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

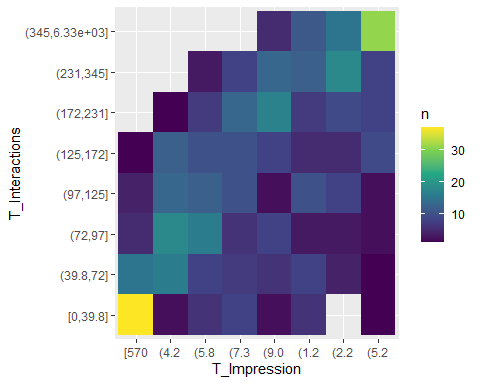
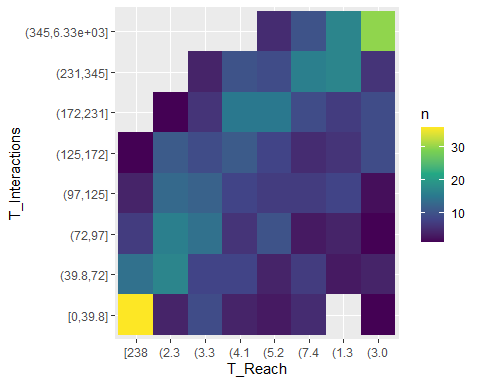
Using T\_Interactions as the response variable, we construct heatmaps for our potential covariates to identify meaningful relationships.

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 in T\_Interactions. In other words, they are simply a function of T\_Interactions.

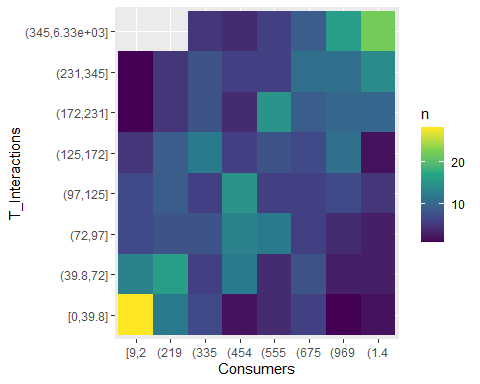
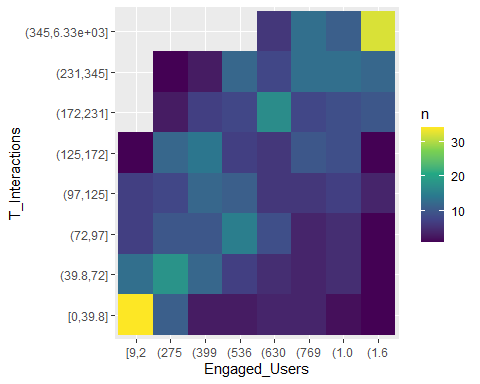
# T\_Reach heatmap  
fb %>% transmute(T\_Reach = cut\_number(T\_Reach, 8), T\_Interactions = cut\_number(T\_Interactions, 8)) %>% count(T\_Reach, T\_Interactions) %>% ggplot(aes(T\_Reach, T\_Interactions)) + geom\_tile(aes(fill = n)) + scale\_x\_discrete(labels = abbreviate) + scale\_fill\_viridis\_c()

# 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()



# 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()

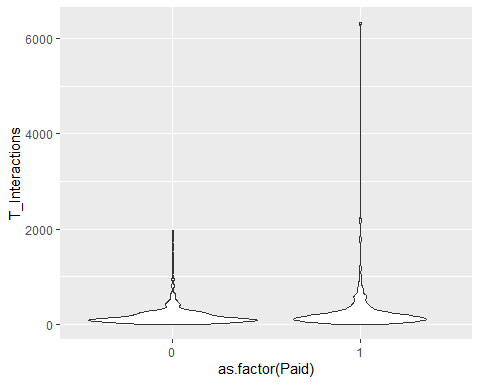
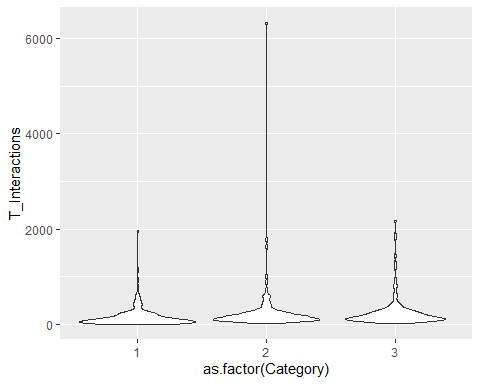
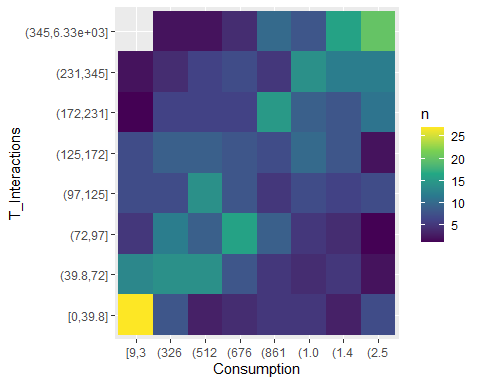
# Consumers heatmap  
fb %>% transmute(Consumers = cut\_number(Consumers, 8), T\_Interactions = cut\_number(T\_Interactions, 8)) %>% count(Consumers, T\_Interactions) %>% ggplot(aes(Consumers, T\_Interactions)) + geom\_tile(aes(fill = n)) + scale\_x\_discrete(labels = abbreviate) + scale\_fill\_viridis\_c()



# 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)

# Paid plot  
fb %>% ggplot(aes(as.factor(Paid), T\_Interactions)) + geom\_violin() + scale\_x\_discrete(labels = abbreviate)



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 differences. 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)

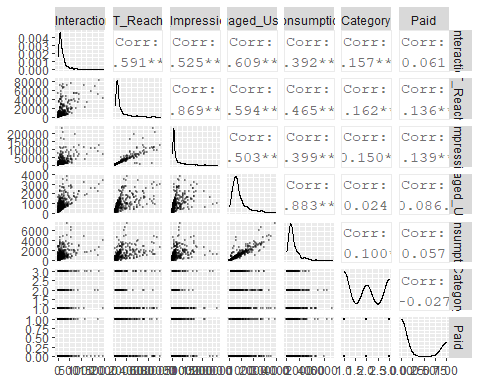
## # A tibble: 1 x 2  
## Variable Percentage\_Removed  
## <chr> <dbl>  
## 1 Overall 5.45

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

## Scatter Matrix

Now, we proceed to calculate MLS

# scatter matrix  
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)))



From the scatter matrix, we noticed that there are two pairs of variable 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)

图表, 散点图

描述已自动生成图表, 散点图

描述已自动生成

Here are the two 3D plots for our datasets. The one above is from actual values of T\_Interactions, whereas the one below 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

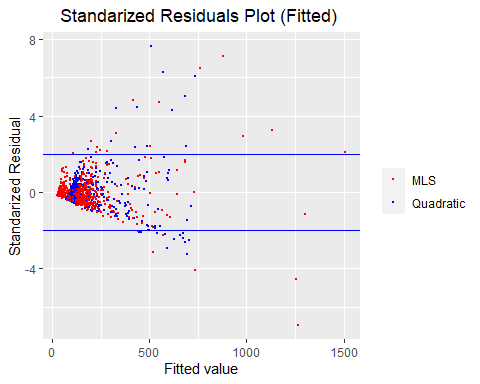
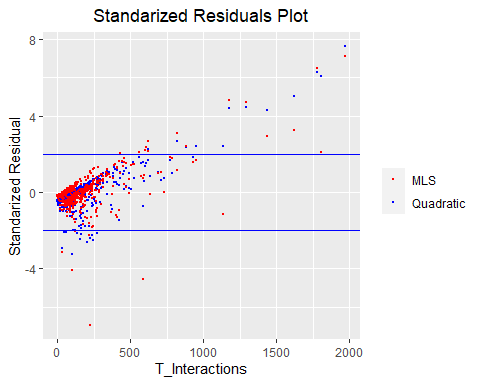
From this output, we see that we can no longer reject the null hypothesis for both the Intercept and T\_Reach. In fact, the p value for T\_Reach is extremely large in this model, and we lost confidence on Engaged\_Users as well. We do have confidence in the interaction term as well as a higher R-Square value, but this model is not useful due to its lack of confidence for T\_Reach and Engaged\_Users as predictors.

## 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")

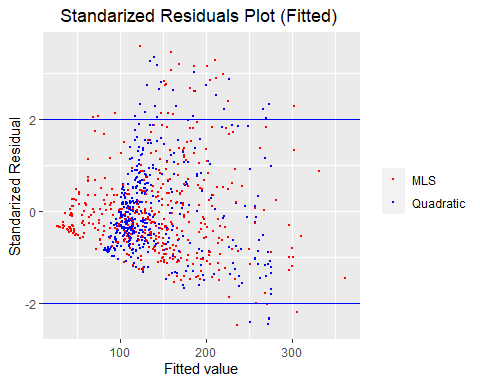
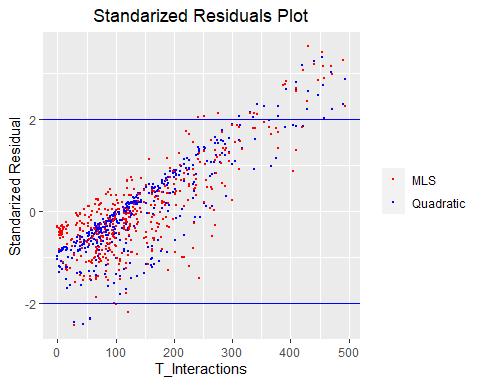
# fitted residuals  
qfit = fitted(m.qls)  
mfit = fitted(m.mls)  
  
fb.clean %>% ggplot() + geom\_point(aes(qfit, StanResQLS, color = "Quadratic"), size = 0.1) + geom\_point(aes(mfit, StanResMLS, color = "MLS"), size = 0.1) +  
geom\_hline(yintercept=2,color='blue') + geom\_hline(yintercept=-2, color='blue') +  
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 shows 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.