Mehra_Kai_Assignment-3

Kai Mehra

2023-02-17

Libraries

```
# Load the {tidyverse}, {ggplot2}, {regclass}, {GGally}, and {bestNormalize}
# libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(regclass)
```

```
## Loading required package: bestglm
## Loading required package: leaps
## Loading required package: VGAM
## Loading required package: stats4
## Loading required package: splines
## Loading required package: rpart
## Loading required package: randomForest
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
##
## Important regclass change from 1.3:
```

```
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(bestNormalize)
```

Data

```
# Load the swift_spotify_data dataset using read_csv
swift_spotify_data <-
  read_csv("../Data/spotify_taylorswift.csv")
```

```
## New names:
## Rows: 171 Columns: 16
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (3): name, album, artist dbl (12): ...1, length, popularity, danceability,
## acousticness, energy, ins... date (1): release_date
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
dim(swift_spotify_data)
```

```
## [1] 171  16
```

```
colnames(swift_spotify_data)
```

```
##  [1] "...1"            "name"            "album"           "artist"
##  [5] "release_date"    "length"          "popularity"      "danceability"
##  [9] "acousticness"    "energy"          "instrumentalness" "liveness"
## [13] "loudness"        "speechiness"     "valence"         "tempo"
```

The `swift_spotify_data` data set is a data set from Kaggle user "JAN LLENZL DAGOHOY". The data set was taken from Spotify's API and contains song characteristic data of Taylor Swift's songs from the beginning of her career in 2006 to November 6th, 2021. There 168 songs in the data set and 12 characteristics were tracked for each song.

Research Quetion

Is the popularity of a Taylor Swift song related to the release year, length (in seconds), danceability, acousticness, energy, loudness, valence, or tempo of the song?

Taylor Swift is one of the most popular artists in the world, yet her music spans decades, genre, and tone. I want to empirically study the characteristics of her music to better understand what determines the popularity of her songs, or discover if it is even possible to figure out what makes her songs popular. In the future, I would want to apply this analysis to other artists to see if there are overarching trends in the popularity of music on Spotify.

Hypotheses

- H_0 - The popularity of a Taylor Swift song is not related to the release year, length (in seconds), danceability, acousticness, energy, loudness, valence, or tempo of the song
- H_1 - The popularity of a Taylor Swift song is related to at least one of the release year, length (in seconds), danceability, acousticness, energy, loudness, valence, or tempo of the song

Variables of Interest

Dependent Variable:

popularity: popularity is measured on a scale of 1-100, with 1 be very unpopular and 100 being very popular. This popularity measure is based upon Spotify's classified algorithm meaning the true understanding of the measure is not possible to fully comprehend.

Independent Variables:

- year: The year the song was released
- length_sec: The length of the song in seconds
- danceability: how suitable a track is for dancing based on a combination of musical elements. 0.0 is least danceable and 1.0 is most danceable.
- acousticness: a confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
- energy: a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.
- loudness: The overall loudness of a track in decibels (dB). Values typical range between -60 and 0 db.
- valence: A measure from 0.0 (very sad) to 1.0 (very happy) describing the musical positiveness conveyed by a track.
- tempo: The overall estimated tempo of a track in beats per minute (BPM).

These variables provide a comprehensive understanding of the general characteristics of a specific song. Additionally, since they are all continuous, numeric variables, they lend themselves to analysis using linear regression. Intuitively, these variables should have some impact on popularity as songs fitting into certain genres have higher levels of popularity than others. These variables help define genres; for example, a highly danceable, energetic, valent, and loud song would likely be a club/party song which can be incredibly popular.

Data Wrangling

```
# selcting the primary variables of interest
song_chars <-
  swift_spotify_data %>%
  select(
    name,
    popularity,
    release_date,
    length,
    danceability,
    acousticness,
    energy,
    loudness,
    valence,
    tempo
  )
```

```
# splitting the release_date variable into year, month, and day variabes
song_chars <-
  song_chars %>%
   separate(release_date, into = c("year", "month", "day"),
            sep = "-", convert = TRUE) %>%
  mutate(length_sec = length/1000) %>% # converting length from milsecs to secs
  select(-length)
```

```
song_chars <-
  song_chars %>%
  filter(!grepl("Voice Memo",name)) # filter out the "Voice Memo" observations
```

Swift recorded a few song explanation recordings under the "Voice Memo" title which do not classify as real songs.

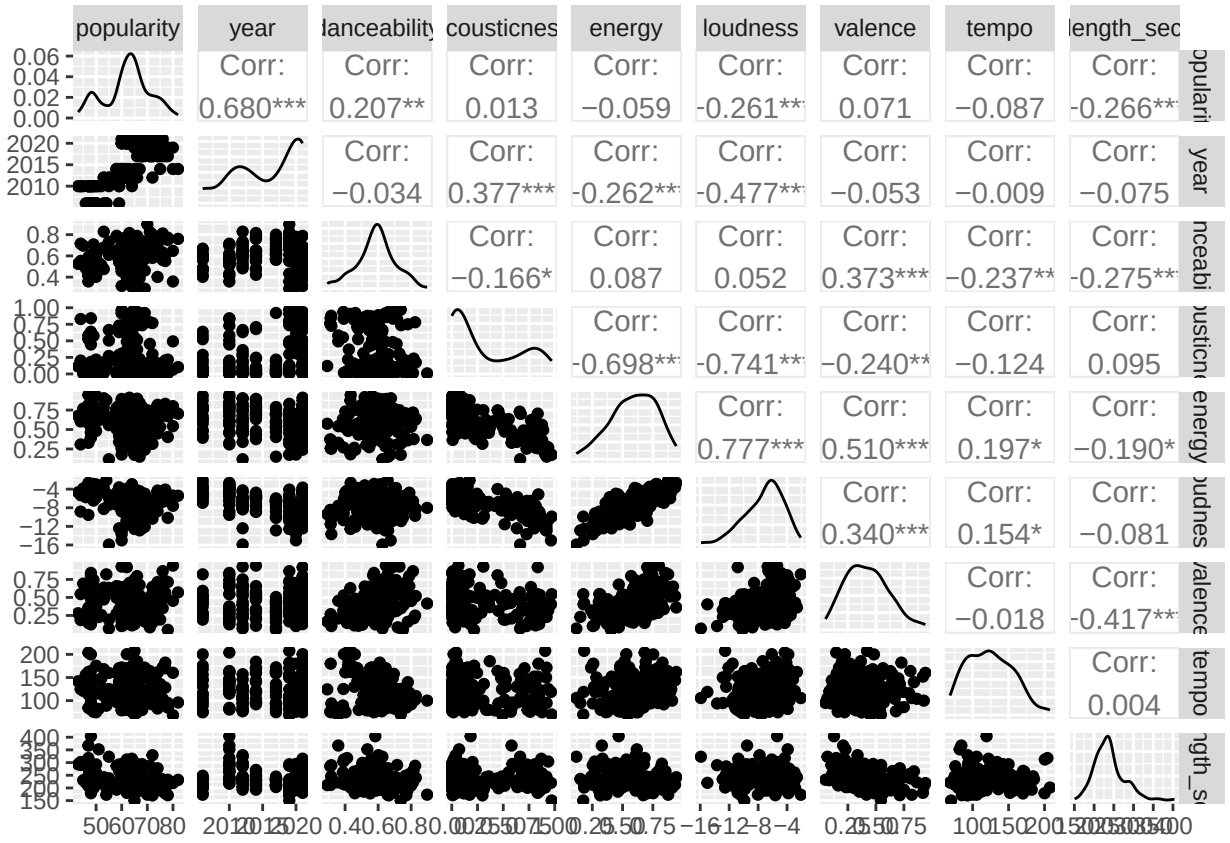
Creating the Model

Checking Distributions

Checking Normality of the Variables

```
song_chars_interest <-
  song_chars %>%
  select(-name,
         -month,
         -day) # removing variables that are not of interest


song_chars_interest %>%
  ggpairs() # using the GGally library to plot correlations and distributions
```



From the ggpairs() output, I noticed that most of the distributions are not normalized. Additionally, some of the variables namely loudness, energy, and acousticness have high correlations in magnitude potentially leading to issues with multicollinearity.

```
# using the apply function to ensure that all variables are treated as numeric
song_chars_interest <-
  as.data.frame(
  apply(
    song_chars_interest,
    2,
    function(x){
      return(as.numeric(x))
    }
  ))
```

Normalizing the variables using bestNormalize

```
set.seed(1234) # set seed to ensure the same outcome for stochastic processes
```

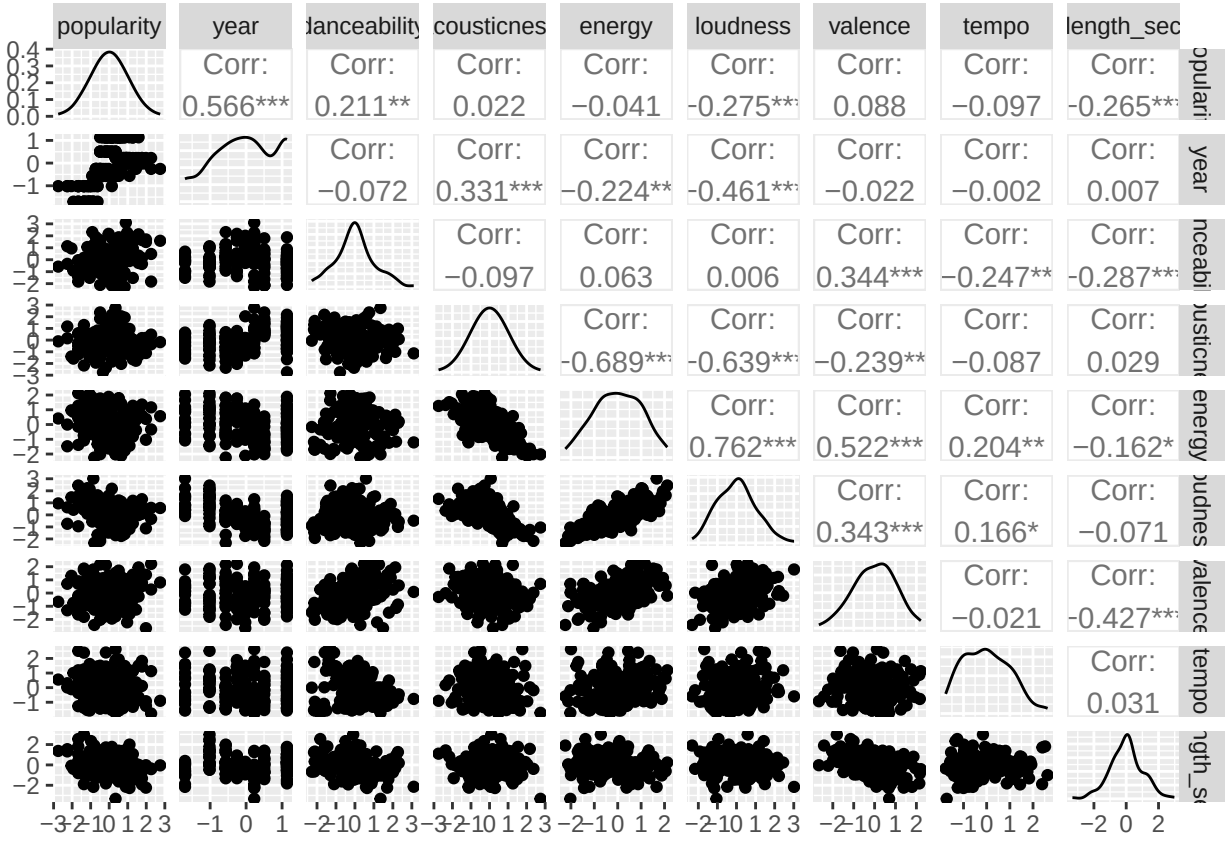
```
song_chars_normal <-
  apply(
  song_chars_interest, 2,
  function(x){
    bestNormalize(x)$x.t # use the best normalize function to normalize all vars
```

```
  }
)

song_chars_normal <- as.data.frame(song_chars_normal) # put the normalized
# values in a data frame
```

```
song_transforms <- lapply(
  1:ncol(song_chars_interest),
  function(i){
    bestNormalize(
      song_chars_interest[,i]
    )
  }
) # storing the transforms taken on the variables to enable interpretation
```

```
song_chars_normal %>%
  ggpairs() # looking at the normality and correlation of the normalized vars
```



After normalizing the variables with bestNormalize, the variable distribibutions are much more normal allowing for better analysis and understanding. year is still a bit not normal, but that is expected as it is not as continuous as the other variables. There are still potential multicollinearity issues with energy, loudness, and accousticness that need to be explored using VIF.

Setting up the Model

```
lm_popularity <-
  song_chars_interest %>%
  lm(
    formula = popularity ~ . # including all independent variables
    )
```

```
summary(lm_popularity) # extracting the coeffs and values from the regression
```

```
##
## Call:
## lm(formula = popularity ~ ., data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1062  -3.8890  -0.5559   2.8701  18.6512
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.430e+03  2.016e+02 -12.054  < 2e-16 ***
## year          1.239e+00  1.005e-01  12.322  < 2e-16 ***
## danceability  9.700e+00  4.368e+00   2.221  0.02778 *
## acousticness -1.008e+01  2.131e+00  -4.733 4.86e-06 ***
## energy        7.876e-01  4.474e+00   0.176  0.86047
## loudness     -7.104e-01  3.155e-01  -2.252  0.02572 *
## valence      -1.770e+00  3.074e+00  -0.576  0.56563
## tempo        -1.872e-02  1.445e-02  -1.296  0.19702
## length_sec   -3.846e-02  1.248e-02  -3.082  0.00242 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.563 on 159 degrees of freedom
## Multiple R-squared:  0.6109, Adjusted R-squared:  0.5913
## F-statistic: 31.21 on 8 and 159 DF,  p-value: < 2.2e-16
```

energy, valence, and tempo are not significant. The initial R^2 of the model is 0.6109 which shows that the variation in the independent variables are explaining about 61% of the variation in popularity.

Checking Multicollinearity

```
VIF(lm_popularity)
```

```
##         year danceability acousticness       energy     loudness      valence
##     1.377381     1.360054     2.686281     3.698117     3.698328     1.881446
##        tempo   length_sec
##     1.123163     1.255518
```

As expected, the VIF of acousticness, energy, and loudness are high (around 2-3), but they are not large enough to cause issues with multicollinearity.

Removing Non-Significant Predictors

```
song_chars_interest_sig <-
  song_chars_interest %>%
  select(
    popularity,
    year,
    danceability,
    acousticness,
    loudness,
    length_sec
  ) # new data frame with only significant predictors
```

Rerunning the Model with significant predictors

```
lm_popularity <-
  song_chars_interest_sig %>%
  lm(
    formula = popularity ~ . # rerunning with significant predictors
    )
```

```
summary(lm_popularity)
```

```
##
## Call:
## lm(formula = popularity ~ ., data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.3368  -4.0221  -0.2519   3.3497  18.9671
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.411e+03  1.977e+02 -12.193  < 2e-16 ***
## year          1.227e+00  9.816e-02  12.502  < 2e-16 ***
## danceability  1.021e+01  3.955e+00   2.582  0.01071 *
## acousticness -1.016e+01  1.967e+00  -5.164 7.02e-07 ***
## loudness     -7.609e-01  2.591e-01  -2.936  0.00380 **
## length_sec   -3.548e-02  1.169e-02  -3.037  0.00279 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.545 on 162 degrees of freedom
## Multiple R-squared:  0.6061, Adjusted R-squared:  0.5939
## F-statistic: 49.84 on 5 and 162 DF,  p-value: < 2.2e-16
```

Now, all of the predictors are significant, but the R^2 stayed about the same at 0.6061.

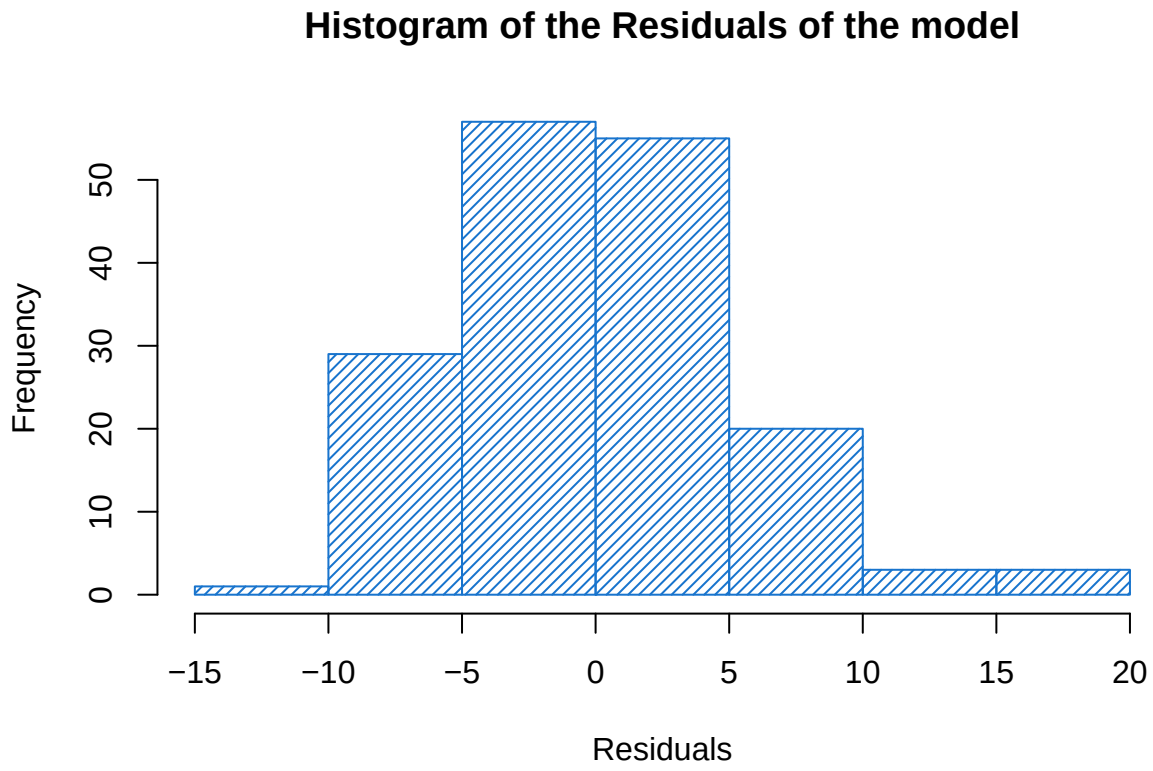
Rechecking Multicollinearity

```
VIF(lm_popularity)
```

```
##         year danceability acousticness     loudness   length_sec
##     1.321369     1.122172     2.304153     2.510469     1.108267
```

None of the variables have a VIF > 5, so there are no explicit issues with multicollinearity. # Analysis of Residuals and Outliers

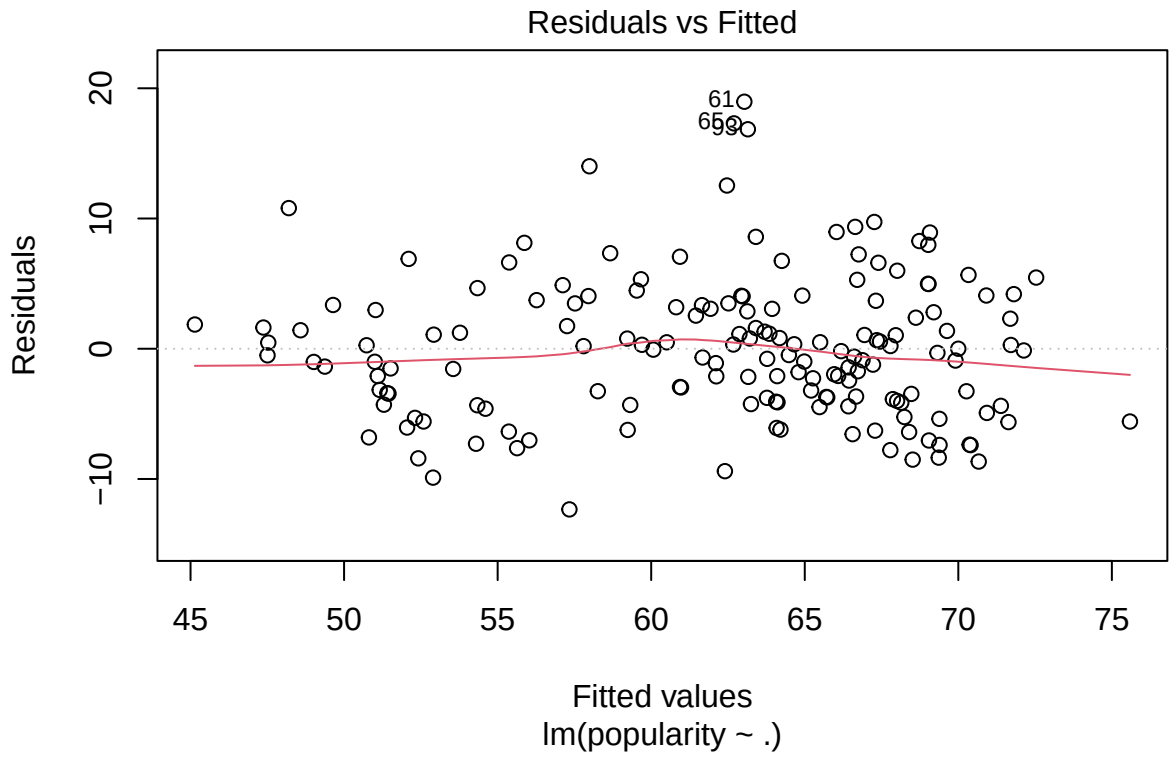
```
hist(residuals(lm_popularity), # base R histogram plot
     xlab = "Residuals",
     main = "Histogram of the Residuals of the model",
     col="dodgerblue3",
     density=25) # crossed lines in the bars
```



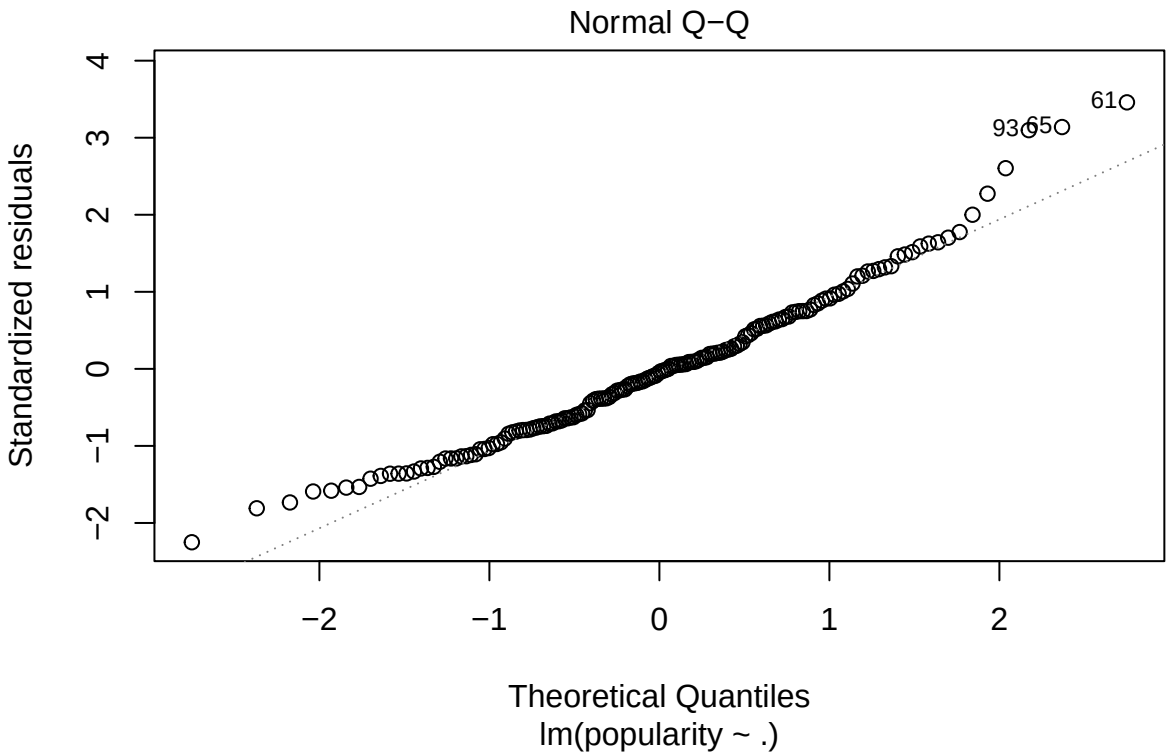

```
shapiro.test(residuals(lm_popularity))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lm_popularity)
## W = 0.97187, p-value = 0.001716
```

```
plot(lm_popularity, which = 1)
```




```
plot(lm_popularity, which = 2)
```



The histogram of the residuals is somewhat normal, but it is a bit positively skewed. This is supported by the shapiro test which concludes that the residuals are not normal since $0.0017 < 0.05$. The Residuals vs Fitted plot also supports this conclusion. While the red line closely lines up with normality, the data is clumped up and there are clear outliers. The Q-Q plot confirms this with there being three main outliers at the top of the popularity spectrum. Thus, her three most popular songs (`Blank Space`, `Shake it Off`, `Lover`) are considered outliers in the model.

Removing Outliers

```
song_chars_interest_outliers <-
  song_chars_interest[-c(61, 65, 93),] # removing the outliers identified above
```

Rerunning the model without outliers

```
lm_popularity_outliers <-
  song_chars_interest_outliers %>%
  lm(
    formula = popularity ~ . # rerunning the model on each var without outliers
    )
```

```
summary(lm_popularity_outliers)
```

```
##
## Call:
## lm(formula = popularity ~ ., data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1965  -3.6010  -0.6269   3.3123  14.0318
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.417e+03  1.820e+02 -13.283  < 2e-16 ***
## year          1.232e+00  9.077e-02  13.569  < 2e-16 ***
## danceability  1.156e+01  4.050e+00   2.854 0.004897 **
## acousticness -9.673e+00  1.923e+00  -5.030 1.33e-06 ***
## energy        1.644e+00  4.038e+00   0.407 0.684462
## loudness     -7.223e-01  2.848e-01  -2.536 0.012192 *
## valence      -4.710e+00  2.841e+00  -1.657 0.099439 .
## tempo        -1.310e-02  1.334e-02  -0.982 0.327381
## length_sec   -3.958e-02  1.129e-02  -3.506 0.000594 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.014 on 156 degrees of freedom
## Multiple R-squared:  0.6623, Adjusted R-squared:  0.645
## F-statistic: 38.24 on 8 and 156 DF,  p-value: < 2.2e-16
```

energy, valence, and tempo are still not significant predictors, but the R^2 has increased up to 0.66 by removing outliers.

Checking Multicollinearity

```
VIF(lm_popularity_outliers)
```

```
##         year danceability acousticness       energy     loudness      valence
##     1.376487     1.382981     2.673193     3.669965     3.687961     1.882918
##        tempo   length_sec
##     1.140466     1.261330
```

Similar to before, acousticness, energy, and loudness are close to having issues with multicollinearity.

Removing Non-Significant Predictors

```
song_chars_interest_outliers_sig <-
  song_chars_interest_outliers %>%
  select(
    popularity,
```

```
    year,
    danceability,
    acousticness,
    loudness,
    length_sec
  ) # removing insignificant variables
```

Rerunning Model with significant predictors

```
lm_popularity_outliers <-
  song_chars_interest_outliers_sig %>%
  lm(
    formula = popularity ~ . # rerunning the model with only sig vars
    )
```

```
summary(lm_popularity_outliers)
```

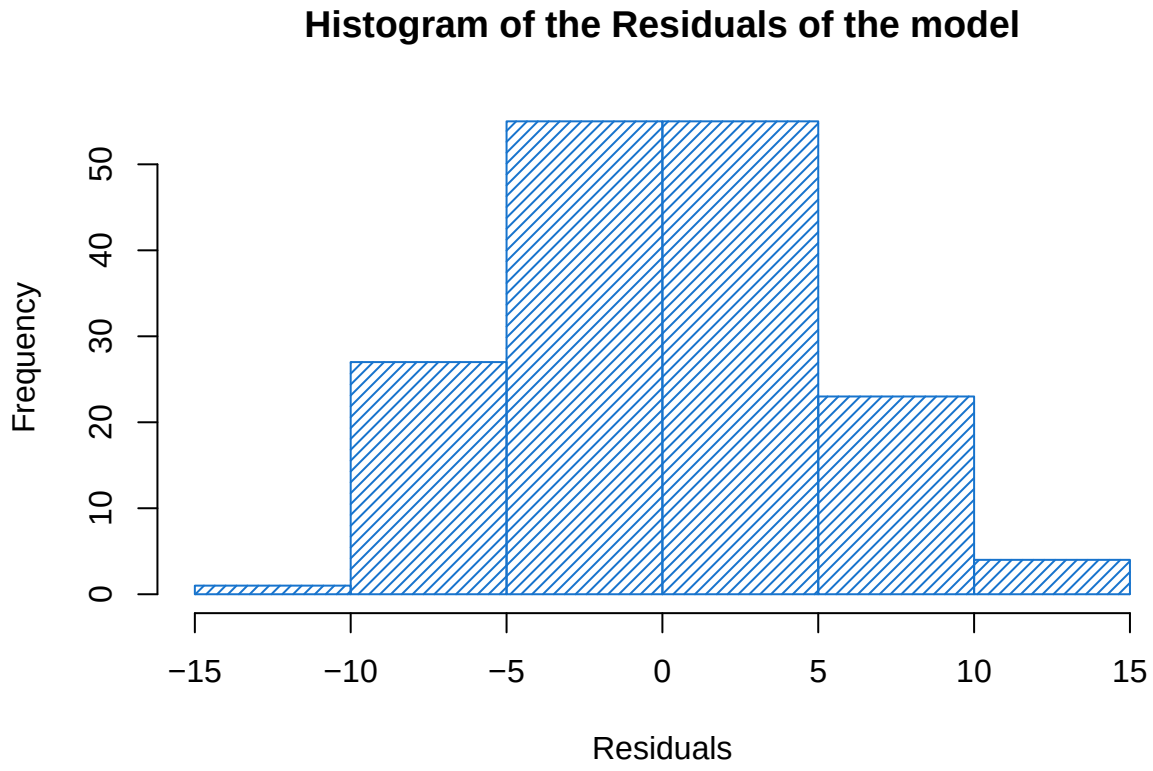
```
##
## Call:
## lm(formula = popularity ~ ., data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8746  -3.7180  -0.2226   3.5365  14.0109
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.390e+03  1.793e+02 -13.326  < 2e-16 ***
## year          1.216e+00  8.903e-02  13.658  < 2e-16 ***
## danceability  1.028e+01  3.658e+00   2.811 0.005557 **
## acousticness -1.017e+01  1.783e+00  -5.701 5.65e-08 ***
## loudness     -8.196e-01  2.351e-01  -3.487 0.000632 ***
## length_sec   -3.284e-02  1.063e-02  -3.090 0.002362 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.026 on 159 degrees of freedom
## Multiple R-squared:  0.6542, Adjusted R-squared:  0.6434
## F-statistic: 60.17 on 5 and 159 DF,  p-value: < 2.2e-16
```

Now all of the predictors are significant, and the adjusted R^2 increased by removing insignificant variables which is good. This means that the removed variables were not adding a significant predicitve value to the model.

Analysis of Residuals and Outliers

```
hist(residuals(lm_popularity_outliers), # base R histogram
     xlab = "Residuals",
```

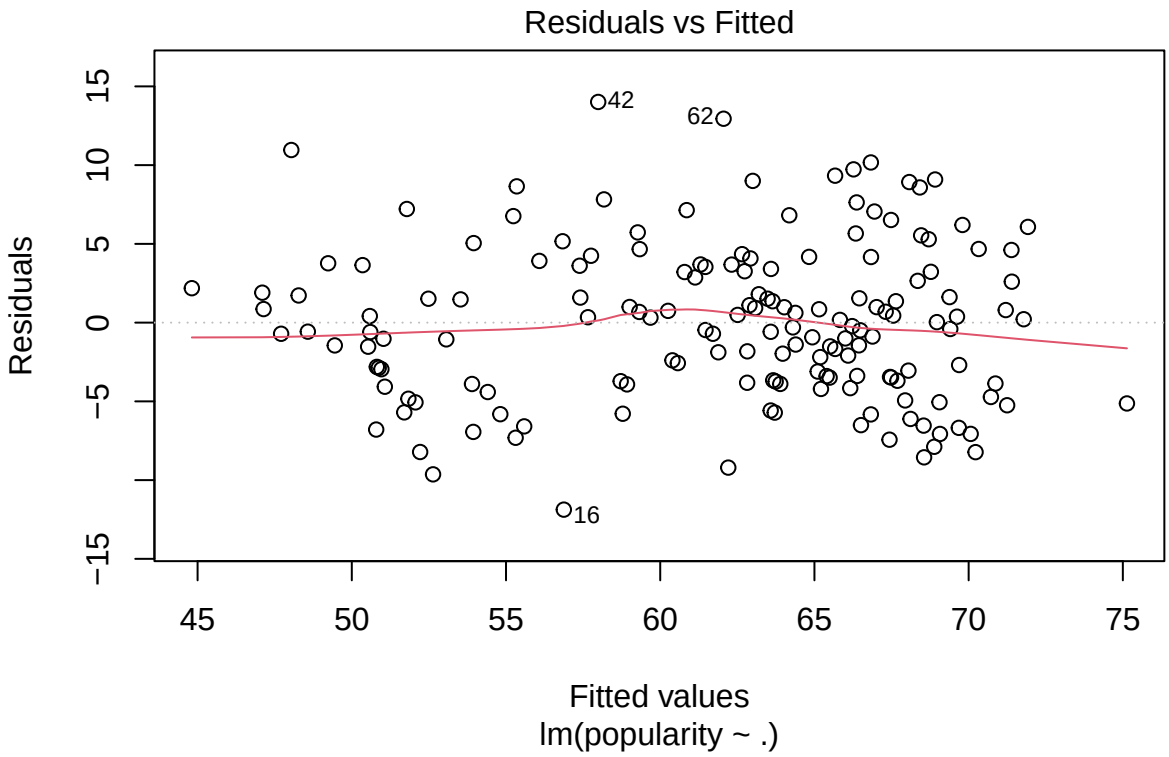
```
     main = "Histogram of the Residuals of the model",
     col="dodgerblue3",
     density=25) # crossed lines filing the bar
```



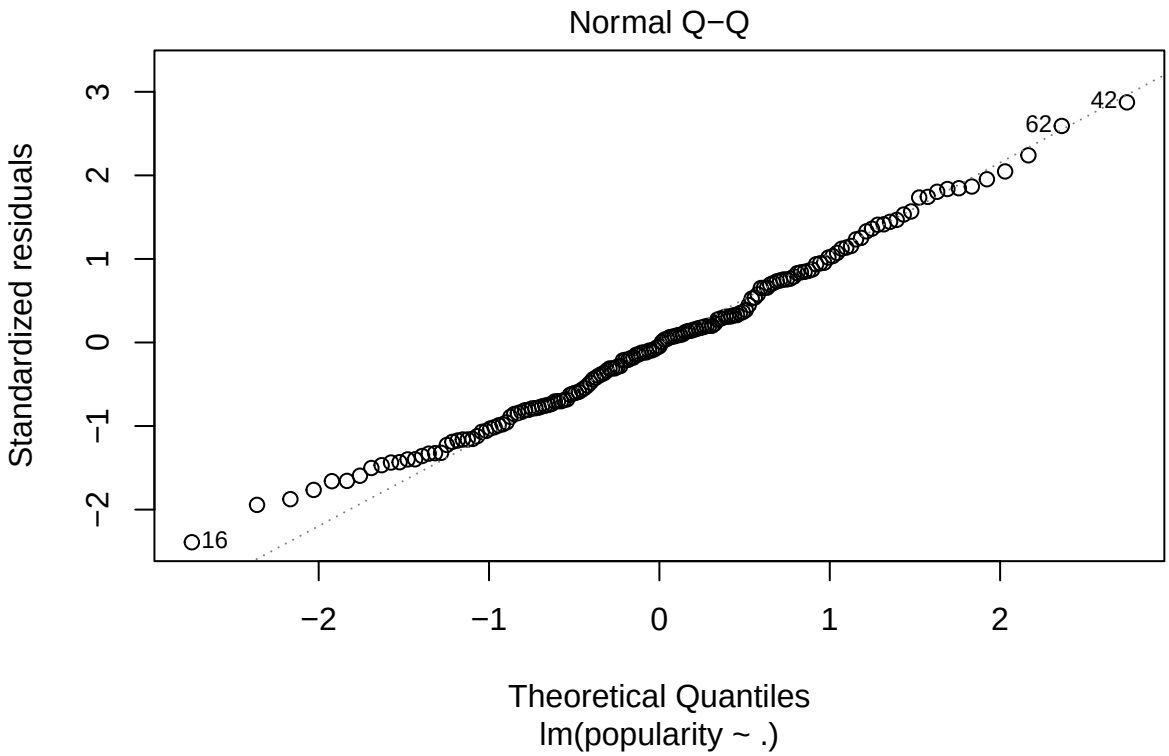

```
shapiro.test(residuals(lm_popularity_outliers))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lm_popularity_outliers)
## W = 0.99043, p-value = 0.3326
```

```
plot(lm_popularity_outliers, which = 1)
```




```
plot(lm_popularity_outliers, which = 2)
```



The residuals in the histogram are more normal and show little evidence of skew. Both the residuals vs fitted and Q-Q plot show that the model is mostly normal, and there is little evidence of impactful outliers. Finally, the shapiro test confirms that the residuals are normal as $0.33 > 0.05$, so this model satisfies the assumption of residual normality.

Final Model

There is no need to backtransform the popularity data as it was output in the same units as the original data set.

Thus, this model satisfies the assumptions of linear regression.

- The outcome variable, popularity, is continuous.
- The relationship between popularity and the song characteristics is mostly linear.
- The residuals of the model are approximately normal.
- The variance of the residuals is not correlated with the independent variables.
- There is not significant multicollinearity.
- Influential outliers were removed.

```
# Predicted Values
predicted_values <- predict(lm_popularity_outliers) # final model

# Actual values
actual_values <- song_chars_interest_outliers_sig$popularity
```

The final model was used to generate the final predictions for the popularity of Taylor Swift songs based on their characteristics. The actual popularity values were also stored for evaluation purposes.

```
final_df <- data.frame(
  actual = actual_values,
  predicted = predicted_values
) # compile the actual and predicted values into a dataframe
```

Evaluate Model

Compute RMSE and Correlation

```
sqrt(mean(residuals(lm_popularity_outliers)^2))
```

```
## [1] 4.933355
```

The RMSE is approximately 4.93 which means that the model incorrectly predicts song popularity by about 5 points on average. This shows that the model is not incredibly accurate as the range of popularities for Swift's songs go from 43 - 82.

Computing Correlation

```
cor(predicted_values, actual_values)^2
```

```
## [1] 0.6542296
```

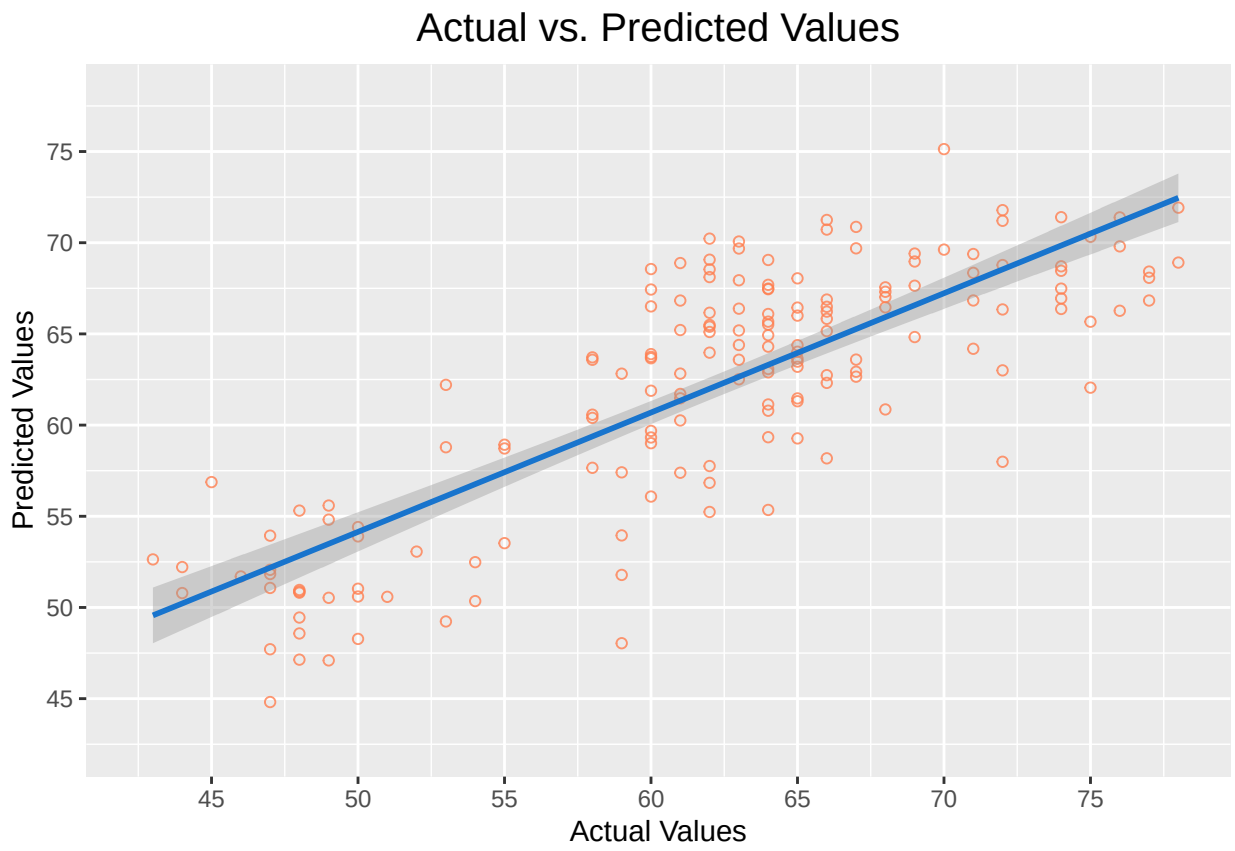
The final correlation coefficient or R^2 is approximately 0.654 which is slightly above average. This aligns with the RMSE values as the model can moderately accurately predict song popularity.

Plot Fitted vs. Actual Data

```
final_df %>%
ggplot(
  aes(
    x = actual,
    y = predicted
    )
  ) +
  geom_point(
    color = "coral",
    alpha = 0.8,
    shape = 1 # empty hole points
  ) + # scatterplot
  geom_smooth(method = "lm",
              color = "dodgerblue3") +
  labs(
```

```
    x = "Actual Values",
    y = "Predicted Values",
    title = "Actual vs. Predicted Values"
  ) + # better lables
  scale_x_continuous(
    limits = c(42.5, 78),
    breaks = seq(40, 80, 5)
  ) + # better x scale
  scale_y_continuous(
    limits = c(42.5, 78),
    breaks = seq(40, 75, 5)
  ) + # better y scale
  theme(
    plot.title = element_text(size = 15, hjust = 0.5)
  ) # center and increase title size
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



This plot reflects the moderate accuracy of the linear regression model. The blue linear regression line shows the overall trend of the song popularities, but it does not do an incredible job of having extreme accuracy.

Discussion

Using Spotify's data on Taylor Swift songs, I was able to create a moderately accurate model using song characteristics to predict the popularity of her songs. The model had a final RMSE of 4.93 and an R^2 of

0.654. This means that on avearge the model was off by about 4.93 points of popularity (out of 100), but the variation in the independent variables explained about 65% of the variation in popularity. Since the coefficients of the variables are significant, I can reject the null hypothesis and concluded that at least one of the release year, length (in seconds), danceability, acousticness, energy, loudness, valence, or tempo of the song are related to popularity. While this model did satisfy all of the assumptions of linear regression, it is important to understand the the popularity of songs is very complicated and needs to be analyzed by more predictors and data points. Two songs can have the exact same characteristics but have very different popularity results. If songs could be designed to be as popular as possible, major labels and artists would patent and mass produce mega-popular songs to be consumed by the masses. However, opinion is subjective, and I appreciate that songs of any genre or background can be popular and enjoyable to listen to. With that being said, the model predicted the following result for each characteristic of Taylor Swift song between 2006 and 2021:

- year: A one standard deviation increase in year corresponds with a 1.22 point increase in popularity.
- danceability: A one standard deviation increase in danceability corresponds with a 1.03 point increase in popularity.
- acousticness: A one standard deviation increase in acousticness corresponds with a 1.02 point decrease in popularity.
- loudness: A one standard deviation increase in loudness corresponds with an 8.2 point decrease in popularity.
- length_sec: A one standard deviation increase in the length of the song in seconds corresponds with a 3.28 point decrease in popularity.

Thus, according to my model, the most popular Taylor Swift song would be one with high levels of danceability, limited acoustic instruments, low loudness levels, a short song, and one released as close to present day as possible. Many of these things contradict each other or with Swift's main identity. Swift is most popularly known for using an acoustic guitar, having long songs, and she has released songs over multiple decades. While this analysis can show what has worked in the past, if I were Swift's agent or producer, I would focus on her releasing songs she enjoys writing and performing and try to keep genuine to her core audience. It has brought her much success so far, and I do not see why it would not continue.