S&DS 230 Project

2022-05-07

Introduction

Music is a big part of our lives! We stream music all the time, and we occasionally peek at charts to see if our favorite artist made the list. We may also wonder how certain variables, like rank, total number of listeners, and length of lyrics, may be related to each other. For our report, we specifically looked at Last.fm. We wanted to see if the total number of listeners for each song can be predicted by variables such as rank, length of lyrics, year, and the number of artist appearances.

Data

- Response Variable
  - Number of listeners on Last.fm (i.e. the cumulative number of listeners for each track since the inception of Last.fm or the release of the track)
- Categorical Variables
  - Gender
  - Region (i.e. the country where the song was produced)
  - Type (band or soloist)
- Continuous Variables
  - Year (i.e. the year in which the chart rankings are taken from)
  - Rank (i.e. a value between 1-100 representing its place the first time it appears on a year-end chart)
  - forcast (i.e. a measure of a lyric's readability measured by the proportion of one syllable words in a song)
  - mean_word_syllables (i.e. the average number of syllables per word in the lyrics)

Prereqs

```
library(tidyverse)
library(RSQLite)
library(RecordLinkage)
library(stringdist)
library(devtools)
library(quanteda)
library(quanteda.textstats)
library(car)
library(leaps)
library(corrplot)
library(PerformanceAnalytics)
source("http://www.reuningscherer.net/s&ds230/Rfuncs/regJDRS.txt")
```

```
conn <- dbConnect(RSQLite::SQLite(), "22-04-29-playback-fm-top-pop.db")
dfSongs <- dbGetQuery(conn, 'SELECT * FROM tracks')
dim(dfSongs)

## [1] 3595   12

dfArtists <- dbGetQuery(conn, 'SELECT * FROM artists')
dbDisconnect(conn)
```

We used RSQLite to separate the raw data into two dataframes: dfSongs and dfArtists. dfSongs contains information about the songs, and dfArtists stores information about the artists.

Data Cleaning

Artist Dataset

Cleaning + Subset of Interest

```
dfArtists[dfArtists == "nan"] <- NA
dfArtistsInterest <- dfArtists %>%
   dplyr::select(artist_id, type, area.name, gender) %>%
   as.data.frame()
```

Then, we created a new dataframe dfArtistsInterest, which is a subset of the dataframe dfArtists. dfArtistsInterest contains four columns: the id of the artist, the type (whether the artist is a band or soloist), the area/region the song was produced/released, and the gender of the artist.

Merge Artists with Songs

```
dfSongsArtists <- merge(dfSongs,dfArtistsInterest,by="artist_id")
dim(dfSongsArtists)

## [1] 3595   15
```

We then merged the dfSongs dataframe with the dfArtistsInterest dataframe.

Filter out NA's in Important Parameters

```
cleaned_df <-dfSongsArtists %>%
  filter(!is.na(lyrics))  %>%
  filter(!is.na(artist))  %>%
  filter(!is.na(area.name)) %>%
  filter(!is.na(type))  %>%
  filter(!is.na(last_fm_listeners))  %>%
  as.data.frame()
dim(cleaned_df)

## [1] 2862   15
```

In cleaned_df, we kept only the rows in which lyrics, artist, area.name, type, and last_fm_listeners are not NA.

Gender

```
unique(cleaned_df$gender)
cleaned_df <- cleaned_df %>%
    mutate(gender = replace(gender, gender == "other", "non-binary"))
cleaned_df <- cleaned_df %>%
    mutate(gender = replace(gender, is.na(gender), "group"))
cleaned_df %>% count(gender)
cleaned_df <- cleaned_df %>% filter(gender != "non-binary")
dim(cleaned_df)
```

In this section, we cleaned the variable Gender. Before cleaning, the possible values for Gender are NA, male, female, other, and non-binary. We changed all "other" values to "non-binary." Then, for the NAs, we changed their value to "group" since a gender of NA probably means the artist is a band, not a soloist. We found that most artists were bands (groups). Out of the soloists, male artists were the most prevalent. Since the number of non-binary singers is small, we unfortunately removed them from our dataset.

area.name -> country

```
cleaned_df <- cleaned_df %>%
    mutate(country = dplyr::recode(area.name, "Los Angeles" = "United
States", "Boston" = "United States", "Malvern" = "United States", "Olympia
Fields" = "United States", "Alpharetta" = "United States", "Atlanta" =
"United States", "Nordrhein-Westfalen" = "Germany", "Saddle River" = "United
States", "Florida" = "United States", "Hollywood" = "United States",
"Manhattan" = "United States", "Vancouver" = "Canada", "Portland" = "United
States", "Quebec" = "Canada", "New York" = "United States", "Hawaii" =
"United States", "Devon" = "United States", "Toronto" = "Canada",
"[Worldwide]" = "Worldwide", "London" = "United Kingdom", "British Virgin
Islands" = "United Kingdom", "Brooklyn" = "United States", "Ann Arbor" =
"United States", "Salt Lake City" = "United States", "Rome" = "Italy",
"Nashville" = "United States", "Chicago" = "United States", "Houston" =
"United States", "Scotland" = "United Kingdom", "England" = "United Kingdom",
"Puerto Rico" = "United States"))
cleaned_df <- cleaned_df %>%
    mutate(region = dplyr::recode(country, "Austria" = "Non-UK Europe",
"Belgium" = "Non-UK Europe", "Austria" = "Non-UK Europe", "Denmark" = "Non-UK
Europe", "Finland" = "Non-UK Europe", "France" = "Non-UK Europe", "Germany" =
"Non-UK Europe", "Greece" = "Non-UK Europe", "Iceland" = "Non-UK Europe",
"Ireland" = "Non-UK Europe", "Italy" = "Non-UK Europe", "Moldova" = "Non-UK
Europe", "Netherelands" = "Non-UK Europe", "Norway" = "Non-UK Europe",
"Romania" = "Non-UK Europe", "Russia" = "Non-UK Europe", "Switzerland" =
"Non-UK Europe", "Sweden" = "Non-UK Europe", "Spain" = "Non-UK Europe",
"Netherlands" = "Non-UK Europe", "Senegal" = "Africa", "Guinea" = "Africa",
"Jamaica" = "Africa", "Morocco" = "Africa", "South Africa" = "Africa",
"Bahamas" = "Non-Canada/US Americas", "Barbados" = "Non-Canada/US Americas",
"Jamaica" = "Non-Canada/US Americas", "Colombia" = "Non-Canada/US Americas",
"Panama" = "Non-Canada/US Americas", "Saint Vincent and The Grenadines" =
"Non-Canada/US Americas", "Japan" = "Misc", "South Korea" = "Misc",
"Worldwide" = "Misc", "Philippines" = "Misc", "Australia" = "Oceania", "New
```

```
Zealand" = "Oceania"))
# cleaned_df %>% count(region)
cleaned_df <- cleaned_df %>%
    mutate(region = dplyr::recode(region, "Africa" = "Misc", "Non-Canada/US
Americas" = "Misc", "Oceania" = "Misc"))
cleaned_df %>% count(region)

##           region    n
## 1         Canada  131
## 2           Misc  129
## 3  Non-UK Europe  259
## 4 United Kingdom  641
## 5  United States 1688
```

Next, we condensed the variable Region so as to limit the number of different categories. We started by replacing the names of cities and states in the variable to their respective countries (e.g: Boston → United States). We then went on to categorize countries with less frequent appearances in the data into their larger affiliated regions (e.g: Austria → Non-UK Europe) and others without clear regional affiliations into Misc. Eventually, we decided to combine the regions Africa, Non-Canada/US Americas and Oceania and place them under Misc as well. By the end of this, we ended up with 5 main regions: Canada, United States, United Kingdom, Non-UK Europe and Misc.

Type

```
unique(cleaned_df$type)

## [1] "Group"  "Person" "Other"

cleaned_df %>% count(type)

##     type    n
## 1  Group 1161
## 2  Other    1
## 3 Person 1686

dim(cleaned_df)

## [1] 2848   17

cleaned_df <- cleaned_df %>% filter(type %in% c("Group", "Person"))
dim(cleaned_df)

## [1] 2847   17
```

For this step, we wanted to categorize the types of artists into two distinct types: soloists (Person) and bands (Group). However, while doing this, we encountered a bug in which, despite cleaning the data so that the variable Type would only contain Person or Group, there was one row that had "Other" as its data Type. We took out this "Other," though we were unable to figure out why it was there.

Filter out Mismatches in Lyrics

```
#
cleaned_df$lyrics <- str_replace_all(cleaned_df$lyrics,"[\\s]+", " ")
cleaned_df$cleaned_lyrics <-
  str_replace_all(cleaned_df$lyrics, 'Chap\\. [0-9]', NA_character_) %>%
  str_replace_all(., 'Listening Log', NA_character_) %>%
  str_replace_all(., 'Favorite Songs Of', NA_character_) %>%
  str_replace_all(., 'Chapter [0-9]', NA_character_) %>%
  str_replace_all(., 'New Music ', NA_character_) %>%
  str_replace_all(., 'Nominees', NA_character_) %>%
  str_replace_all(., 'Best Songs of ', NA_character_) %>%
  str_replace_all(., "[0-9]+ U S", NA_character_) %>% # Court Cases
  str_replace_all(., "[0-9]+ U.S", NA_character_) %>% # Court Cases
  str_replace_all(.,"[ ]+", " ") %>%
  str_replace(., ".*Lyrics", "") %>%
  str_replace(., "[0-9]*Embed$", "")

cleaned_df <- cleaned_df %>%
  filter(!is.na(cleaned_lyrics)) %>%
  filter(levenshteinSim(track, str_match(lyrics, "(.*)Lyrics")[,2]) > .5) %>%
# There are some false positives, when there are other languages
  as.data.frame()

# Filter away songs with less than 100 chars, must have been a mistake in
lyrics
cleaned_df <- cleaned_df %>% filter(nchar(cleaned_lyrics) > 100)
dim(cleaned_df)

## [1] 2669   18
```

Now on to cleaning the lyrics! For this, we started by identifying the "problematic" lyrics in the data, particularly ones that did not contain the actual lyrics. We then replaced these strings with NA characters and proceeded to remove them. Another method we used to clean the lyrics was to use the levenshteinSim function. The variable lyrics is formatted such that it should start with the name of the track, followed by the string 'Lyrics' and then the actual lyrics of the song. The function allowed us to compare for similarity between the title of the track and the initial part of the "lyrics". If the two strings were not at least 50% similar, we removed them.

Readability

```
cleaned_corpus <- corpus(
  cleaned_df,
  docid_field = "track_id",
  text_field = "cleaned_lyrics",
  unique_docnames = TRUE
)
```

Then, we quantified the variable lyrics so that we could use it for our data analysis. One such method is to use the function textstat_readability, which could only read in the corpus data

type. Thus, we used the corpus function to convert our cleaned_df dataframe into a corpus data type.

```
# We use FORCAST because it's one of the only readability metrics that don't look for sentences
cleaned_df$forcast <- textstat_readability(
  cleaned_corpus,
  measure = "FORCAST",
  min_sentence_length = 3,
  max_sentence_length = 10000,
)$FORCAST
cleaned_df$mean_word_syllables <- textstat_readability(
  cleaned_corpus,
  measure = "meanWordSyllables",
  remove_hyphens = TRUE,
  min_sentence_length = 3,
  max_sentence_length = 10000,
)$meanWordSyllables
```

Then, using the function textstat_readability, we calculated the FORCAST readability of the lyrics, which is basically the proportion of one syllable words to the total number of words. We chose the FORCAST metric over other metrics because it is one of the only metrics that don't look for sentences. Since most lyrics do not contain fully formed grammatical sentences with periods, the FORCAST metric is suitable for our data. We additionally add a second readability metric, which is simply the mean number of syllables for a word in a song.

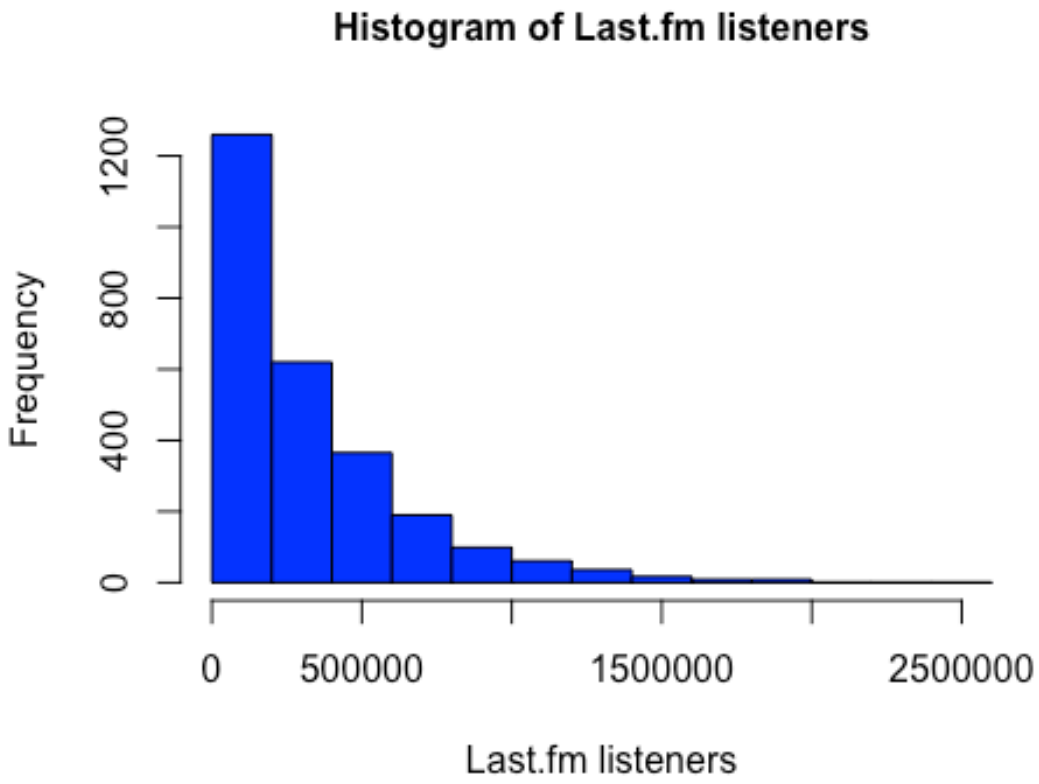
Filter to Relevant Details

```
subset_df <- cleaned_df %>%
   dplyr::select(year, type, region, gender, last_fm_listeners,
artist_appearances, rank, forcast, mean_word_syllables)
subset_df$log_listeners <- log(subset_df$last_fm_listeners)
attach(subset_df)
```

Descriptive Plots

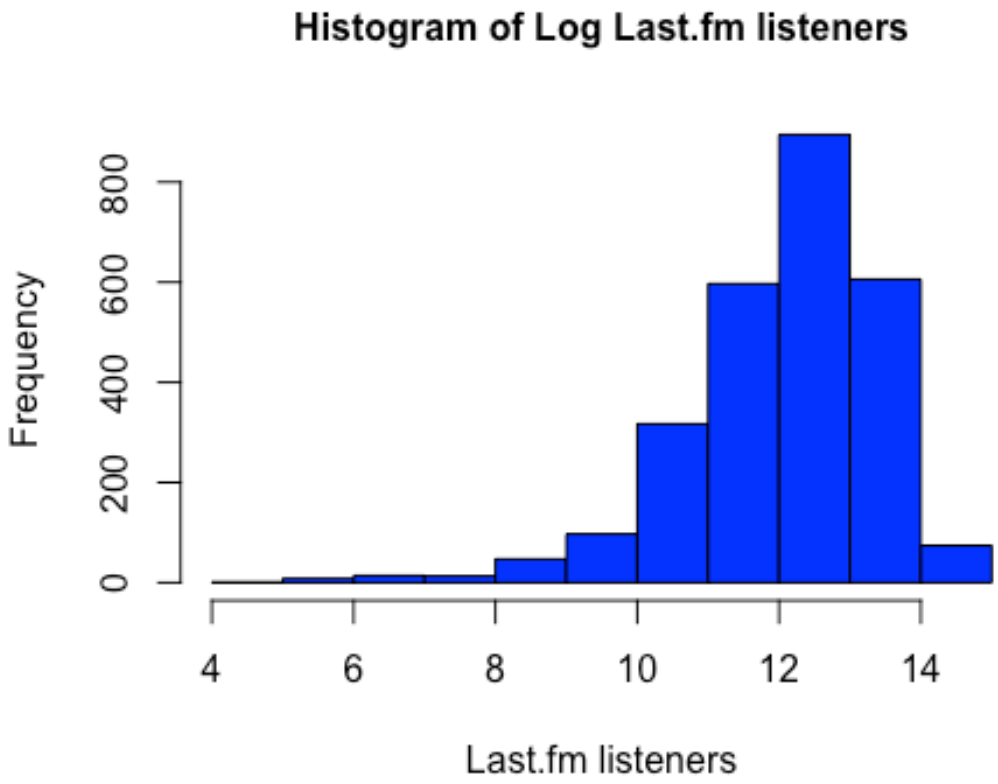
Histograms

```
hist(last_fm_listeners, main="Histogram of Last.fm listeners", xlab="Last.fm listeners", ylab = "Frequency", cex.main = 1, cex.lab = 1, col = "blue")
```



The histogram of last.fm listeners is heavily right-skewed.

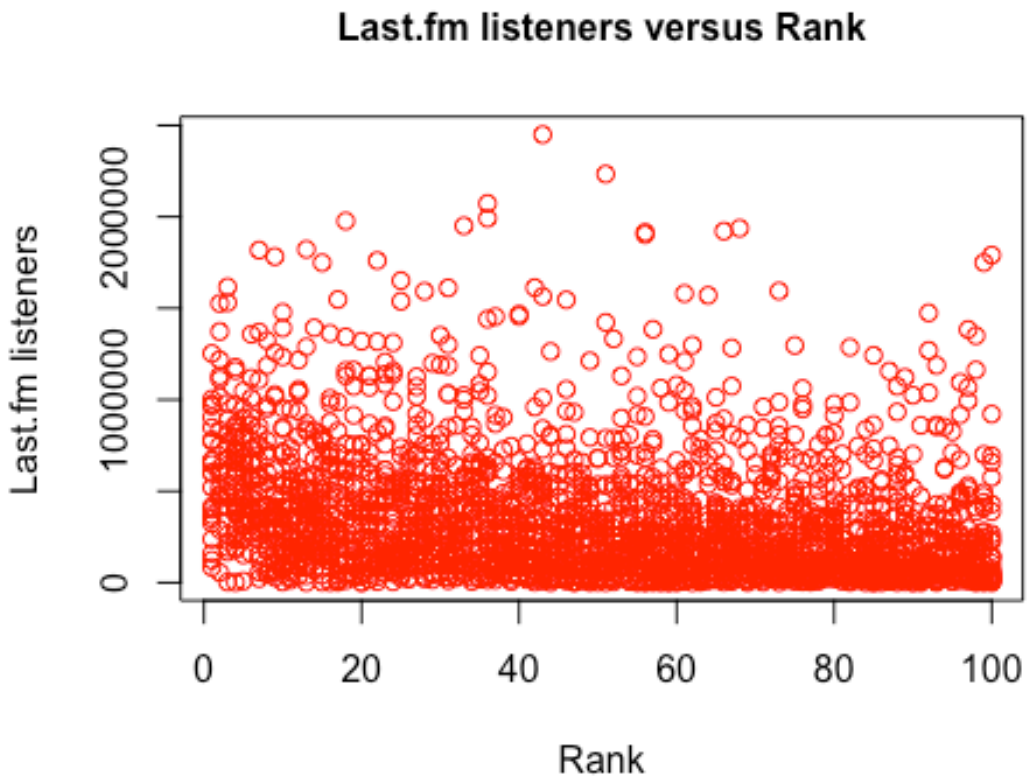
```
hist(log_listeners, main="Histogram of Log Last.fm listeners", xlab="Last.fm
listeners", ylab = "Frequency", cex.main = 1, cex.lab = 1, col = "blue")
```



Thus, we applied a log transformation on last.fm listeners to see if it might solve the skew problem. However, the log of last.fm listeners turns out to be left-skewed. This suggests we might need to do a different transformation on last.fm listeners.

Scatterplots

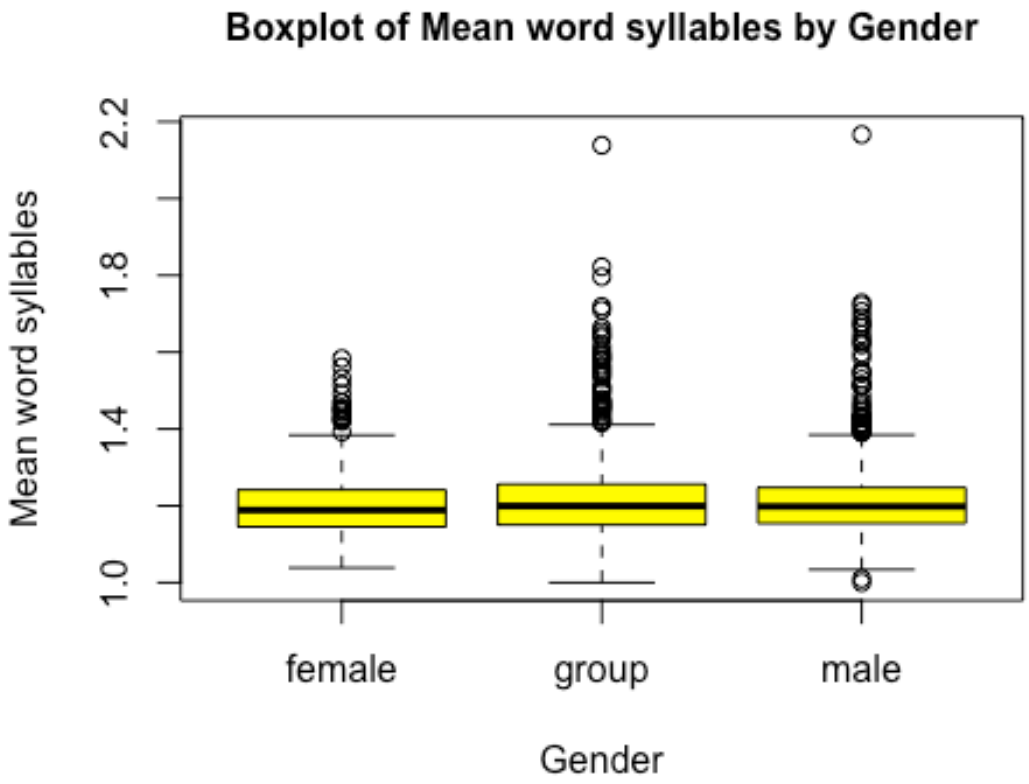
```
plot(last_fm_listeners ~ rank, main = "Last.fm listeners versus Rank", xlab =
"Rank", ylab = "Last.fm listeners", cex.main = 1,
     cex.lab = 1, col = "red")
```



In this scatterplot, it appears that most of the data seems to be concentrated around the less than 500,000 number of listeners mark. There also does not appear to be any indication of a clearly defined relationship between last.fm listeners and rank. However, when looked at carefully, there seems to be a slight downward trend as the rank increases in numeric value (i.e. as the song places lower on the chart). It is also quite interesting that there are data points beyond the 1 million mark for ranks numerically above 50, which is surprising considering how we might assume that songs lower on the chart ranks would have fewer cumulative listeners. This might hint to what we could refer to as the "staying power" of a song, which refers to the popularity of the song across its existence on last.fm.

Boxplots

```
boxplot(mean_word_syllables ~ gender, main = "Boxplot of Mean word syllables
by Gender", ylab = "Mean word syllables", xlab = "Gender", cex.main = 1,
        cex.lab = 1, col = "yellow")
```

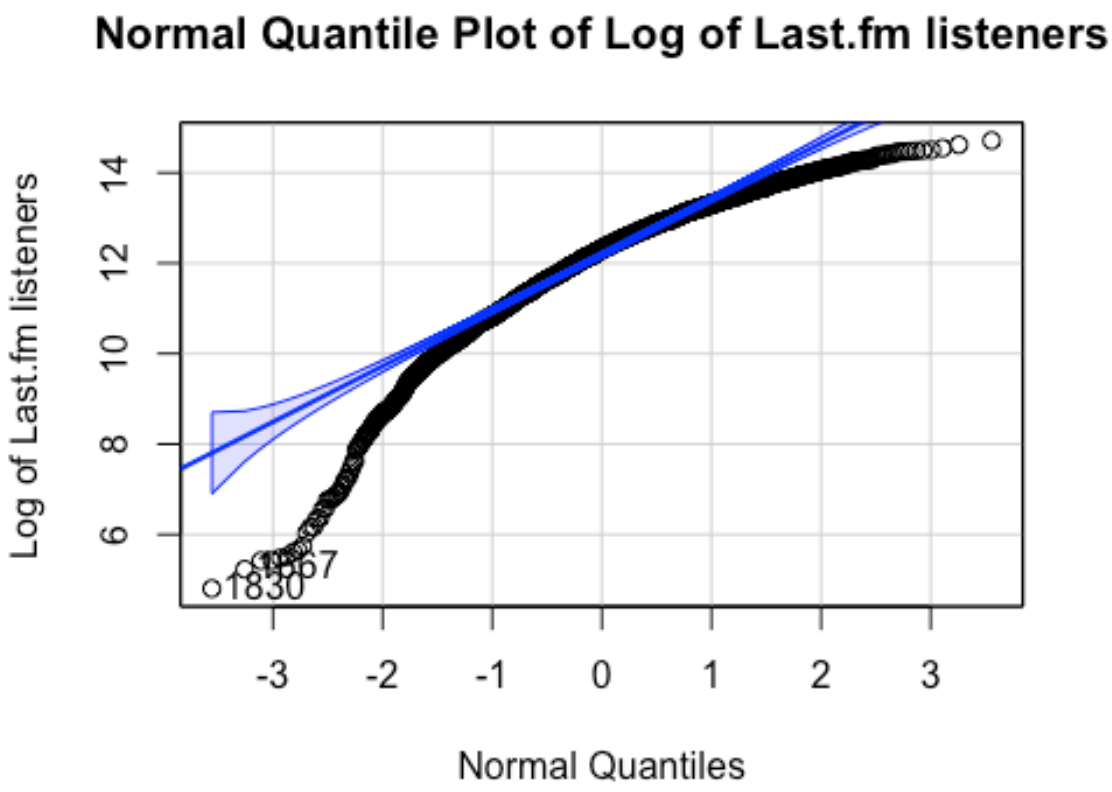


The boxplots of mean word syllables by gender show that the mean word syllables across genders have similar medians and interquartile ranges. The genders Group (Band) and Male, however, seem to have a lot more outliers than Female.

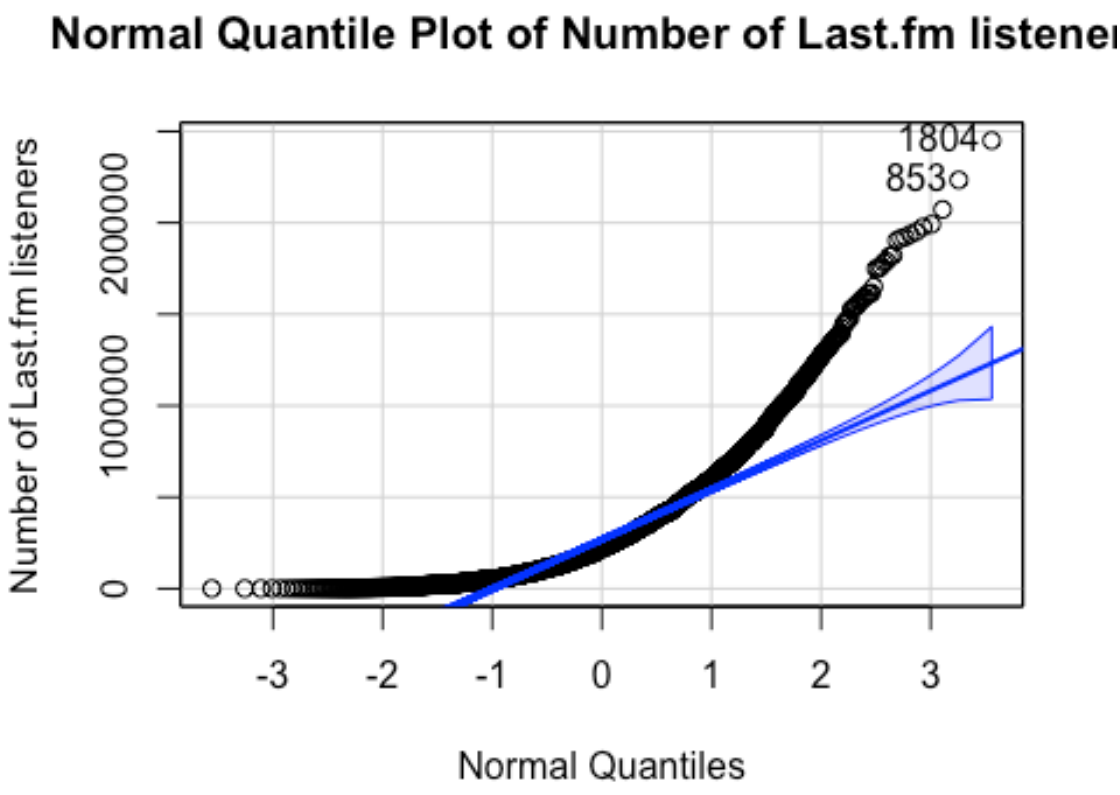
Analysis/Tests

Normal quantile plots

```
qqPlot(log_listeners, xlab = "Normal Quantiles", ylab = "Log of Last.fm
listeners",
       main = "Normal Quantile Plot of Log of Last.fm listeners")
```




```
## [1] 1830 1667

qqPlot(last_fm_listeners, xlab = "Normal Quantiles", ylab = "Number of
Last.fm listeners",
       main = "Normal Quantile Plot of Number of Last.fm listeners")
```

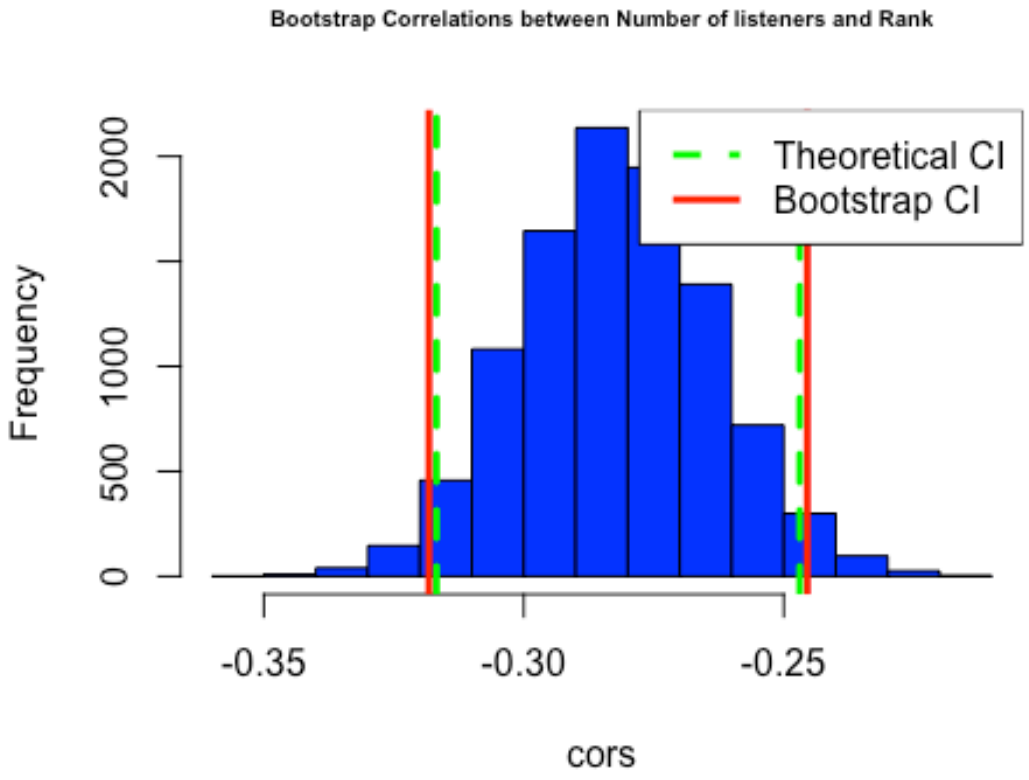



```
## [1] 1804  853
```

Correlation between number of listeners and rank (correlation, bootstrapped correlation)

```
# Calculate and report the correlation along with the results of a parametric
test of the significance of the correlation
(cor1 <- cor(last_fm_listeners, rank))

## [1] -0.2822321

cor1test <- cor.test(last_fm_listeners, rank)

# Calculate a 95% bootstrap confidence interval for the true correlation
N <- 10000
cors <- rep(NA, N)
for (i in 1:N)
{
  # listeners and rank have the same length, so we can use either
  s <- sample(1:length(last_fm_listeners), length(last_fm_listeners), replace
= TRUE)

  cors[i] = cor(last_fm_listeners[s], rank[s])
}
```

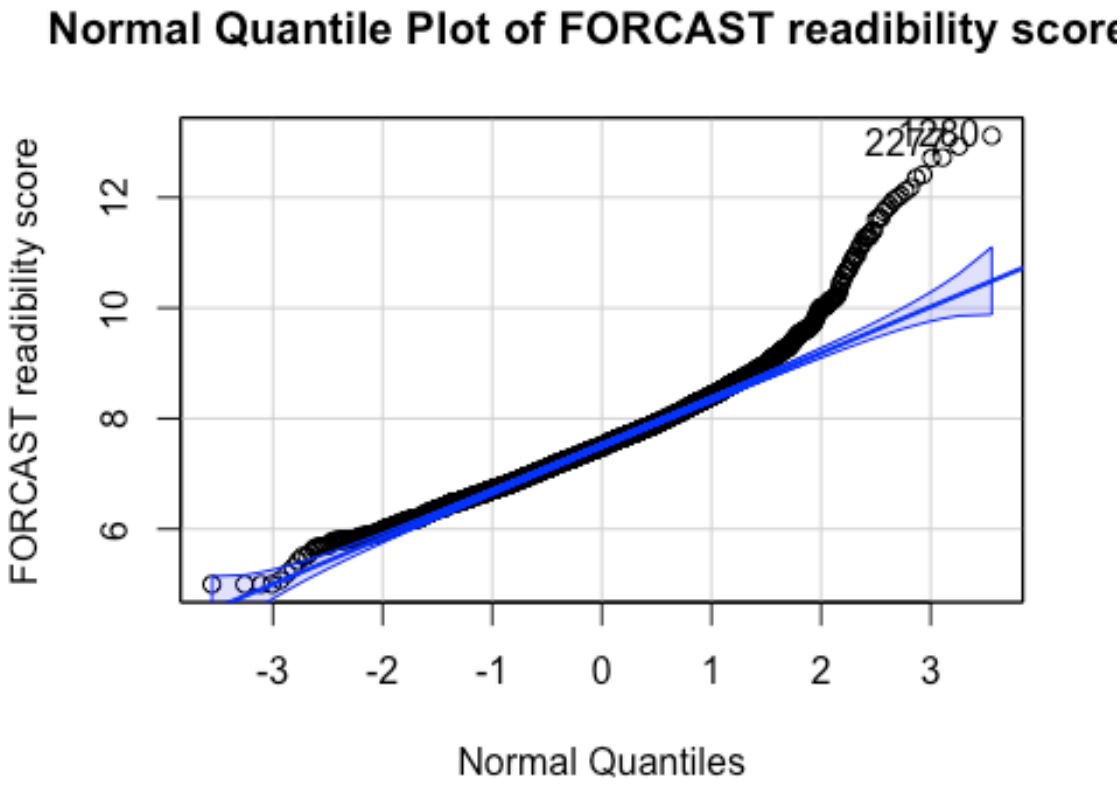
```r
corsCI <- quantile(cors, c(0.025, 0.975))
print("95% confidence interval for the true correlation")

## [1] "95% confidence interval for the true correlation"

corsCI

##       2.5%      97.5%
## -0.3182347 -0.2454717

# Display the results on a histogram (bootstrapped sample correlations,
bootstrap confidence interval, and theoretical confidence interval)
hist(cors, main = "Bootstrap Correlations between Number of listeners and
Rank", col = "blue", cex.main = .6)
abline(v = corsCI, lwd = 3, col = "red")
abline(v = cor1test$conf.int, lwd = 3, col = "green", lt = 2)
legend("topright", c("Theoretical CI", "Bootstrap CI"), lwd = 3, col =
c("green", "red"), lty = c(2, 1))
```



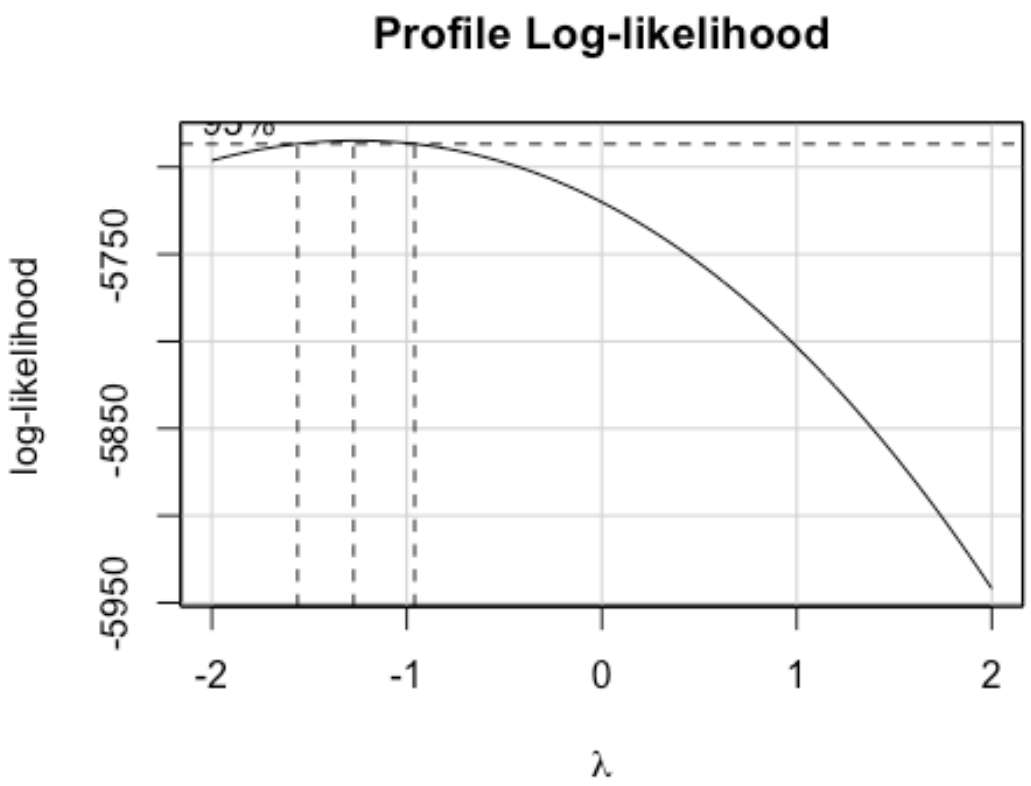
There is a weak negative correlation between the raw number of listeners and rank. We also calculated the bootstrapped correlation between these two variables. The bootstrapped confidence intervals and theoretical confidence intervals appear to be the same, though the theoretical CI is slightly narrower.

Difference in FORCAST readibility between male and female artists (t-test)

```
# Check the normality of FORCAST data
qqPlot(forcast, main = "Normal Quantile Plot of FORCAST readibility scores",
ylab = "FORCAST readibility score", xlab = "Normal Quantiles")
```




```
## [1] 1280 2277

# The normal quantile plot is not exactly linear, so we perform a Box Cox
transformation
trans1 <- boxCox(aov(forcast[gender != "group"] ~ gender[gender != "group"]))
```

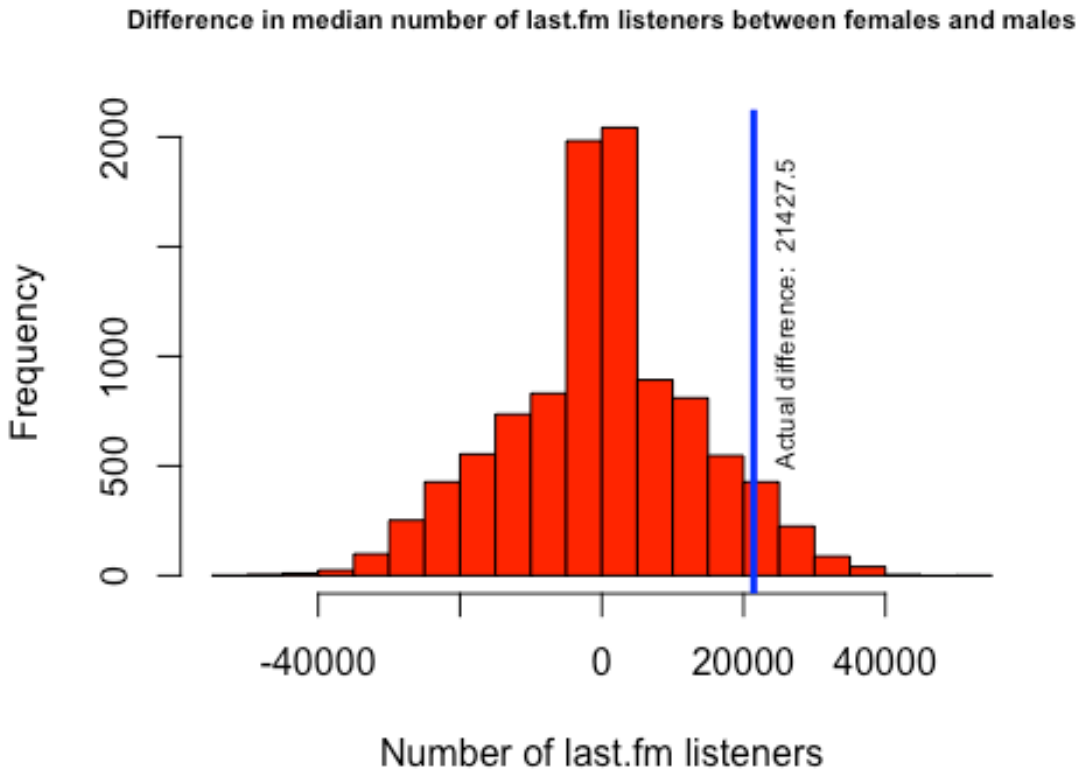



```
(pow1 <- trans1$x[which.max(trans1$y)])

## [1] -1.272727

# Apply the transformation to FORCAST
trans_forcast <- forcast^-1

# Omit "Group" as a type to compare female and male artists
test1 <- t.test(trans_forcast[gender != "group"] ~ gender[gender != "group"])
test1

##
##  Welch Two Sample t-test
##
## data:  trans_forcast[gender != "group"] by gender[gender != "group"]
## t = 2.5885, df = 1595.9, p-value = 0.009727
## alternative hypothesis: true difference in means between group female and
group male is not equal to 0
## 95 percent confidence interval:
##  0.0004790204 0.0034758690
## sample estimates:
## mean in group female   mean in group male
##            0.1348332            0.1328558
```

We then wanted to see if there was a significant difference in FORCAST readability between male and female artist produced songs. First, we checked to see if FORCAST is normally distributed. The normal quantile plot is not linear, suggesting that FORCAST is not normally distributed. Thus, we performed a box-cox transformation and got a lambda of around -1. Thus, we transformed the FORCAST variable by raising it to -1 power. The p-value from the Welch's t-test is 0.010, which is less than 0.05. Thus, we can reject the null hypothesis and conclude that there is a significant difference in the transformed FORCAST readability between male and female artists.

Difference in number of listeners between female and male artists (permutation test)

```
# Transformed version of gender containing only solo artists (omitting the
type "group")
soloGen <- gender[gender != "group"]

actualDiff <- median(last_fm_listeners[soloGen == "female"]) -
median(last_fm_listeners[soloGen == "male"])

diffValues <- rep(NA, N)
for (i in 1:N)
{
  fakeGender <- sample(soloGen)
  diffValues[i] <- median(last_fm_listeners[fakeGender == "female"]) -
median(last_fm_listeners[fakeGender == "male"])
}

hist(diffValues, xlab = "Number of last.fm listeners", main = "Difference in
median number of last.fm listeners between females and males",
     col = "red", breaks = 20, cex.main = 0.7)
abline(v = actualDiff, col = "blue", lwd = 3)
text(actualDiff + 4500, 1200, paste("Actual difference: ", round(actualDiff,
3)), srt = 90, cex = 0.7)
```

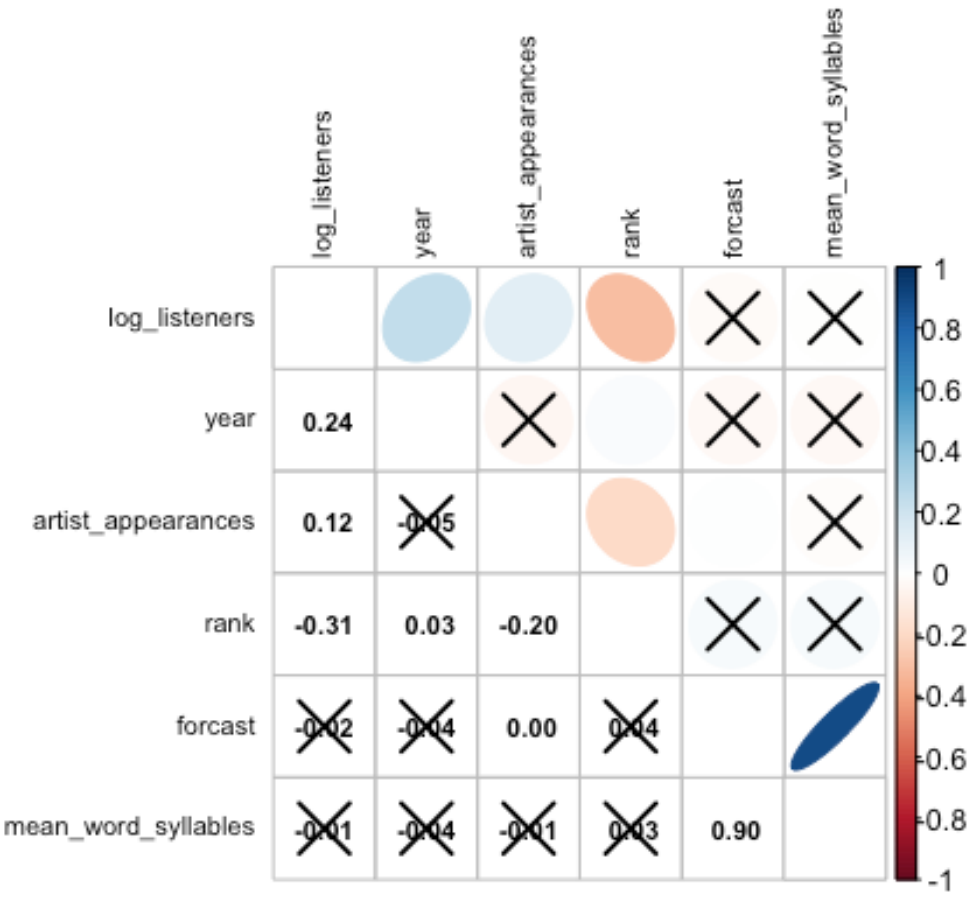



```
mean(abs(diffValues) >= abs(actualDiff))

## [1] 0.1374
```

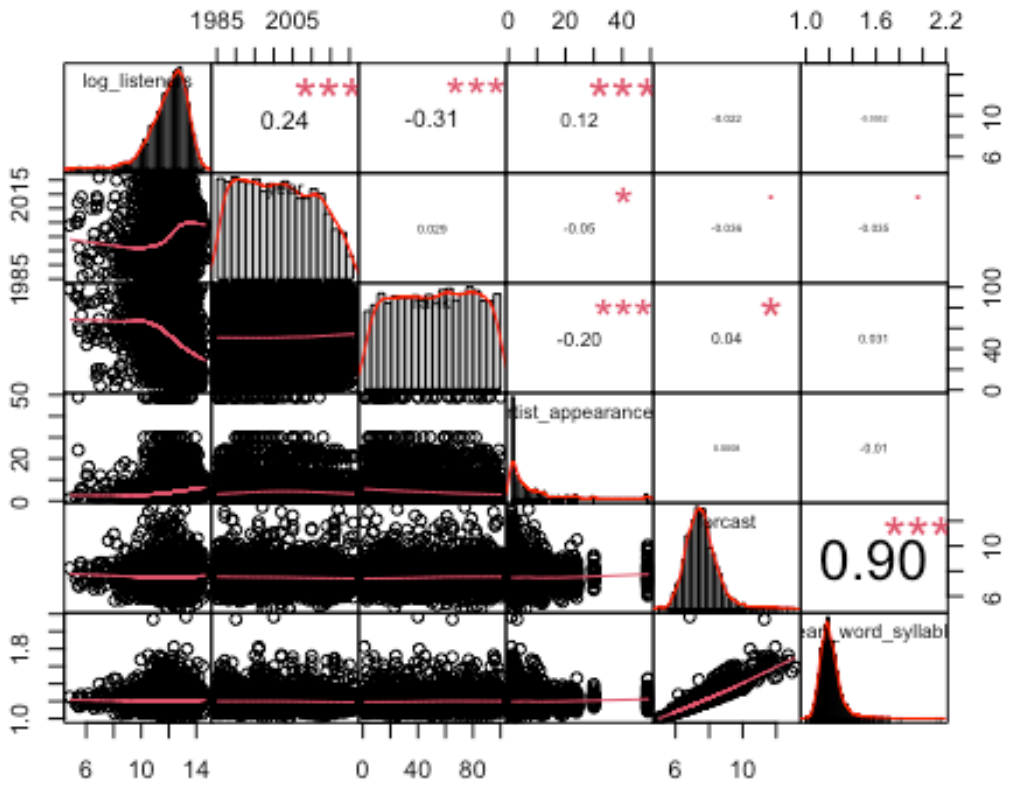
The p-value is greater than any alpha value, so we fail to reject the null hypothesis. Thus, there is no significant difference in last.fm listeners between male and female artists.

Mixed Correlation Plots

```
# Calculate the pairwise correlations
cor1 <- cor(subset_df[ , c("log_listeners", "year", "artist_appearances",
"rank", "forcast", "mean_word_syllables")])

# Calculate the pairwise correlation significances
sigcorr <- cor.mtest(subset_df[ , c("log_listeners", "year", "rank",
"artist_appearances", "forcast", "mean_word_syllables")], conf.level = .95)

#Use corrplot.mixed to display confidence ellipses, pairwise correlation
values, and put on 'X' over non-significant values.
corrplot.mixed(cor1, lower.col="black", upper = "ellipse", tl.col = "black",
number.cex=.7,
               tl.pos = "lt", tl.cex=.7, p.mat = sigcorr$p, sig.level = .05)
```



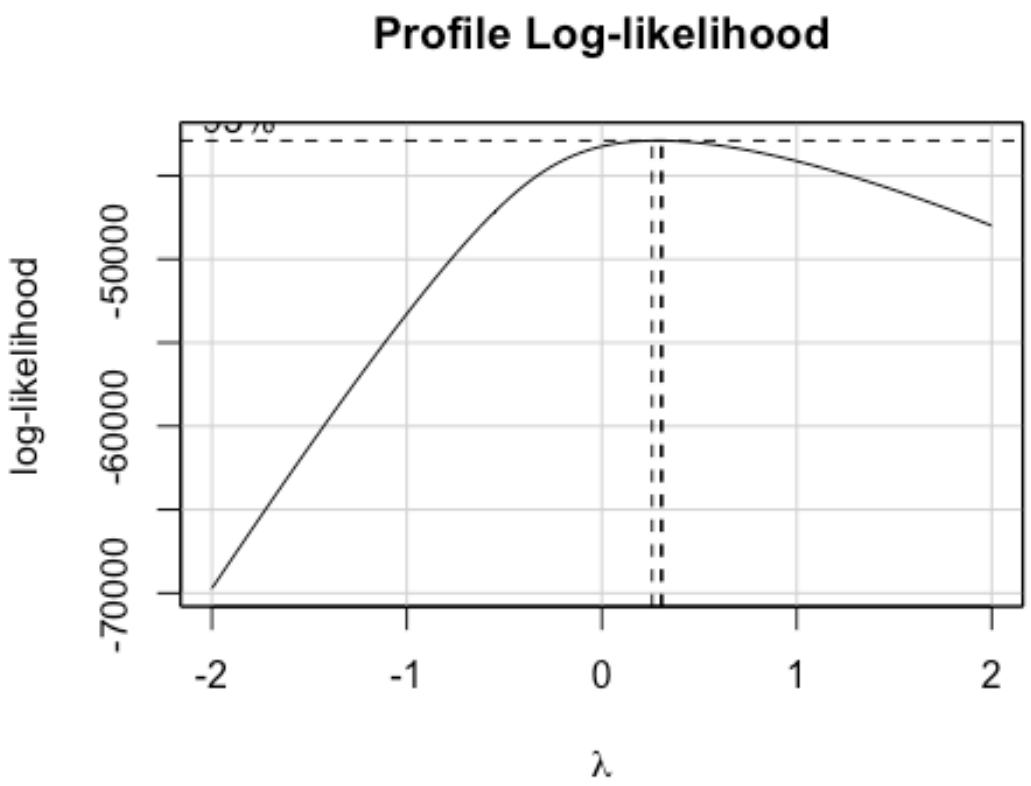

```
chart.Correlation(subset_df[ , c("log_listeners", "year", "rank",
"artist_appearances", "forcast", "mean_word_syllables")], histogram = TRUE,
pch = 19)
```



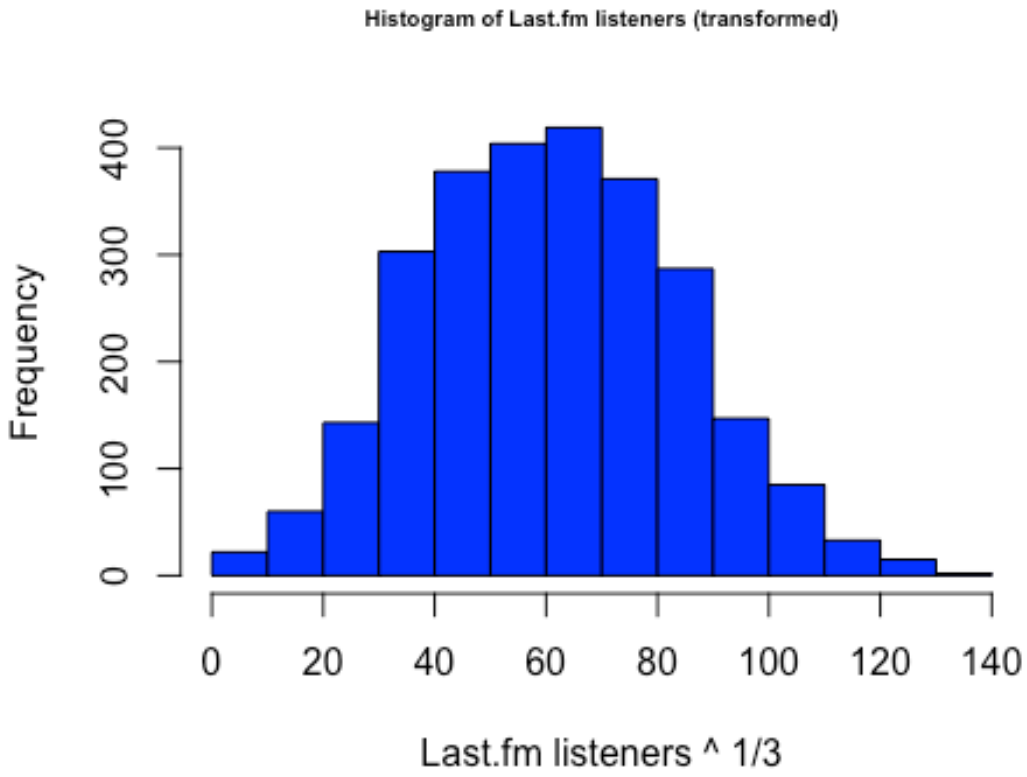
The only somewhat strong(er) positive correlation that exists is between the log number of listeners and the year. On the other hand, the only somewhat strong(er) negative correlation is between the log number of listeners and the rank. There also appears to be a highly strong correlation between forcast and mean word syllables, which makes sense since the two predictor variables are directly related. This is an example of collinearity.

Box Cox transformation

```
# Perform a Box Cox transformation
trans2 <- boxCox(lm(last_fm_listeners ~ gender + year + rank + region +
artist_appearances + gender*year + year*rank + region*gender, data =
subset_df))
```



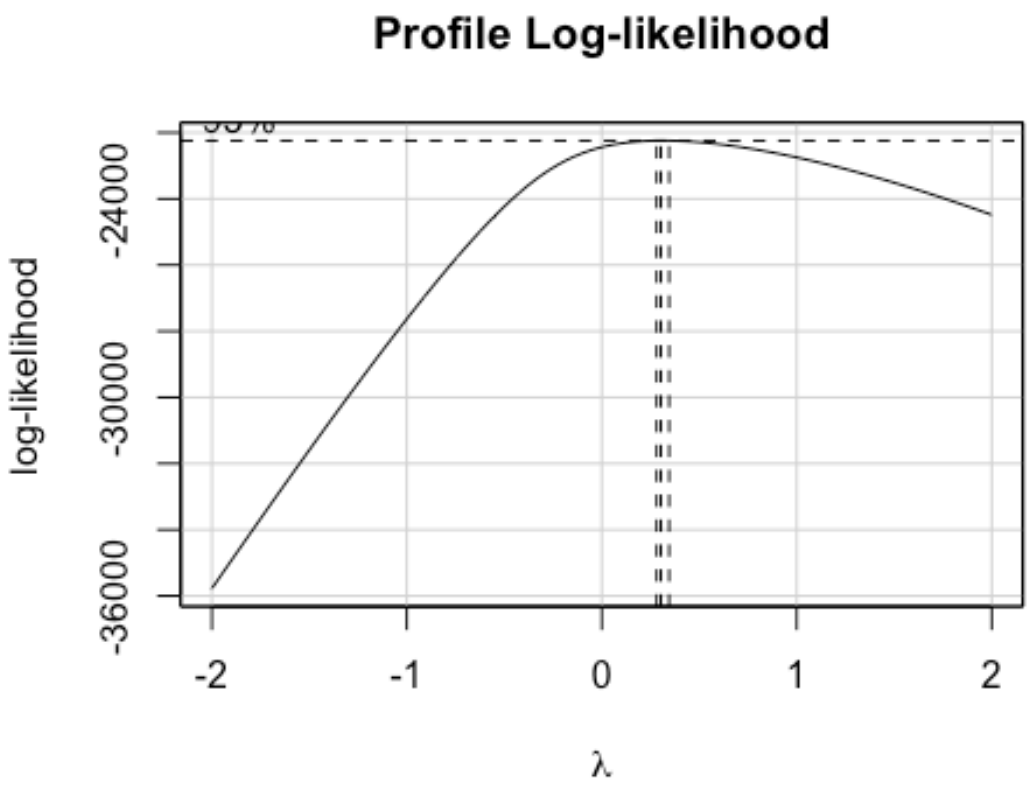

```
(pow2 <- trans2$x[which.max(trans1$y)])

## [1] -1.272727

# Apply the transformation to the variable
trans_listeners <- last_fm_listeners^(1/3)

hist(trans_listeners, main="Histogram of Last.fm listeners (transformed)",
xlab="Last.fm listeners ^ 1/3", ylab = "Frequency", cex.main = 1, cex.lab =
1, col = "blue", cex.main = .6)
```



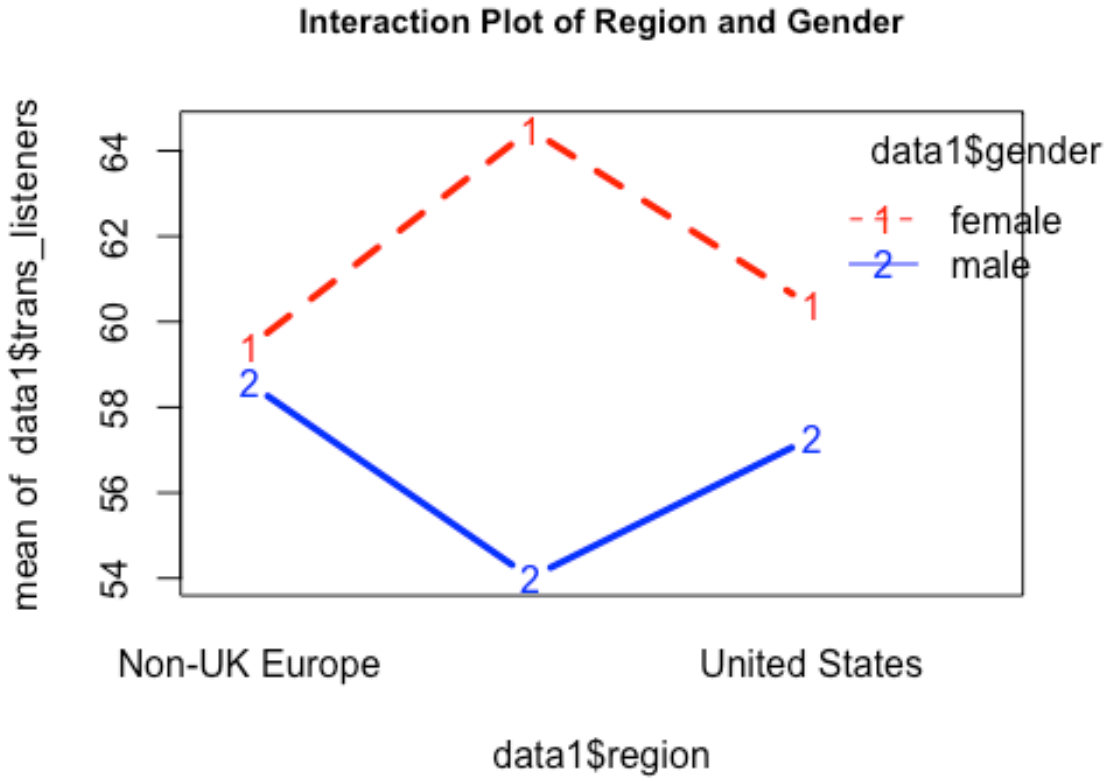
Earlier, we found that both the raw data and log transformed data of last.fm listeners produced skewed histograms. Here, we performed a Box Cox transformation on last.fm listeners to hopefully obtain a normal distribution. We got a lambda value of 0.303, and so we took the cube root of last.fm listeners. The histogram now looks a lot more normally distributed.

Two-way ANOVA

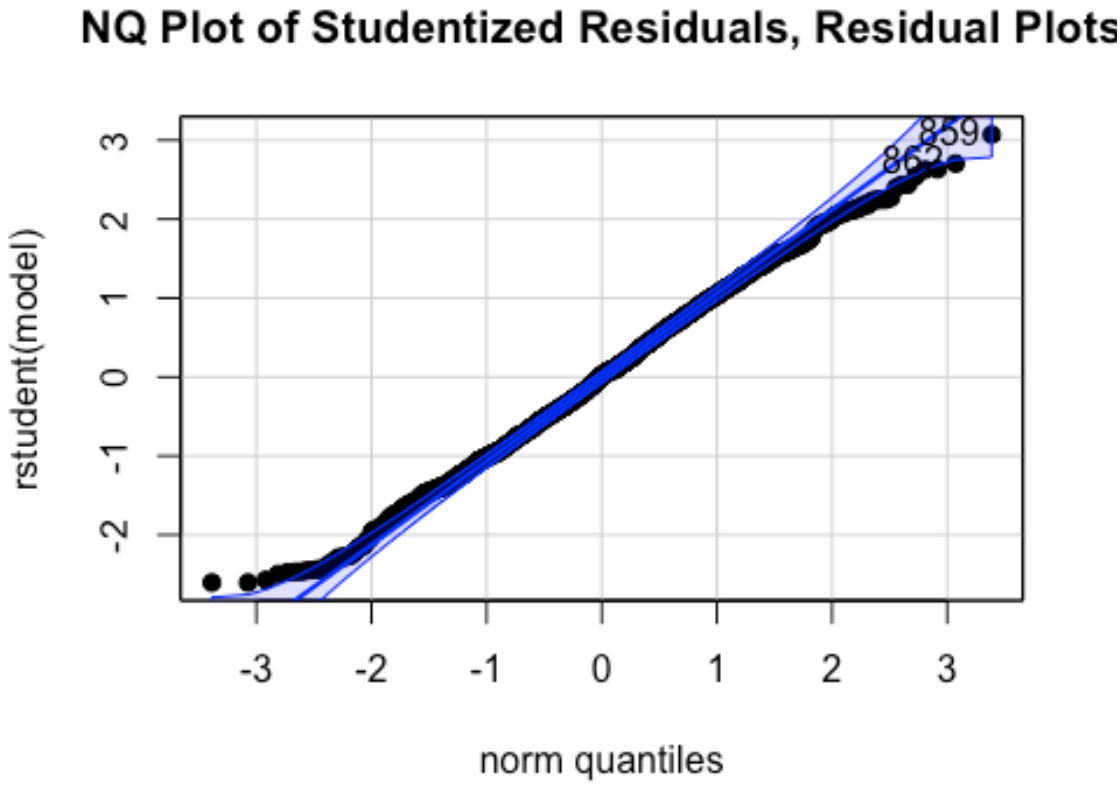
```
# We will perform a Two-way ANOVA, predicting listeners (transformed) by
Region (limited to U.S., U.K., and Non-UK Europe) and Gender (limited to solo
artists) as well as their interaction
# Transformed data set
data1 <- subset_df %>% filter(region %in% c("United States", "United
Kingdom", "Non-UK Europe")) %>% filter(gender != "group")

# Perform a Box Cox transformation
trans3 <- boxCox(aov(data1$last_fm_listeners ~ data1$region + data1$gender +
data1$region*data1$gender))
```

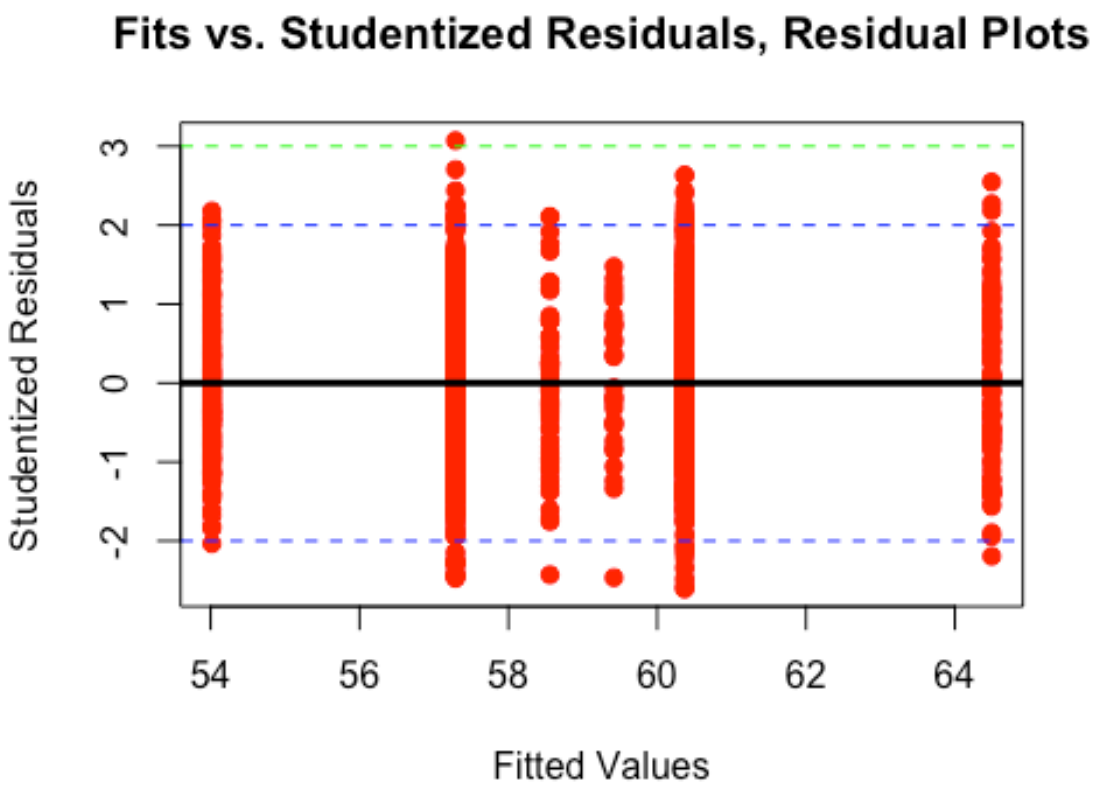



```
(pow3 <- trans3$x[which.max(trans3$y)])

## [1] 0.3030303

# The suggested lambda value is, once again, close to 1/3, so we use the
cube-root transformed version of last.fm listeners once again
data1$trans_listeners <- data1$last_fm_listeners^(1/3)

# Before performing a Two-way ANOVA, we will examine the interaction plots to
see if we expect a significant interaction
interaction.plot(data1$region, data1$gender, data1$trans_listeners, type =
'b', lwd = 3,
                 col = c("red", "blue", "black", "orange", "brown", "violet",
"dark green"),
                 main = "Interaction Plot of Region and Gender", cex.main =
.9)
```




```
# Perform the Two-way ANOVA and get results
test2 <- aov(data1$trans_listeners ~ data1$region + data1$gender +
data1$region*data1$gender)
Anova(test2, type = 3)

## Anova Table (Type III tests)
##
## Response: data1$trans_listeners
##                            Sum Sq   Df  F value  Pr(>F)
## (Intercept)                 98855    1 226.2202 < 2e-16 ***
## data1$region                 1455    2   1.6652 0.18953
## data1$gender                   14    1   0.0322 0.85755
## data1$region:data1$gender    3062    2   3.5038 0.03035 *
## Residuals                  614837 1407
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# We performed Tukey comparisons and discussed the results below, though we
omitted the code as the results take up too much space
# plot(TukeyHSD(test2), las = 1, cex = 0.8)

myResPlots2(test2)
```


NQ Plot of Studentized Residuals, Residual Plots
rstudent(model)
norm quantiles
859
862
-3
-2
-1
0
1
2
3



We then performed a Two-way ANOVA to predict the cube-root of last.fm listeners (transformed version) by gender, region, and their interaction. The interaction plot indicates that there is an interaction between region and gender, as the lines are not parallel. In the Two-way ANOVA, we find that region and gender are not significant indicators by themselves (p-value above 0.05); however, their interaction has a significant p-value at the 0.05 confidence level (approx. 0.030). The plot of two-way Tukey comparisons shows that there is a significant difference in three factor combinations: female is preferred over male in the U.K., female is preferred over male in the U.S., and female in the U.S. has more listeners than male in the U.K. The residuals seem to be approximately normally distributed as the normal quantile plot is linear, and the plot of fits versus residuals has only one significant outlier, though since there are many observations, this is negligible. There is also no evidence of heteroskedasticity as the standard deviation is more or less constant across different fitted values. Overall, the model assumptions have been reasonably met.

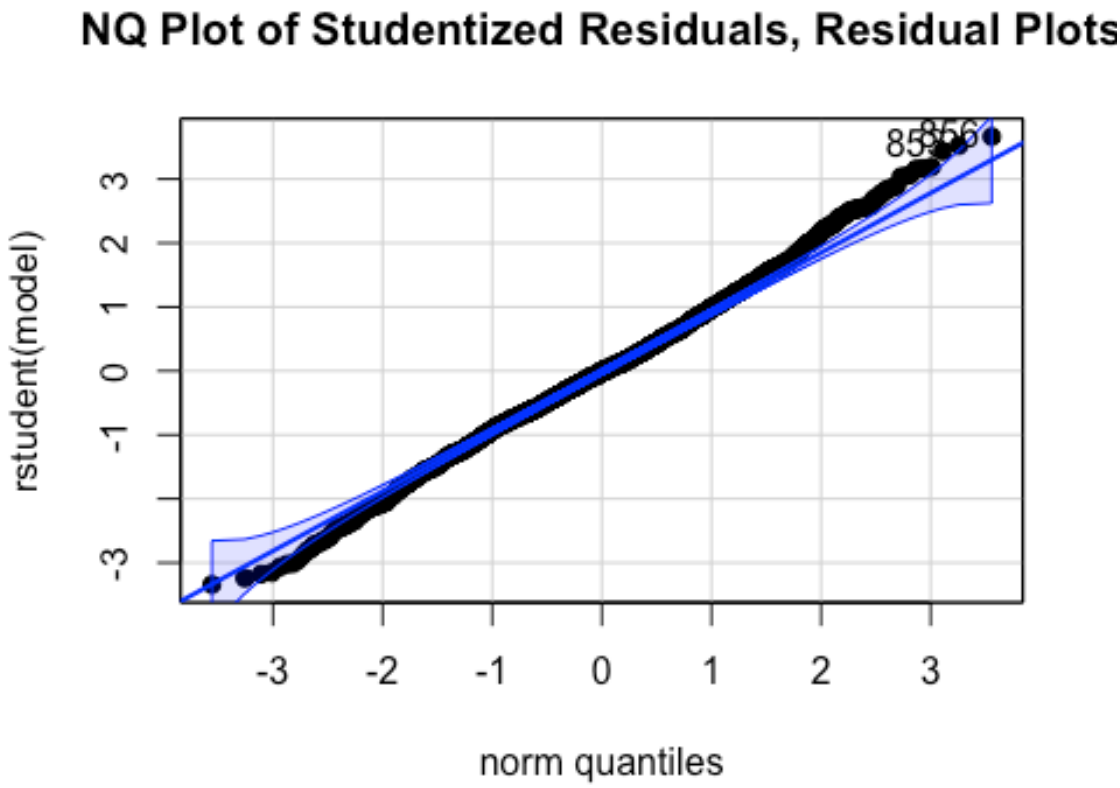
GLM and Backwards Stepwise Regression

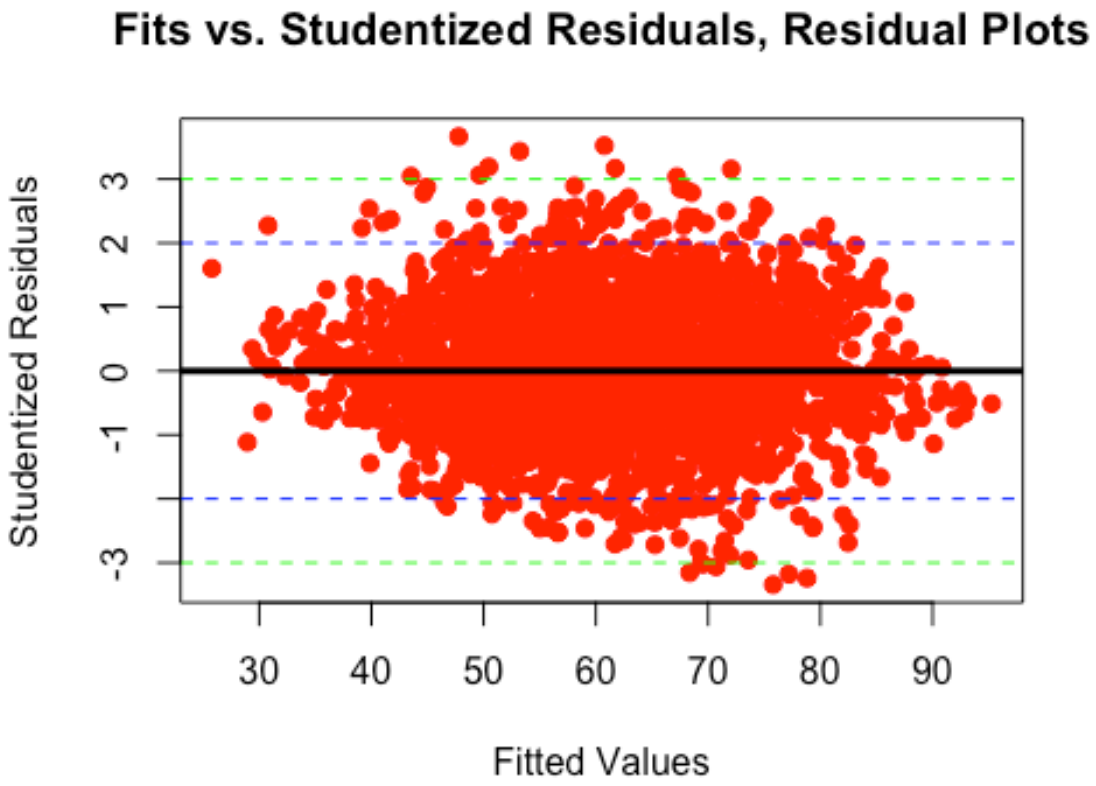
Lastly, we performed a backwards stepwise regression to assess which variables and their interactions are significant predictors of the transformed last.fm listeners. We included the variables gender, year, rank, region, and the number of artist appearances, as well as the interaction between gender and year, year and rank, and region and gender. Due to the strong positive correlation between mean_word_syllables and FORCAST, we omitted mean_word_syllables before performing backwards stepwise regression. We used a

significance of 0.001 since most of the p-values are very low to begin with. We started out by looking at the p-values of the interaction terms, eliminating first the interaction term with the highest p-value.

```
mod1 <- lm(trans_listeners ~ gender + year + rank + region +
artist_appearances + gender*year + year*rank + region*gender)
Anova(mod1, type = 3)

## Anova Table (Type III tests)
##
## Response: trans_listeners
##                      Sum Sq   Df  F value    Pr(>F)
## (Intercept)           52711    1 133.0798 < 2.2e-16 ***
## gender                13666    2  17.2514 3.599e-08 ***
## year                  56093    1 141.6177 < 2.2e-16 ***
## rank                   2714    1   6.8529  0.008900 **
## region                 4557    4   2.8765  0.021619 *
## artist_appearances    19076    1  48.1603 4.920e-12 ***
## gender:year           13871    2  17.5097 2.789e-08 ***
## year:rank              2926    1   7.3861  0.006616 **
## gender:region         12567    8   3.9658  0.000111 ***
## Residuals           1048839 2648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

mod2 <- lm(trans_listeners ~ gender + year + rank + region +
artist_appearances + gender*year + region*gender)
summary(mod2)

##
## Call:
## lm(formula = trans_listeners ~ gender + year + rank + region +
##     artist_appearances + gender * year + region * gender)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.263 -12.830  -0.293  12.180  72.717
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.986e+03  1.557e+02 -12.757  < 2e-16
***
## gendergroup                    4.892e+02  2.067e+02   2.367   0.0180 *
## gendermale                     1.186e+03  2.019e+02   5.875 4.75e-09
***
## year                           1.029e+00  7.773e-02  13.237  < 2e-16
***
## rank                          -2.703e-01  1.388e-02 -19.465  < 2e-16
***
## regionMisc                    -6.436e-01  4.364e+00  -0.147   0.8828
## regionNon-UK Europe           -1.721e+00  4.650e+00  -0.370   0.7113
```

```
## regionUnited Kingdom             -1.262e-01  3.407e+00  -0.037   0.9705
## regionUnited States              -5.959e+00  2.860e+00  -2.084   0.0373 *
## artist_appearances                3.538e-01  5.191e-02   6.816 1.15e-11
***
## gendergroup:year                 -2.383e-01  1.032e-01  -2.308   0.0211 *
## gendermale:year                  -5.933e-01  1.007e-01  -5.894 4.25e-09
***
## gendergroup:regionMisc           -9.916e+00  7.246e+00  -1.368   0.1713
## gendermale:regionMisc            -1.038e+00  5.940e+00  -0.175   0.8613
## gendergroup:regionNon-UK Europe  -1.375e+01  7.021e+00  -1.959   0.0503 .
## gendermale:regionNon-UK Europe    6.781e-01  5.913e+00   0.115   0.9087
## gendergroup:regionUnited Kingdom -4.253e+00  6.146e+00  -0.692   0.4890
## gendermale:regionUnited Kingdom  -4.643e+00  4.567e+00  -1.017   0.3094
## gendergroup:regionUnited States   6.842e-01  5.809e+00   0.118   0.9062
## gendermale:regionUnited States    2.994e+00  4.030e+00   0.743   0.4576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.93 on 2649 degrees of freedom
## Multiple R-squared:  0.2511, Adjusted R-squared:  0.2457
## F-statistic: 46.74 on 19 and 2649 DF,  p-value: < 2.2e-16

myResPlots2(mod2)
```


NQ Plot of Studentized Residuals, Residual Plots
rstudent(model)
norm quantiles
-3
-2
-1
0
1
2
3



The interaction between year and rank had a p-value of 0.01, greater than our significance 0.001. Thus, we took this term out of our model. In our new model, the remaining two interaction terms had p-values less than 0.001. Thus, we kept these two terms and ended the backwards regression. Our final model thus contains all variables we started out with except for the interaction between year and rank. Looking at the significant interactions, we see that cube-root of listeners is smaller for U.S. artists than for Canadian artists, the popularity (implied by cube-root of listeners) of groups decreases with year, and the popularity of males decreases with year. The The R-squared of our final model is 0.2502. Thus, about 25% of the variability in transformed last.fm listeners can be explained by the predictor variables. The normal quantile plot of the studentized residuals is linear, suggesting that the residuals are normally distributed. The fits vs studentized residuals shows no heteroskedasticity and contain minimal outliers. Thus, our final model fit is great!

Conclusion

We found that on average, female artists have a higher inverse FORCAST readibility score than male artists, where a higher inverse score corresponds to less complexity. We also found that there is no statistically significant difference in number of last.fm listeners between women and men. Additionally, we found that gender and region, taken alone, are not significant predictors of the cube-root of number of listeners. However, their interaction is significant, and we observe effects such as female artists being more popular than male artists

in the U.K as well as the U.S. or female artists in the U.S. having more listeners than male artists in the U.K. Finally, we fit a generalized linear model to predict the cube-root of number of listeners by relevant variables and interactions, and we performed backwards-stepwise regression to get a model with all predictors significant. The results showed us that we can predict the cube-root of number of listeners by taking into account gender, year, rank, region, artist appearances, as well as the interactions gender-year, year-rank, and gender-region. Overall, we now have a sense of what variables influence song popularity (in terms of number of listeners), and we have knowledge of some of the interesting interactions that occur between different factors.