Final Project Predicting Song Popularity

Syracuse University - IST 707 - Spring 2021

https://www.kaggle.com/edalrami/19000-spotify-songs/discussion/73524

Project Introduction

The music industry has evolved a lot over the last several decades and will continue to evolve. This is in part due to advancements in technology. It is hard to believe that not too long ago CD players were still commonplace for listening to music, while nowadays, companies such as Spotify provide on demand streaming of music directly to any device. Not only has the method of listening to music changed due to technology, but the sound of the music itself has changed as well. For example, a lot of recordings are now digitally enhanced and new genres have emerged based on pure digital production such as electronic dance music and dubstep.

More recent developments in technology allow for attributes about songs to be extracted from an audio file. For example, the Organize Your Music tool, is an online tool that takes a song or a selection of songs and gathers information on them. It gathers attributes such as the song tempo, energy, and danceability, among others. This enables a new perspective of music to be explored and analyzed, which leads into the purpose of this project.

The following project will explore a Kaggle dataset consisting of thousands of songs with these attributes. Here are some examples of questions that will attempt to be answered. What makes a song popular? Are there certain attributes that are correlated with popular songs? How have popular songs changed overtime? Can the popularity of a song be predicted based on its attributes? The primary objective will be predicting the popularity of a song.

Song popularity is important because having a popular song can mean many different things to an artist. One benefit it has is the influence and trajectory it can take an artists' career. One song can make or break a new artist. Many artists strive to go 'viral' with a song that can bring the spotlight that they are looking for. Producers of music and labels can also benefit from understanding what makes a song popular. Having that knowledge can help them create a hit song, and thus, improve their profitability and chances of breeding successful artists.

In the era of the internet and social media any song could pop off at any time if it has the right sound. As the music industry and consumer listening preferences continue to change, it is very important for the creators of music to stay on top of these trends and prepare for them.

Data Preparation

Two original datasets from Kaggle, song_data and song_info, are used along with two additional datasets, master_song_T and master_artist_T, which were collected via web scraping wikepedia.

Web scraping was conducted to collect additional data about the songs and artists which is not included in the original datasets from Kaggle. For example, when a song was released is a crucial piece of information that was missing.

The code for the web scraping is not provided in the output of this report, however it is available in the source code. As mentioned before, the data from the web scraping has been packaged into two csv files for convenience.

The goal of this section is to provide the reader with the necessary background of the data. In the following section, each of the datasets will be described in more detail along with any data preprocessing that took place.

```
#------------------------------------------------------------------------#

# load the data
# make sure the files are saved in current working directory

# the first two data sets can be downloaded from kaggle
# https://www.kaggle.com/edalrami/19000-spotify-songs/discussion/73524

song_data         <- read.csv("song_data.csv")
song_info         <- read.csv("song_info.csv")

# the second two datasets are provided seperately
# these two datasets come from web scraping wikipedia

master_artist_T   <- read.csv("master_artist_T.csv")
master_song_T     <- read.csv("master_song_T.csv")

#------------------------------------------------------------------------#
```

table: song_data

The song_data table is an original dataset from Kaggle, which consists of a collection of songs that were parsed via the Organize Your Music tool.

Note: there were many duplicate song names, without a way of uniquely identifying them. For this reason, duplicate song names were removed.

```
#------------------------------------------------------------------------#

## Original Table Dimensions
##  18835  rows by  15  cols
```

```
## Cleaned Table Dimensions
##  10174  rows by  15  cols
##
##  Removed 8661 Duplicate song_name

## [1] "~~~~~~~~~~~~~~~~~~~~~~"
## [1] "song_name: character"
## [1] "song_popularity: numeric"
## [1] "song_duration_ms: numeric"
## [1] "acousticness: numeric"
## [1] "danceability: numeric"
## [1] "energy: numeric"
## [1] "instrumentalness: numeric"
## [1] "key: numeric"
## [1] "liveness: numeric"
## [1] "loudness: numeric"
## [1] "audio_mode: numeric"
## [1] "speechiness: numeric"
## [1] "tempo: numeric"
## [1] "time_signature: numeric"
## [1] "audio_valence: numeric"
## [1] "~~~~~~~~~~~~~~~~~~~~~~"

#-------------------------------------------------------------------------#

# meta data from http://organizeyourmusic.playlistmachinery.com/

# song_name         - the name of the song
# song_popularity   - the higher the value the more popular the song is
# song_duration_ms  - the length of the song measured in milliseconds
# acousticness      - the higher the value the more acoustic the song is
# danceability      - the higher the value, the easier it is to dance to
# energy            - the higher the value, the more energtic the song is
# instrumentalness  - the higher the value, the more instrumental the song is
# key               - description not provided
# liveness          - the higher the value, more likely a live recording
# loudness          - the higher the value, the louder, measured in dB
# audio_mode        - description not provided
# speechiness       - the higher the value the more spoken word in the song
# tempo             - the tempo of the song measured in beats per minute
# time_signature    - description not provided
# audio_valence     - the higher the value, the more positive mood

#-------------------------------------------------------------------------#
```

table: song_info

The song_info table is an original dataset from Kaggle which contains the corresponding artist, album, and playlist for each song in the song_data table.

Note: as in the song_data table, there were duplication errors in this table as well. This is resolved in the same manner by removing duplicate song names.

```
#-------------------------------------------------------------------------#

##
##  Original Table Dimensions
##  18835  rows by  4  cols

## Cleaned Table Dimensions
##  10174  rows by  4  cols
##
##  Removed 8661 Duplicate Song Names

## [1] "~~~~~~~~~~~~~~~~~~~~~~~"
## [1] "song_name: character"
## [1] "artist_name: character"
## [1] "album_names: character"
## [1] "playlist: character"
## [1] "~~~~~~~~~~~~~~~~~~~~~~~"

#-------------------------------------------------------------------------#

# song_name         - the name of the song
# artist_name       - the name of the corresponding artist(s)
# album_names       - the name of the corresponding album(s)
# playlist          - the name of the corresponding playlist(s)

#-------------------------------------------------------------------------#
```

table: master_song_T

The master_song_T table was collected by web scraping wikipedia pages for additional information about the songs.

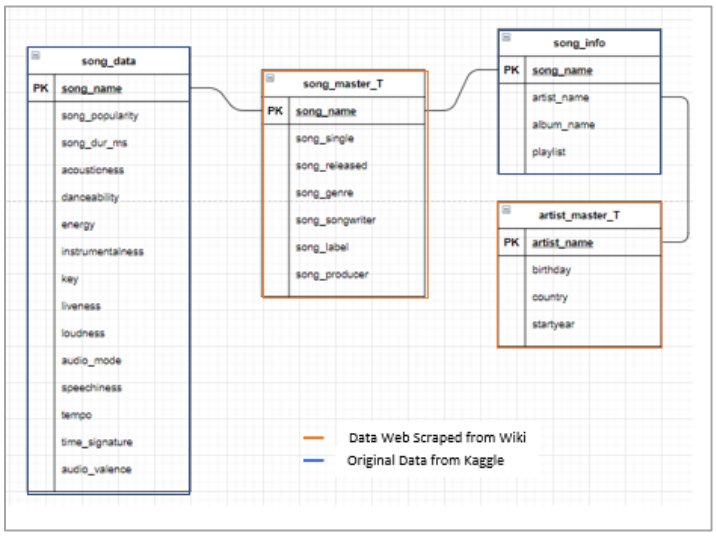
```
#-------------------------------------------------------------------------#

## 10171  rows by  7  cols

## [1] "~~~~~~~~~~~~~~~~~~~~~~~"
## [1] "song_name: character"
## [1] "song_single: numeric"
## [1] "song_released: numeric"
## [1] "song_genre: character"
## [1] "song_label: character"
## [1] "song_songwriter: character"
## [1] "song_producer: character"
## [1] "~~~~~~~~~~~~~~~~~~~~~~~"

#-------------------------------------------------------------------------#
```

```
#-------------------------------------------------------------------------#

# song_name          - the name of the song
# song_single        - binary whether the song is a single or not
# song_released      - the year that the song was released in
# song_genre         - the corresponding genre(s) for the song
# song_label         - the corresponding label(s) for the song
# song_songwriter    - the corresponding songwriter(s) for the song
# song_producer      - the corresponding producer(s) for the song

#-------------------------------------------------------------------------#
```

table: master_artist_T

The master_artist_T table was constructed by web scraping wikipedia pages for additional artist information.

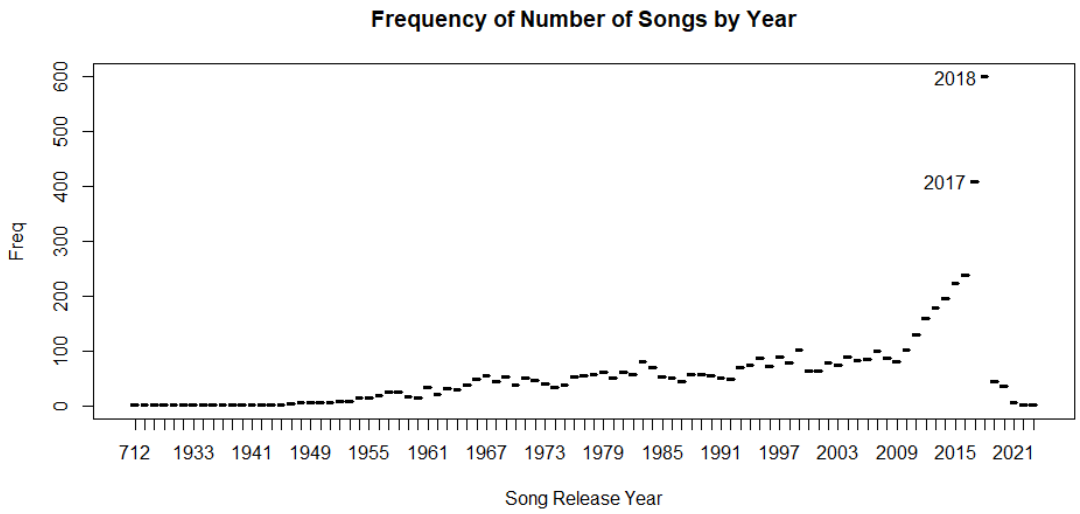
```
#-------------------------------------------------------------------------#

## 7564  rows by  4  cols

## [1] "~~~~~~~~~~~~~~~~~~~~~~~~"
## [1] "artist_name: character"
## [1] "birthday: character"
## [1] "country: character"
## [1] "startyear: numeric"
## [1] "~~~~~~~~~~~~~~~~~~~~~~~~"

#-------------------------------------------------------------------------#

# artist_name        - the name of the artist
# birthday           - the date that the artist was born
# country            - 2 values - either USA or foreign
# startyear          - the year that the artist started making music

#-------------------------------------------------------------------------#
```



Song Release Year

The song release year was one of the attributes that was retrieved from web scraping Wikipedia. Here is how the distribution of the song release years looks.

Note: there are a large number of missing song years due to the data not being available on Wikipedia which are not displayed on the graph below.



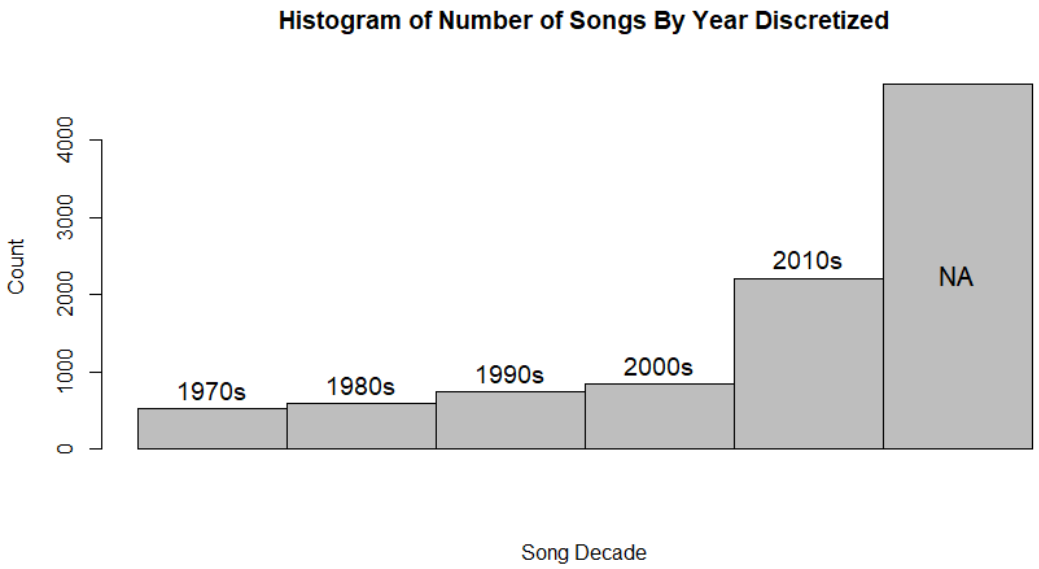
Observations
- there are more newer generation songs than older generation songs
- there is a particularly larger number of songs between 2010 and 2020
- there is a large spike in the number of songs released in 2017-2018
- the bias is probably due to the preferences of the creator of the data
- there are some outliers, such as year 712, which is not a valid year

```
#-------------------------------------------------------------------------#

# years prior to 1970 and later than 2020 thrown out due to lack of data
# this will also take care of any outliers outside of the date range
# NA's will be kept because although the year is not known there a lot of dat
a

cat('Number of Songs Removed:',
    nrow(master_song_T) - nrow(master_song_T[which(
                            (master_song_T$song_released >= 1970  &
                            master_song_T$song_released <= 2020) |
                            is.na(master_song_T$song_released)), ]))

## Number of Songs Removed: 554

#-------------------------------------------------------------------------#
```

Discretize Song Release Year into Song Decade

The song years will be discretized into bins containing one decade per bin. This is based on (a) the assumption that decades of music tend to have similar sounds and (b) inferring song decade rather than inferring song year will be more accurate.

```
#-------------------------------------------------------------------------#

# look at the distribution of the new song years after discretizing

table(new_song_T$song_released)

##
## 1970s 1980s 1990s 2000s 2010s    NA
##   474   585   731   801  2280  5114

#-------------------------------------------------------------------------#
```

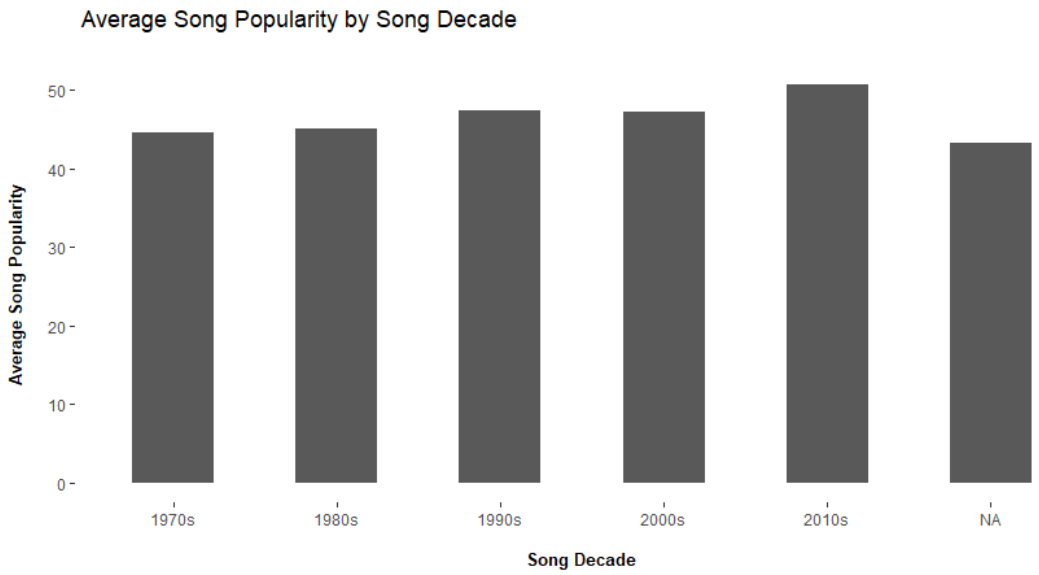


Observations

- as mentioned before, there are a lot of unknown song years, which are shown here
- the NA values are due to the information not being available on Wikipedia
- it may be possible to infer the missing song decade - having the genre would be helpful for this, so the genre will be explored next

Song Decade and Song Popularity

Now that the song release years have been discretized into song decades, the following chart compares the average song popularity for each song decade.



Observations
- there is a slight positive correlation between song decade and song popularity
- the most popular songs are in the 2010s, the least popular songs are in the NA
- this makes sense because songs that do not have information on Wikipedia are probably less popular

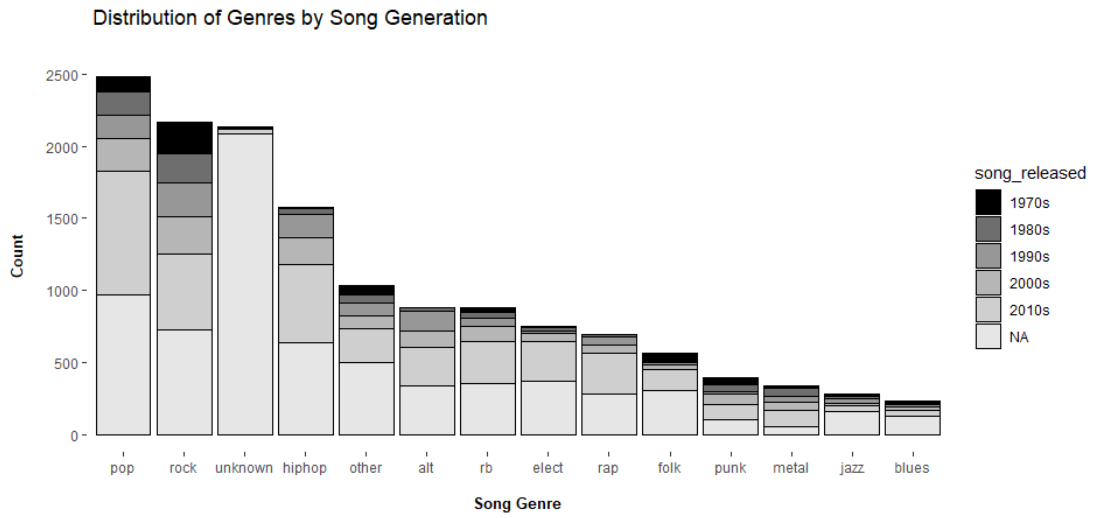
Song Genres

The song genre was one of the attributes that was retrieved from web scraping Wikipedia. Note: there can be multiple genres for one song.

There was a wide range of genres that came out of Wikipedia, but these were aggregated into higher level parent genres.

```
# here is the list of all of the parent genres
# note that hip and hop will get put together into hiphop

parent_genre_list <- c('alt', 'rock', 'metal', 'punk', 'pop', 'hip', 'hop', '
r&b', 'rap', 'jazz', 'blues', 'folk', 'country', 'elect', 'other')
```

Below chart shows the distribution of the song genres and how many of them there are for each song release year.



Below table shows the percentage total of the song genres for each song year, note that the percentages will not add up to 100 due to being multiple genres for some songs.

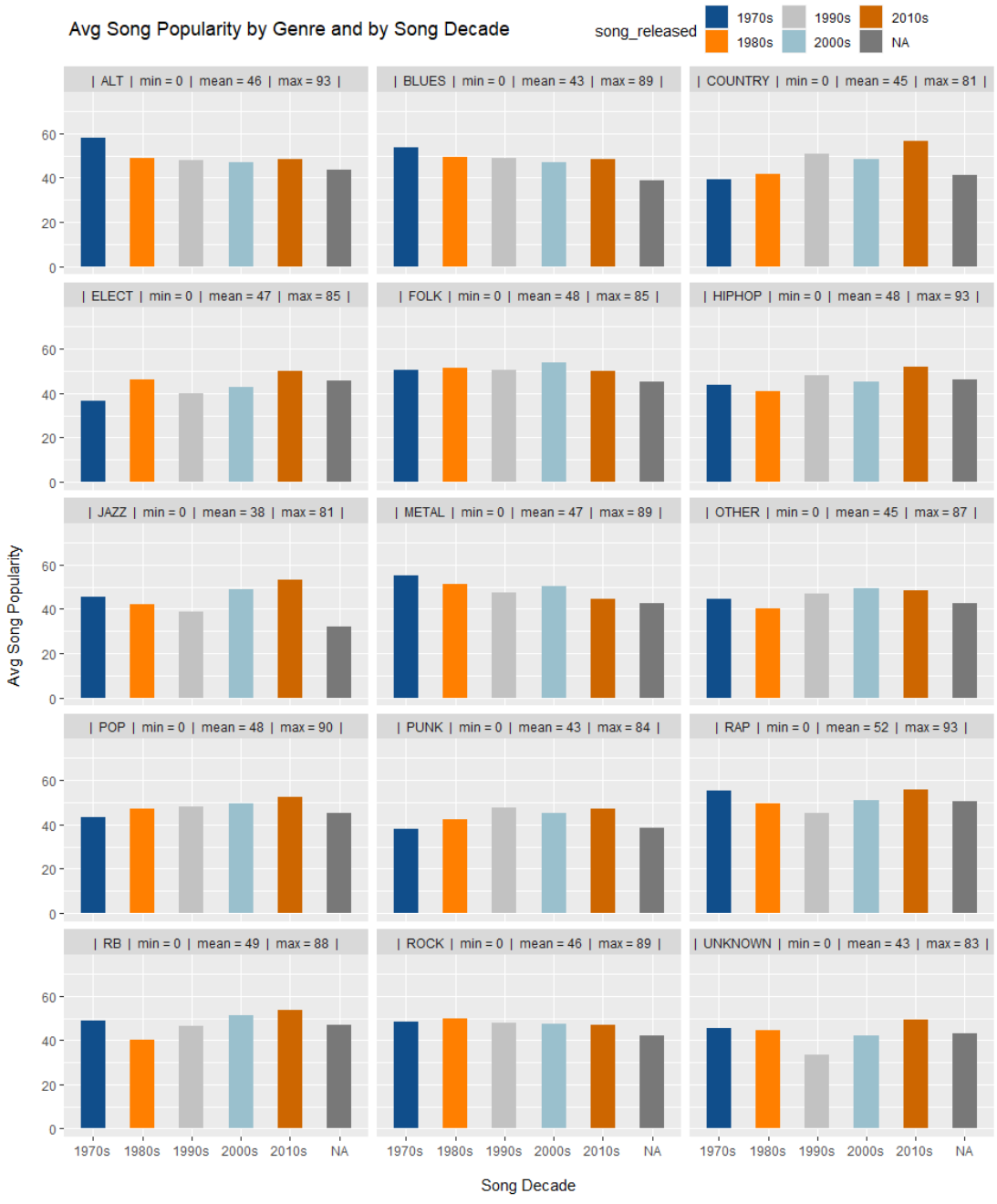
	1970s	1980s	1990s	2000s	2010s	NA
alt	1%	4%	18%	15%	12%	7%
rock	46%	35%	32%	32%	23%	14%
metal	3%	10%	5%	8%	5%	1%
punk	10%	9%	2%	9%	5%	2%
pop	21%	28%	21%	29%	37%	19%
hiphop	1%	8%	22%	23%	24%	12%
rb	8%	7%	8%	12%	13%	7%
rap	0%	2%	8%	8%	12%	5%
jazz	5%	2%	4%	3%	2%	3%
blues	4%	2%	2%	3%	2%	3%
folk	10%	2%	3%	4%	6%	6%
elect	1%	3%	3%	7%	12%	7%
other	12%	10%	12%	11%	10%	10%
unknown	1%	1%	1%	1%	1%	41%

Observations

- there are some noticeable correlations with genre and generation
- for example not many hiphop or electro songs in earlier generations
- there are more folk and rock songs in earlier gens
- pop, rock, hiphop are some of the most frequent genres as would be expected because these genres are mostly newer generation

Song Genre and Song Popularity

The following series of charts displays the average song popularity for each genre broken down by song decade.



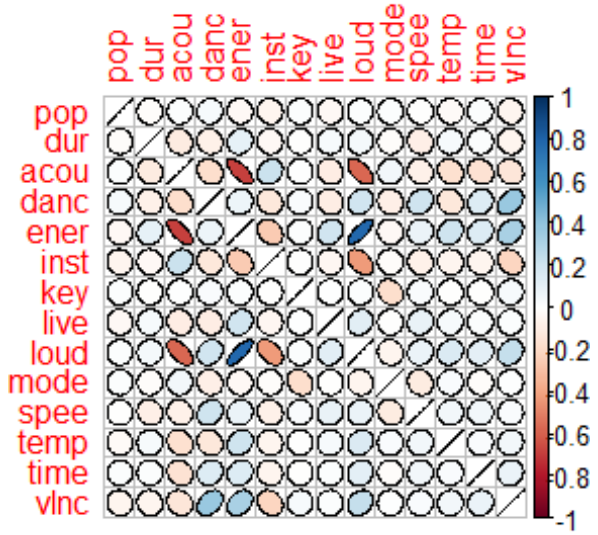
Observations

- rap is most popular genre (52 avg) and jazz is the least popular (38 avg)
- some genres have decreased in popularity over time, such as alt and metal
- some genres have increased in popularity over time, such as country and pop

Correlation of Song Attributes

The song attributes were included as part of the original Kaggle datasets. These will be explored next.

```
#-------------------------------------------------------------------------#

# transform the song attributes with min max
temp_song_T_transformed <- min_max_transform(temp_song_T, seq(19, 32, 1))

# truncate the column names
colnames(temp_song_T_transformed) <- substr(colnames(temp_song_T_transformed)
, 1, 4)

# fix some of the column names manually
colnames(temp_song_T_transformed) [c(19, 20, 28, 32)] <- c('pop', 'dur', 'mod
e', 'vlnc')

# compute the correlation matrix
temp_cor <- round(cor(temp_song_T_transformed[, 19:32], temp_song_T_transform
ed[, 19:32]), 2)

# plot the correlation matrix
corrplot(temp_cor,
         method = 'ellipse',
         outline = TRUE)
```



Observations
- there are not any very strong correlations with song popularity
- song acousticness has a strong negative correlation with song energy
- song energy, loudness, and audio valence have a strong positive correlation

```
#-------------------------------------------------------------------------#
```

Song Attribute Importance

Using the boruta package, the importance of each song attribute for predicting song popularity will be tested.

```
## Boruta performed 14 iterations in 56.33384 secs.
##  9 attributes confirmed important: acou, danc, dur, ener, inst and 4
## more;
##  2 attributes confirmed unimportant: key, mode;
##  2 tentative attributes left: live, time;

# remove the variables that are not significant
temp_song_T_transformed <- temp_song_T_transformed[, which(
  ! colnames(temp_song_T_transformed) %in% c('key', 'live', 'mode', 'time'))]
```

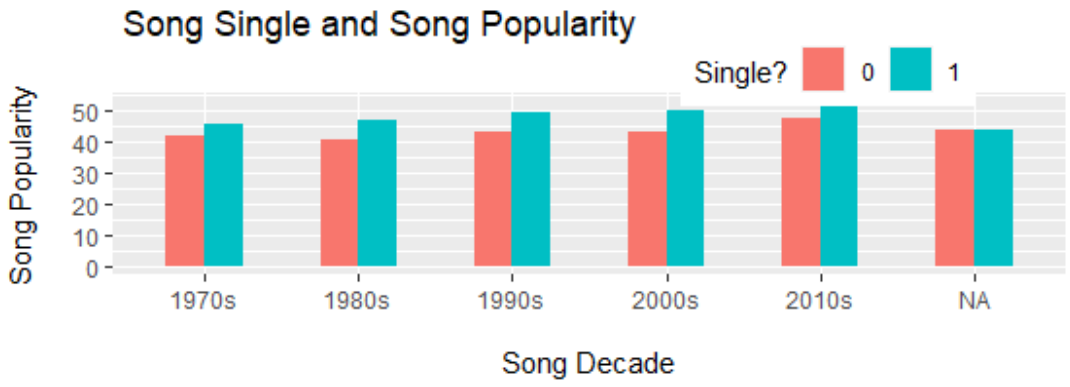
Observations
- key and audio mode are not significant for predicting song popularity
- liveness and time signature are tentative for predicting song popularity
- based on the results, all four of these columns will be dropped

```
#-------------------------------------------------------------------------#
```

Exploration of Song Attributes

Each remaining song attribute will be explored in more detail along with any data preprocessing that takes place.

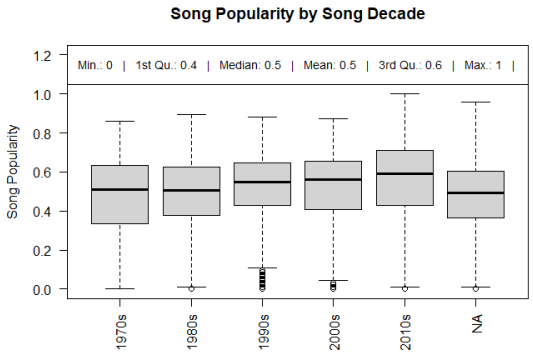
```
# what effect does a song being a single have on song popularity?
```

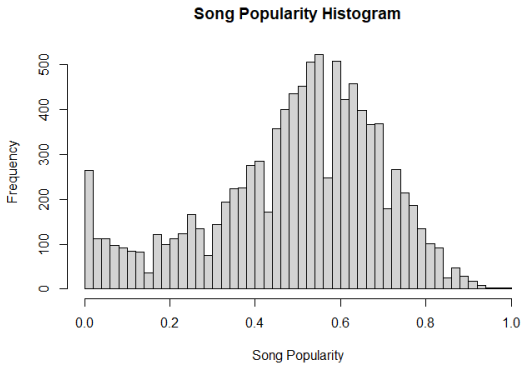



```
#-------------------------------------------------------------------------#
```

Observations
- there is a positive correlation between song single and song popularity
- this is not surprising because singles are usually more popular songs
- interestingly, there is little to no difference seen in the NA group

```
#-------------------------------------------------------------------------#
```





Observations
- the distribution of song popularity is normal but slightly left skewed
- this means that there are more less popular songs than there are popular songs
- the majority of songs fall within a song popularity of 0.4 to 0.6
- there is an unusual number of very unpopular songs in the 1990s and 2000s

```
#-------------------------------------------------------------------------#

# a sample of some of the outliers from 1990s

kable(head(x.Outliers[, c(1, 30, 31, 2, 33, 34)], 5))
```

song	artist_name	album_names	song.1	song_single	song_released
I'll Be There For You/You're All I Need To Get By	Method Man	More Music From 8 Mile	hip,hop	1	1995
All Apologies	SinÃ©ad O'Connor	Universal Mother	alt,rock	1	1993
Hey Leonardo (She Likes Me For Me)	Blessid Union Of Souls	Walking Off The Buzz	alt,rock,pop	1	1999
Into Your Arms	The Lemonheads	Come On Feel The Lemonheads	pop	1	1993

New Age Girl	Deadeye Dick	Sk8terboy Rock!, Vol. 1	alt,rock	1	1994

```
#-------------------------------------------------------------------------#

# a sample of some of the outliers from 2000s

kable(head(x.Outliers[, c(1, 30, 31, 2, 33, 34)], 5))
```

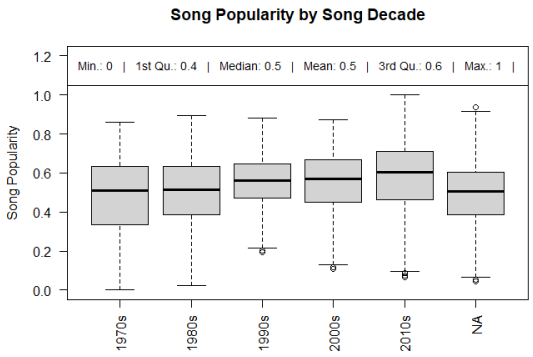
song	artist_name	album_names	song.1	song_single	song_released
Long Way Home	Tom Waits	Orphans	rock	0	2006
For A Dancer	Jackson Browne	The Very Best Of Jackson Browne (US & International Release)	rock	0	2004
Heaven Can Wait	We The Kings	Smile Kid	rock,punk,pop	0	2009
I Am The Message	Fightstar	One Day Son, This Will All Be Yours	alt,rock,metal	1	2008
Teddy Picker	Arctic Monkeys	Teddy Picker	punk	1	2007

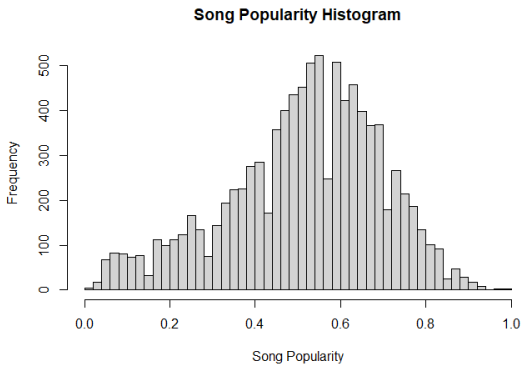
```
#-------------------------------------------------------------------------#
```

Observations

- some of these songs should not have such low song popularity
- for example, “I’ll Be There For You/You’re All I Need To Get By” was a single by Method Man and sold over 800,000 copies.
- these songs will be removed due to the potential adverse effect that they could have on the prediction models later on.

```
#-------------------------------------------------------------------------#

## Number of Songs Removed: 456
```

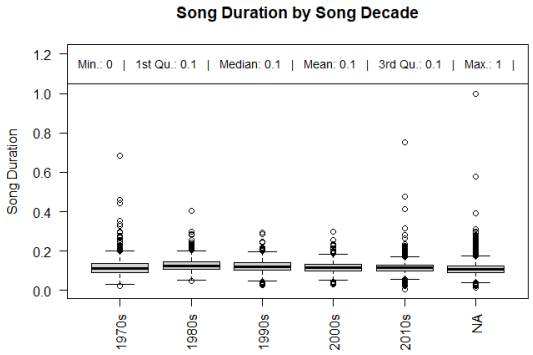


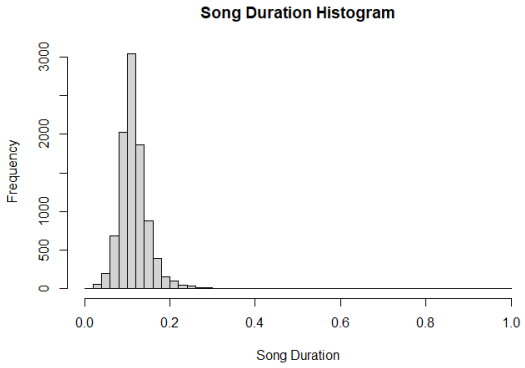


Observations

- the distribution of song popularity after removing outliers is more normal
- there are still some strange pockets of low popularity but will leave for now

```
#-------------------------------------------------------------------------------#
```






```
# a sample of some of the shortest songs
kable(
  head(
    sqldf('
      select
        song_name,
        (sum(song_duration_ms) / 1000) as song_duration_seconds
      from
        temp_song_T
      group by
        song_name
      order by
        song_duration_ms asc'
    )
  )
)
```

song_name	song_duration_seconds

The Avengers	26
Shirley Chisholm - 1972	35
Twins	50
Tykky Interludium	50
Taiko Drumming	50

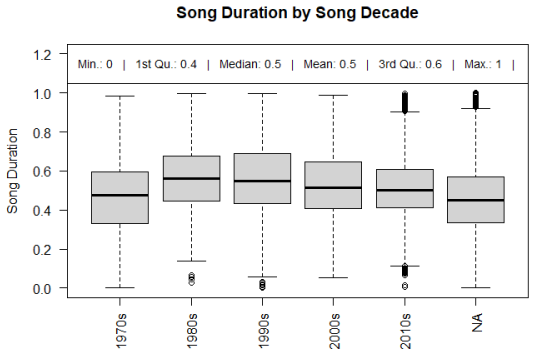
```
# a sample of some of the longest songs
kable(
  head(
    sqldf('
      select
        song_name,
        (sum(song_duration_ms) / 1000) as song_duration_seconds
      from
        temp_song_T
      group by
        song_name
      order by
        song_duration_ms desc'
    )
  )
)
```

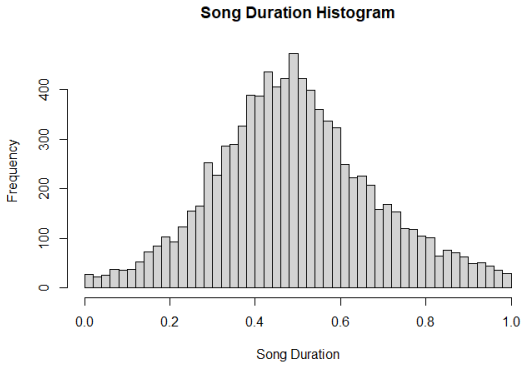
song_name	song_duration_seconds
Army Arrangement	1799
Play	1355
2112: Overture / The Temples Of Syrinx / Discovery / Presentation / Oracle / Soliloquy / Grand Finale - Medley	1233
I Have a Dream - The Complete Speech of Martin Luther King Jr.	1047
Autobahn - 3-D	866
Do You Feel Like We Do	836

```
#--------------------------------------------------------------------------#
```

Observations
- there is little difference in avg song duration between decades
- there are some outlier songs in terms of their song duration
- some are either intros, outros, transitions, or not real songs
- or they could just be really short or really long songs
- these songs can be removed since the main focus is hit songs

```
#--------------------------------------------------------------------------#

## Number of Songs Removed: 896
```

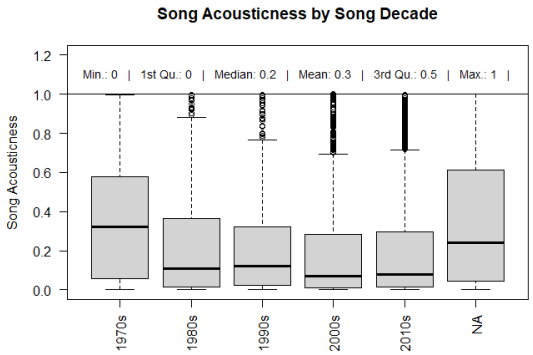


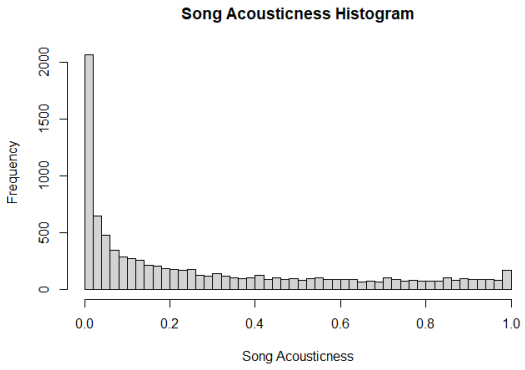


Observations

- the distribution of song duration after removing outliers is normal
- the new range of song duration is between about 1.5 min to 5.5 min
- the majority of songs fall tend to be around 3 to 4 minutes

```
#-------------------------------------------------------------------------#
```





Observations

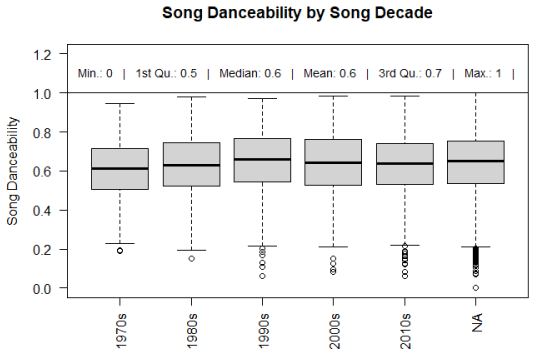
- the distribution of song acousticness is skewed
- this could adversely affect song popularity prediction later on

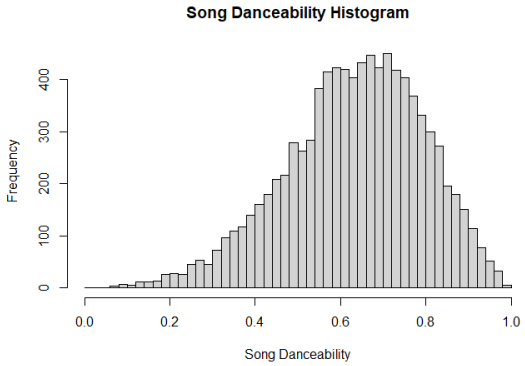
```
#-------------------------------------------------------------------------#
```

```
##    Var1 Freq
## 1     0 2949
## 2   0.1 1527
## 3   0.2  958
## 4   0.3  628
## 5   0.4  532
## 6   0.5  469
## 7   0.6  465
## 8   0.7  414
## 9   0.8  397
## 10  0.9  461
## 11    1  289
```

Observations
- the song acousticness does not seem to directly affect song popularity
- to simplify this attribute, a song can have 3 levels of acousticness
- no acousticness (0), some acousticness (0.5), acousticness (1.0)

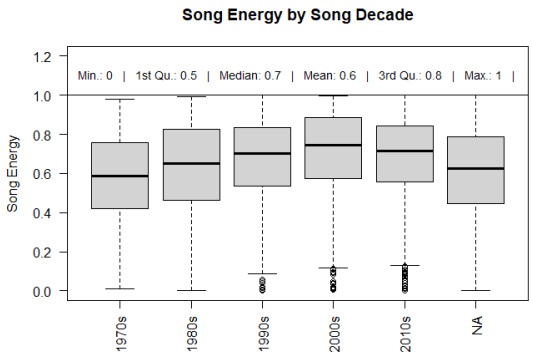
```
# the frequency distribution after transformation
table(temp_song_T_transformed$acou)

##
##    0  0.5    1
## 5434 2508 1147

#-------------------------------------------------------------------------#
```

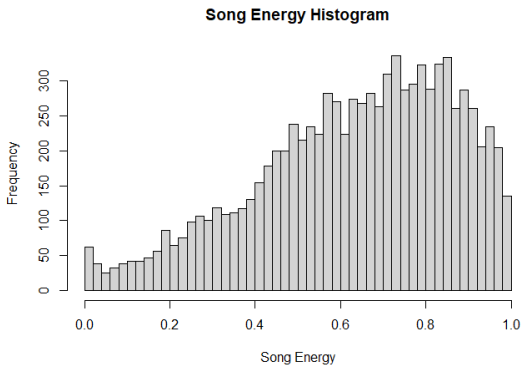




Observations
- the danceability attribute has a normal distribution
- no preprocessing is necessary at this time

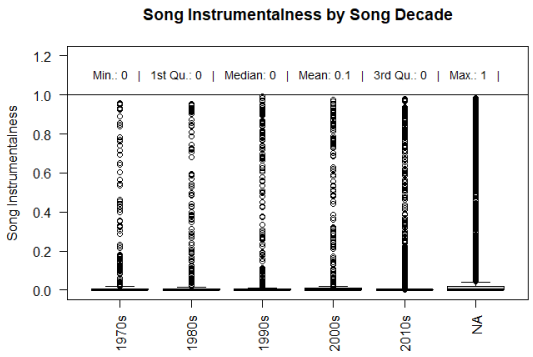
```
#-------------------------------------------------------------------------#
```

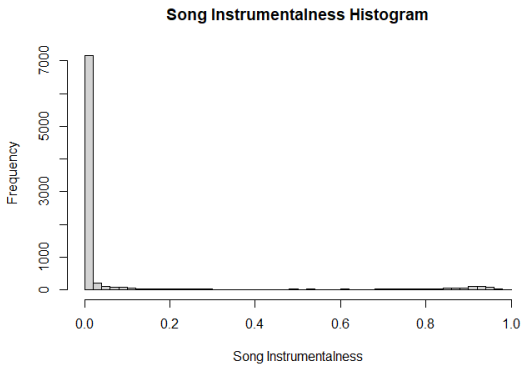




Observations
- the danceability attribute has a skewed distribution
- newer generation songs tend to have higher energy
- no preprocessing is necessary at this time

```
#-------------------------------------------------------------------------#
```

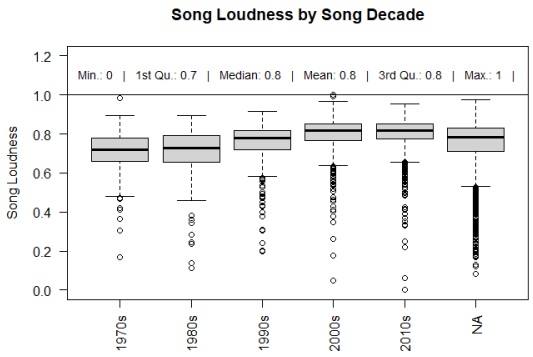


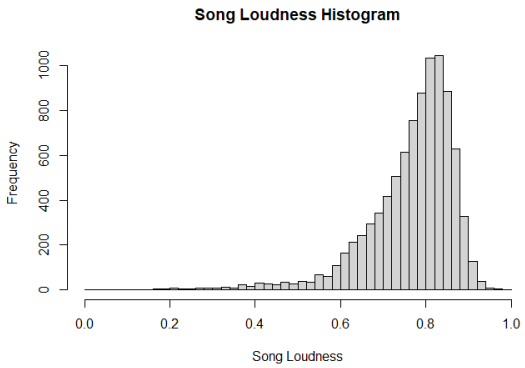


Observations

- the song instrumentalness attribute is highly skewed
- the large majority of songs do not have any instrumentalness
- the same transformation from song acousticness will be applied

```
#-------------------------------------------------------------------------------#

##    Var1 Freq
## 1     0 7426
## 2   0.1  326
## 3   0.2  161
## 4   0.3  111
## 5   0.4   85
## 6   0.5   91
## 7   0.6   93
## 8   0.7  118
## 9   0.8  188
## 10  0.9  398
## 11    1   92

# the frequency distribution after transformation
table(temp_song_T_transformed$inst)

##
##    0  0.5    1
## 7913  498  678

#-------------------------------------------------------------------------------#
```

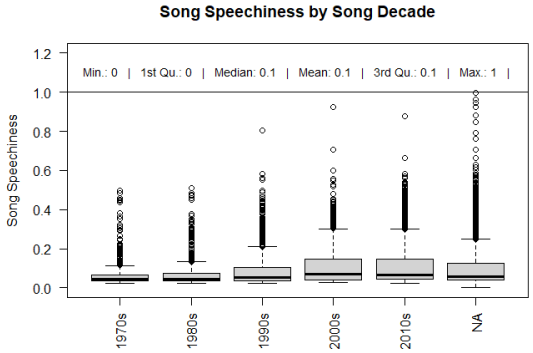


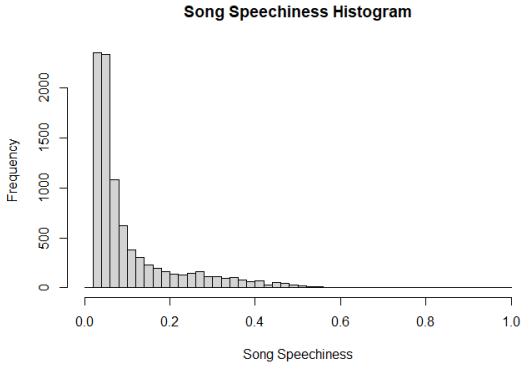


Observations

- the loudness attribute has a skewed distribution
- newer generation songs tend to have higher loudness
- no preprocessing is necessary at this time

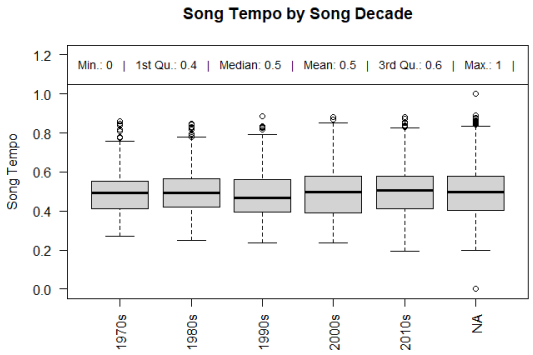
```
#-------------------------------------------------------------------------------#
```

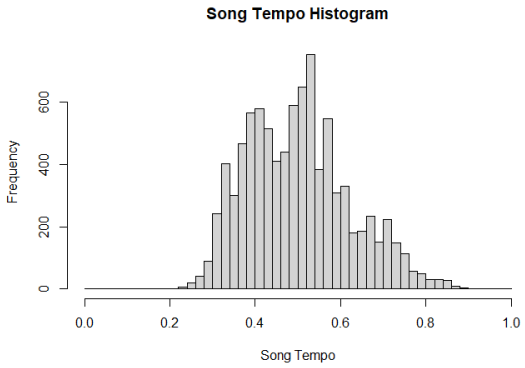




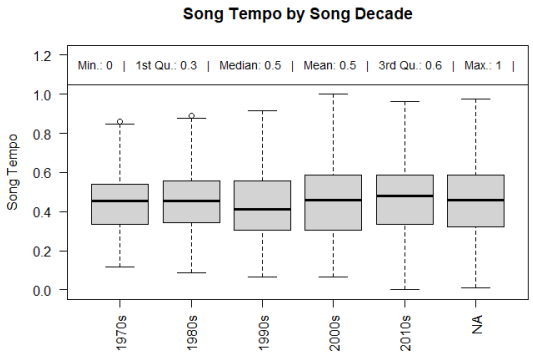

```
#-------------------------------------------------------------------------------#

##    Var1 Freq
## 1     0 3770
## 2   0.1 3426
## 3   0.2  815
## 4   0.3  598
## 5   0.4  313
## 6   0.5  129
## 7   0.6   21
## 8   0.7    4
## 9   0.8    5
## 10  0.9    6
## 11    1    2
```

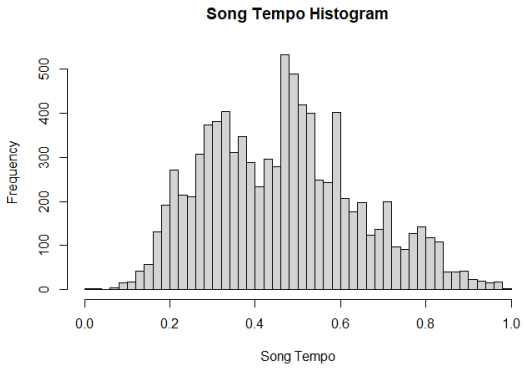
```
# frequency distribution after transformation
table(temp_song_T_transformed$spee)

##
##    0  0.5    1
## 8011 1065   13

#-------------------------------------------------------------------------------#
```



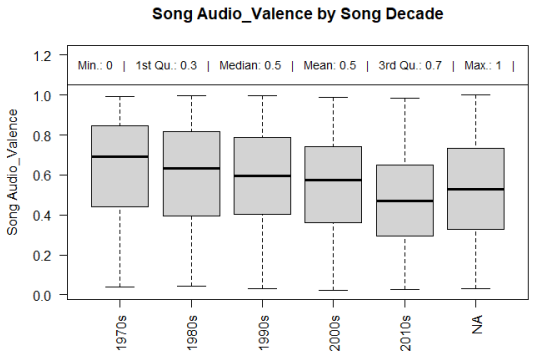


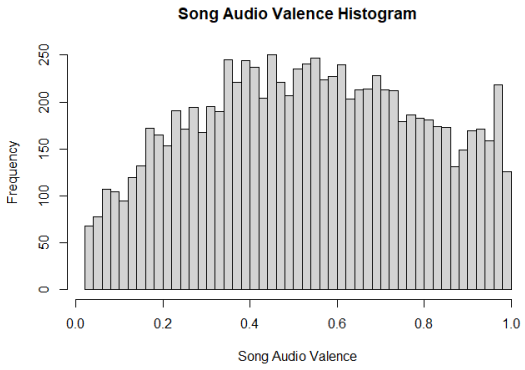

```
## Number of Songs Removed: 61
```






```
#-------------------------------------------------------------------------------#
```



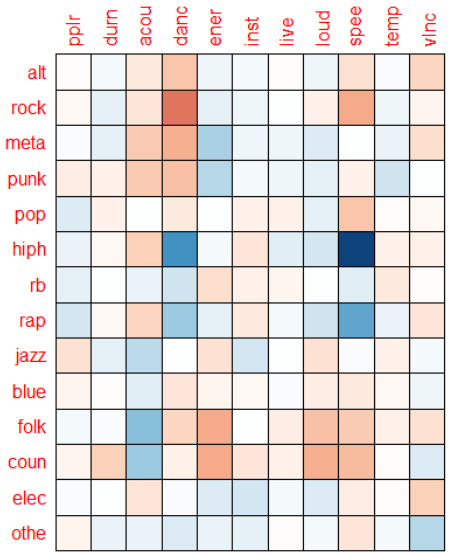



```
#-------------------------------------------------------------------------------#
```

Song Genre and Song Attribute Correlation Matrix

The following pearson correlation matrix shows the correlations between each of the song genres and song attributes.

The darker the blue color means higher positive correlation (closer to 1), and the darker the orange color means lower negative correlation (closer to -1).



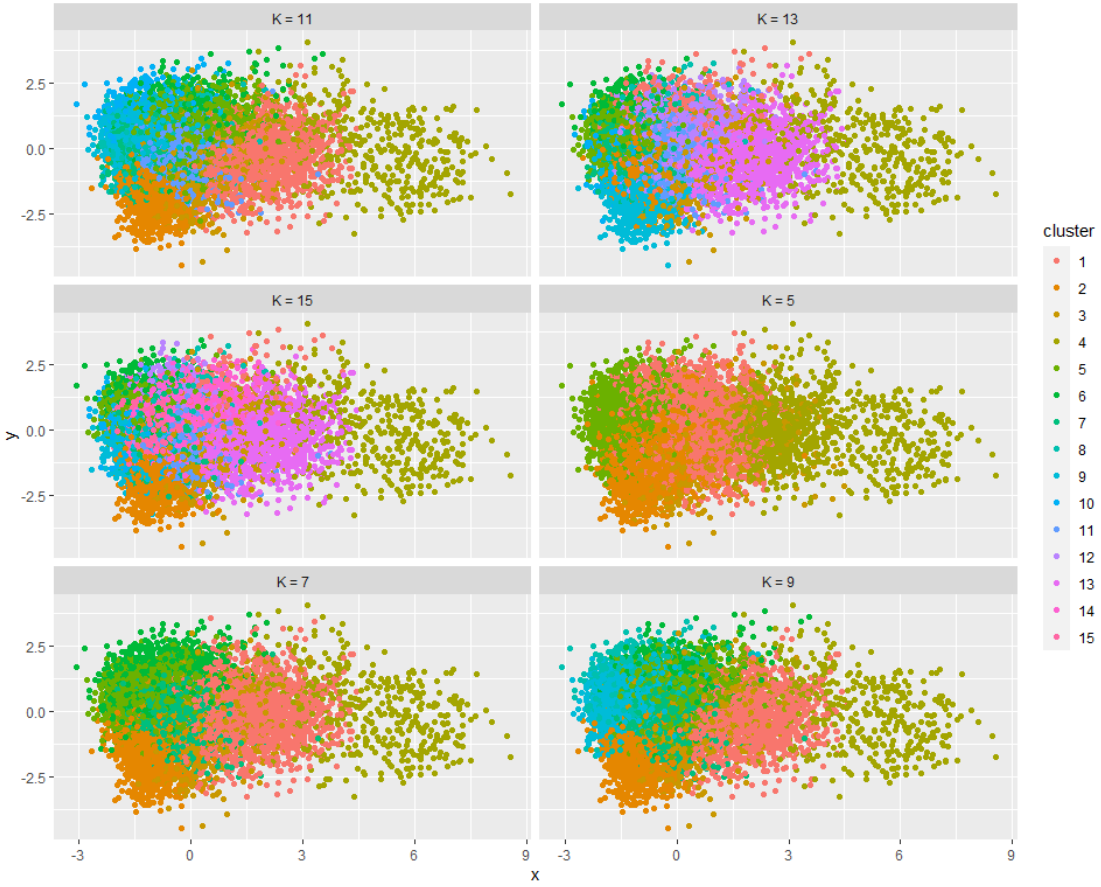
Observations

- there are some interesting correlations between song attributes and song genres
- example 1: alt / rock / metal / punk are less popular than pop / hiphop / rap / r&b
- example 2: jazz / blues / folk / country tend to have more acousticness
- example 3: metal / punk tend to have higher energy than the other genres
- example 4: hiphop / rap have significantly more speechiness than the other genres
- example 5: other has significantly more audio valence than the other genres
- this is encouraging because it means it might be possible to effectively infer song genres

Song Genre Clustering

Based on the song attributes, an unsupervised clustering model will be implemented to create clusters of songs.

These clusters of songs could represent genres or they could represent other similarities between the songs.



Observations

- there must be some level of disctinction based on the song attributes
- it is possible that these clusters could represent genres
- for example, genres have their distinct sounds but do overlap

Association Rules mining

Association rules mining will be performed primarily in the interest of understanding what correlations exist with song popularity.

```
#------------------------------------------------------------------------#

## What Makes a Song Popular in the 1970s?
##

## Top 3 Rules Sorted by Support
##
## Rule Number 1
##  {instr__med,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 1    0.08       0.28     0.27 1.11    33
##
##
## Rule Number 2
##  {instr__med,speech__med} => {poplr__vhigh}
##   support confidence coverage lift count
## 2    0.06       0.23     0.27 1.37    27
##
##
## Rule Number 3
##  {instr__med,tempo__high} => {poplr__high}
##   support confidence coverage lift count
## 3    0.06       0.26     0.24 1.01    27
##
##
## Top 3 Rules Sorted by Confidence
##
## Rule Number 1
##  {instr__med,tempo__high,vlnc__low} => {poplr__high}
##   support confidence coverage lift count
## 1    0.02       0.69     0.03  2.7     9
##
##
## Rule Number 2
##  {dur__med,vlnc__low} => {poplr__high}
##   support confidence coverage lift count
## 2    0.02       0.53     0.04 2.06     9
##
##
## Rule Number 3
##  {tempo__high,vlnc__low} => {poplr__high}
##   support confidence coverage lift count
## 3    0.02       0.53     0.04 2.05    10
##
##
## Top 3 Rules Sorted by Lift
```

```
##
## Rule Number 1
##  {instr__med,tempo__high,vlnc__low} => {poplr__high}
##   support confidence coverage lift count
## 1    0.02       0.69     0.03  2.7     9
##
##
## Rule Number 2
##  {dur__vhigh,ener__high,instr__med} => {poplr__vhigh}
##   support confidence coverage lift count
## 2    0.02       0.43     0.05 2.53     9
##
##
## Rule Number 3
##  {acou__low,instr__med,speech__med} => {poplr__vhigh}
##   support confidence coverage lift count
## 3    0.02       0.41     0.05 2.41     9
##
##
```

Observations

- songs with medium instrumentalness were the most popular
- low to medium speechiness is common in popular songs
- low audio valence was associated with high confidence

```
#-------------------------------------------------------------------------#

## What Makes a Song Popular in the 1980s?
##

## Top 3 Rules Sorted by Support
##
## Rule Number 1
##  {instr__med,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 1    0.09       0.35     0.27 1.19    48
##
##
## Rule Number 2
##  {instr__med,tempo__high} => {poplr__high}
##   support confidence coverage lift count
## 2    0.08       0.36     0.23 1.24    44
##
##
## Rule Number 3
##  {instr__med,vlnc__high} => {poplr__high}
##   support confidence coverage lift count
## 3    0.08       0.35     0.24  1.2    43
##
##
```

```
## Top 3 Rules Sorted by Confidence
##
## Rule Number 1
##  {acou__low,ener__vhigh,loud__vhigh,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 1    0.02       0.67     0.03 2.28    12
##
##
## Rule Number 2
##  {acou__low,loud__vhigh,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 2    0.03       0.65     0.04 2.22    13
##
##
## Rule Number 3
##  {ener__vhigh,loud__vhigh,vlnc__high} => {poplr__high}
##   support confidence coverage lift count
## 3    0.02       0.65     0.03 2.21    11
##
##
## Top 3 Rules Sorted by Lift
##
## Rule Number 1
##  {dur__high,acou__med,speech__low} => {poplr__vhigh}
##   support confidence coverage lift count
## 1    0.03       0.43     0.06 2.65    13
##
##
## Rule Number 2
##  {acou__med,instr__low,speech__low} => {poplr__vhigh}
##   support confidence coverage lift count
## 2    0.02       0.43     0.05 2.62    12
##
##
## Rule Number 3
##  {acou__med,vlnc__low} => {poplr__vhigh}
##   support confidence coverage lift count
## 3    0.02       0.39     0.05  2.4    11
##
##
```

Observations

- some of the same qualities from the 1970s appear in the 1980s
- medium instrumentalness and low to medium speechiness

```
#-----------------------------------------------------------------------#

## What Makes a Song Popular in the 1990s?
##
```

```
## Top 3 Rules Sorted by Support
##
## Rule Number 1
##  {instr__med,vlnc__high} => {poplr__high}
##   support confidence coverage lift count
## 1    0.08       0.39     0.21 1.03    53
##
##
## Rule Number 2
##  {acou__med,vlnc__high} => {poplr__high}
##   support confidence coverage lift count
## 2    0.08       0.36     0.23 0.94    52
##
##
## Rule Number 3
##  {acou__med,dance__high} => {poplr__high}
##   support confidence coverage lift count
## 3    0.08        0.4     0.19 1.04    48
##
##
## Top 3 Rules Sorted by Confidence
##
## Rule Number 1
##  {dance__high,ener__high,speech__low,tempo__high} => {poplr__high}
##   support confidence coverage lift count
## 1    0.02       0.68     0.03  1.8    13
##
##
## Rule Number 2
##  {dance__high,instr__low,loud__low} => {poplr__high}
##   support confidence coverage lift count
## 2    0.02       0.67     0.03 1.75    14
##
##
## Rule Number 3
##  {acou__high,loud__low} => {poplr__high}
##   support confidence coverage lift count
## 3    0.02       0.65     0.03 1.71    13
##
##
## Top 3 Rules Sorted by Lift
##
## Rule Number 1
##  {dur__vhigh,dance__vhigh} => {poplr__vhigh}
##   support confidence coverage lift count
## 1    0.02       0.41     0.06 2.27    15
##
##
## Rule Number 2
##  {dur__vhigh,instr__med,speech__low} => {poplr__vhigh}
```

```
##   support confidence coverage lift count
## 2    0.02        0.4     0.06 2.24    14
##
##
## Rule Number 3
##  {dur__vhigh,vlnc__med} => {poplr__vhigh}
##   support confidence coverage lift count
## 3    0.02       0.38     0.06  2.1    15
##
##
```

Observations

- there seems to be a shift in which song attributes correlate with song popularity
- for example, medium acousticness and high audio valence were not seen until now

```
#------------------------------------------------------------------------#

## What Makes a Song Popular in the 2000s?
##

## Top 3 Rules Sorted by Support
##
## Rule Number 1
##  {instr__med,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 1     0.1       0.41     0.23 1.21    67
##
##
## Rule Number 2
##  {dur__high,instr__med} => {poplr__high}
##   support confidence coverage lift count
## 2    0.08       0.42     0.18 1.23    53
##
##
## Rule Number 3
##  {instr__low,vlnc__high} => {poplr__high}
##   support confidence coverage lift count
## 3    0.08       0.41     0.18 1.19    53
##
##
## Top 3 Rules Sorted by Confidence
##
## Rule Number 1
##  {dur__vhigh,acou__med,instr__med} => {poplr__high}
##   support confidence coverage lift count
## 1    0.03       0.72     0.04 2.11    18
##
##
## Rule Number 2
##  {speech__med,tempo__low,vlnc__high} => {poplr__high}
##   support confidence coverage lift count
```

```
## 2    0.03       0.61     0.05 1.77    23
##
##
## Rule Number 3
##  {ener__vlow,vlnc__low} => {poplr__high}
##   support confidence coverage lift count
## 3    0.03        0.6     0.04 1.76    18
##
##
## Top 3 Rules Sorted by Lift
##
## Rule Number 1
##  {dur__high,acou__med,dance__high,instr__low} => {poplr__vhigh}
##   support confidence coverage lift count
## 1    0.02       0.59     0.04 2.59    16
##
##
## Rule Number 2
##  {ener__high,instr__low,speech__med} => {poplr__vhigh}
##   support confidence coverage lift count
## 2    0.02       0.55     0.04 2.42    16
##
##
## Rule Number 3
##  {dur__high,dance__high,instr__low} => {poplr__vhigh}
##   support confidence coverage lift count
## 3    0.03       0.54     0.05 2.38    19
##
##
```

Observations

- not much stands out in this decade compared to the others
- it seems like more of a blend of the previous decades

```
#------------------------------------------------------------------------#

## What Makes a Song Popular in the 2010s?
##

## Top 3 Rules Sorted by Support
##
## Rule Number 1
##  {acou__med,instr__low} => {poplr__vhigh}
##   support confidence coverage lift count
## 1    0.09       0.44     0.21 1.39   195
##
##
## Rule Number 2
##  {instr__med,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 2    0.09       0.35     0.25 1.15   188
```

```
##
##
## Rule Number 3
##  {instr__low,loud__vhigh} => {poplr__vhigh}
##   support confidence coverage lift count
## 3    0.09       0.45     0.19  1.4   182
##
##
## Top 3 Rules Sorted by Confidence
##
## Rule Number 1
##  {dance__high,ener__high,instr__low,loud__vhigh} => {poplr__vhigh}
##   support confidence coverage lift count
## 1    0.03       0.67     0.04 2.09    54
##
##
## Rule Number 2
##  {dance__high,instr__low,loud__vhigh,speech__med} => {poplr__vhigh}
##   support confidence coverage lift count
## 2    0.02       0.63     0.03 1.97    46
##
##
## Rule Number 3
##  {dance__high,instr__low,loud__vhigh,vlnc__high} => {poplr__vhigh}
##   support confidence coverage lift count
## 3    0.02       0.61     0.04  1.9    48
##
##
## Top 3 Rules Sorted by Lift
##
## Rule Number 1
##  {dance__high,ener__high,instr__low,loud__vhigh} => {poplr__vhigh}
##   support confidence coverage lift count
## 1    0.03       0.67     0.04 2.09    54
##
##
## Rule Number 2
##  {dance__high,instr__low,loud__vhigh,speech__med} => {poplr__vhigh}
##   support confidence coverage lift count
## 2    0.02       0.63     0.03 1.97    46
##
##
## Rule Number 3
##  {dance__high,instr__low,loud__vhigh,vlnc__high} => {poplr__vhigh}
##   support confidence coverage lift count
## 3    0.02       0.61     0.04  1.9    48
##
##
```

Observations
- low instrumentalness not medium is associated with high popularity
- high danceability appears and is associated with high confidence + lift

```
#------------------------------------------------------------------------#

## What Makes a Song Popular in the unknown years?
##

## Top 3 Rules Sorted by Support
##
## Rule Number 1
##  {instr__med,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 1    0.07       0.31     0.24 1.04   344
##
##
## Rule Number 2
##  {acou__med,instr__med} => {poplr__high}
##   support confidence coverage lift count
## 2    0.06       0.33     0.17  1.1   256
##
##
## Rule Number 3
##  {instr__med,tempo__high} => {poplr__high}
##   support confidence coverage lift count
## 3    0.05       0.31     0.17 1.03   243
##
##
## Top 3 Rules Sorted by Confidence
##
## Rule Number 1
##  {acou__vhigh,speech__med,vlnc__low} => {poplr__high}
##   support confidence coverage lift count
## 1    0.02       0.51     0.04 1.69    99
##
##
## Rule Number 2
##  {acou__vhigh,ener__vlow,loud__vlow,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 2    0.02       0.47     0.04 1.57    96
##
##
## Rule Number 3
##  {acou__vhigh,ener__vlow,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 3    0.02       0.47     0.05 1.56   109
##
##
## Top 3 Rules Sorted by Lift
```

```
##
## Rule Number 1
##  {acou__vhigh,speech__med,vlnc__low} => {poplr__high}
##   support confidence coverage lift count
## 1    0.02       0.51     0.04 1.69    99
##
##
## Rule Number 2
##  {acou__vhigh,ener__vlow,loud__vlow,speech__med} => {poplr__high}
##   support confidence coverage lift count
## 2    0.02       0.47     0.04 1.57    96
##
##
## Rule Number 3
##  {instr__low,vlnc__high} => {poplr__vhigh}
##   support confidence coverage lift count
## 3    0.02       0.18     0.14 1.56   112
##
##
```

Observations

- very high acousticness appears in several rules
- this is odd because most songs had low acousticness
- there is definitely something different about these songs

```
#------------------------------------------------------------------------#
```

Perform KNN Trials to determine a K Value

An iteration of trials is performed using a KNN model with different K values on data for which the genre is known.

Since it is possible for a song to have multiple genres, iterate through each song and each genre one by one, and determine if the song fits in that genre.

First the data is split 80 / 20 training and testing and then the KNN model is implemented, the accuracy is measured based on the percent correct of the test data.

```
## Overall Accuracy by Number of K Nearest Neighbors
##

##  K3  K5  K7  K9 K11 K13 K15 K17 K19 K21 Avg
##  87  88  88  88  88  89  89  89  89  89  88
```

Genre	K3	K5	K7	K9	K11	K13	K15	K17	K19	K21	Avg
alt	86	87	88	89	89	89	89	89	89	89	88
blues	96	97	97	97	97	97	97	97	97	97	97
country	90	91	91	92	92	92	92	92	92	92	92
elect	89	90	90	90	90	90	90	90	90	90	90
folk	91	93	93	94	94	94	94	94	94	94	94
hiphop	82	83	83	84	84	84	84	85	84	84	84
jazz	96	97	97	97	97	97	97	97	97	97	97
metal	96	96	96	96	96	96	96	96	96	96	96
other	85	85	86	87	87	87	87	87	87	87	87
pop	61	62	65	66	66	66	68	66	66	67	65
punk	94	95	95	95	95	95	95	95	95	95	95
rap	90	90	90	91	90	91	90	90	91	91	90
rb	86	88	89	89	89	89	89	89	89	89	88
rock	71	73	72	73	73	74	74	74	74	73	73

Observations

- able to effectively determine song genre with 88% on avg accuracy
- K values of 9-21 all produced very similar accuracies
- moving forward with a conservative middle of the pack k value of 11
- do not want to risk selecting too many k nearest neighbors

Infer Missing Song Genres with KNN model K = 11

All of the missing song genres are inferred using a KNN model with a K value of 11. The training data used is all of the data with known genres.

```
##   alt rock metal punk pop hiphop rb rap jazz blues folk elect other
## 1  12  185     4    7 355    222  9  30    1     0   12     6    61
```

Perform KNN Trials to determine a K Value for Song Decade

An iteration of trials is performed using a KNN model with different K values on data for which the song decade is known.

First the data is split 80 / 20 training and testing and then the KNN model is implemented, the accuracy is measured based on the percent correct of the test data.

```
## Overall Accuracy by Number of K Nearest Neighbors
##
```

NumK	Accuracy
3	46.22
5	48.74
7	50
9	50.46
11	50
13	50.34
15	50.69
17	51.6
19	52.06
21	52.97

Observations
- the accuracy for inferring the song genre is very good
- the accuracy for inferring the song decade is not good
- it might be better to model with the missing song decades

Data Modeling

Now that the data has been explored and cleaned, several models will be implemented to predict the song popularity and the results will be compared.

The song popularity will be predicted via both classification and regression. Song popularity will need to be binned prior to classification models.

The models that will be implemented include, decision tree, naive bayes, k nearest neighbors, support vector machine, and neural net.

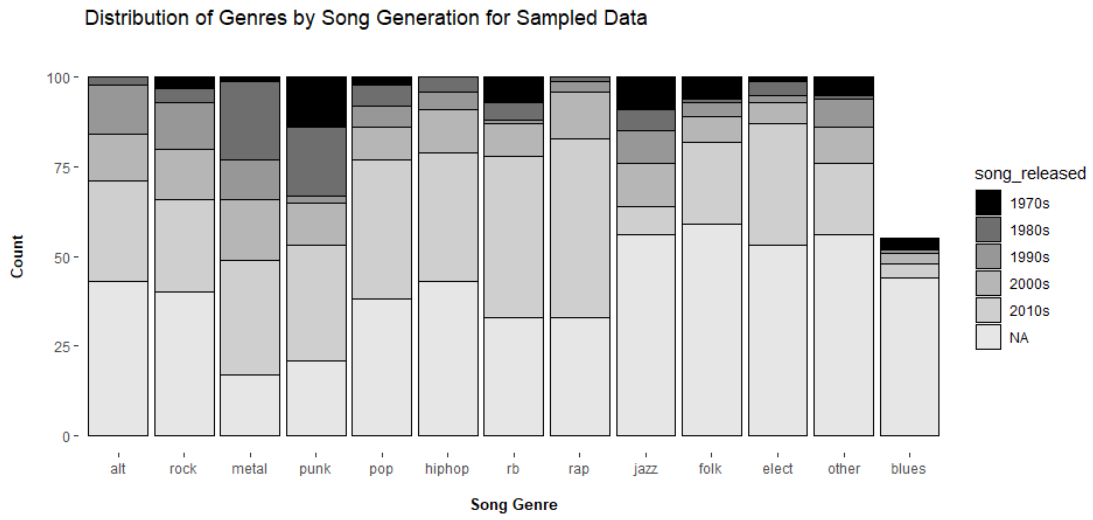
The objective is to find a model that is suitable for accurately predicting the song popularity.

```
#-------------------------------------------------------------------------#
```

Sampling

Due to the distribution of song genres being skewed, a sample will be taken with an even amount of songs from each genre.

```
## Using song_released as id variables
```



Observations
- did not get full 100 songs from song genre = blues due to lack of data
- this makes sense becuase blues is not a very popular genre
- moving forward regardless of the reduced sample size for the blues genre

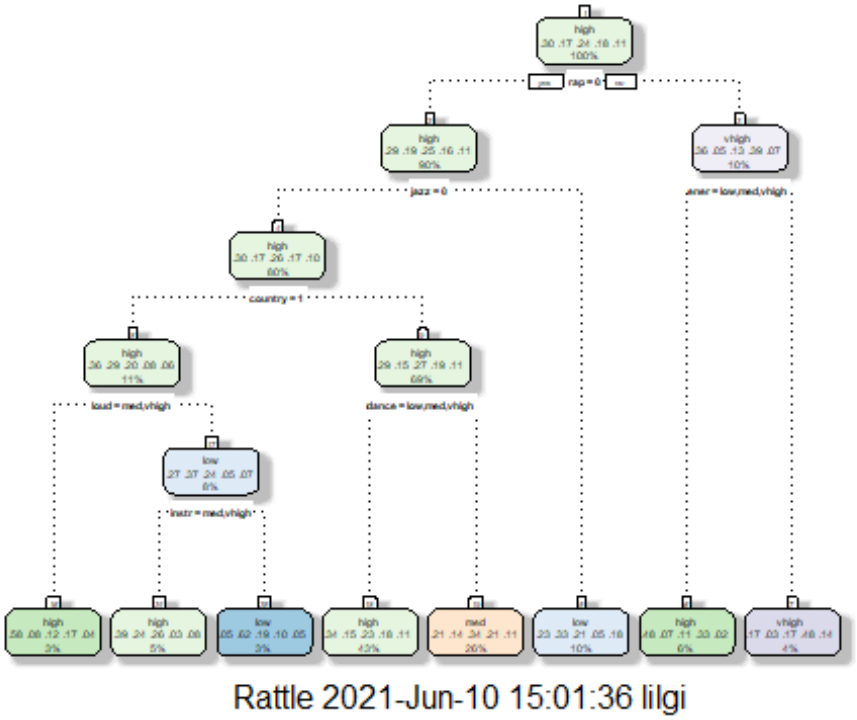
```
#-------------------------------------------------------------------------#
```

```
#-----------------------------------------------------------------------#
```

Decision Tree Model 1

Decision tree classification model using the rpart package.

```
## Decision Tree Model 1 - Accuracy = 33.7 %
##
```




```
##
##          high low med vhigh vlow
##    high    40   3  11     1    0
##    low     16   6   9     0    0
##    med     23   7  11     4    0
##    vhigh   17   0  12     5    0
##    vlow     9   5   5     0    0
```

Observations

- the model did not perform well
- possible causes - noise, discretization method, attributes, model

```
#-----------------------------------------------------------------------#
```

```
#------------------------------------------------------------------------#
```

Decision Tree Model 2

Decision tree classification model using the C5.0 package.

	Metric
Accuracy	0.2580645
Kappa	0.0369926
AccuracyLower	0.1968169
AccuracyUpper	0.3271952
AccuracyNull	0.3118280
AccuracyPValue	0.9537133
McnemarPValue	0.3295917

```
##
##  Decision Tree 2 Confusion Matrix
```

	high	low	med	vhigh	vlow
high	19	8	20	5	4
low	4	8	12	4	4
med	18	5	11	8	3
vhigh	11	6	6	9	2
vlow	6	5	7	0	1

```
##
##  Decision Tree 2 Precision and Recall

##                Precision     Recall
## Class: high  0.33928571 0.32758621
## Class: low   0.25000000 0.25000000
## Class: med   0.24444444 0.19642857
## Class: vhigh 0.26470588 0.34615385
## Class: vlow  0.05263158 0.07142857
```

Observations

- the model did not perform well
- decision tree is probably not effecive for this prediction task

```
#------------------------------------------------------------------------#
```

```
#-------------------------------------------------------------------------#
```

Naive Bayes Model 1

Naive Bayes classification model using the e1071 package.

	Metric
Accuracy	0.3387097
Kappa	0.1494851
AccuracyLower	0.2710797
AccuracyUpper	0.4115750
AccuracyNull	0.2849462
AccuracyPValue	0.0631416
McnemarPValue	0.0579646

Observations
- results about the same as decision tree
- possible causes - noise, discretization method, attributes, model

```
#-------------------------------------------------------------------------#
```

Naive Bayes Model 2

Same as previous but with various discretization methods.

```
## Results from Different Number of Bins
##
##  38.17%   25.81%   33.33%   19.46%   31.89%   15.14%   22.16%   19.89%   2
3.37%   11.89%   17.84%   17.49%   16.94%   11.96%   18.58%   10.44%   9.29%
10.5%   12.57%   11.41%   15.3%   9.34%   9.94%   7.18%   9.39%   7.14%   7.7
8%   4.47%   5%   4.42%   8.33%   7.73%
```

Observations
- changing the bins did not improve the result
- naive bayes does not seem to be effective either

```
#-------------------------------------------------------------------------#
```

K Nearest Neighbors

K nearest neighbors model with a K value of 21.

	Metric
Accuracy	0.3440860
Kappa	0.1367596
AccuracyLower	0.2761134
AccuracyUpper	0.4171185
AccuracyNull	0.3333333
AccuracyPValue	0.4046248
McnemarPValue	0.0068896

```
##
##  KNN Confusion Matrix
```

	high	low	med	vhigh	vlow
high	26	3	15	11	1
low	6	7	15	4	0
med	14	6	18	7	0
vhigh	9	1	11	13	0
vlow	7	6	3	3	0

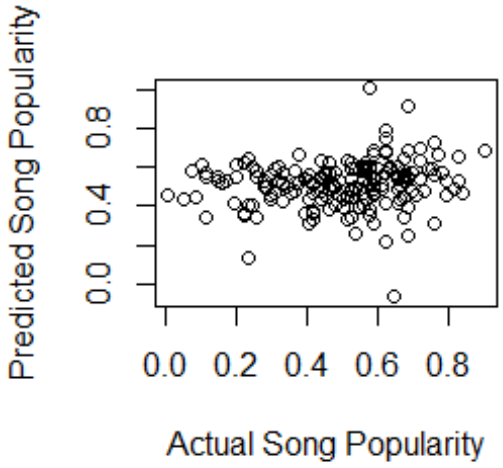
```
##
##  KNN Precision and Recall

##              Precision    Recall
## Class: high  0.4642857 0.4193548
## Class: low   0.2187500 0.3043478
## Class: med   0.4000000 0.2903226
## Class: vhigh 0.3823529 0.3421053
## Class: vlow  0.0000000 0.0000000
```

Observations
- the knn model did not perform well
- none of the classification models have produced good results

SVM Model

A regression SVM model will be implemented to predict song popularity.

```
##  Accuracy = 68.02%
##  Avg Abs Error = 16.12
##  Prediction Bias = 2.12%
```




```
# kernel = polynomial, cost = 1, degree = 3 -----> accuracy 66%
# kernel = polynomial, cost = 5, degree = 3 -----> accuracy 59%
# kernel = polynomial, cost = 10, degree = 3 -----> accuracy 55%
# kernel = polynomial, cost = 1, degree = 4 -----> accuracy 66%
# kernel = polynomial, cost = 1, degree = 5 -----> accuracy 64%
# kernel = polynomial, cost = 1, degree = 6 -----> accuracy 62%
# type = nu-regression, kernel = linear, cost = 1 -----> accuracy 64%
# type = nu-regression, kernel = linear, cost = 25 -----> accuracy 62%
# type = nu-regression, kernel = linear, cost = 50 -----> accuracy 66%
# type = nu-regression, kernel = linear, ost = 500 -----> accuracy 66%
# type = nu-regression, kernel = polynomial, cost = 5 -----> accuracy 60%
```

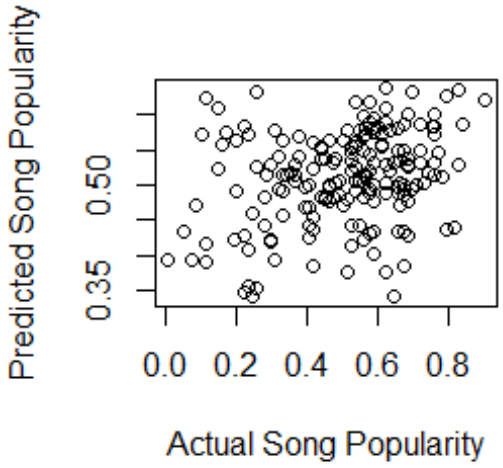
Observations

- the polynomial kernel with a cost of 1 and degree of 3 gave best results
- accuracy was 66% when predicting a continuous song popularity
- the SVM model has better results than the decision tree and naive bayes

Random Forest Model

A regression random forest model will be implemented to predict song popularity.

```
##  Accuracy = 72.21%
##  Avg Abs Error = 14.01
##  Prediction Bias = 1.51%
```

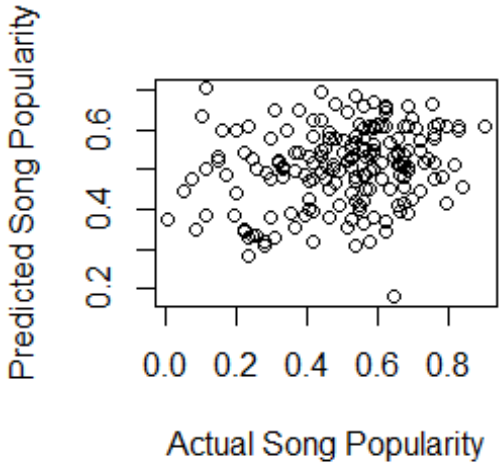


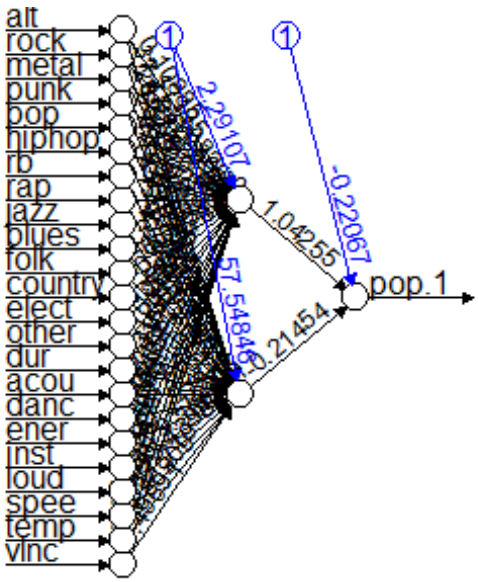
Observations
- the random forest with 100 trees gave best results
- after 100 trees the accuracy begins to flatten
- the random forest was a little better than the SVM

Neural Net

A regression neural net will be implemented to predict song popularity.

```
##  Accuracy = 70.43%
##  Avg Abs Error = 14.9
##  Prediction Bias = 0.26%
```





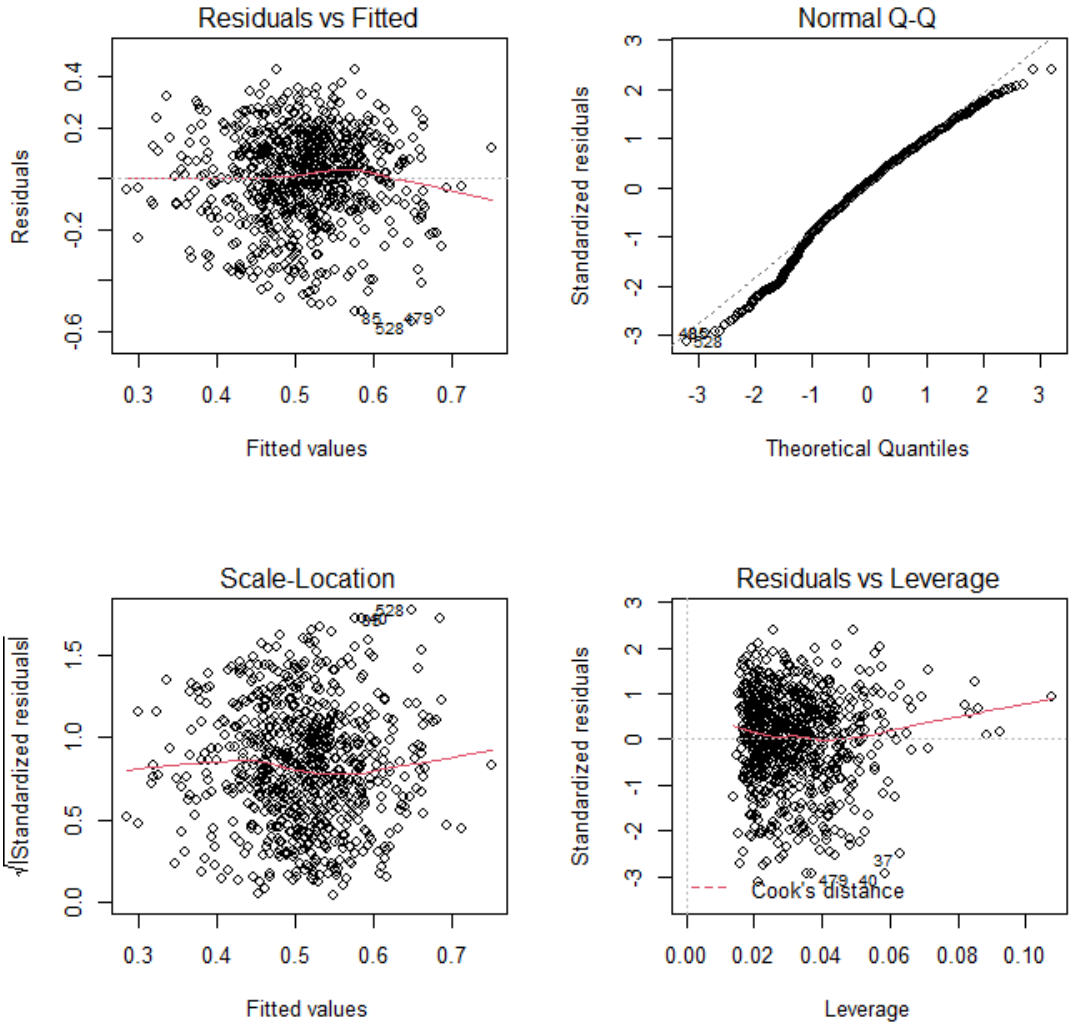
Observations

- the neural net produced similar results to SVM and random forest
- adding additional hidden layers did not increase the accuracy

Linear Model

A regression linear model will be implemented to predict song popularity.

```
##  Accuracy = 72.01%
##  Avg Abs Error = 14.11
##  Prediction Bias = 0.8%
```



Observations

- the linear model performed similar to the other models
- the regression models had similar performance

Project Conclusion

Through the analysis of the two original datasets from Kaggle along with the two datasets retrieved from Wikipedia, the conclusion will answer the original questions stated in the introduction.

How have popular songs change overtime?

As noted in the introduction, music has evolved over time. Song attributes such as acousticness, danceability, loudness, tempo, etc. have changed with culture and generations. The findings in this project reflect that song popularity has increased positively over time, with the highest frequency of popular songs in the 2010s. Over time the types songs that are popular have also changed. For example, alternative and Metal started off popular but over time have declined in popularity. Some genres have increased in popularity over time, such as Country and Pop. In the 1970s and 1980s, songs with low to medium instrumentalness and low to medium speechiness correlated with higher popularity. In the 1990s and 2000s, attributes such as acousticness and audio valence correlate with popular songs. Additionally, in the 2000s, there was a shift to a new song attribute, danceability, which was found to be correlated with song popularity.

Are there certain attributes that correlate with popular songs?

The findings indicate that there are certain attributes that correlate with popular songs. One of which was whether or not the song was a single. A single is typically a song that is released earlier and separately from an album, and then later, appear on an album. A single is important because it gives the consumer a taste of the artists' new music and helps promote the album because consumers have gotten familiar to the music from the single song release. The data shows that there is a positive correlation between a song single and song popularity. Songs that come out as singles generally have 5%-10% greater song popularity. The data also shows that popular songs are largely defined by the generation, and what a particular generation is listening to and liking at the moment. In 1970 and 1980s, popular possessed qualities such as medium instrumentalness, high tempo, low audio valence and medium speechiness. In the 1990s- 2010s, there's a shift in song attributes related to popular songs with attributes such as medium to high acousticness, high energy, high tempo, low speechiness, and high danceability. From a general music knowledge, these findings make sense to see a shift in song attributes of popular songs between decades. Music groups like boy bands like *NSYNC or Backstreet Boys that promote songs that have high energy, high danceability originated in the mid 1990s and these groups are believed to be two of the most popular bands with many popular songs then and still now.

Can the popularity of a song be predicted based on its attributes?

The overall conclusion is yes, to a certain extend song popularity can be predicted. It became evident, however, after applying several different prediction techniques predicting song popularity is a regression problem and not a classification problem. The classification models - decision tree, naive bayes, and k nearest neighbors - did not perform well. Furthermore, even after removing outliers, tuning model parameters, and discretizing the

data in multiple ways, the classification models still did not perform well. On the other hand, the regression models - support vector machine, random forest, neural network, and general linear models - did perform well. These models were able to predict song popularity with about 70% accuracy.

This is all taking into consideration major challenges with data quality. The data was cleaned up as much as possible, including removing outliers, transforming the song attributes, inferring missing song genres, and removing non-significant variables, but yet was far from perfect. When taking that into consideration, it is actually pretty impressive that the song popularity was able to be predicted as accurately as it was. Not to mention, there were large biases in the types of songs in the dataset due to the listening preferences and playlists of the creator of the dataset, but this was mitigated prior to the predictive modeling by sampling the data. The quality of the prediction can only be as good as the quality of the data that goes into it.

Final thoughts and next steps

The results from this project were promising. This project shows that it is possible to predict the popularity of a song. If more resources were invested to improve the quality of the data, or to collect more data such as additional song attributes, the accuracy of predicting the song popularity could only increase and it may very well prove to be worthwhile for music production company to do. This project only scratches the surface of what can potentially be done in the music industry, but it does provide a foundation for analytics in music that can continue to built on.