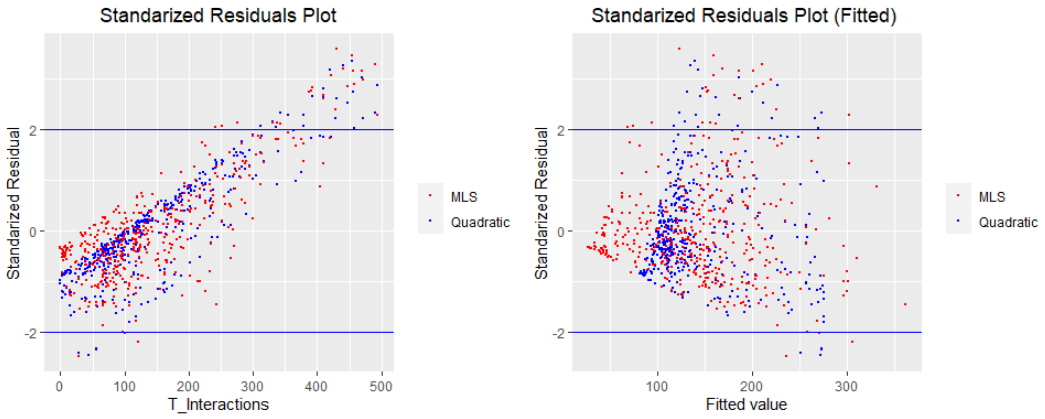



```
# fitted residuals
qfit = fitted(m.qls)
mfit = fitted(m.mls)

fb.clean %>% ggplot() + geom_point(aes(qfit, StanResQLS, color =
"Quadratic"), size = 0.1) + geom_point(aes(mfit, StanResMLS, color =
"MLS"), size = 0.1) +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2,
color='blue') +
scale_color_manual(name = element_blank(), labels =
c("MLS","Quadratic"), values = c("red","blue")) +
```

```
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals Plot (Fitted) ")
```

It is very difficult to visualize pattern in the residual due to the domain of our response variables. There seems to be a linear trend leading to larger values of T_Impression. To make our analysis better, consider only T_Interactions below 500 for now.

```
# first build residual from quadratic model to compare against mls
m.qls1 <- fb.clean %>% filter(T_Interactions <= 500) %>%
lm(T_Interactions ~ T_Reach + I(T_Reach^2), .)
StanResQLS1 <- rstandard(m.qls1)

# standardize for mls and plotting
m.mls1 <- fb.clean %>% filter(T_Interactions <= 500) %>%
lm(T_Interactions ~ T_Reach + Engaged_Users + as.factor(Category) +
T_Reach:Engaged_Users, .)
StanResMLS1 <- rstandard(m.mls1)
fb.clean %>% filter(T_Interactions <= 500) %>% ggplot() +
geom_point(aes(T_Interactions, StanResQLS1, color = "Quadratic"), size
= 0.1) + geom_point(aes(T_Interactions, StanResMLS1, color = "MLS"),
size = 0.1) +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2,
color='blue') +
scale_color_manual(name = element_blank(), labels =
c("MLS","Quadratic"), values = c("red","blue")) +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals
Plot")
```




```
# fitted residuals
qfit1 = fitted(m.qls1)
mfit1 = fitted(m.mls1)

fb.clean %>% filter(T_Interactions <= 500) %>% ggplot() +
geom_point(aes(qfit1, StanResQLS1, color = "Quadratic"), size = 0.1) +
geom_point(aes(mfit1, StanResMLS1, color = "MLS"), size = 0.1) +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2,
```

```
color='blue') +
scale_color_manual(name = element_blank(), labels =
c("MLS","Quadratic"), values = c("red","blue")) +
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals Plot (Fitted) ")
```

These two plots show a lot more detail than the previous ones. On the standardized residual plot, we see a linear trend in both models, indicating that error increases as T_Interactions increases. We don't have good predictions for larger values of T_Impression. Since the linearity is a positive trend, it means that we tend to overestimate for larger value of T_Impression. The majority of residuals roughly lie within the horizontal band, and there is no trend for unequal variances (as the graph doesn't form a cone). Even though residuals in both models exhibit linearity, MLS model is actually better than the quadratic model. This is because the linear trend is more scattered compared to the very concentrated line in the quadratic form. MLS residuals distributes itself evenly on both sides of the quadratic residuals. The fitted model has a very nice-looking residual plot. MLS model does not exhibit any linear or conical trend, and it is mostly distributed within the horizontal bands. It can be seen from our 3D plot that we indeed have linearity in prediction. Fitted residuals confirms that the errors are quite acceptable.