Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

Dataset

Why did we choose it

The dataset that we chose was Top 50 Spotify Songs - 2019 from the website Kaggle. This dataset lists 50 songs in their ranking order, 3 string, and 11 integer columns. The three strings columns are names of songs, artists, and genres. The 11 integer columns include Beats.Per.Minute, Energy, Danceability, and more. We chose this dataset because we all love music and thought it would be interesting to look into. We also saw that this dataset was clean and not too big so it would be easy to work with.

Big Question

The big question we wanted to look at was whether there was any relationship in the data that linked which songs ranked higher in the top 50. This dataset had lots of characteristics of the songs that would give us insight into the type of songs and hopefully show something that explained why some songs ranked higher than others. All of the columns described different aspects of the song to give us a better picture of each song. We also created our own dataset to go with the original dataset of all the lyrics of each song. This will allow us to see if there are any words or phrases that are frequent in the songs and can correlate to the ranking of songs. There were also many Spanish songs included in the data set, so we decided to further question what made these songs make the list, and if they had any similarities or differences with the English ones.

Processing

When initially looking at the dataset there were no processing problems that stood out right away. The only thing that was a little odd was the Loudness..dB.. was recorded as negative numbers ranging from -2 to -11. This did not affect our analysis of the data so we left it. We added a few columns to work with. One of those columns was tuning the genres into broader ranges. We took all the different styles of pop and just labeled them pop and did the same thing with the rap songs and other genres. Another column we created was trying to divide the ranks into clusters to look at. We convert the rank into clusters of 2,3,4, and 5. We were not sure which one would work best with the data so we tested all of them. A column was also added into the excel file to flag if the song contained Spanish or not.

Analysis

Excel-Pivot Table

Before we uploaded the data into RStudio we ran a few tests in excel by creating a pivot chart. We took a deeper look at the 3 original strings and the extra general column we made. The name of the songs did not lead to anything insightful because they were all listed only once. The artist's name brought more insight with 10 artists

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

having more than one song in the top 50 and Ed Sheeran having 4 shown in figure 1. When taking a look into the genres, 13 of them showed up more than once. Dance-pop was the most with 8 and pop with second with 7 reappearances shown in figure 2. With the top two both being pop we make that second column of the shortened genres. Looking into genre 2.0 the different pop songs equal 23 of the 50 songs shown in figure 3.

Row Labels	Count of Artist.Name
Ed Sheeran	4
Marshmello	2
Ariana Grande	2
The Chainsmokers	2
J Balvin	2
Shawn Mendes	2
Billie Eilish	2
Sech	2
Post Malone	2
Lil Nas X	2
Jonas Brothers	1
ROSALÍA	1
Drake	1
MEDUZA	1
Ali Gatie	1
Martin Garrix	1
Y2K	1
Tones and I	1
Daddy Yankee	1
Maluma	1

Figure 1

Row Labels	Count of Genre
dance pop	8
pop	7
latin	5
canadian hip hop	3
edm	3
reggaeton	2
panamanian pop	2
reggaeton flow	2
canadian pop	2
brostep	2
country rap	2
dfw rap	2
electropop	2
pop house	1
trap music	1
r&b en espanol	1
atl hip hop	1
boy band	1
big room	1
australian pop	1
escape room	1
Grand Total	50

Figure 2

Row Labels	Count of Genre2.0
pop	23
latin	5
rap	4
reggaeton	4
hip hop	4
edm	3
brostep	2
escape room	1
boy band	1
trap	1
big room	1
r&b en espanol	1
Grand Total	50

Figure 3

Rstudio-Summary and Correlation

To start in Rstudio we did a summary of the dataset to get a closer look at the data and see if there are any outliers that we needed to look into. The summary gave no red flags so we did not look deeper into any of the columns. Next, we ran a ggpairs plot to see all of the numeric column's correlations to each other. We wanted to look specifically at how they all related to the rank. There was less of a correlation than we thought there was going to be. There were only 3 with a higher than .2 correlation to the

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

rank shown in figure 4. Taking a look at the rest of the correlation we did not find as many highly correlated pairs then we would think. Before starting we thought danceability and energy would be highly correlated or loudness and energy along with many others. Only two correlated above .5. Speechineess and Beats.per.mintue correlated to .557 and loudness and energy correlate to .671. The majority of them correlated below .1 shown in figure 4.

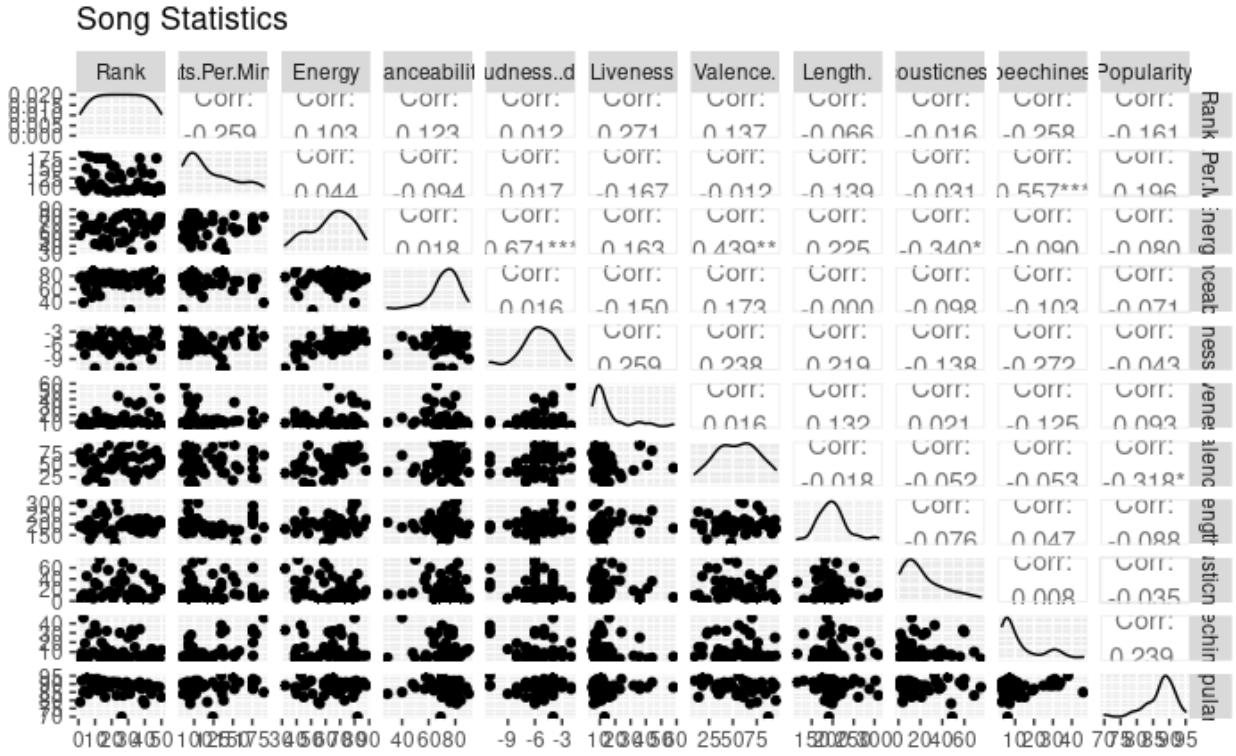

Figure 4

Scatter Plots

Our next step was to look into the top three correlated matches and convert them into scatter plots with relation to the rank on a color scale. All three of these scatter plots did not lead to anything showing that their correlation could be drawn back to the ranking of the song. The top-ranking songs were scattered all over each of these 3 plots. With these three plots not showing anything we took a look and made scatter plots with the beats per minute vs the rest of the columns. Three of the plots stood out with the coloring of the ranks. The three plots were Length. vs. Beats.Per.Minute, Liveness vs. Beats.Per.Minute, and Speechiness. vs. Beats.Per.Minute. The liveness plot was the most oblivious with the ranking. The top ranks show to be between the 5 and 20 loudness shown in figure 5. This makes sense because liveness has the highest collation to rank.

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

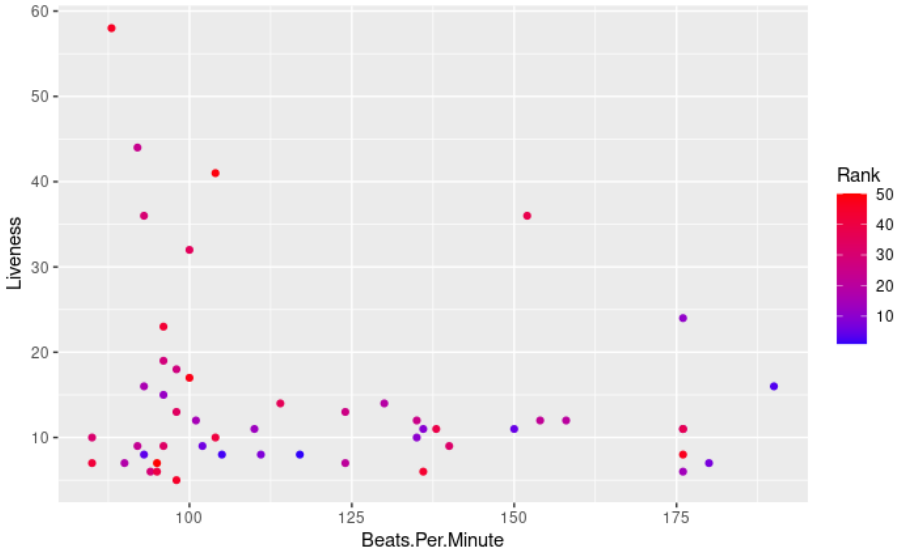

Figure 5

The next two are not as obvious of the relation between the rank. They each have at least one outlier. We wanted to look at the other high correlation to rank being Speechiness. The high ranking songs show between 0 to 10 with one interesting outlier at the top of the speechiness shown in figure 6. Taking a deeper look that song was number 3 boyfriend by Ariana Grande it also has the highest beats per minute.

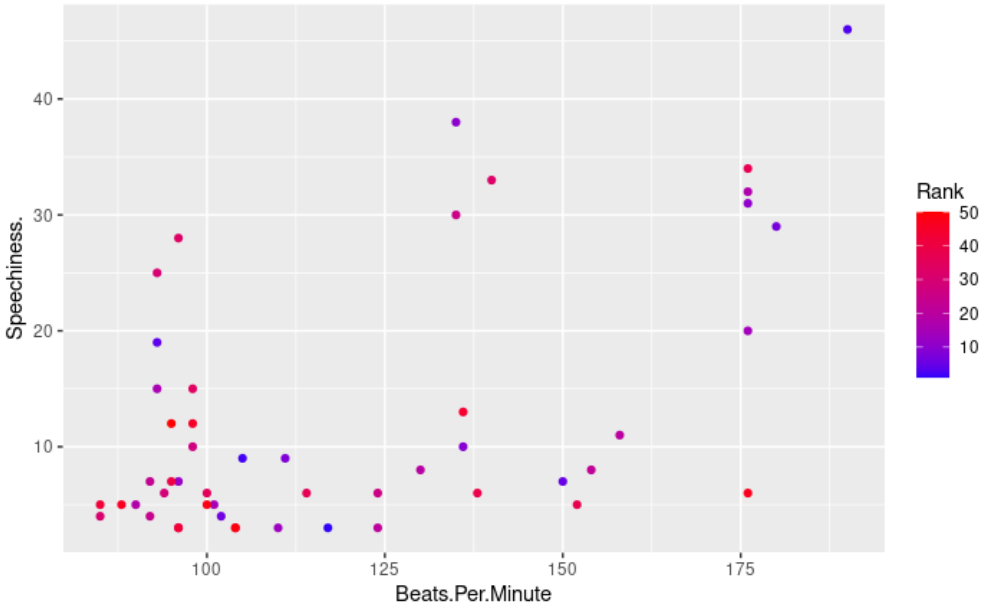

Figure 6

Length was not very highly correlated to rank but the graph shows some relation to the rank. As shown in the figure 7 it seems the top ranks are between length of 175 to 225. There is also one outlier with the length being over 300. That song was number two China by Anuel AA.

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

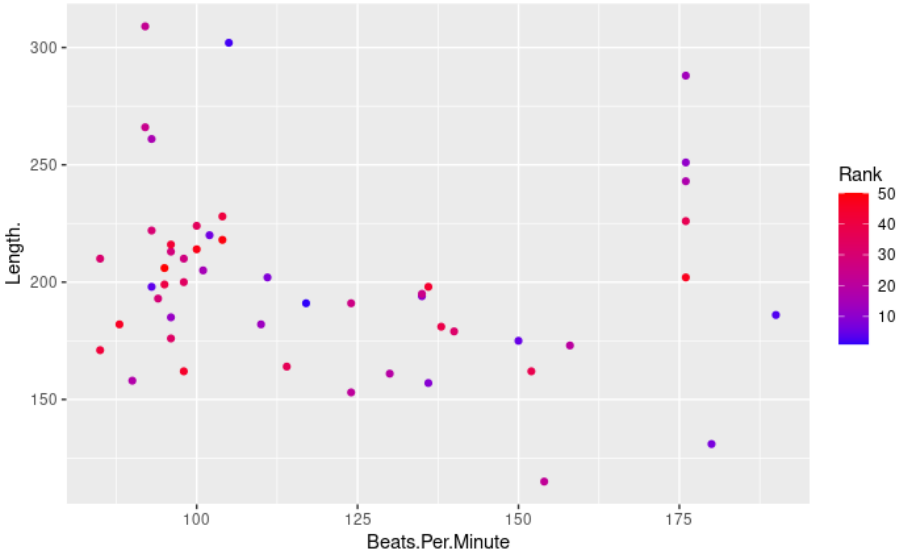

Figure 7

Analyzing the Spanish Songs

After we created a column in the excel file notifying if the song contained Spanish or not, we calculated the averages for the other variables based on those two groups, and placed them in a table as seen in figure 8. It was our initial prediction that the Spanish songs would have a higher average for variables, such as BPM, Energy, and Danceability due to their prevalence of Latin, Reggaeton, and Pop songs, while the English songs had a greater variety of genres. This is seen in figure 8, where the Spanish BPM, Danceability, and Energy are all greater than their English counterparts.

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

Variable	English	Spanish
BPM	117.8	125.9
Energy	60.6	72.85
Danceability	69.9	75.2
Loudness	-6.2	-4.3
Liveness	14.8	14.2
Valence	50.8	64.3
Length	189.8	229.7
Accousticness	22.9	20.4
Speechiness	11.1	16
Popularity	87.2	88.35

However when we used R to get a closer look at the distributions, the data started to tell a different story. Using R, we created 2 subsets of the original data frame, one including the songs that contained Spanish, and the ones that did not. We then used these data frames to make numerous histograms representing the different variables. See figure 9.

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

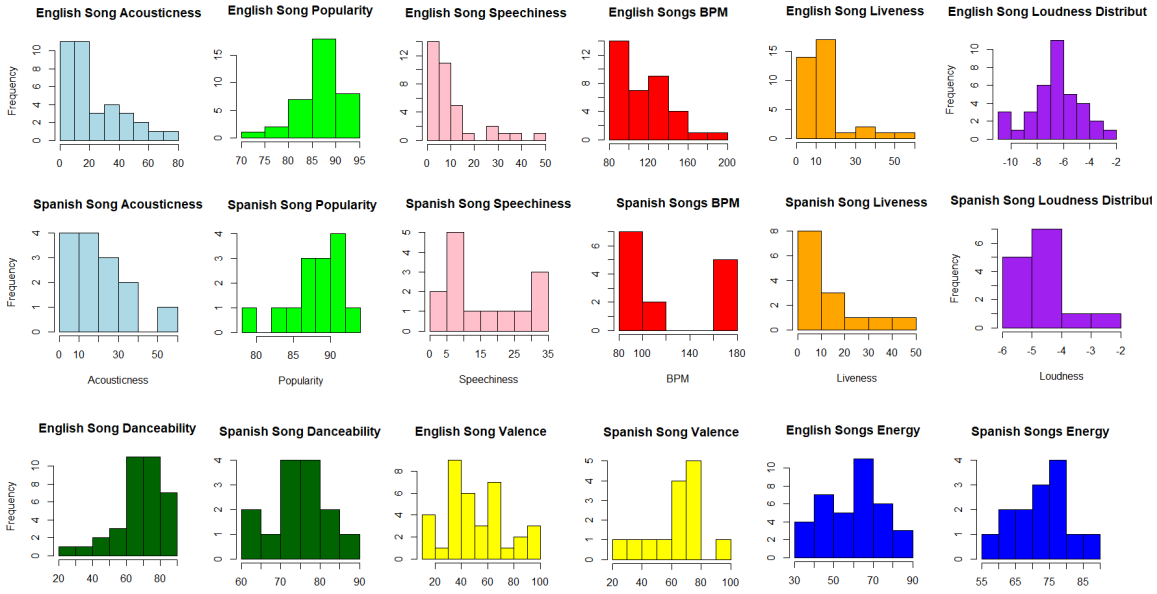

Figure 9

With BPM, the English songs had a positive skew, but the Spanish histogram was much more interesting. It is bimodal with 2 peaks, with the songs either having a BPM in the 80 - 90 range, or 160-180 bpm range. We wondered if this was due to the fact that the Spanish songs have fewer genres, and that maybe a specific genre correlated to a certain BPM, but that did not seem to be the case when further analyzing the values in Excel. The English songs also never got as fast as the Spanish ones seemed to.

The energy curves for both categories seemed to be pretty close to a symmetrical bell curve, but the English ones had a higher range. Again, this could possibly be due to the greater variety in genres, and the fact that they represented a larger portion of the dataset.

For danceability, the English songs had a negative, with most songs in the 60-90 range. The Spanish ones had a bell curve, with the songs also in the 60-90 range, so the two categories are actually very similar in this aspect. Even though the Spanish songs had an overall higher average for danceability, it seems that a few outliers is what is mainly causing the English average to be dragged down.

A few other interesting histograms were for Valence, Loudness, and Speechiness. Spanish songs tended to have a higher valence than English ones, suggesting that they

are more positive. With Loudness, English songs had a bell curve, while Spanish ones had a positive skew. English songs also had a larger range for this, while the Spanish ones tended to all be around the same volume. Speechiness was also an interesting distribution to examine. The Spanish histogram had a bimodal distribution with either speechiness between the 5-10 or 30-35 range. There was little variation between. However, the English Songs distribution had a positive skew, and their songs were overall less speechy, lying within the 0 -20 range. This suggests that their songs included in the list tended to be more lyrical, and musical, than the Spanish ones.

Keyword Analysis

Although the lyrical analysis of our dataset was a relatively simple task, the collection of the data was a lengthy process. First, we searched up each individual song to find its lyrics. Many of the songs in our dataset were in a foreign language, as Latin-American listeners comprise a large amount of global music consumers. Therefore, we decided to translate the lyrics of those songs to English using various websites in order to better interpret the data. We pasted this data into a new .csv file, and de-concatenated the lyrics so each word was an individual row. Then we recompiled the data from our original dataset so that each keyword was associated with that data. All in all, this resulted in almost 20,000 rows of data being created for this project.

Finally, we sorted the keywords into four categories – “stop”, “filler”, “vulgar”, and “none”. “Stop” words are the words that are usually omitted from search engines. We categorized stop words according to this website’s guidelines, for the purposes of omitting irrelevant or unhelpful words when we wish to take a deeper look at our keywords. “Filler” words are words that aren’t necessarily descriptive in nature, but rather guttural or unintelligible in nature. These words are generally used for the purposes of keeping the pentameter of the lyrics consistent with the rhythm without having to think of additional and meaningful lyrical content. Since these words don’t particularly contribute any kind of meaning, we wanted to distinguish these words from other keywords. “Vulgar” words are words that are considered vulgar or crass in nature, and we denoted these for the purposes of omitting these to avoid any potential offense in our visualization. Finally, the “none” category is any word that cannot be categorized in any of these other categories, and most often is where we can find our most useful keywords.

Next was our visualization process. For this stage in the process, we made use of various packages and libraries, such as wordcloud and plotrix, in order to suit our visualization purposes. The first visualization we made is a simple, straightforward pie

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto
chart to demonstrate how our four categories of lyrical keywords were distributed amongst our data.

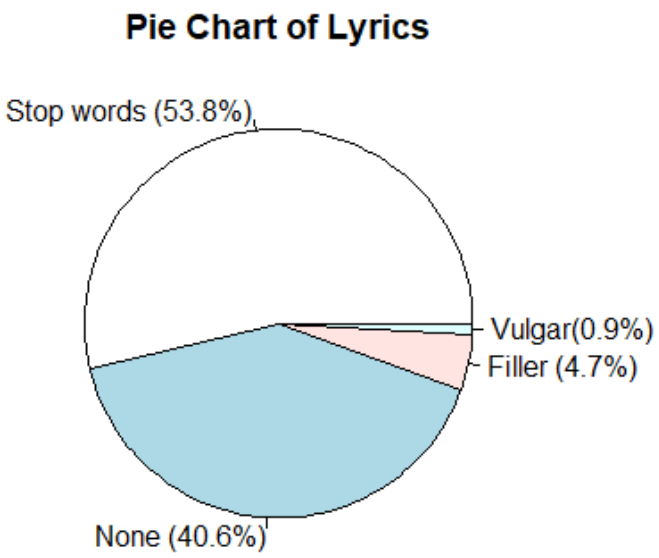

Figure 10

Next, we focused on words in our none category, omitting words we felt were not useful or beneficial to include in our visualization. With the remaining keywords, we created a word cloud for an easy-to-interpret visualization without too much quantified data to distract from the main and most recurring keywords.

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

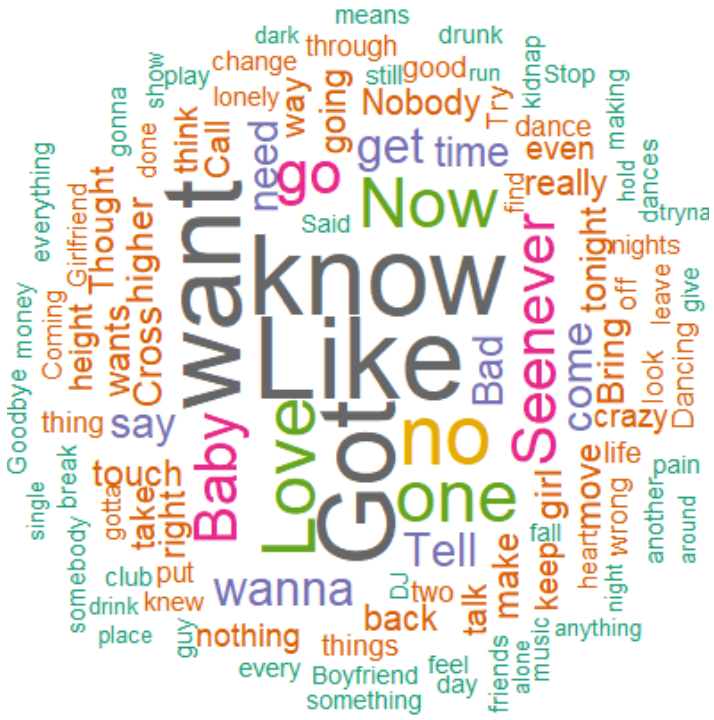

Figure 11

One last and quite interesting aspect we took a look at for our keyword data was inspecting which keywords were correlated with the highest ranking. This was done quite simply using a pivot chart to compute the average ranking for each keyword.

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

Row Labels	Count of Keyword	Average of Ranking
Boyfriend	19	2.947368421
girlfriend	21	4.047619048
ya	27	4.925925926
club	20	5.3
Dancing	24	7.666666667
Goodbye	20	8
Nobody	34	8.352941176
uah	16	9
Eh	31	9.870967742
She's	21	11
Who	20	12.95
was	63	13.96825397
ooh	94	14.03191489
somebody	18	14.72222222
Ain't	56	15.26785714
baby	73	15.36986301
guy	19	15.68421053
Try	26	16.11538462
Yeh	19	16.31578947
bad	44	16.31818182
up	54	16.68518519
feel	19	17
gotta	21	17.0952381
fall	17	18.41176471
way	31	18.70967742
keep	30	18.86666667
very	21	19
here	21	19.0952381
Drunk	18	19.22222222
he	20	19.3
by	20	19.35
'Cause	40	19.375
take	33	19.45454545
play	18	19.55555556

Figure 12

Methodologies

Pros and Cons

Our question is not an overly deep question so we used some of the simpler methodologies. We used Tukey's 5 numbers to get a closer look at the dataset. The pro to this methodology is that it gives you in-depth knowledge of the dataset. A con is it only gives you the numbers you have to look closer into the data if there seems to be

outliers. Another tool we used was a correlation table and the pro of this is the usefulness of the information it provides. A con is that it is a lot of numbers and is sometimes hard to read. We also constructed a lot of scatter plots with points colored. The pro of this technique as it is easy to read and has a good visualization of the data. A con of it is when using the color scale to show the rank it isn't always clear where the data point is in the rank order. The colors blend together in the middle of the ranks.

Alternative methods

We looked into and used K-means clustering, QQ plot of the residuals, and PCA but none of them showed anything useful to answer our big question. For the cluster, we used the new column we made and let R generate the cluster and neither provided anything useful. We also ran out of time to try and dive deeper into each one.

Conclusion

Our answer

At the end of all of our analysis of the dataset, we were disappointed to say that there was nothing that stood out as to why the songs ranked where they did or why songs ranked higher than others. A few things we can say is if your name is Ed Sheeran you have a pretty good chance of making it in the top 50. Also if your song's genre is any style of pop whether that be dance-pop or Canadian pop you also have a greater chance of being in the top 50. For the different characteristics of songs, we found nothing that is correlated enough where we could say with great confidence that it would relate to a higher ranking if you did it.

If a Spanish song wants to have a chance of making the list, its genre would have to be either pop, latin, or reggaeton. It would also have to have danceability lying somewhere within the 60-90 range, have lyrics more on the positive side, be high energy, and have BPM between either 80-90 or 160-180. The overall lack of bell curves in the distributions suggests that these restrictions are a lot more strict and rigid, than for English songs. It is possible that this could just be due to the fact that English songs represent a greater portion of the dataset, therefore making it more likely for a standard normal curve to appear, but further analysis would have to be taken.

Based on our keyword analysis, we can infer that there are some common themes amongst Spotify's most popular songs. Often these themes revolve around love, heartbreak, dancing, feelings, and filler words. Perhaps many top artists may incorporate these themes into their songs as a form of expressing their own internal turmoil, or as a way of creating content that consumers can find powerfully relatable. Perhaps this is a reflection of how many in our society crave the vulnerability that is so uniquely articulated through the lyrics of our pop culture songs.

Additional analyses and data

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

We thought about doing a Compositional analysis but realized this would be very time-consuming and hard to find the information. There are multiple different versions of each song and not a lot of consistency between each site. One question that came up when reviewing our findings was how rank and popularity weren't more correlated than they were. When looking up how Spotify ranks these songs it showed that it was based on what songs were the most listened to around the world. But that also seems like it would be how popularity is determined, essentially rank and popularity would be the same and definitely much more closely related than they were shown to be. This is one of those things where the evidence and data found does not match what logically makes sense. We decided to put our time into better, more useful things.

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

Appendices

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(GGally)
```

```{r song}
song <- read.csv("top 50 CVS.csv")
head(song)
```
```{r new column}
song1<-song %>% mutate(N =
                (case_when(Rank <= 17 ~ "A",
                        Rank <= 34 ~ "B",
                        Rank <= 50 ~ "C")))
##Rank <= 30 ~ "C", Rank <= 40 ~ "D", Rank <= 50 ~ "E",)))

head(song1)
```
```{r outliers}
song1[song$Speechiness.>40,]
song1[song$Length.>300,]
song1[song$Acousticness..>40,]
```
```{r analy}
summary(song)
```
```{r cor}
ggpairs(song1, columns = c(1,5,6,7,8,9,10,11,12,13,14), title="Song Statistics")

```
```{r model}
model <- lm(Rank ~ Beats.Per.Minute + Liveness + Speechiness., data = song)
summary(model)
```
```{r histo}
GenrePlot <-ggplot(song, aes(y = Genre)) +
  geom_bar()
 print(GenrePlot + ggtitle("Number of Genre types"))
```
```{r scatter L vs E}
ggplot(data = song, mapping = aes(Loudness..dB.., Energy, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue")
```

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

```
ggplot(song, aes(x=Loudness..dB..,y=Energy)) +geom_point(color='blue') +
geom_smooth(method="lm",formula=y~x,color='green')
```

```{r scatter top}
ggplot(data = song, mapping = aes(Loudness..dB.., Energy, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue")
```

```{r scatter 3rd}
ggplot(data = song, mapping = aes(Valence., Energy, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue")
```

```{r scatter 2nd}
ggplot(data = song, mapping = aes(Speechiness., Energy, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue")
```

```{r scatter rank}
ggplot(data = song, mapping = aes(Danceability, Beats.Per.Minute, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```

```{r scatter energy}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Energy, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```

```{r scatter Danceability}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Danceability, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```

```{r scatter Loudness..dB..}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Loudness..dB.., color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```

```{r scatter Liveness}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Liveness, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```

```{r scatter Valence.}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Valence., color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

```
```{r scatter Length.}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Length., color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```
```{r scatter Acousticness..}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Acousticness.., color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```
```{r scatter Speechiness.}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Speechiness., color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
ggplot(song, aes(x=Beats.Per.Minute,y=Speechiness.)) +geom_point(color='blue') +
geom_smooth(method="lm",formula=y~x,color='green')
```
```{r scatter Popularity}
ggplot(data = song, mapping = aes(Beats.Per.Minute, Popularity, color=Rank))+
  geom_point()+
  scale_color_gradient2( high = "red", mid = "blue", low = "blue")
```
```{r cluster}
ggplot(data = song1, mapping = aes(x=Loudness..dB.., y=Energy,  color = N, )) + geom_point(size=2)
```
```{r cluster1}
ggplot(data = song1, mapping = aes(x=Speechiness., y=Beats.Per.Minute,  color = N, )) +
geom_point(size=2)
```
```{r cluster2}
ggplot(data = song1, mapping = aes(x=Valence., y=Energy,  color = N, )) + geom_point(size=2)
```
```{r cluster3}
ggplot(data = song1, mapping = aes(x=Rank, y=Liveness,  color = N, )) + geom_point(size=2)

songN<- song1[,5:14]
song1PCA <- prcomp(songN,center=TRUE,scale=TRUE)
song1$pc1 <- song1PCA$x[,1]
song1$pc2 <- song1PCA$x[,5]
ggplot(song1,aes(x=pc1,y=pc2,color=as.factor(N))) +geom_point()
```
```{r kmean}
songk2 <- kmeans(song[,1],3)

song$cluster2 <- songk2$cluster
```
```

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto
ggplot(song,aes(x=Loudness..dB..,y=Energy,color=as.factor(cluster2))) + geom_point()
```
```{r qq plot}
linearH <- lm(Loudness..dB.. ~ Energy, song)
par(mfrow=c(2,2))
plot(linearH)
par(mfrow=c(1,1))
```
```{r qq plot2}
linearHi <- lm(Energy ~ Rank, song)
par(mfrow=c(2,2))
plot(linearHi)
par(mfrow=c(1,1))
```
---
title: "Group Dataset"
author: "Grace Okamoto"
date: "12/4/2021"
output: pdf_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

I have already checked the data for any required cleaning, and I have de-concatenated keywords into separate sheets. Let's load those sheets in with the original data. We'll also need to load in a variety of packages for future visualizations.
```{r load}
ST50 <- read.csv("top50.csv")
keywords <- read.csv("Lyrics Deconcatenated.csv")
WC <- read.csv("Wordcloud.csv")

library(ggplot2)
library(GGally)

library(wordcloud)
library(RColorBrewer)

library(plotrix)
```
Let's summarize some of these tables:
```{r summary}
summary(ST50)
```
Not a terribly clear way of interpreting data, but it gives a relatively decent idea of the average values we're going to be looking at.
```

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

Let's try making a correlation table to see if we can pick out any values that might prove to be usefully correlated to our ranking.

```{r correlation}
ggpairs(ST50, columns = c(5:14), title="Spotify Top 50 Correlation")
cor(ST50[,5:14])
```

Although the table is a bit difficult to read, we can see that there aren't very many strong correlations between many of the values listed, making a regression model quite difficult to obtain. Let's take a different approach to our data.

I've used a wordcloud package to visualize some of the most common keywords that appear in the cleaned data, once things such as stop words, filler words, and vulgarities have been censored out.

```{r wordcloud}
wordcloud(words = WC$ï..keyword, freq = WC$count, min.freq = 15, random.order=FALSE, rot.per=.5, colors=brewer.pal(8, "Dark2"))
```

I've also counted the number of stop words, filler words, vulgarities, and other words and created a pie chart to demonstrate the distribution of these words.

```{r piechart}
slices <- c(10521, 7930, 928, 175)
lbls <- c("Stop words (53.8%)", "None (40.6%)", "Filler (4.7%)", "Vulgar(0.9%)")
pie(slices, labels = lbls, main="Pie Chart of Lyrics")
```

Code for Spanish Analysis

```
library("tidyverse")

#Read in the data
directory <- "C:/Users/Clara Brigitta/Desktop/WSU Fall 2021/Data 115/Final Project"
top50 <- file.path(directory,"top50.csv")
SpotifyDF <- read.csv(top50, header = TRUE)
view(SpotifyDF)

#Create Data Frames for the Spanish and English Songs
SpanishSongDF <- subset.data.frame(SpotifyDF, SpotifyDF$Contains.Spanish == "Y")
view(SpanishSongDF)

EnglishSongDF <- subset.data.frame(SpotifyDF, SpotifyDF$Contains.Spanish == "N")
view(EnglishSongDF)

# Histogram of BPM
# English BPM
hist(EnglishSongDF$Beats.Per.Minute, main = "English Songs BPM", col = "red", xlab = "BPM")
# Spanish Song BPM
hist(SpanishSongDF$Beats.Per.Minute, main = "Spanish Songs BPM", col = "red", xlab = "BPM")
```

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

```
# Histogram of Energy
# English Energy
hist(EnglishSongDF$Energy, main = "English Songs Energy", col = "blue", xlab = "Energy")
# Spanish Energy
hist(SpanishSongDF$Energy, main = "Spanish Songs Energy", col = "blue", xlab = "Energy")

# Histogram of Danceability
# English Danceability
hist(EnglishSongDF$Danceability, main = "English Song Danceability", col = " Dark green", xlab =
"Danceability")
# Spanish Danceability
hist(SpanishSongDF$Danceability, main = "Spanish Song Danceability", col = " Dark green", xlab =
"Danceability")

# Histogram of Loudness
#English Histogram
hist(EnglishSongDF$Loudness..dB.., main = "English Song Loudness", col = "purple", xlab = "Loudness")
# Spanish Histogram
hist(SpanishSongDF$Loudness..dB.., main = "Spanish Song Loudness", col = "purple", xlab =
"Loudness")

#Histogram Liveness
# English Histogram
hist(EnglishSongDF$Liveness, main = "English Song Liveness", col = "orange", xlab = "Liveness")
# Spanish Histogram
hist(SpanishSongDF$Liveness, main = "Spanish Song Liveness", col = "orange", xlab = "Liveness")


# Histogram Valence
# English Histogram
hist(EnglishSongDF$Valence., main = "English Song Valence", col = "yellow", xlab = "Valence")
#Spanish Histogram
hist(SpanishSongDF$Valence., main = "Spanish Song Valence", col = "yellow", xlab = "Valence")


# Histogram Acousticness
# English Hist
hist(EnglishSongDF$Acousticness.., main = "English Song Acousticness", xlab = "Acousticness", col =
"Light Blue")
# Spanish Hist
hist(SpanishSongDF$Acousticness.., main = "Spanish Song Acousticness", xlab = "Acousticness", col =
"Light Blue")


# Histogram Speechiness
# English Hist
```

Gabrielle Isaak
Clare
Ian Miller
Grace Okamoto

```
hist(EnglishSongDF$Speechiness., main = "English Song Speechiness", col = "pink", xlab =
"Speechiness")
# Spanish Hist
hist(SpanishSongDF$Speechiness., main = "Spanish Song Speechiness", col = "pink", xlab =
"Speechiness")


# Histogram Popularity
# English Hist
hist(EnglishSongDF$Popularity, main = "English Song Popularity", col = "green", xlab = "Popularity")
# Spanish Hist
hist(SpanishSongDF$Popularity, main = " Spanish Song Popularity", col = "green", xlab = "Popularity")


# Histogram
# English Hist
hist(EnglishSongDF$Length, main = "English Song Length", col = "Dark Green", xlab = "Length")
hist(SpanishSongDF$Length, main = "Spanish Song Length", col = "Dark Green", xlab = "Length")
```