R Final Project

Xiao Yuheng

2023-04-27

Part 1: a: Background. This data set is about Donald Trump's all insult Tweeters. In case Trump has been arrested, analyze his Tweeters will make contribution to understand his behaviors. b: Objective. Analyze what impacted his behavior. c: Data set. I am using the Trump's insult tweeters data set. The source is Kaggle. link: https://www.kaggle.com/datasets/ayushggarg/all-trumps-twitter-insults-20152021 d: Methods. Methods of chi-sq test, t-test(non-parametric), anova, linear regression, etc. are going to be used.

Part 2: Data Analysis. a: Data Preview

```
insult <- read.csv("Trump_Insult_Tweeter.csv",encoding = "UTF-8")
dim(insult)#Data set contains 5 variables and 10360 rows.
```

```
## [1] 10360     5
```

```
head(insult,20)
```

```
##     X       date                    target
## 1   1 2014-10-09           thomas-frieden
## 2   2 2014-10-09           thomas-frieden
## 3   3 2015-06-16              politicians
## 4   4 2015-06-24               ben-cardin
## 5   5 2015-06-24               neil-young
## 6   6 2015-06-24 rockin-in-the-free-world
## 7   7 2015-06-25             willie-geist
## 8   8 2015-06-25                 jeb-bush
## 9   9 2015-06-25               molly-sims
## 10 10 2015-06-25           nicole-wallace
## 11 11 2015-06-25                 the-view
## 12 12 2015-06-25                 the-view
## 13 13 2015-06-25           nicole-wallace
## 14 14 2015-06-25                 the-view
## 15 15 2015-06-25        lawrence-o-donnell
## 16 16 2015-06-25        lawrence-o-donnell
## 17 17 2015-06-25              rick-scott
## 18 18 2015-06-26             john-roberts
## 19 19 2015-06-26                univision
## 20 20 2015-06-26             john-roberts
##                                                         insult
## 1                                                         fool
## 2                                                         DOPE
## 3                                           all talk and no action
## 4        It’s politicians like Cardin that have destroyed Baltimore.
## 5                                                     total hypocrite
## 6                                                      didn’t love it
## 7                                               uncomfortable looking
## 8                                 will NEVER Make America Great Again
## 9                                                          a disaster
## 10                                                         a disaster
## 11                                                          dead T.V.
## 12                                                    put it to sleep
## 13                                                doesn’t have a clue
## 14                                                     close to death
## 15                                             dopey political pundit
## 16                             one of the dumber people on television
## 17                                    did really poorly on television
## 18                                                        let us down
## 19                                  controlled by Mexican government?
## 20 my judicial appointments will do the right thing unlike... Roberts
##
## 1  Can you believe this fool, Dr. Thomas Frieden of CDC, just stated, "anyone with fever should be a
## 2  Can you believe this fool, Dr. Thomas Frieden of CDC, just stated, "anyone with fever should be a
## 3                     Big time in U.S. today - MAKE AMERICA GREAT AGAIN! Politicians are all talk and
## 4  Politician @SenatorCardin didn’t like that I said Baltimore needs jobs & spirit. It’s politicians
## 5                    For the nonbeliever, here is a photo of @Neilyoung in my office and his $$ requ
## 6                     .@Neilyoung’s song, "Rockin’ In The Free World" was just one of 10 songs used
## 7  Uncomfortable looking NBC reporter Willie Geist calls me to ask for favors and then mockingly smil
## 8   Just out, the new nationwide @FoxNews poll has me alone in 2nd place, closely behind Jeb Bush-but
## 9        The ratings for The View are really low. Nicole Wallace and Molly Sims are a disaster. Get
## 10       The ratings for The View are really low. Nicole Wallace and Molly Sims are a disaster. Get
## 11       The ratings for The View are really low. Nicole Wallace and Molly Sims are a disaster. Get
## 12       The ratings for The View are really low. Nicole Wallace and Molly Sims are a disaster. Get
## 13     .@WhoopiGoldberg had better surround herself with better hosts than Nicole Wallace, who doesn
## 14     .@WhoopiGoldberg had better surround herself with better hosts than Nicole Wallace, who doesn
## 15 I hear that dopey political pundit, Lawrence O’Donnell, one of the dumber people on television,
## 16 I hear that dopey political pundit, Lawrence O’Donnell, one of the dumber people on television,
## 17                                                   Governor Rick Scott of Florida did really poorly on
## 18                             Once again the Bush appointed Supreme Court Justice John Roberts has
## 19                           .@Univision cares far more about Mexico than it does about the U.S. Are
## 20                 If I win the presidency, my judicial appointments will do the right thing unlike
```

```
sum(is.na(insult))#Just 1 missing value.
```

```
## [1] 1
```

```
sum(is.na(insult$target))#Missing value comes from target column.
```

```
## [1] 1
```

```
#It seems the missing value does not matter.
```

b: Data Cleaning

```r
insult$length <- nchar(insult$tweet)#Get the length of each Tweeter.
insult$year <- substr(insult$date,1,4)#Extract the year.
media_list <- list("the-media","the-new-york-times","cnn",
                   "washington-post","fox-news","nbc-news",
                   "msnbc","twitter","abc-post-poll",
                   "saturday-night-live","abc-news"
                   )
insult$target_media_or_not <- insult$target %in% media_list
#This column indicates the tweeter is about media or not, respectively.
insult$target_media_or_not <- factor(insult$target_media_or_not)
library(lubridate)
insult$datetime <- as.Date(insult$date)#Convert to the desired form.
insult$tweet_char_setting_changed <- insult$datetime >= as.Date("2017-11-08")
insult$date_before_presidency <- insult$datetime <=
  as.Date("2017-01-20")
insult$date_before_presidency <- factor(insult$date_before_presidency)
#Use the presidency time to divide the data set into two parts.
#Twitter expanded the largest character from 140 to 280 on 2017-11-08.
insult$tweet_char_setting_changed <- factor(insult$tweet_char_setting_changed)
insult$was_impeached <- insult$datetime >= as.Date("2019-12-18")
#Trump was impeached on 2019-12-18, this is other criteria to divide data
insult$was_impeached <- as.factor(insult$was_impeached)
```

c: Preview after data cleaning.

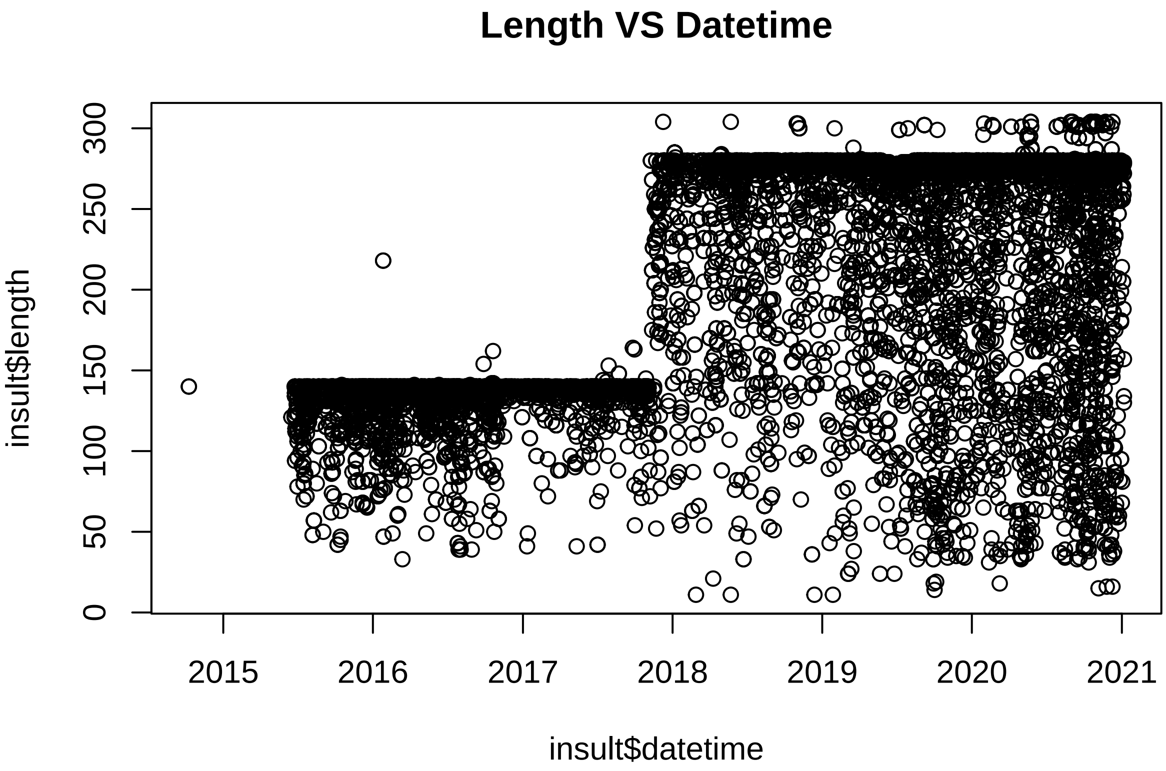
```r
head(insult, 20)
```

```
##     X       date                   target
## 1   1 2014-10-09           thomas-frieden
## 2   2 2014-10-09           thomas-frieden
## 3   3 2015-06-16              politicians
## 4   4 2015-06-24               ben-cardin
## 5   5 2015-06-24               neil-young
## 6   6 2015-06-24 rockin-in-the-free-world
## 7   7 2015-06-25             willie-geist
## 8   8 2015-06-25                 jeb-bush
## 9   9 2015-06-25               molly-sims
## 10 10 2015-06-25           nicole-wallace
## 11 11 2015-06-25                 the-view
## 12 12 2015-06-25                 the-view
## 13 13 2015-06-25           nicole-wallace
## 14 14 2015-06-25                 the-view
## 15 15 2015-06-25       lawrence-o-donnell
## 16 16 2015-06-25       lawrence-o-donnell
## 17 17 2015-06-25               rick-scott
## 18 18 2015-06-26             john-roberts
## 19 19 2015-06-26                univision
## 20 20 2015-06-26             john-roberts
##                                                                 insult
## 1                                                                 fool
## 2                                                                 DOPE
## 3                                               all talk and no action
## 4         It’s politicians like Cardin that have destroyed Baltimore.
## 5                                                      total hypocrite
## 6                                                       didn’t love it
## 7                                                uncomfortable looking
## 8                                  will NEVER Make America Great Again
## 9                                                           a disaster
## 10                                                          a disaster
## 11                                                           dead T.V.
## 12                                                     put it to sleep
## 13                                                 doesn’t have a clue
## 14                                                      close to death
## 15                                              dopey political pundit
## 16                              one of the dumber people on television
## 17                                     did really poorly on television
## 18                                                         let us down
## 19                                   controlled by Mexican government?
## 20 my judicial appointments will do the right thing unlike... Roberts
##
## 1  Can you believe this fool, Dr. Thomas Frieden of CDC, just stated, "anyone with fever should be as
## 2  Can you believe this fool, Dr. Thomas Frieden of CDC, just stated, "anyone with fever should be as
## 3                     Big time in U.S. today - MAKE AMERICA GREAT AGAIN! Politicians are all talk and
## 4  Politician @SenatorCardin didn’t like that I said Baltimore needs jobs & spirit. It’s politicians
## 5                    For the nonbeliever, here is a photo of @Neilyoung in my office and his $$ reque
## 6                     .@Neilyoung’s song, "Rockin’ In The Free World" was just one of 10 songs used
## 7  Uncomfortable looking NBC reporter Willie Geist calls me to ask for favors and then mockingly smil
## 8   Just out, the new nationwide @FoxNews poll has me alone in 2nd place, closely behind Jeb Bush-but
## 9        The ratings for The View are really low. Nicole Wallace and Molly Sims are a disaster. Get
## 10       The ratings for The View are really low. Nicole Wallace and Molly Sims are a disaster. Get
## 11       The ratings for The View are really low. Nicole Wallace and Molly Sims are a disaster. Get
## 12       The ratings for The View are really low. Nicole Wallace and Molly Sims are a disaster. Get
## 13     .@WhoopiGoldberg had better surround herself with better hosts than Nicole Wallace, who doesn
## 14     .@WhoopiGoldberg had better surround herself with better hosts than Nicole Wallace, who doesn
## 15 I hear that dopey political pundit, Lawrence O’Donnell, one of the dumber people on television, is
## 16 I hear that dopey political pundit, Lawrence O’Donnell, one of the dumber people on television, is
## 17                                               Governor Rick Scott of Florida did really poorly on
## 18                                 Once again the Bush appointed Supreme Court Justice John Roberts has
## 19                               .@Univision cares far more about Mexico than it does about the U.S. Are
## 20                     If I win the presidency, my judicial appointments will do the right thing unlike
##    length year target_media_or_not   datetime tweet_char_setting_changed
## 1     140 2014               FALSE 2014-10-09                      FALSE
## 2     140 2014               FALSE 2014-10-09                      FALSE
## 3     121 2015               FALSE 2015-06-16                      FALSE
## 4     140 2015               FALSE 2015-06-24                      FALSE
## 5     122 2015               FALSE 2015-06-24                      FALSE
## 6     121 2015               FALSE 2015-06-24                      FALSE
## 7     140 2015               FALSE 2015-06-25                      FALSE
## 8     139 2015               FALSE 2015-06-25                      FALSE
## 9     134 2015               FALSE 2015-06-25                      FALSE
## 10    134 2015               FALSE 2015-06-25                      FALSE
## 11    134 2015               FALSE 2015-06-25                      FALSE
## 12    134 2015               FALSE 2015-06-25                      FALSE
## 13    136 2015               FALSE 2015-06-25                      FALSE
## 14    136 2015               FALSE 2015-06-25                      FALSE
## 15    140 2015               FALSE 2015-06-25                      FALSE
## 16    140 2015               FALSE 2015-06-25                      FALSE
## 17     94 2015               FALSE 2015-06-25                      FALSE
## 18    112 2015               FALSE 2015-06-26                      FALSE
## 19    114 2015               FALSE 2015-06-26                      FALSE
## 20    124 2015               FALSE 2015-06-26                      FALSE
##    date_before_presidency was_impeached
## 1                    TRUE         FALSE
## 2                    TRUE         FALSE
## 3                    TRUE         FALSE
## 4                    TRUE         FALSE
## 5                    TRUE         FALSE
## 6                    TRUE         FALSE
## 7                    TRUE         FALSE
## 8                    TRUE         FALSE
## 9                    TRUE         FALSE
## 10                   TRUE         FALSE
## 11                   TRUE         FALSE
## 12                   TRUE         FALSE
## 13                   TRUE         FALSE
## 14                   TRUE         FALSE
## 15                   TRUE         FALSE
## 16                   TRUE         FALSE
## 17                   TRUE         FALSE
## 18                   TRUE         FALSE
## 19                   TRUE         FALSE
## 20                   TRUE         FALSE
```

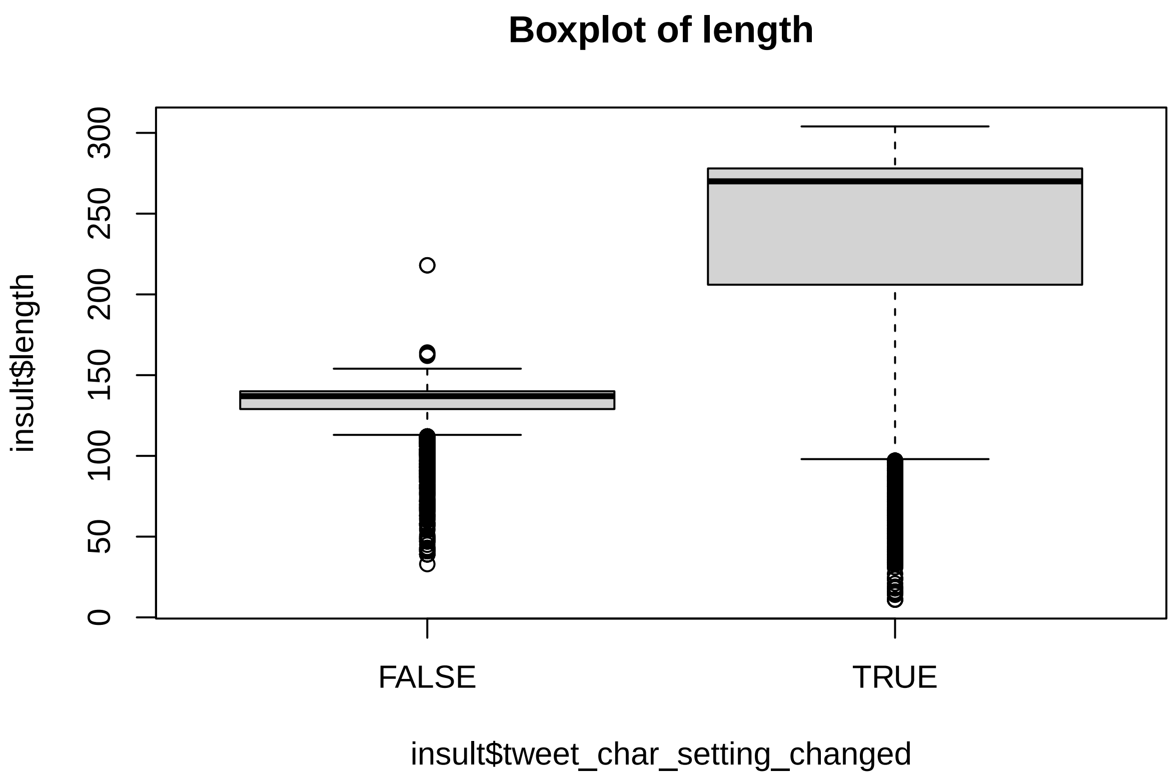
```r
#Below is an introduction about the data set.
#X: number.
#date: exact date.
#Target: tweeter's target.
#insult: offensive words.
#Tweeter: main body.
#length: Tweeter's length.
#year: exact date's year.
#target_media_or_not: FALSE if not media while TRUE when target media.
#date_before_presidency: TRUE if date before presidency and vice versa.
#datetime: date in correct format.
#was_impeached: FALSE before the impeachment and TRUE after.
dim(insult)#Data set contains 12 variables and 10360 rows.
```

```
## [1] 10360    12
```

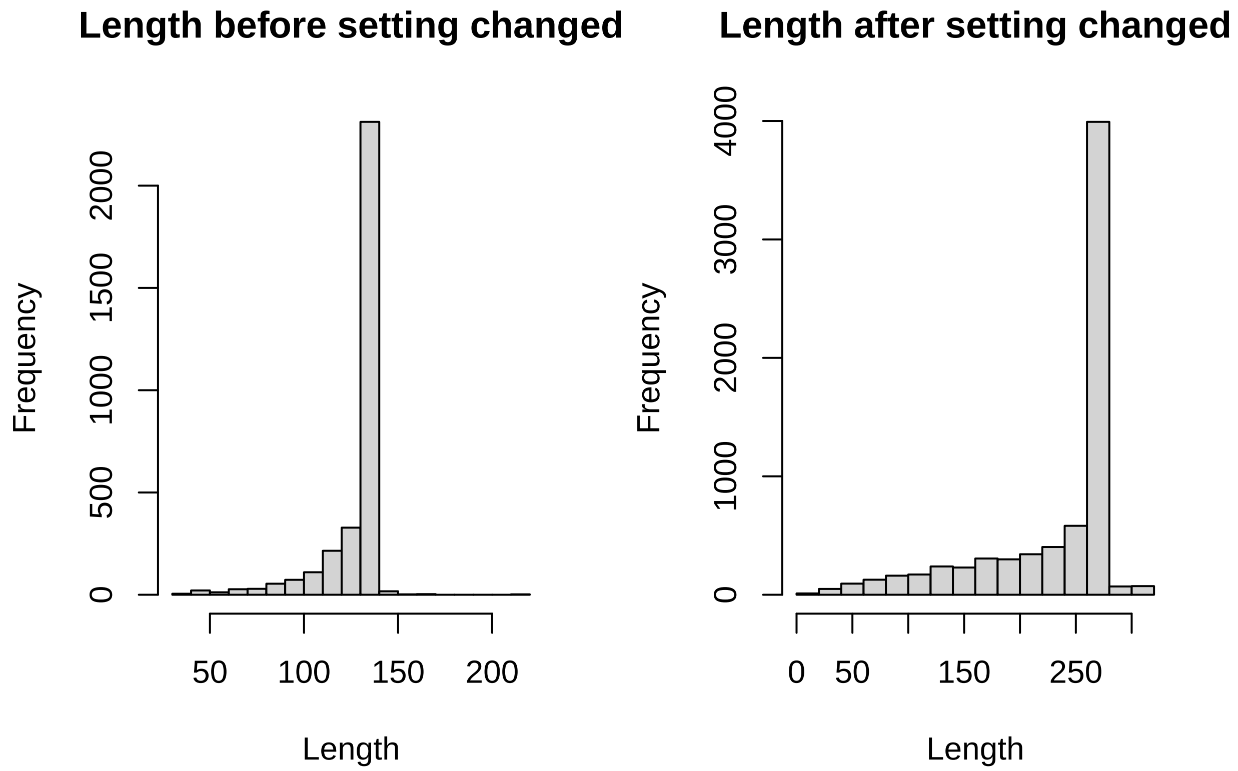
```r
plot(insult$datetime, insult$length, main = "Length VS Datetime")
```



```
#From the graph we can see that Trump always reach the max character.
boxplot(insult$length~insult$tweet_char_setting_changed,
        main = "Boxplot of length")
```



```
#Again, we can see that Trump always reach the max character.
par(mfrow=c(1,2))
hist(insult$length[insult$tweet_char_setting_changed==FALSE],
     breaks=20, main = "Length before setting changed",
     xlab = "Length")
hist(insult$length[insult$tweet_char_setting_changed==TRUE],
     breaks=20, main = "Length after setting changed",
     xlab = "Length")
```



d: Get some statistics.

```
#Remind:
#After inspection, I found that the length before twitter expanded
#the largest character from 140 to 280 sometimes exceed 140 is because
#Trump's twitter sometimes include a long website, which the website
#is not counted as character.
#However, some statistics such as mean and median retain their power.
library(doBy)
summaryBy(length~tweet_char_setting_changed, data=insult, FUN=summary)
```

```
##   tweet_char_setting_changed length.Min. length.1st Qu. length.Median
## 1                      FALSE          33            129           137
## 2                       TRUE          11            206           270
##   length.Mean length.3rd Qu. length.Max.
## 1    130.0754            140         218
## 2    234.6534            278         304
```

```
table(insult$target_media_or_not)
```

```
##
## FALSE  TRUE
##  7943  2417
```

```
table(insult$tweet_char_setting_changed)
```

```
##
## FALSE  TRUE
##  3210  7150
```

```
table(insult$date_before_presidency)
```

```
##
## FALSE  TRUE
##  8003  2357
```

```
table(insult$was_impeached)
```

```
##
## FALSE  TRUE
##  7542  2818
```

Remark before I start: during the preview section I have inspected that some tweeters are long because they contain web link, and almost all of his tweeters are with max characters, so the characters of tweeter can not be an accurate evaluation.

e: Question 1. Has Trump hated the media more since he took office?

```
#H0: the variable "date_before_presidency" and the variable
#"target_media_or_not" are independent.
#Ha: the variable "date_before_presidency" and the variable
#"target_media_or_not" are not independent.
table_media <- xtabs(~date_before_presidency+target_media_or_not,
                     data=insult)
media_chisq<-chisq.test(table_media)
media_chisq$p.value#1.037685e-33, which is super small.
```

```
## [1] 1.037685e-33
```

```
#Reject H0
#We concluded that the variable "date_before_presidency" and the variable
#"target_media_or_not" are not independent.
#Media has an impact on Trump's behavior during his presidency.
```

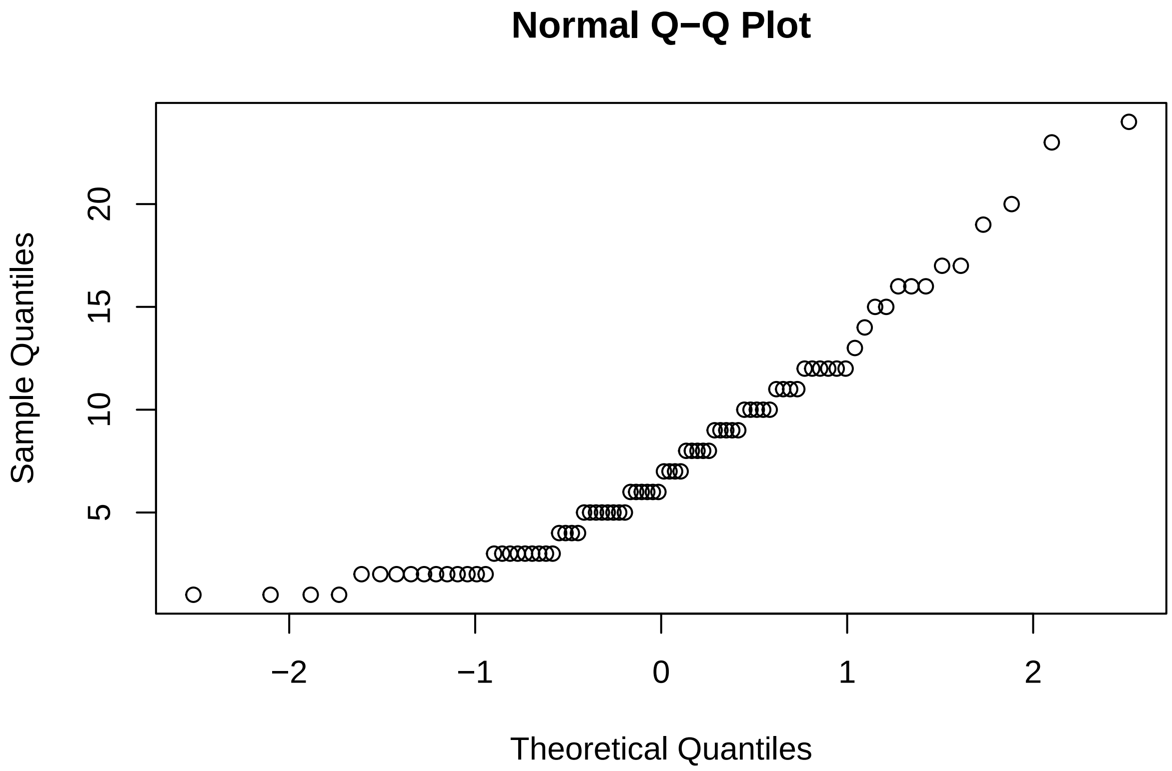
f: Question 2. We know that the impeachment happened at the end of 2019. Has Trump been significantly impacted by the impeachment?

```
#H0: Trump has not been significantly impacted.
#Ha: Trump has been significantly impacted.
#Use t test.
#Alpha: 0.05
#Assumptions:
#1. The two samples should follow normal distribution.
#2. The samples should have equal variance.
```

```
#Take the six months interval to do research.
#Because the impact of an emergency is usually most pronounced within a #month
tem_form_1 <- insult[year(insult$datetime)==2019
                   &insult$tweet_char_setting_changed=="TRUE"
                   &month(insult$datetime)%in%list(10,11,12),]
tem_form_2 <- insult[year(insult$datetime)==2020
                   &insult$tweet_char_setting_changed=="TRUE"
                   &month(insult$datetime)%in%list(1,2,3),]
tweeter_per_day_1 <- aggregate(tem_form_1$tweet,
                               by=list(tem_form_1$datetime),
                               FUN=length)
tweeter_per_day_2 <- aggregate(tem_form_2$tweet,
                               by=list(tem_form_2$datetime),
                               FUN=length)
#Check Assumptions:
sample_1 <- tweeter_per_day_1$x
sample_2 <- tweeter_per_day_2$x
shapiro.test(sample_1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample_1
## W = 0.91781, p-value = 5.025e-05
```

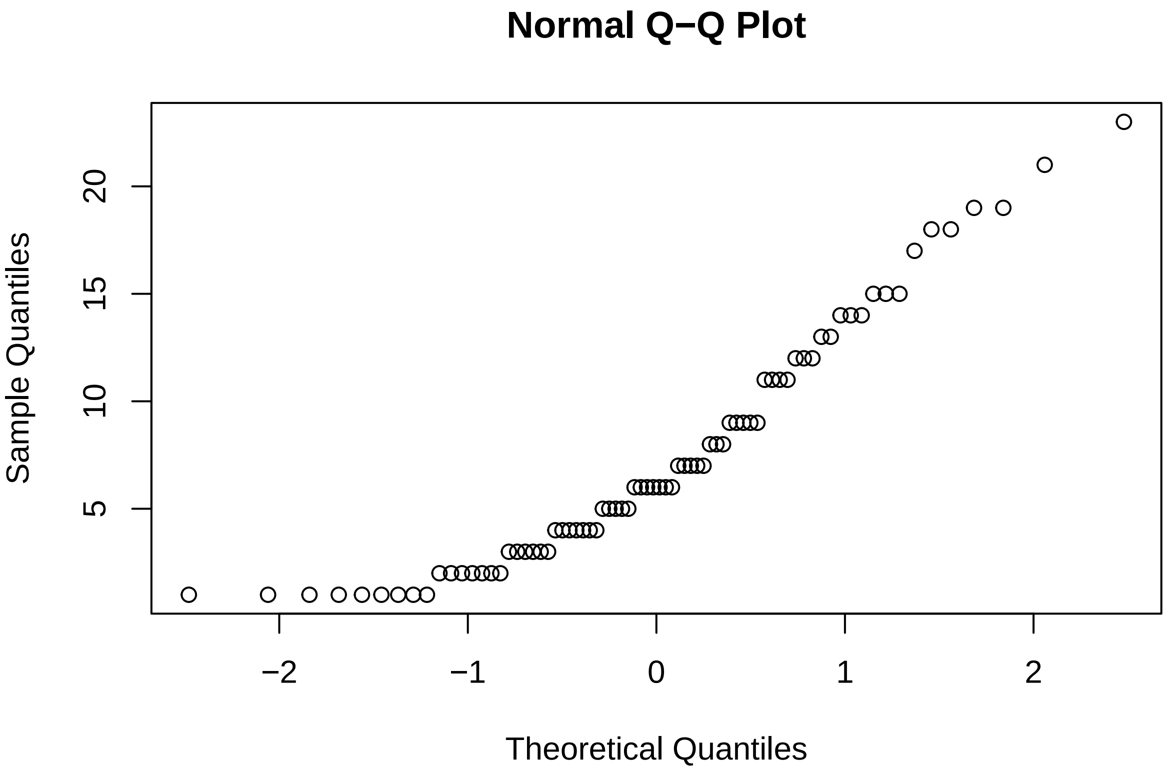
```
qqnorm(sample_1)
```



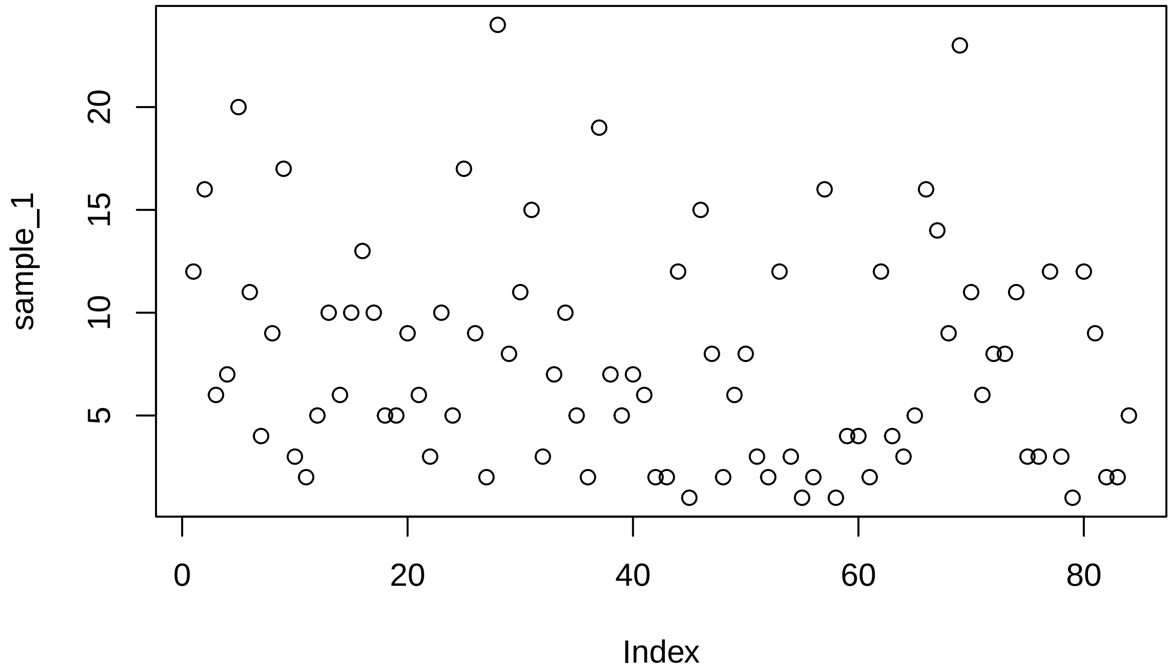
```
shapiro.test(sample_2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample_2
## W = 0.91077, p-value = 5.49e-05
```

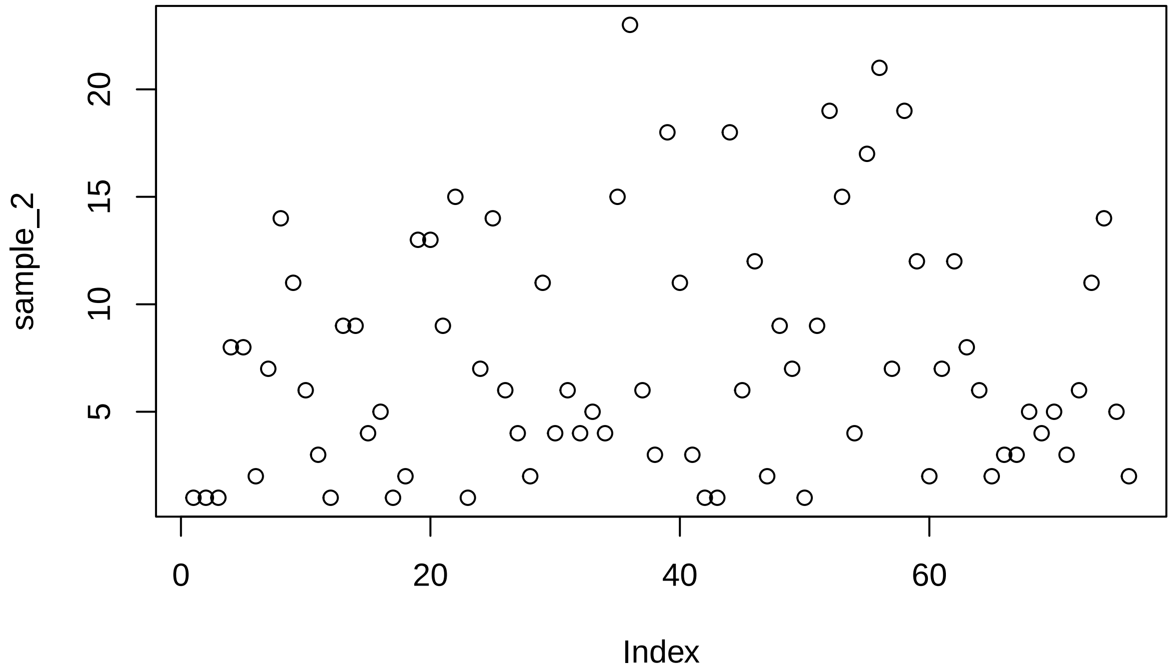
```
qqnorm(sample_2)
```



```
plot(sample_1)
```



```
plot(sample_2)
```



```
#Those samples violate the normal distribution assumption
#under alpha = 0.01.
#I concluded it as severely violated.
var.test(sample_2,sample_1)
```

```
##
##  F test to compare two variances
##
## data:  sample_2 and sample_1
## F = 1.0753, num df = 75, denom df = 83, p-value = 0.745
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6909596 1.6819355
## sample estimates:
## ratio of variances
##           1.075303
```

```
#Variance is not significantly unequal,
#because p is 0.745, which is large.
#Use alternative method.
#To use the non-parametric test,
#sample_1 and sample_2 should be in same length to be paired.
length(sample_1)
```

```
## [1] 84
```

```
length(sample_2)
```

```
## [1] 76
```

```
set.seed(1234)
tweeter_per_day_1 <- tweeter_per_day_1[sample(84,76),]
t.test(tweeter_per_day_1$x,
       tweeter_per_day_2$x, paired = TRUE)$p.value
```

```
## [1] 0.6359715
```

```
#p is 0.6359715, which is not too small.
#Fail to reject H0.
#The impeachment at the end of 2019 did not impact him significantly.
#(I guess maybe he knew the impeachment long before 2019-12-18)
```

g: Question 3 Remark 1: since most of the variables in this data set are categorical, the best and the most suitable method should be anova if assumptions are satisfied. Remark 2: since every tweeter is about insult someone or something, the count of his tweeter can definitely reflect his aggressiveness. Goal: Use the average tweeter per day to evaluate: Does Trump become more aggressive after his presidency? If the answer is yes, what is the relationship? (Donald Trump's tenure as the 45th president of the United States began with his inauguration on January 20, 2017. source: https://en.wikipedia.org/wiki/Presidency_of_Donald_Trump).

```
average_per_day <- aggregate(insult$tweet,
                                   by=list(insult$datetime),
                                   FUN=length)
average_per_day_df <- as.data.frame(average_per_day)
average_per_day_df$Group.1 <- as.Date(average_per_day_df$Group.1)
average_per_day_df$year <- substr(average_per_day_df$Group.1,1,4)
average_per_day_df$presidency <-
  average_per_day_df$Group.1>=as.Date("2017-01-20")
table(average_per_day_df$year)
```

```
##
## 2014 2015 2016 2017 2018 2019 2020 2021
##    1  146  259  231  279  326  325    6
```

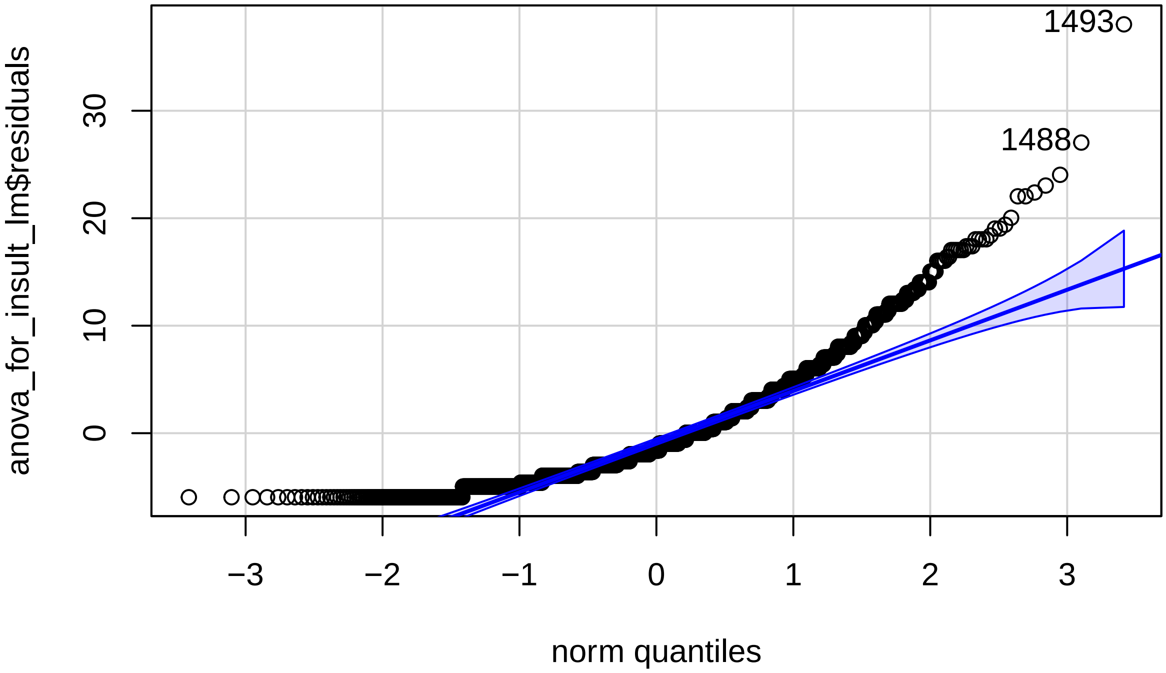
```
table(average_per_day_df$presidency)
```

```
##
## FALSE  TRUE
##   421  1152
```

```
#2014 and 2021's data are not completed.
#Dump them.
average_per_day_df <- average_per_day_df[average_per_day_df$year
                                             %in% c(2015,2016,2017,2018,
                                                    2019,2020),]
#Take a roughly look.
aggregate(average_per_day_df$x, by=list(average_per_day_df$presidency),
          FUN=mean)
```

```
##   Group.1        x
## 1   FALSE 5.607143
## 2    TRUE 6.959860
```

```
#The above code indicates that this question is meaningful.
#H0: Trump's aggressiveness degree stays the same.
#H0: u2015=u2016=u2017=u2018=u2019=u2020.
#Ha: Trump's aggressiveness degree changed.
#Try to do one-way anova:
#Checking Assumptions:
library(car)
anova_for_insult_lm <- lm(x~presidency, data=average_per_day_df)
qqPlot(anova_for_insult_lm$residuals)
```



```
## 1493 1488
## 1492 1487
```

```
shapiro.test(anova_for_insult_lm$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  anova_for_insult_lm$residuals
## W = 0.86387, p-value < 2.2e-16
```

```
#Unfortunately, the normality assumption is violated.
bartlett.test(x~presidency, data=average_per_day_df)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  x by presidency
## Bartlett's K-squared = 11.512, df = 1, p-value = 0.0006914
```

```
#Unfortunately, the equal variance assumption is also violated.
#Try to use non-parametric test.
#Recall the in the previous parts, I force the length of sample to be
#same, but for here, repeat this process will trigger the loss of
#accuracy.
#More than 2 groups and with violation of normality:
kruskal.test(x ~ year, data=average_per_day_df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  x by year
## Kruskal-Wallis chi-squared = 77.06, df = 5, p-value = 3.455e-15
```

```
#Accept Ha, reject H0.
```

h: Question 3 Alternative Method. It is said that anova, especially one-way anova has a robustness. The one-way ANOVA is considered a robust test against the normality assumption (https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide-3.php). ANOVA is robust to heterogeneity of variance so long as the largest variance is not more than 4 times the smallest variance (https://stats.stackexchange.com/questions/56971/alternative-to-one-way-anova-unequal-variance). Since the variable is not independent with the variable presidency, the interaction must exist. Anocova can't be applied here.

```
#H0: Trump's aggressiveness degree stays the same.
#H0: u2015=u2016=u2017=u2018=u2019=u2020.
#Ha: Trump's aggressiveness degree changed.
average_per_day_df$year <- as.factor(average_per_day_df$year)
anova_for_insult <- aov(x~year, data=average_per_day_df)
summary(anova_for_insult)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## year           5   2348   469.6   16.87 2.89e-16 ***
## Residuals   1560  43435    27.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

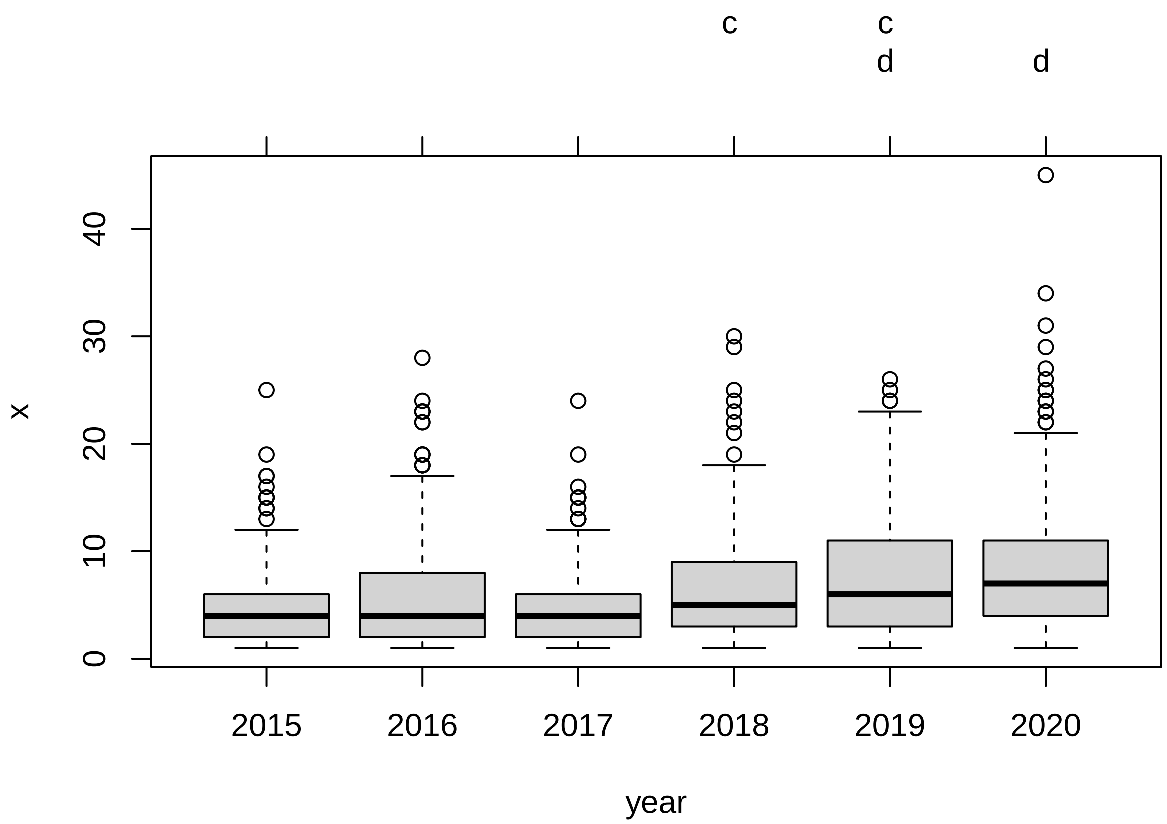
```
#The p value indicates the same result of part g: reject H0.
#Multiple Comparison:
library(multcomp)
TukeyHSD(anova_for_insult)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = x ~ year, data = average_per_day_df)
##
## $year
##                 diff        lwr       upr     p adj
## 2016-2015  0.7571534 -0.8009557 2.3152625 0.7352395
## 2017-2015 -0.3364467 -1.9282335 1.2553402 0.9908336
## 2018-2015  1.1842441 -0.3536010 2.7220892 0.2395235
## 2019-2015  2.2567863  0.7575074 3.7560652 0.0002678
## 2020-2015  3.1596839  1.6596916 4.6596761 0.0000000
## 2017-2016 -1.0936001 -2.4561074 0.2689072 0.1985960
## 2018-2016  0.4270907 -0.8719891 1.7261705 0.9366174
## 2019-2016  1.4996328  0.2464457 2.7528200 0.0085820
## 2020-2016  2.4025304  1.1484900 3.6565708 0.0000008
## 2018-2017  1.5206908  0.1814038 2.8599778 0.0154479
## 2019-2017  2.5932329  1.2984127 3.8880532 0.0000002
## 2020-2017  3.4961305  2.2004844 4.7917766 0.0000000
## 2019-2018  1.0725422 -0.1553591 2.3004434 0.1269230
## 2020-2018  1.9754398  0.7466676 3.2042119 0.0000711
## 2020-2019  0.9028976 -0.2772510 2.0830461 0.2462503
```

```
tuk_insult <- glht(anova_for_insult, linfct=mcp(year="Tukey"))
cld(tuk_insult, level=.05)
```

```
## 2015 2016 2017 2018 2019 2020
## "ab" "ab"  "a" "bc" "cd"  "d"
```

```
plot(cld(tuk_insult, level=.05), col="lightgrey")
```



```
#It can be seen that the differences between adjacent years tend to be #insignificant. That is to say,
```

i: Linear Regression. My conclusion in the previous part is that the data has a trend, so it is natural to use linear regression to analyze the trend. Since the difference between before and after presidency is significant, it is necessary to fit two models.

```
#Step 1: .Data Cleaning.
average_per_day_df$year_month <- substr(average_per_day_df$Group.1,1,7)
average_per_month_df <- aggregate(
  average_per_day_df$x, by = list(average_per_day_df$year_month),
  FUN = mean
)
#Set 2015-06 as starting point.
average_per_month_df <- average_per_month_df[order(
  average_per_month_df$Group.1,decreasing = FALSE
),]
dim(average_per_month_df)
```

```
## [1] 67  2
```

```
average_per_month_df$timestamp_as_label <- seq(1,67,1)
average_per_month_df$presidency <-
  substr(average_per_month_df$Group.1,1,4)%in%c(
  2017,2018,2019,2020
)
```

```r
#Step 2: Transform for henceforth usage.
powerTransform(average_per_month_df$x)
```

```
## Estimated transformation parameter
## average_per_month_df$x
##              0.2430473
```

```r
boxTidwell(x~timestamp_as_label, data=average_per_month_df[
  average_per_month_df$presidency==TRUE,
])
```

```
##  MLE of lambda Score Statistic (z) Pr(>|z|)
##        0.43246             -0.5853   0.5584
##
## iterations =  2
```

```r
#Strong evidence to transform.
#Step 3: Fit two models. Firstly without transforming.
#Fit a model based on before taking office.
insult_lm_1 <- lm(x~timestamp_as_label, data=average_per_month_df[
  average_per_month_df$presidency==FALSE,
])
insult_lm_1
```

```
##
## Call:
## lm(formula = x ~ timestamp_as_label, data = average_per_month_df[average_per_month_df$presidency ==
##     FALSE, ])
##
## Coefficients:
##        (Intercept)  timestamp_as_label
##            4.92044             0.04579
```

```r
summary(insult_lm_1)
```

```
##
## Call:
## lm(formula = x ~ timestamp_as_label, data = average_per_month_df[average_per_month_df$presidency ==
##     FALSE, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4827 -1.1568 -0.0322  0.9155  4.9385
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.92044    0.89731   5.484 4.04e-05 ***
## timestamp_as_label  0.04579    0.07870   0.582    0.568
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
```

```
## Residual standard error: 1.879 on 17 degrees of freedom
## Multiple R-squared:  0.01952,    Adjusted R-squared:  -0.03815
## F-statistic: 0.3385 on 1 and 17 DF,  p-value: 0.5683
```

```
#Unfortunately, the F test is not significant.
#This model is meaningless. Dump it.
#Fit a model based on after taking office.
insult_lm_2 <- lm(x~timestamp_as_label, data=average_per_month_df[
  average_per_month_df$presidency==TRUE,
])
insult_lm_2
```

```
##
## Call:
## lm(formula = x ~ timestamp_as_label, data = average_per_month_df[average_per_month_df$presidency ==
##     TRUE, ])
##
## Coefficients:
##        (Intercept)  timestamp_as_label
##             2.2199              0.1018
```

```
summary(insult_lm_2)
```

```
##
## Call:
## lm(formula = x ~ timestamp_as_label, data = average_per_month_df[average_per_month_df$presidency ==
##     TRUE, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2204 -1.0353 -0.2506  0.6534  3.5420
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.21988    0.67685   3.280  0.00198 **
## timestamp_as_label  0.10177    0.01483   6.864 1.46e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.423 on 46 degrees of freedom
## Multiple R-squared:  0.506,  Adjusted R-squared:  0.4952
## F-statistic: 47.11 on 1 and 46 DF,  p-value: 1.458e-08
```

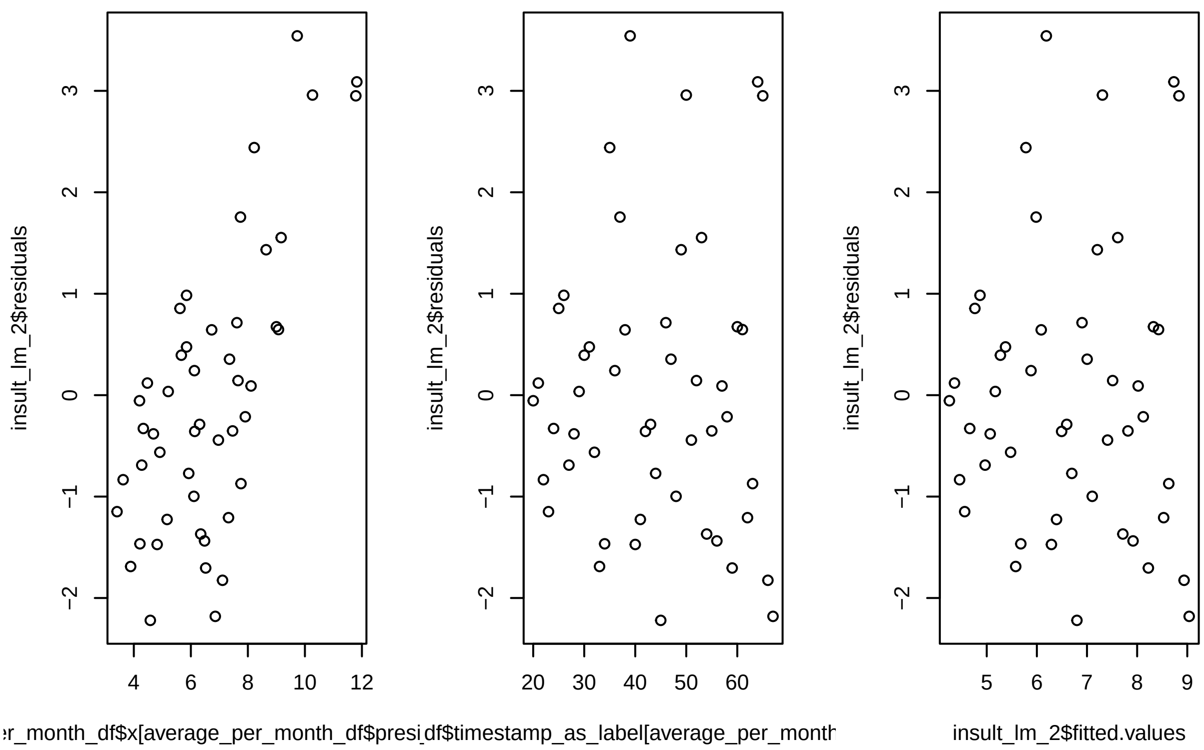
```
#F test indicates that this one is super significant.
#Step 4: Check Assumptions.
#Normality:
shapiro.test(insult_lm_2$residuals)#Violated.
```

```
##
##  Shapiro-Wilk normality test
##
## data:  insult_lm_2$residuals
## W = 0.94426, p-value = 0.02375
```

```
#Equal Variance.
ncvTest(insult_lm_2)#Violated.
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 4.204626, Df = 1, p = 0.040314
```

```
#Independence.
par(mfrow=c(1,3))
plot(average_per_month_df$x[average_per_month_df$presidency==TRUE],
     insult_lm_2$residuals)
plot(average_per_month_df$timestamp_as_label
     [average_per_month_df$presidency==TRUE],
     insult_lm_2$residuals)
plot(insult_lm_2$fitted.values, insult_lm_2$residuals)
```



```
#The first graph indicates the violation, the rest are good.
#Linearity.
crPlots(insult_lm_2)#Not violated.
#Outlier.
outlierTest(insult_lm_2)#Seemed no.
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
```

```
##    rstudent unadjusted p-value Bonferroni p
## 39 2.682575           0.010182      0.48873
```

```
#Step 5: Fit again after transforming.
#Try 0.5.
insult_lm_3 <- lm(x**0.5~timestamp_as_label, data=average_per_month_df[
  average_per_month_df$presidency==TRUE,
])
insult_lm_3
```

```
##
## Call:
## lm(formula = x^0.5 ~ timestamp_as_label, data = average_per_month_df[average_per_month_df$presidency
##     TRUE, ])
##
## Coefficients:
##        (Intercept)  timestamp_as_label
##            1.68079             0.01998
```

```
summary(insult_lm_3)
```

```
##
## Call:
## lm(formula = x^0.5 ~ timestamp_as_label, data = average_per_month_df[average_per_month_df$presidency
##     TRUE, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44020 -0.21459 -0.02982  0.12870  0.65927
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.680791   0.125901  13.350  < 2e-16 ***
## timestamp_as_label 0.019984   0.002758   7.246  3.9e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.2647 on 46 degrees of freedom
## Multiple R-squared:  0.533,  Adjusted R-squared:  0.5229
## F-statistic: 52.51 on 1 and 46 DF,  p-value: 3.898e-09
```

```
#F test is still significant.
#Step 6: Check Assumptions.
#Normality:
shapiro.test(insult_lm_3$residuals)#Not violated.
```

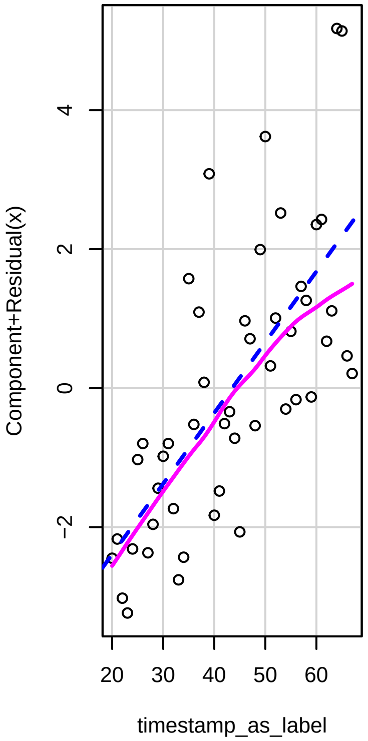
```
##
##  Shapiro-Wilk normality test
##
## data:  insult_lm_3$residuals
## W = 0.96645, p-value = 0.1835
```

```
#Equal Variance.
ncvTest(insult_lm_3)#Not violated.
```

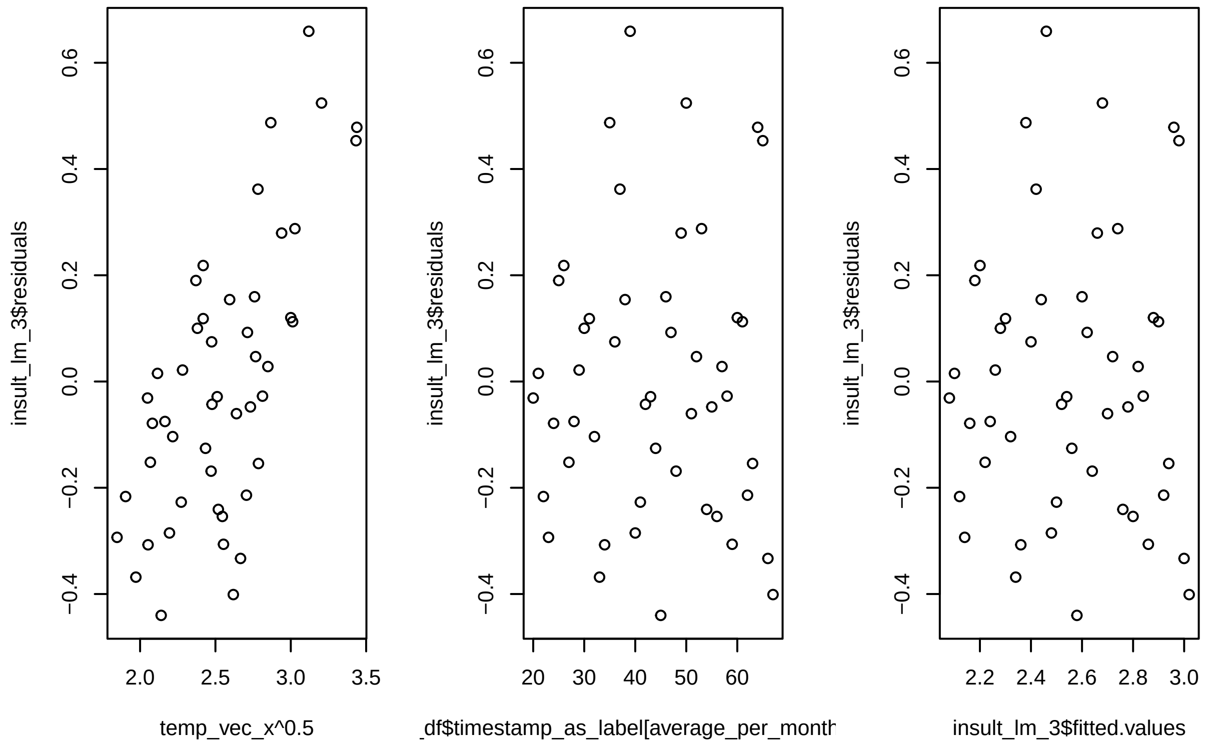
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.346689, Df = 1, p = 0.24586
```

```
#Independence.
temp_vec_x <-
  average_per_month_df[average_per_month_df$presidency==TRUE,]$x
par(mfrow=c(1,3))
```

.



```
plot(temp_vec_x**0.5,insult_lm_3$residuals)
plot(average_per_month_df$timestamp_as_label
     [average_per_month_df$presidency==TRUE],
     insult_lm_3$residuals)
plot(insult_lm_3$fitted.values, insult_lm_3$residuals)
```

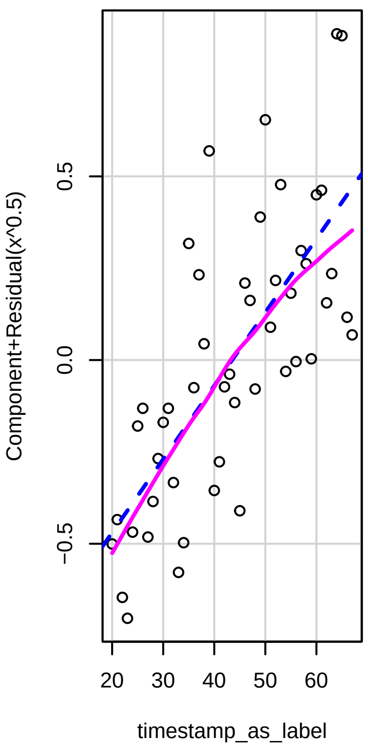


```
#The first graph indicates the violation, the rest are good.
#Linearity.
crPlots(insult_lm_3)#Not violated.
#Outlier.
outlierTest(insult_lm_3)#Seemed no.
```

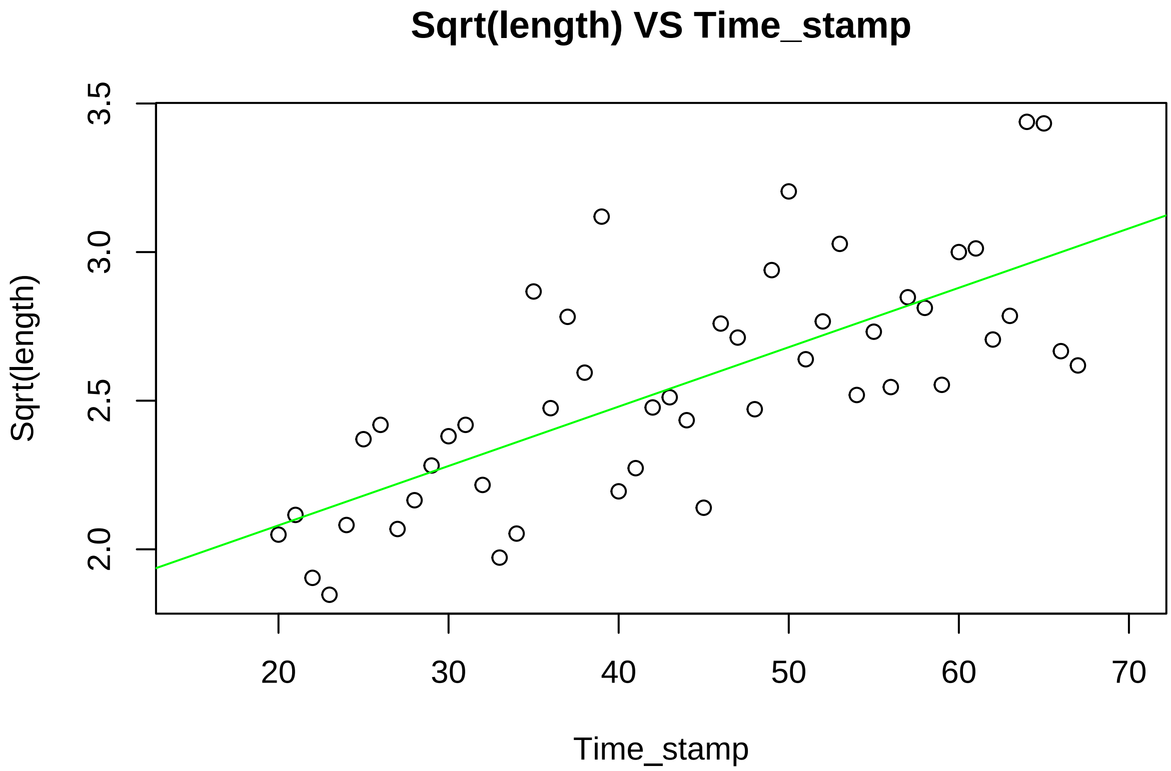
```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferroni p
## 39 2.684517           0.010131       0.4863
```

```
#Step 7: Visualization.
par(mfrow=c(1,1))
```

.



```
plot(
  average_per_month_df$timestamp_as_label
  [average_per_month_df$presidency==TRUE],
  temp_vec_x**0.5,
     main = "Sqrt(length) VS Time_stamp",
     xlab = "Time_stamp", ylab = "Sqrt(length)",
  xlim = c(15,70))
abline(insult_lm_3, col="green")
```



```
#Step 8: Relationship.
insult_lm_3$coefficients
```

```
##        (Intercept) timestamp_as_label
##         1.68079058         0.01998357
```

```
#Sqrt(length) = 1.68+0.02timestamp_as_label.
#Length = 2.82 + 0.067timestamp_as_label + 0.004(timestamp_as_label^2).
```

Part 3: Conclusion. a: Conclusion It seems that Trump has been more talkative and aggressive after taking office, which is against common sense because president is deemed to be too busy to attack others. One of the main cause is the media. The impeachment seems to have little impact to him.