Facebook Comments

Final Project: Analysis on Metrics of Facebook Comment Data

-

Jason Yu (T 3:00 - 3:50 PM)

Brandon Lee (T 5:00 - 5:50 PM)

6.7.20

PSTAT 126: Regression Analysis

1. **Introduction**

The main goal of our project will be to study the number of comments per hour (***comments/hr***) on a post from a Facebook page given the variables in the "Facebook Comment Volume" data set from the UC Irvine Machine Learning Repository that initially contained 100 observations. More specifically, we will be investigating whether or not our response variable, which is the number of comments per hour on a Facebook page, can be predicted by the five following possible predictors: ***page popularity*** which represents the number of likes the page received; ***page check-ins*** which represents the number of individuals that have visited the page so far; ***page talking about*** which represents the number of people that actually come back to the page after liking it; ***post length*** which represents the character count of the page post; and ***post share count*** which represents the number of times a person shared the page post to their timeline. After finding the coefficients of the final predictive model, we will also investigate whether or not the number of comments per hour is significantly linearly related to the number of people that actually come back to a page after liking it. Lastly, we will also find the expected number of comments per hour for a page with average values in the predictors of our final model as well as a 95% confidence interval for said prediction.

2. Questions of Interest

1. What are the coefficients of our final model?
2. Can a post's number of comments per hour be predicted by at least one of our selected variables?
3. Is the number of comments per hour significantly linearly related to the number of people who actually come back to the page after liking it (page talking about)?
4. What is the expected number of comments per hour for a page with average values in the predictor values of our final model? What is the appropriate 95% confidence interval for this prediction?

3. Regression Method

In order to answer our questions of interest, we need to construct a model that meets all four LINE conditions.

We first look at the overall summary of the data and notice two significant outliers in the Post Share Count column. The median and average of the column are 20.50 and 91 respectively. Thus, we consider the two values of 2094 and 1074 to be outliers and remove them. After

finding the initial set of outliers, we find and remove additional outliers by using the rstandard() function in R to compute the internally studentized residuals and determining which of these observations has a studentized residual greater than 3 or less than -3 to remove them. In all, we remove a total of 4 outliers.

We begin by finding the best predictors for the model via stepwise regression using F-tests. This will ensure we have the best model for predicting our targeted response, the number of comments per hour.

We check to see if any interaction terms are needed by using the general linear F-test.

We then use residual analysis to determine whether transformations to the response or predictors are required to meet the LINE assumptions. We draw the residual vs fitted plot to check for the linearity and equal variance, the individual residual vs predictor plots to check each predictor's linearity and variance, and the normal Q-Q plot to check for normality. After the checks, we discover that the LINE assumptions were not met. We transform the response by using the boxcox() function in R to find the correct transformation, then transform the predictors by using the log transformation. After applying all of the transformations, we check the conditions again and discover that all four LINE assumptions are met.

To answer the first question of interest, we use the summary() function in R to obtain the model's coefficients.

For question 2, we use the general linear F-test to test if both slope parameters in our model are zero. We can also use the model's overall p-value as an indicator for how well the model fits our data.

For question 3, we again use the general linear F-test. We are interested in testing whether the slope parameter for one of the predictors (page talking about) is equal to zero.

For question 4, we construct a data frame that contains the average values of the predictors in our final model and use the predict() function in R to find both the expected number of comments per hour for a page with average values (in page popularity and page talking about) and the 95% confidence interval for that prediction. After finding this prediction, we perform the opposite transformation on it to get the original units.

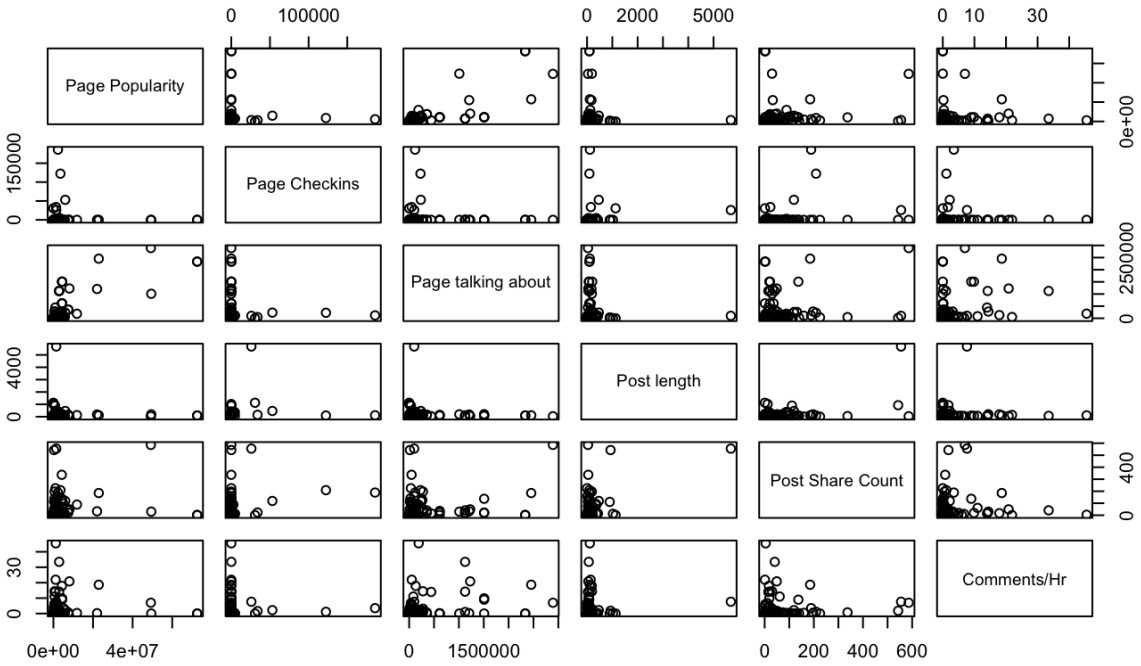
4. Regression Analysis, Results, and Interpretation

We start the analysis aspect of our project by investigating the relationships between our targeted response, the number of comments per hour, and the following predictors:

Y = Comments/Hr
X1 = Page Popularity
X2 = Page Checkins
X3 = Page talking about
X4 = Post Length
X5 = Post Share Count

We discussed including an additional categorical variable, X6 = Page Category, but decided against it since the variable contained 54 different values and thus adding it to the model did not make sense.

We begin investigating the relationships between the response and each predictor through constructing a scatterplot matrix by using the pairs() function in R to plot the response against each predictor of interest. We observe no linear trends but do see a logarithmic trend for most of the predictors.



Performing a stepwise regression using the F-test will help us obtain the “best” model for predicting our response of interest, comments/hr. Through this process, we first add page talking about as the first predictor as it has the lowest p-value. Then, we add page popularity as the second predictor as it has the lowest p-value remaining. To make sure page popularity does not affect the significance we check the summary() of the model with just these two predictors. As

the model looks good with low p-values, we proceed to check the p-values of the remaining predictors. The rest of the predictors have p-values greater than 0.61 so we stop here.

```
Call:
lm(formula = comments ~ talkabt + popularity)

Coefficients:
(Intercept)      talkabt   popularity
  2.088e+00    7.957e-06   -2.878e-07
```

From the results of the stepwise regression, we can say the regression model using page talking about and page popularity does the best job at predicting comments/hr.

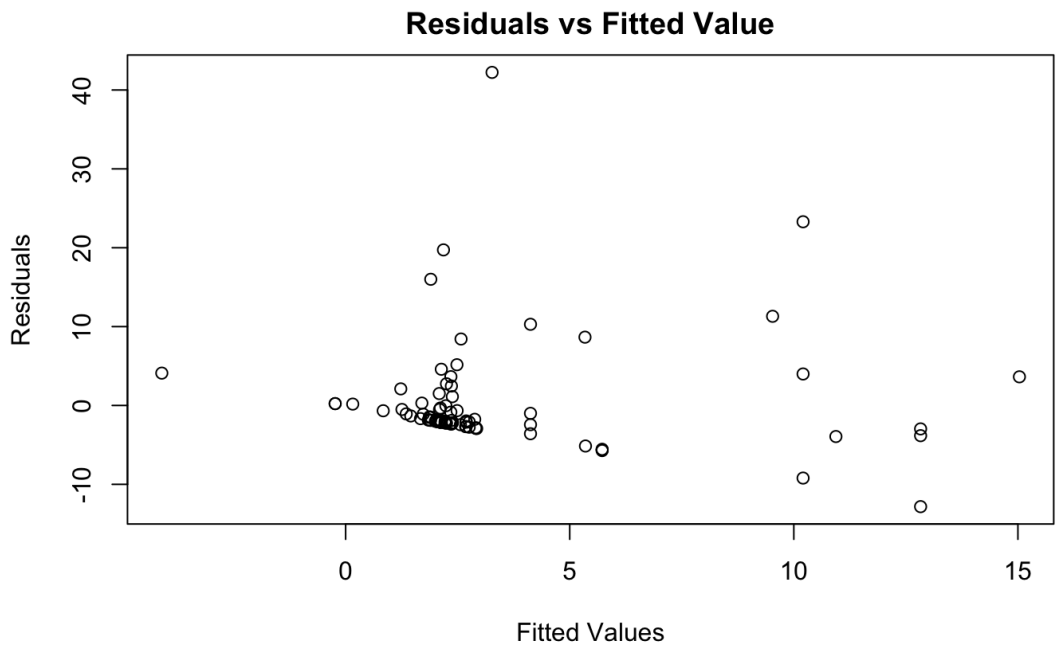
We check to see if an interaction term is needed by using the general linear F-test. We set the full model as the model with the interaction term added and the reduced model as the model without the interaction term. After running anova() on the two models, we obtain a p-value of 0.6794 and fail to reject the reduced model which has no interaction term.

````
```{r}
model.full = lm(comments ~ talkabt + popularity + talkabt*popularity)
model.reduced = lm(comments ~ talkabt + popularity)
anova(model.reduced, model.full)
```
````

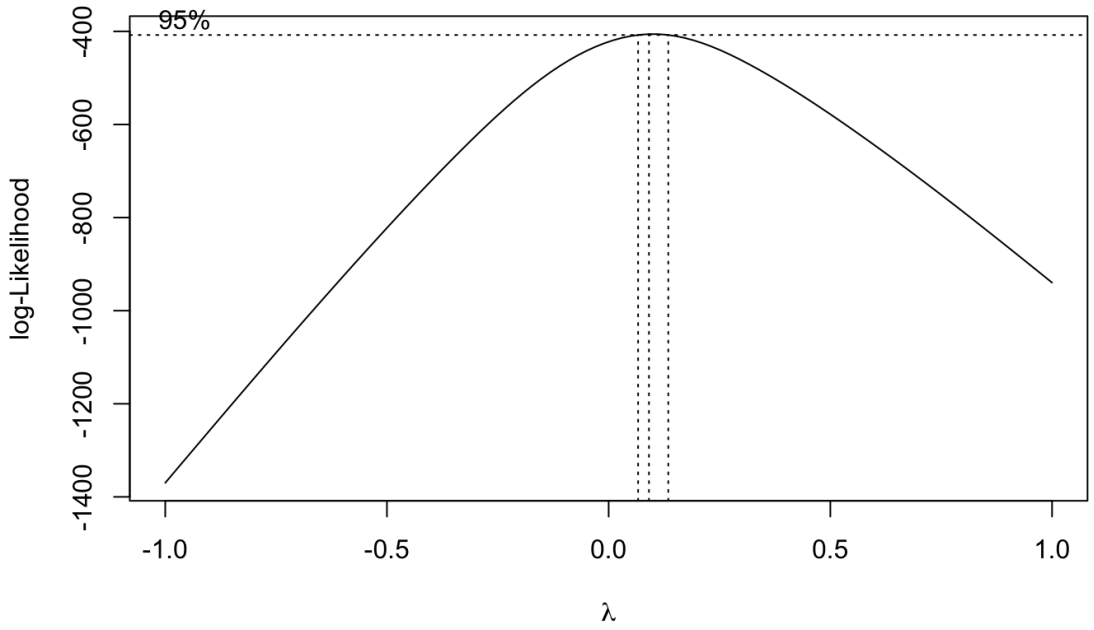
```
Analysis of Variance Table

Model 1: comments ~ talkabt + popularity
Model 2: comments ~ talkabt + popularity + talkabt * popularity
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     93 4146.3
2     92 4138.5  1    7.7344 0.1719 0.6794
```

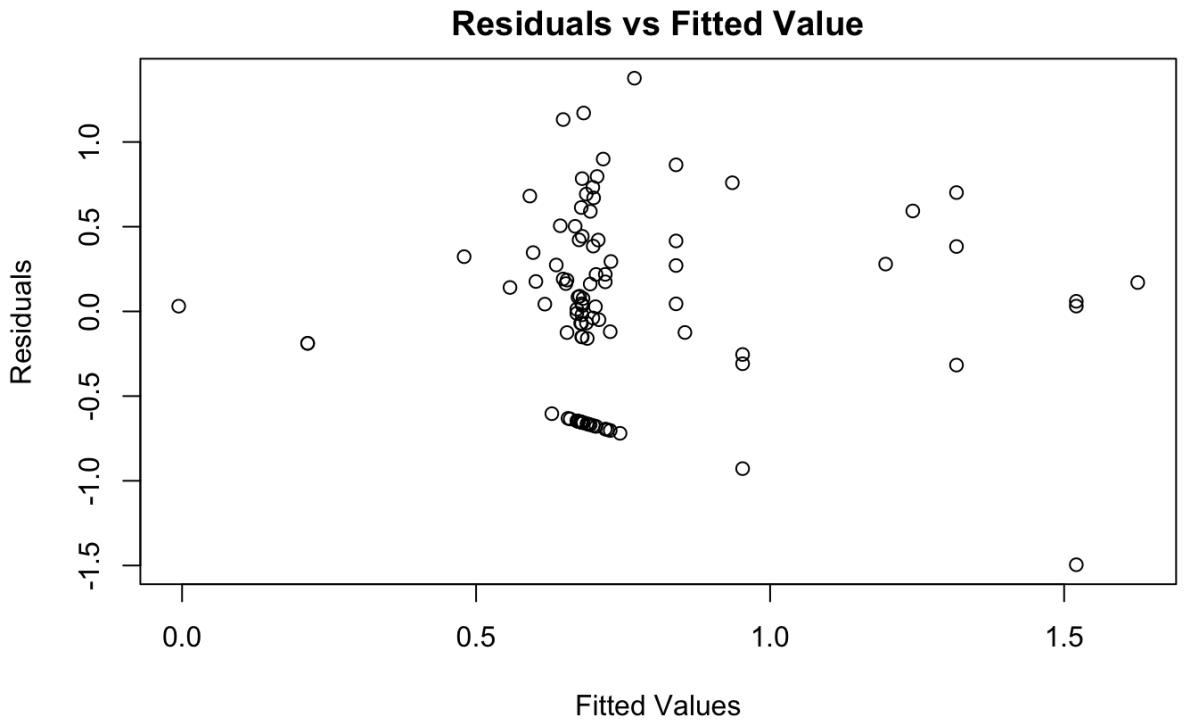
After obtaining the best model, we check our LINE assumptions by first looking at the residual vs fitted values plot. The plot does not look “well-behaved” as the points are very clustered and are not equally distributed.



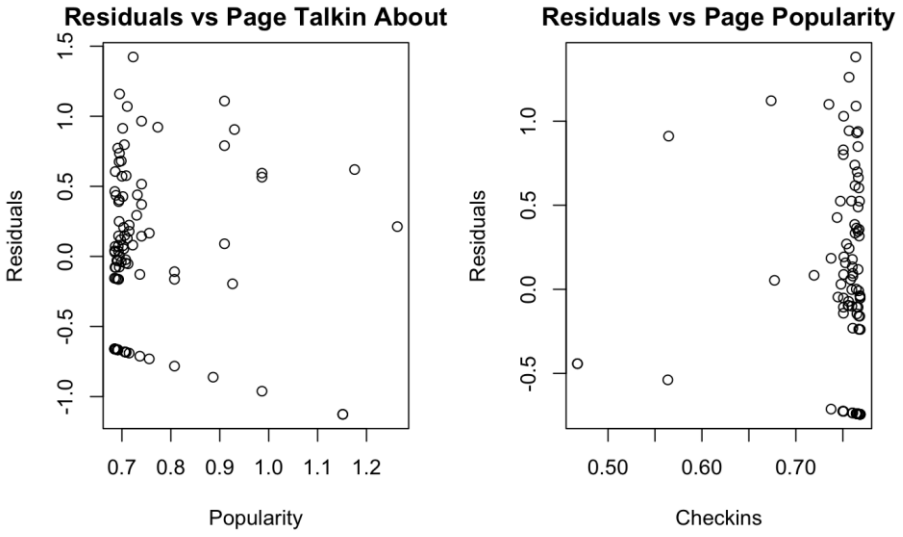
We run a boxcox() transformation on our model to find an appropriate transformation for our response variable. From the Box-Cox graph, we observe lambda to be approximately 0.2 and transform our response as (comments/hr)^0.2.



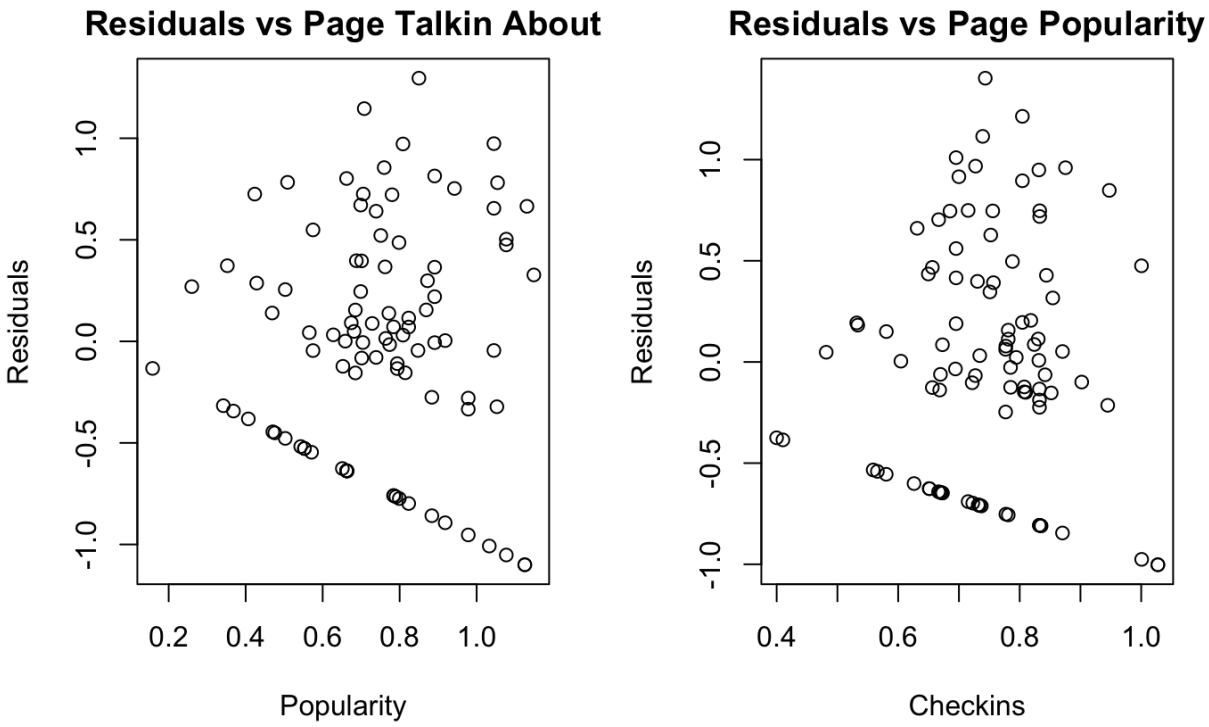
Upon checking the residual vs fitted value plot, the points seem more spread out but are still not equally distributed.



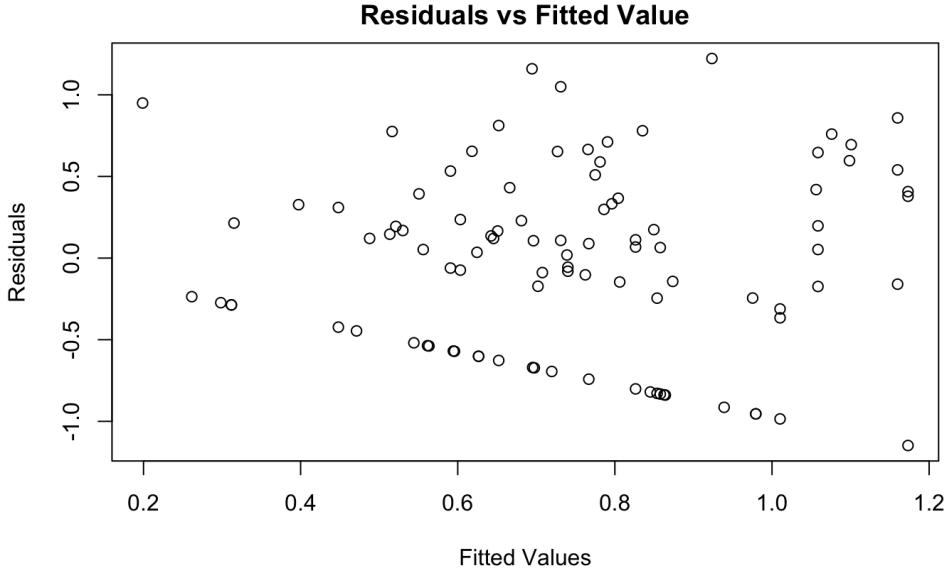
Due to the residual vs fitted value plot still being not "well-behaved", we look to transform the predictors. We plot the individual residual vs predictor plots and see that both plots are not "well-behaved". In both plots, most points are clustering around one side.



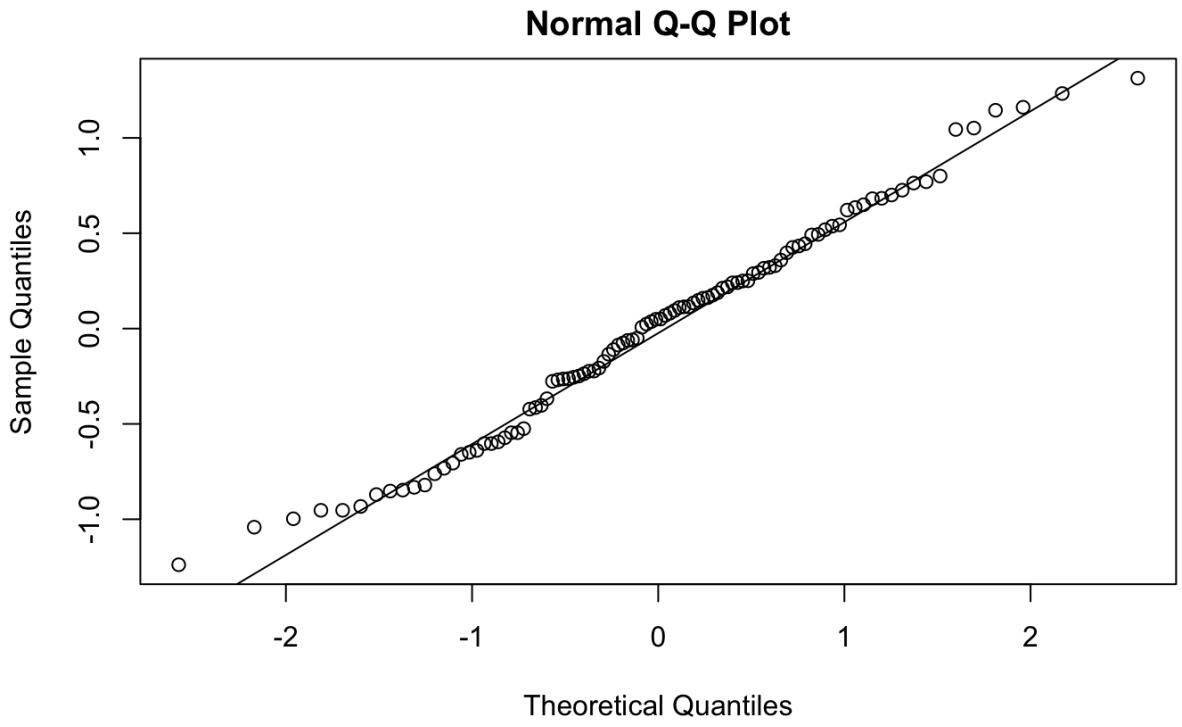
We apply log transformations to both predictors which helps significantly with the clustering and both plots look well-behaved now.



We now look again at the residual vs fitted values plot with all of the transformations we performed and confirm that the residuals look evenly spread out around zero.



Finally, we look at the Q-Q plot to check our transformed model for the normality assumption. All points are close to the line so we can assume normality is met.



After applying the transformations, we decide that our final model meets all four LINE assumptions. Our final model is lm((comments/hr)^0.2 ~ log(page talking about) + log(page popularity).

Question of Interest #1

We compute the coefficients by using the summary() function on the final model. We can now construct the linear regression line which is Y = 0.07223 + 0.18196*x1 - 0.09729*x2. The coefficients are 0.07223 for beta0, 0.18196 for beta1, and -0.09729 for beta2. This means that for every unit increase in log(talkabt), we can expect an increase of 0.18196 increase in (comments/hr)^2. Likewise, for every unit increase in log(popularity), we can expect an increase of 0.18196 increase in (comments/hr)^2.

```
Call:
lm(formula = comments^0.2 ~ log(talkabt) + log(popularity))

Residuals:
     Min       1Q   Median       3Q      Max
-1.26309 -0.40071  0.04754  0.36595  1.29615

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.14312    0.50850   0.281   0.7790
log(talkabt)     0.22812    0.05406   4.219  5.6e-05 ***
log(popularity) -0.13734    0.06095  -2.253   0.0265 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5831 on 95 degrees of freedom
Multiple R-squared:  0.1879,    Adjusted R-squared:  0.1708
F-statistic: 10.99 on 2 and 95 DF,  p-value: 5.08e-05
```

Question of Interest #2

From the summary table of our final regression model, we can see that 17.25% of all variation of (comments/hr)^0.2 is explained after taking into account log(page talkin about) and log(page popularity). In addition, the p-value for the model is 0.0001501 which is very low and indicates that the model fits the data well. To test the hypothesis from question 2 of whether “comments per hour” can be accurately predicted by at least one of our predictors, we will use the general linear F-test. The null and alternative hypothesis are as follows:
H0: beta1 = beta2 = 0
H1: At least one beta_j not equal to 0, where j = (1,2)
Alpha = 0.05

We set the reduced model to be lm((comments)^0.2 ~ 1) and the full model to be lm((comments)^0.2 ~ log(talkabt) + log(popularity)). We compare these two using the anova() function and obtain the F-Statistic and p-value for our hypothesis test.
F-Statistic = 7.8878
p-value = 0.0006851

```r
mod.full2 = lm((comments)^0.2 ~ log(talkabt) + log(popularity))
mod.reduced2 = lm((comments)^0.2 ~ 1)
anova(mod.reduced2, mod.full2)
```

```
Analysis of Variance Table

Model 1: (comments)^0.2 ~ 1
Model 2: (comments)^0.2 ~ log(talkabt) + log(popularity)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     95 33.751
2     93 28.856  2    4.8949 7.8878 0.0006851 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Since the p-value is less than alpha, we reject the null hypothesis. There is sufficient evidence to suggest that at least one of the slope parameters is not equal to 0 and one of our selected predictors is useful in predicting comments/hr. This means that either page talking about or page popularity could be a useful predictor for predicting the number of comments per hour a post gets.

Question of Interest #3

Again, we use the general linear F-test to answer the research question of "Is the number of comments per hour significantly linearly related to the number of people who actually come back to the page after liking it (page talking about)?". The null and alternative hypothesis are:
H0: beta1 = 0
H1: beta1 is not equal to 0
Alpha = 0.05
We assign the reduced model as lm((comments)^0.2 ~ log(popularity)) and the full model as lm((comments)^0.2 ~ log(talkabt) + log(popularity)). We compare these two using the anova() function and obtain the F-Statistic and p-value for our hypothesis test.
F-Statistic = 11.579
P-Value = 0.0009855

```{r}
mod.reduced = lm((comments)^0.2 ~ log(popularity))
mod.full = lm((comments)^0.2 ~ log(talkabt) + log(popularity))
anova(mod.reduced, mod.full)
```

```
Analysis of Variance Table

Model 1: (comments)^0.2 ~ log(popularity)
Model 2: (comments)^0.2 ~ log(talkabt) + log(popularity)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     94 32.449
2     93 28.856  1    3.5926 11.579 0.0009855 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our p-value is less than 0.05 so we reject the null hypothesis and conclude that page talking about is significantly linearly related to the number of comments per hour. This means the log(page talking about) is indeed a useful predictor for predicting our response, comments/hr.

Question of Interest #4

To answer the research questions “What is the expected number of comments per hour for a page with average values in the predictor values of our final model?” and “What is the appropriate 95% confidence interval for this prediction?,” we first construct a data frame that contains the average values for both page talking about and page popularity by using a combination of the data.frame() and mean() functions in R. We then use the predict() function with our final model and the data frame containing the average value while specifying that we want a 95% confidence interval. Lastly, we perform opposite transformations on the prediction and interval values to get these values in their original units by raising them to the power of 1/0.2. Thus, for a Facebook page with an average number of page likes and an average number of people that actually return to the page after liking it, we find that it is expected that a post on an average page will receive approximately 0.5195538 comments/hr with a 95% confidence interval of (0.2087124, 1.123286).

5. Conclusion

Based on our analysis, we find that the number of comments posted per hour on a Facebook page post can in fact be predicted by at least one of the possible predictors. We are also confident that a Facebook page’s popularity and the number of people that actually come back to the page after liking it play a significant role in predicting a post’s number of comments per hour. On the other hand, we correspondingly conclude that the total number of check-ins, post length, and share count do not significantly affect the number of comments per hour on a post. Lastly, we predict that there will be approximately 0.5195538 comments per hour on a Facebook post with an average amount of page likes and an average number of people that return to the page after liking it.

One thing to consider about our project is that our final model could have possibly been improved if we were able to have found and implemented more possible predictors that were not included in the data set, such as the age of the page and the age demographic of users who like the page. All sorts of data are collected for a platform as large as Facebook, so it is reasonable to expect there to be numerous predictors not in the data set that could significantly affect the number of comments per hour. The next thing to consider is that our findings can be generalized to the majority of posts on Facebook pages since our study sample was randomly selected and is representative of the wider population of Facebook page posts. Last but not least, the main insight to take away from our findings is that it seems that the characteristics of the page of a

post are much more significant than the characteristics of the post itself in predicting the number of comments per hour that post receives. This suggests the possibility of people being more inclined to interact with posts from pages that they deem notably popular rather than solely considering the content of a post, which is certainly an interesting thing to consider.

6. Appendix

```
# Import packages
library(tidyverse)
library(MASS)
library(dplyr)

# Read in data and format column names
setwd("/Users/jason/Desktop/PSTAT 126 - Bapat/Project")
data = read.csv(file = 'Test_Case_1.csv', header = F)


cols = c("Page Popularity", "Page Checkins", "Page talking about", "Page Category",
        "C1_0", "C1_1", "C1_2", "C1_3", "C1_4",
        "C2_0", "C2_1", "C2_2", "C2_3", "C2_4",
        "C3_0", "C3_1", "C3_2", "C3_3", "C3_4",
        "C4_0", "C4_1", "C4_2", "C4_3", "C4_4",
        "C5_0", "C5_1", "C5_2", "C5_3", "C5_4",
        "CC1", "CC2", "CC3", "CC4", "CC5",
        "Base time", "Post length", "Post Share Count", "Post Promotion Status", "H Local",
        "Pub0", "Pub1", "Pub2", "Pub3", "Pub4", "Pub5", "Pub6",
        "Dt0", "Dt1", "Dt2", "Dt3", "Dt4", "Dt5", "Dt6",
        "Target Variable")
colnames(data) = cols

# Subset predictors of interest and remove

vars = c("Page Popularity", "Page Checkins", "Page talking about",
        "Post length", "Post Share Count","H Local", "Target Variable")

metrics = data[vars]

# Normalize Comments
metrics["Comments/Hr"] = metrics["Target Variable"]/metrics["H Local"]
```

```
# Remove H Local and Target Variable
vars = c("Page Popularity", "Page Checkins", "Page talking about",
         "Post length", "Post Share Count", "Comments/Hr")
metrics = metrics[vars]

# Set null values to 0
metrics[is.na(metrics)] = 0

# Avoid 0 values in response
metrics$`Comments/Hr` = metrics$`Comments/Hr` + 0.00000001

# Avoid 0 values in predictors
metrics$`Page Checkins` = metrics$`Page Checkins` + 0.00000001
metrics$`Post length` = metrics$`Post length` + 0.00000001

# Remove Outliers
summary(metrics$`Post Share Count`)

metrics = metrics %>% arrange(desc(`Post Share Count`)) %>%
  slice(-1:-2)

outlier.test.model = lm(`Comments/Hr` ~ ., data = metrics)
rs = rstandard(outlier.test.model)
rs #7, 39

pairs(metrics)

# predictors
comments = metrics$`Comments/Hr`
popularity = metrics$`Page Popularity`
checkins = metrics$`Page Checkins`
talkabt = metrics$`Page talking about`
postlen = metrics$`Post length`
sharecount = metrics$`Post Share Count`

# Stepwise Regression
mod0 = lm(comments ~ 1)
add1(mod0, ~.+popularity+checkins+talkabt+postlen+sharecount, test = 'F')
```

```
mod1 = update(mod0, ~.+talkabt)
add1(mod1, ~.+popularity+checkins+postlen+sharecount, test = 'F')

mod2 = update(mod1, ~.+popularity)

# Check that added variable does not affect original predictors
summary(mod2)

add1(mod2, ~.+checkins+postlen+sharecount, test = 'F')

model.full = lm(comments ~ talkabt + popularity + talkabt*popularity)
model.reduced = lm(comments ~ talkabt + popularity)
anova(model.reduced, model.full)

plot(fitted(mod2), resid(mod2), xlab = 'Fitted Values', ylab = 'Residuals', main = 'Residuals vs
Fitted Value')

boxcox(mod2, lambda = seq(-1, 1, length = 10))

model = lm(comments^0.2 ~ talkabt + popularity)
plot(fitted(model), resid(model), xlab = 'Fitted Values', ylab = 'Residuals', main = 'Residuals vs
Fitted Value')

summary(model)

par(mfrow=c(1,2))

plot(fitted(lm(comments^0.2 ~ talkabt)), resid(lm(comments^0.2 ~ talkabt)), xlab = 'Popularity',
ylab = 'Residuals', main = 'Residuals vs Page Talkin About')

plot(fitted(lm(comments^0.2 ~ popularity)), resid(lm(comments^0.2 ~ popularity)), xlab =
'Checkins', ylab = 'Residuals', main = 'Residuals vs Page Popularity')

par(mfrow=c(1,2))

model.t.x1 = lm(comments^0.2 ~ log(talkabt))
model.t.x2 = lm(comments^0.2 ~ log(popularity))

plot(fitted(model.t.x1), resid(model.t.x1), xlab = 'Popularity', ylab = 'Residuals', main =
'Residuals vs Page Talkin About')
```

```
plot(fitted(model.t.x2), resid(model.t.x2), xlab = 'Checkins', ylab = 'Residuals', main = 'Residuals vs Page Popularity')

model.transform = lm(comments^0.2 ~ log(talkabt) + log(popularity))
summary(model.transform)

plot(fitted(model.transform), resid(model.transform), xlab = 'Fitted Values', ylab = 'Residuals', main = 'Residuals vs Fitted Value')

qqnorm(resid(model.transform))
qqline(resid(model.transform))

# Research Question 1:
model.transform

# Research Question 2:
mod.full2 = lm((comments)^0.2 ~ log(talkabt) + log(popularity))
mod.reduced2 = lm((comments)^0.2 ~ 1)
anova(mod.reduced2, mod.full2)

# Research Question 3:
mod.reduced = lm((comments)^0.2 ~ log(talkabt))
mod.full = lm((comments)^0.2 ~ log(talkabt) + log(popularity))
anova(mod.reduced, mod.full)

# Research Question 4:
summary(metrics)
#find average data
avgdata = data.frame(talkabt = mean(talkabt),
                     popularity = mean(popularity))
predict(model.transform, avgdata, interval = "confidence", level = 0.95) #0.8772555 0.730987 1.023524
prediction = 0.8772555^(1/0.2)
prediction
ci.lower = 0.730987^(1/0.2)
ci.lower
ci.upper = 1.023524^(1/0.2)
ci.upper
```