Exploratory Data Analysis of Spotify Songs from 1920- 2020

Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra
Data of Presentation : 24/04/2023

Introduce

→ What is Spotify ?

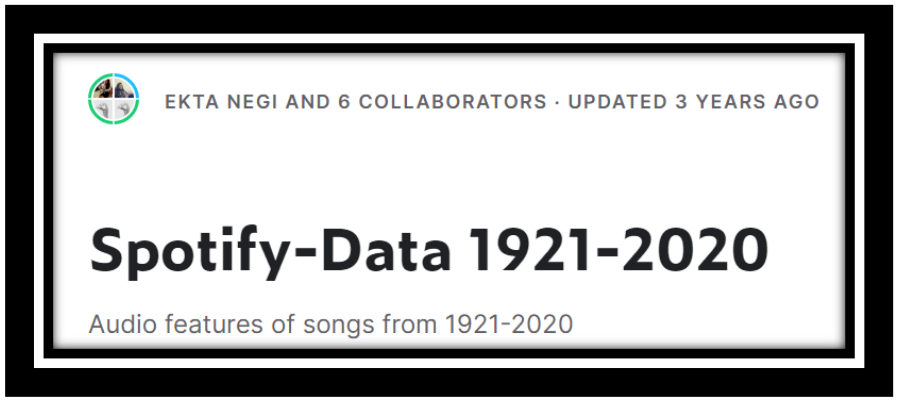
Spotify is a digital music, podcast, and video service that gives you access to millions of songs and other content from creators all over the world.

→ What we did?

In this project, we analyzed the distribution of musical features such as danceability, energy, loudness, liveliness, valence, and duration, as well as their changes across different decades.
We also ranked the popular artists of each decade.

→ Data Source

Data in this report is extracted from Kaggle-Spotify-Data 1921-2020, which contains the top 100 songs in each year from 1921-2020 in Spotify (totally 169k songs) as the description.



Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

About Original Dataset

The file of dataset contains more than 160.000 songs collected from Spotify Web API. The dataset is from Spotify and contains 169k songs from the year 1921 to year 2020. Each year got top 100 songs.

→ bring data into R

```
music <- read.csv("../597DWrangl_23SP/data/Spotify-Data 1921-2020.csv")
glimpse(music) # 169,909 * 19
```

```
## Rows: 169,909
## Columns: 19
## $ acousticness     <dbl> 0.995, 0.994, 0.604, 0.995, 0.990, 0.995, 0.956, 0.98…
## $ artists          <chr> "['Carl Woitschach']", "['Robert Schumann', 'Vladimir…
## $ danceability     <dbl> 0.708, 0.379, 0.749, 0.781, 0.210, 0.424, 0.444, 0.55…
## $ duration_ms      <int> 158648, 282133, 104300, 180760, 687733, 352600, 13662…
## $ energy           <dbl> 0.19500, 0.01350, 0.22000, 0.13000, 0.20400, 0.12000, …
## $ explicit         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ id               <chr> "6KbQ3uYMLKb5jDxLF7wYDD", "6KuQTIu1KoTTkLXKrwlLPV", "…
## $ instrumentalness <dbl> 5.63e-01, 9.01e-01, 0.00e+00, 8.87e-01, 9.08e-01, 9.1…
## $ key              <int> 10, 8, 5, 1, 11, 6, 11, 1, 9, 9, 10, 10, 7, 5, 5, 7, …
## $ liveness         <dbl> 0.1510, 0.0763, 0.1190, 0.1110, 0.0980, 0.0915, 0.074…
## $ loudness         <dbl> -12.428, -28.454, -19.924, -14.734, -16.829, -19.242, …
## $ mode             <int> 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, …
## $ name             <chr> "Singende Bataillone 1. Teil", "Fantasiestücke, Op. …
## $ popularity       <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, …
## $ release_date     <chr> "1928", "1928", "1928", "1928-09-25", "1928", "1928", …
## $ speechiness      <dbl> 0.0506, 0.0462, 0.9290, 0.0926, 0.0424, 0.0593, 0.040…
## $ tempo            <dbl> 118.469, 83.972, 107.177, 108.003, 62.149, 63.521, 80…
## $ valence          <dbl> 0.7790, 0.0767, 0.8800, 0.7200, 0.0693, 0.2660, 0.305…
## $ year             <int> 1928, 1928, 1928, 1928, 1928, 1928, 1928, 1928, 1928, …
```

duration_ms **Need to be modified…**

year **Need to be modified…**

Pre-process of Data

→ bring data into R

```
music <- read.csv("../597DWrangl_23SP/data/Spotify-Data 1921-2020.csv")
glimpse(music) # 169,909 * 19
```

```
## Rows: 169,909
## Columns: 19
## $ acousticness     <dbl> 0.995, 0.994, 0.604, 0.995, 0.990, 0.995, 0.956, 0.98…
## $ artists          <chr> "['Carl Woitschach']", "['Robert Schumann', 'Vladimir…
## $ danceability     <dbl> 0.708, 0.379, 0.749, 0.781, 0.210, 0.424, 0.444, 0.55…
## $ duration_ms      <int> 158648, 282133, 104300, 180760, 687733, 352600, 13662…
## $ energy           <dbl> 0.19500, 0.01350, 0.22000, 0.13000, 0.20400, 0.12000, …
## $ explicit         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ id               <chr> "6KbQ3uYMLKb5jDxLF7wYDD", "6KuQTIu1KoTTkLXKrwlLPV", "…
## $ instrumentalness <dbl> 5.63e-01, 9.01e-01, 0.00e+00, 8.87e-01, 9.08e-01, 9.1…
## $ key              <int> 10, 8, 5, 1, 11, 6, 11, 1, 9, 9, 10, 10, 7, 5, 5, 7, …
## $ liveness         <dbl> 0.1510, 0.0763, 0.1190, 0.1110, 0.0980, 0.0915, 0.074…
## $ loudness         <dbl> -12.428, -28.454, -19.924, -14.734, -16.829, -19.242, …
## $ mode             <int> 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, …
## $ name             <chr> "Singende Bataillone 1. Teil", "Fantasiestücke, Op. …
## $ popularity       <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, …
## $ release_date     <chr> "1928", "1928", "1928", "1928-09-25", "1928", "1928", …
## $ speechiness      <dbl> 0.0506, 0.0462, 0.9290, 0.0926, 0.0424, 0.0593, 0.040…
## $ tempo            <dbl> 118.469, 83.972, 107.177, 108.003, 62.149, 63.521, 80…
## $ valence          <dbl> 0.7790, 0.0767, 0.8800, 0.7200, 0.0693, 0.2660, 0.305…
## $ year             <int> 1928, 1928, 1928, 1928, 1928, 1928, 1928, 1928, 1928, …
```

→ Convert the duration from milliseconds to seconds.

duration_ms → duration_s

Code

```
duration = duration_ms/1000
```

Output

duration
<dbl>
158.648
282.133
104.300
180.760
687.733
352.600

→ Assign each song to its corresponding decades

year → decades

Code

```
music <- music %>%
  mutate(decade = paste0(substr(year, 1, 3), "0s"),
```

Output

year	decade
<int>	<chr>
1928	1920s
1928	1920s

Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Clean Data and Tidy Data

→ Select variables what we need

```
# rearrange the column order
music <- select(music, "name", "artists", "year", "decade","tempo",
                "energy","danceability", "loudness", "liveness",
                "valence","duration", "acousticness", "speechiness",
                "instrumentalness", "key", "popularity")

head(music)
```

Drop those we do not need

```
## Rows: 169,909
## Columns: 19
## $ acousticness     <dbl> 0.995, 0.994, 0.604, 0.995, 0.990, 0.995, 0.956, 0.98…
## $ artists          <chr> "['Carl Woitschach']", "['Robert Schumann', 'Vladimir…
## $ danceability     <dbl> 0.708, 0.379, 0.749, 0.781, 0.210, 0.424, 0.444, 0.55…
## $ duration_ms      <int> 158648, 282133, 104300, 180760, 687733, 352600, 13662…
## $ energy           <dbl> 0.19500, 0.01350, 0.22000, 0.13000, 0.20400, 0.12000, …
## $ explicit         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ id               <chr> "6KbQ3uYMLKb5jDxLF7wYDD", "6KuQTIu1KoTTkLXKrwlLPV", "…
## $ instrumentalness <dbl> 5.63e-01, 9.01e-01, 0.00e+00, 8.87e-01, 9.08e-01, 9.1…
## $ key              <int> 10, 8, 5, 1, 11, 6, 11, 1, 9, 9, 10, 10, 7, 5, 5, 7, …
## $ liveness         <dbl> 0.1510, 0.0763, 0.1190, 0.1110, 0.0980, 0.0915, 0.074…
## $ loudness         <dbl> -12.428, -28.454, -19.924, -14.734, -16.829, -19.242, …
## $ mode             <int> 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, …
## $ name             <chr> "Singende Bataillone 1. Teil", "Fantasiestücke, Op. …
## $ popularity       <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, …
## $ release_date     <chr> "1928", "1928", "1928", "1928-09-25", "1928", "1928", …
## $ speechiness      <dbl> 0.0506, 0.0462, 0.9290, 0.0926, 0.0424, 0.0593, 0.040…
## $ tempo            <dbl> 118.469, 83.972, 107.177, 108.003, 62.149, 63.521, 80…
## $ valence          <dbl> 0.7790, 0.0767, 0.8800, 0.7200, 0.0693, 0.2660, 0.305…
## $ year             <int> 1928, 1928, 1928, 1928, 1928, 1928, 1928, 1928, 1928, …
```

Now we have…

name <chr>	artists <chr>	y… <int>
1 Singende Bataillone 1. Teil	['Carl Woitschach']	1928
2 Fantasiestücke, Op. 111: Più tosto lento	['Robert Schumann', 'Vladimir Horowitz']	1928
3 Chapter 1.18 - Zamek kaniowski	['Seweryn Goszczyński']	1928
4 Bebamos Juntos - Instrumental (Remasterizado)	['Francisco Canaro']	1928
5 Polonaise-Fantaisie in A-Flat Major, Op. 61	['Frédéric Chopin', 'Vladimir Horowitz']	1928
6 Scherzo a capriccio: Presto	['Felix Mendelssohn', 'Vladimir Horowitz']	1928

6 rows | 1-4 of 17 columns

artists <chr>	year <int>	decade <chr>	tempo <dbl>	energy <dbl>	danceability <dbl>
['Carl Woitschach']	1928	1920s	118.469	0.1950	0.708
['Robert Schumann', 'Vladimir Horowitz']	1928	1920s	83.972	0.0135	0.379
['Seweryn Goszczyński']	1928	1920s	107.177	0.2200	0.749
['Francisco Canaro']	1928	1920s	108.003	0.1300	0.781
['Frédéric Chopin', 'Vladimir Horowitz']	1928	1920s	62.149	0.2040	0.210
['Felix Mendelssohn', 'Vladimir Horowitz']	1928	1920s	63.521	0.1200	0.424

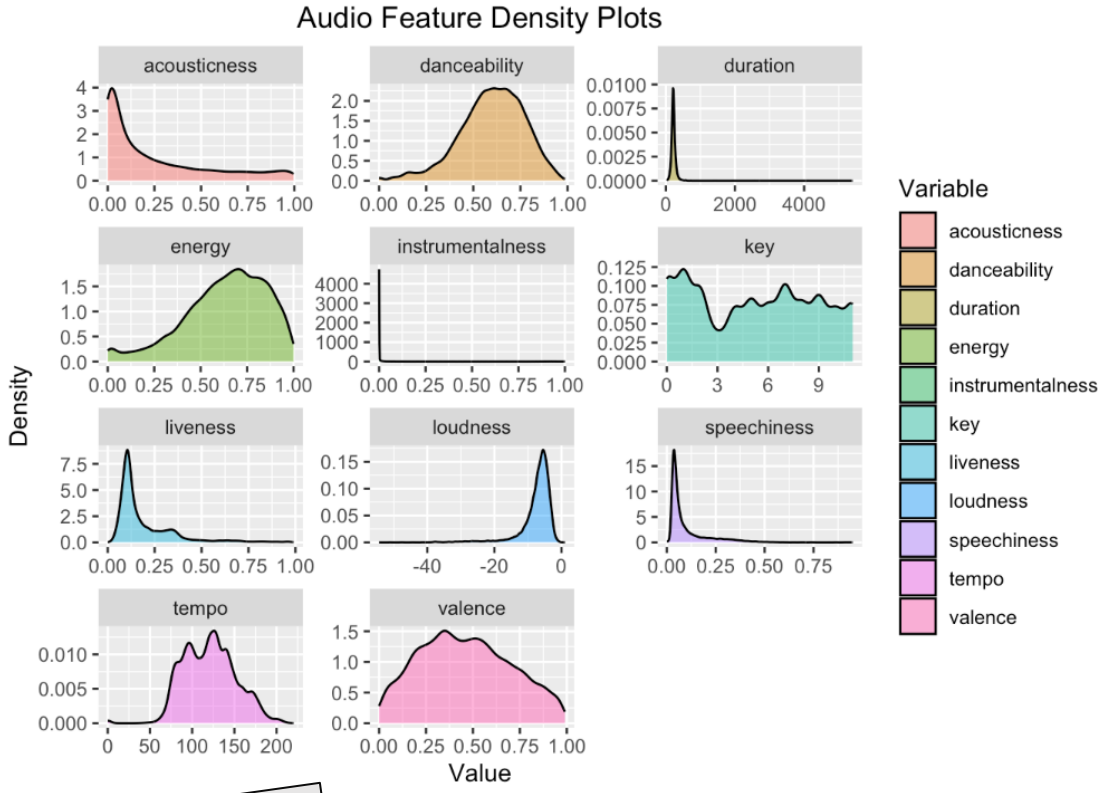
loudness <dbl>	liveness <dbl>	valence <dbl>	duration <dbl>	acousticness <dbl>	speechiness <dbl>	instrumentalness <dbl>	k… <int>	popularity <int>
-12.428	0.1510	0.7790	158.648	0.995	0.0506	0.563	10	0
-28.454	0.0763	0.0767	282.133	0.994	0.0462	0.901	8	0
-19.924	0.1190	0.8800	104.300	0.604	0.9290	0.000	5	0
-14.734	0.1110	0.7200	180.760	0.995	0.0926	0.887	1	0
-16.829	0.0980	0.0693	687.733	0.990	0.0424	0.908	11	1
-19.242	0.0915	0.2660	352.600	0.995	0.0593	0.911	6	0

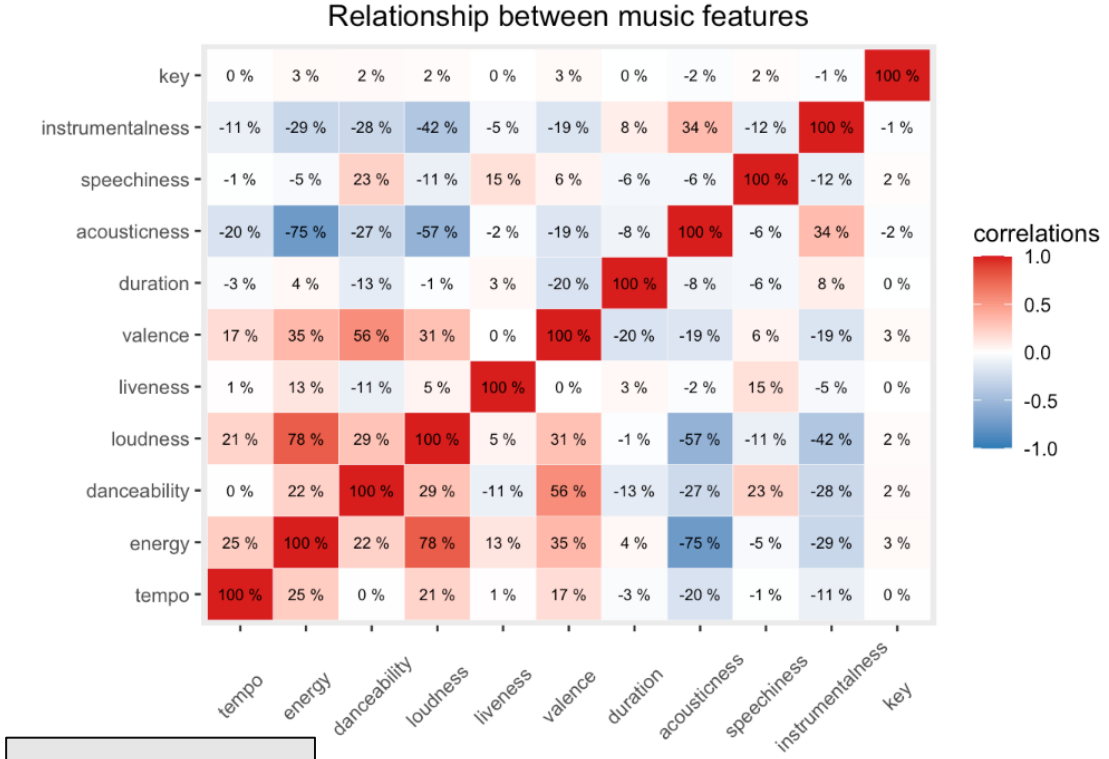
6 rows | 9-17 of 17 columns

Summary of the data…

Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Comparison among variables and Relationship between each two variables





Code Review

→ Density plots with facets

```
# audio features in 2010s
audio_2010s <- music %>%
  filter(decade == "2010s") %>%
  .[,c(5:15)]

# Convert data frame to long format
audio_long <- tidyr::gather(audio_2010s,
                            key = "variable", value = "value")

# Plot density plots with facets
ggplot(audio_long, aes(x = value, fill = variable)) +
  geom_density(alpha = 0.5) +
  theme(plot.title = element_text(hjust = .5)) +
  labs(x = "Value", y = "Density",
       title = "Audio Feature Density Plots") +
  scale_fill_discrete(name = "Variable") +
  facet_wrap(~variable, nrow = 4, ncol = 3, scales = "free")
```

Code Review

→ Step 1 : Normalizing

```
# normalize each variables
audio <- music[,c(5:15)]

normalization <-function(x){
  return( (x - min(x,na.rm = T))/
            ( max(x, na.rm = T) - min(x, na.rm = T)) )
}

for (i in 1:length(audio)){
  audio[,i] = normalization(audio[,i])
}
```

→ Step 2 : tile plot

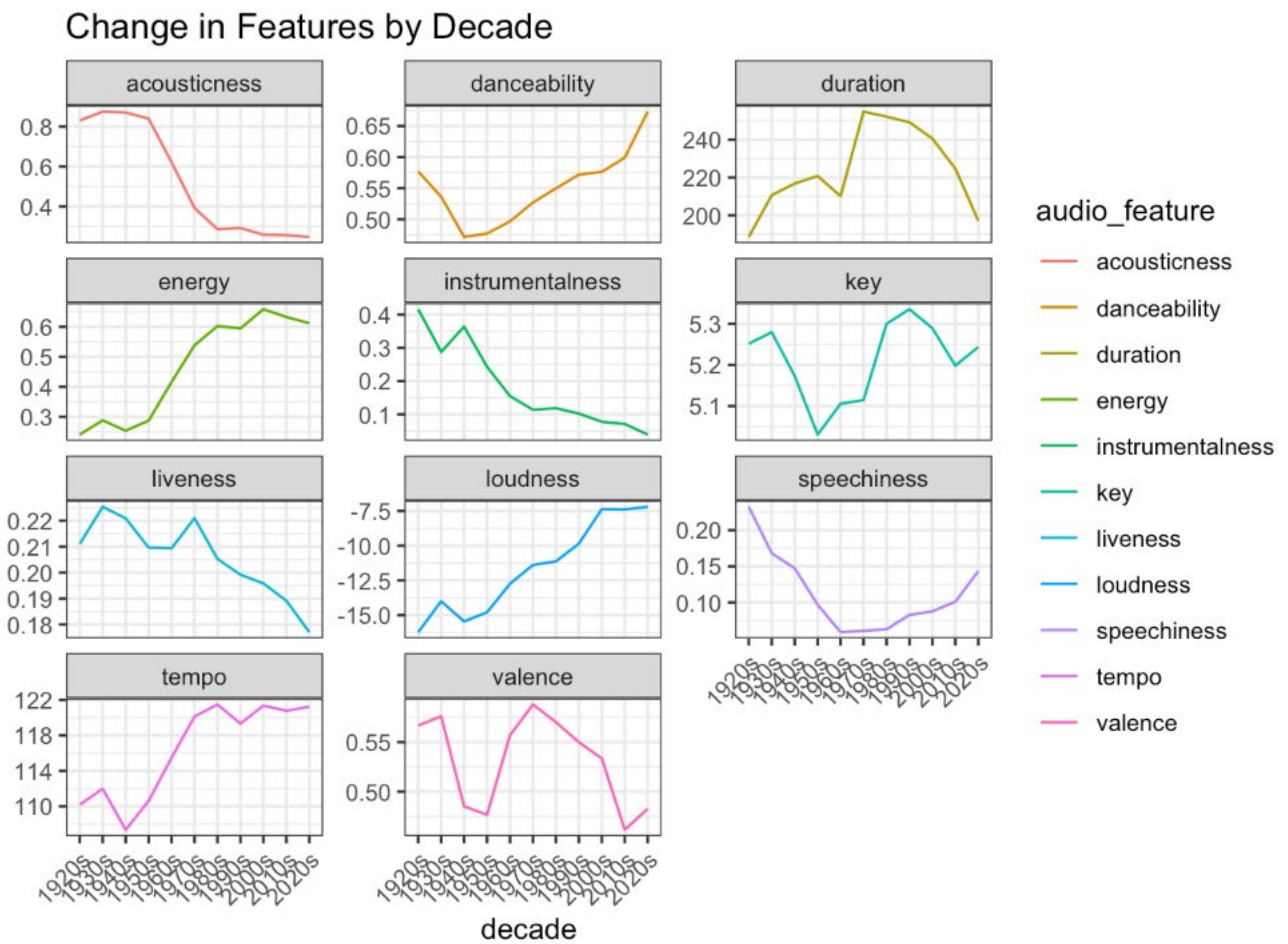
```
audio %>%
  cor() %>%
  melt() %>%
  ggplot(aes(X1, X2, fill=value)) +
  geom_tile(color = 'white') +
  scale_fill_gradient2(low = "#2C7BB6", mid = "white",
                       high = "#D7191C", midpoint = 0,
                       name = "correlations", limits = c(-1, 1),
                       na.value = "gray90", guide = "colorbar",
                       oob = scales::squish) +
  geom_text(aes(label = paste(round(value, 2) * 100, '%')),
            size = 2.5, color = 'black') +
  labs(x = '', y = '',
       title = 'Relationship between music features') +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5),
        plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```


Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Changes from 1921 to 2020

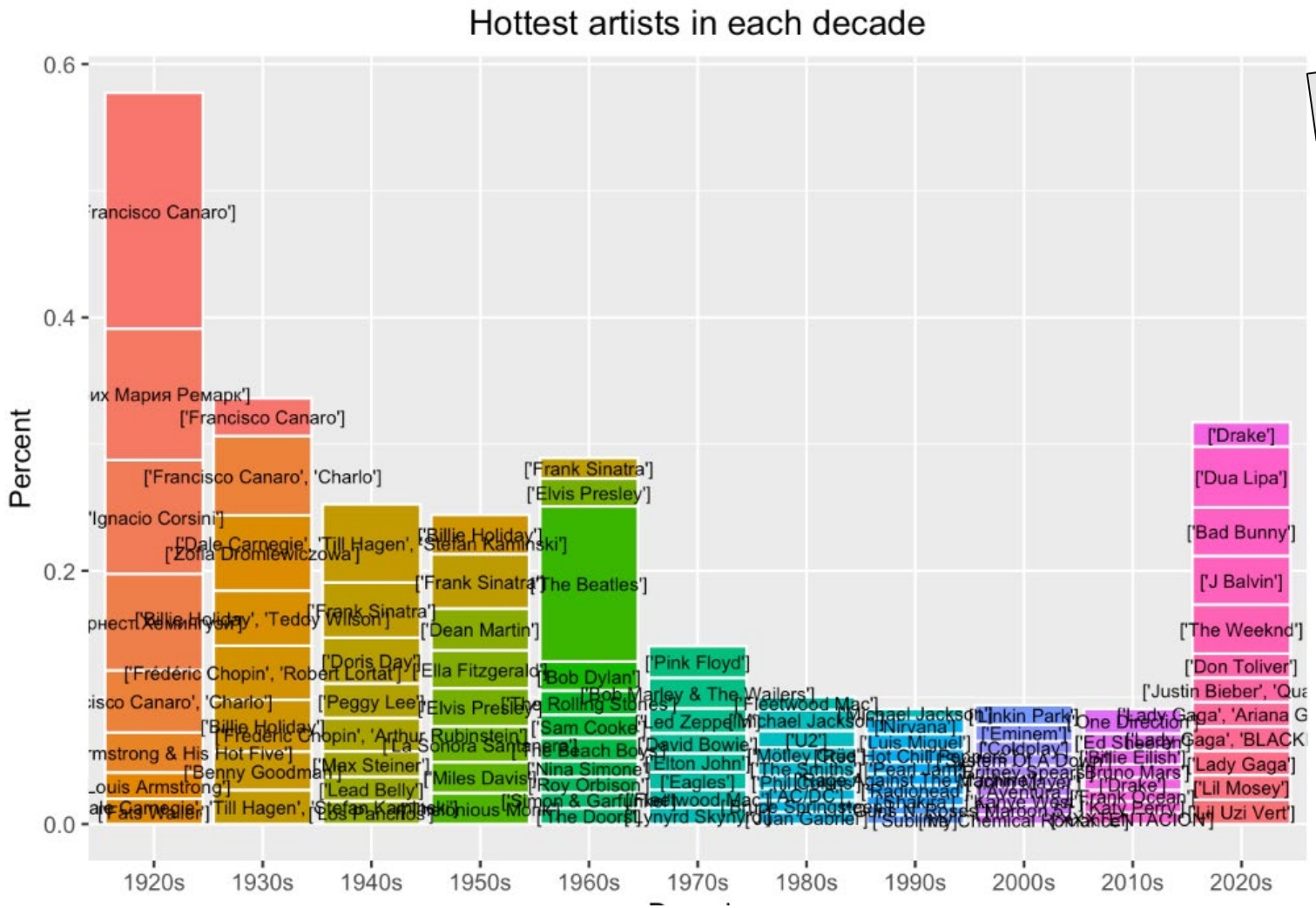
Code Review → **Line plots**

```
audio_decade <- music[,c(4,5:15)]

audio_decade %>%
  group_by(decade) %>%
  summarize(across(starts_with("tempo"), mean),
            across(starts_with("energy"), mean),
            across(starts_with("danceability"), mean),
            across(starts_with("loudness"), mean),
            across(starts_with("liveness"), mean),
            across(starts_with("valence"), mean),
            across(starts_with("duration"), mean),
            across(starts_with("acousticness"), mean),
            across(starts_with("speechiness"), mean),
            across(starts_with("instrumentalness"), mean),
            across(starts_with("key"), mean)) %>%
  pivot_longer(cols = -decade, names_to = "audio_feature",
               values_to = "value") %>%
  ggplot(aes(x = decade, y = value, group = audio_feature,
             color = audio_feature)) +
  geom_line() +
  facet_wrap(~ audio_feature, nrow = 4, ncol = 3,
             scales = "free_y") +
  labs(title = "Change in Features by Decade") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Rank of popular artists of each decade



Code Review → **bar graph**

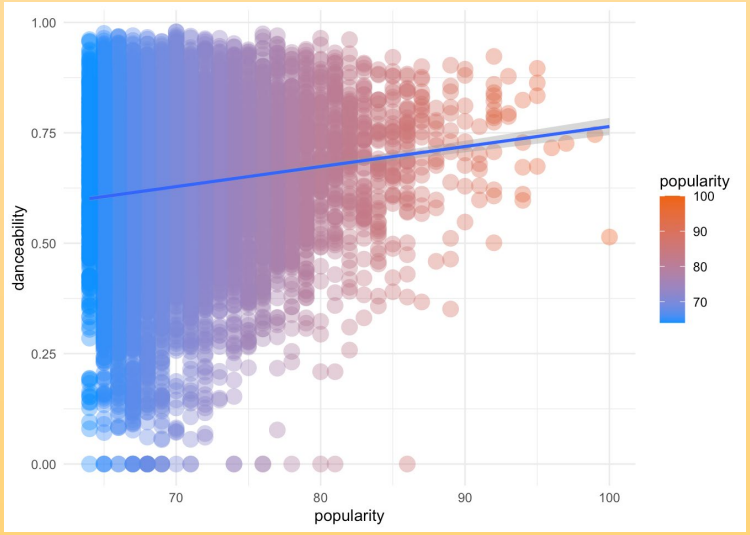
```
# select the top 100 music each year based on the popularity variable

music_top100 <- music %>%
  group_by(year) %>%
  arrange(desc(popularity)) %>%
  top_n(100, popularity) %>%
  ungroup()

# select 8 top artists for each decades
top_artist <- music_top100 %>%
  group_by(decade) %>%
  count(artists) %>%
  mutate(prop = n / sum(n)) %>%
  group_by(decade) %>%
  top_n(8, prop) %>%
  ungroup()

# draw a stacked bar chart
top_artist %>%
  arrange(decade, desc(prop)) %>%
  mutate(artists = factor(artists, unique(artists))) %>%
  ggplot(aes(decade, prop, fill = artists)) +
  geom_bar(stat = 'identity', color = 'white', show.legend = F) +
  geom_text(aes(label = artists), size = 2.5, color = 'black',
            position = position_stack(vjust = .5)) +
  theme(plot.title = element_text(hjust = .5)) +
  labs(title = 'Hottest artists in each decade', y = 'Percent', x = 'Decade')
```


Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

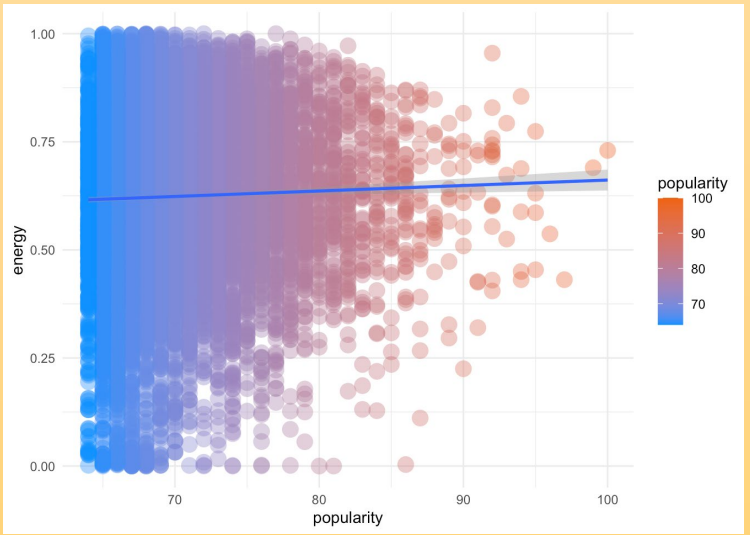
Attributes vs. Popularity

Positive effect

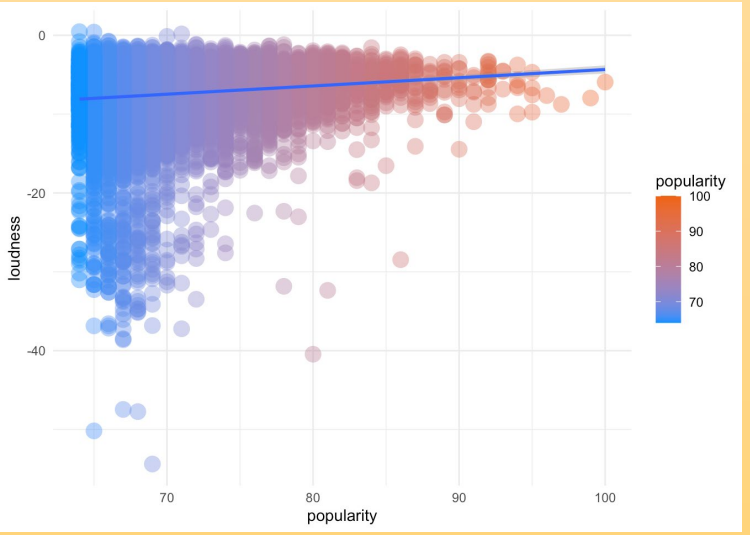
→ Danceability



→ Energy

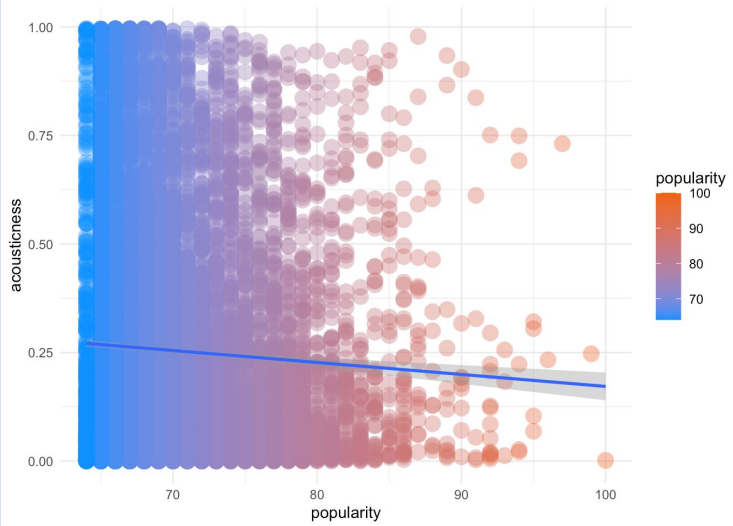


→ Loudness

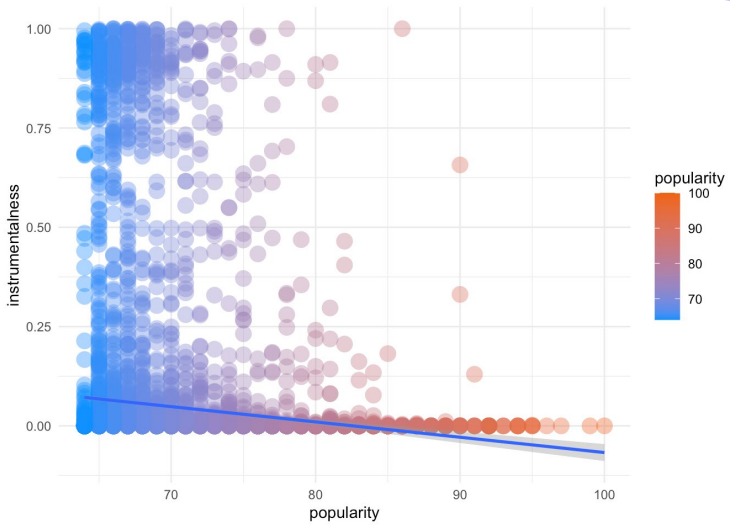


Negative effect

→ Acousticness



→ Instrumentalness



Code Review

```
music_popularity_order <-
  music[order(music$popularity, decreasing = TRUE),]

music_top1000 <- music_popularity_order[1:10000,]

ggplot(music_top1000, aes(popularity, danceability, color = popularity)) +
  geom_point(shape = 16, size = 5, show.legend = FALSE, alpha=.4) +
  theme_minimal() +
  scale_color_gradient(low = "#0091ff", high = "#f0650e")+geom_smooth(method='lm')
```

```
ggplot(music_top1000, aes(popularity,
                          acousticness, color = popularity))
```

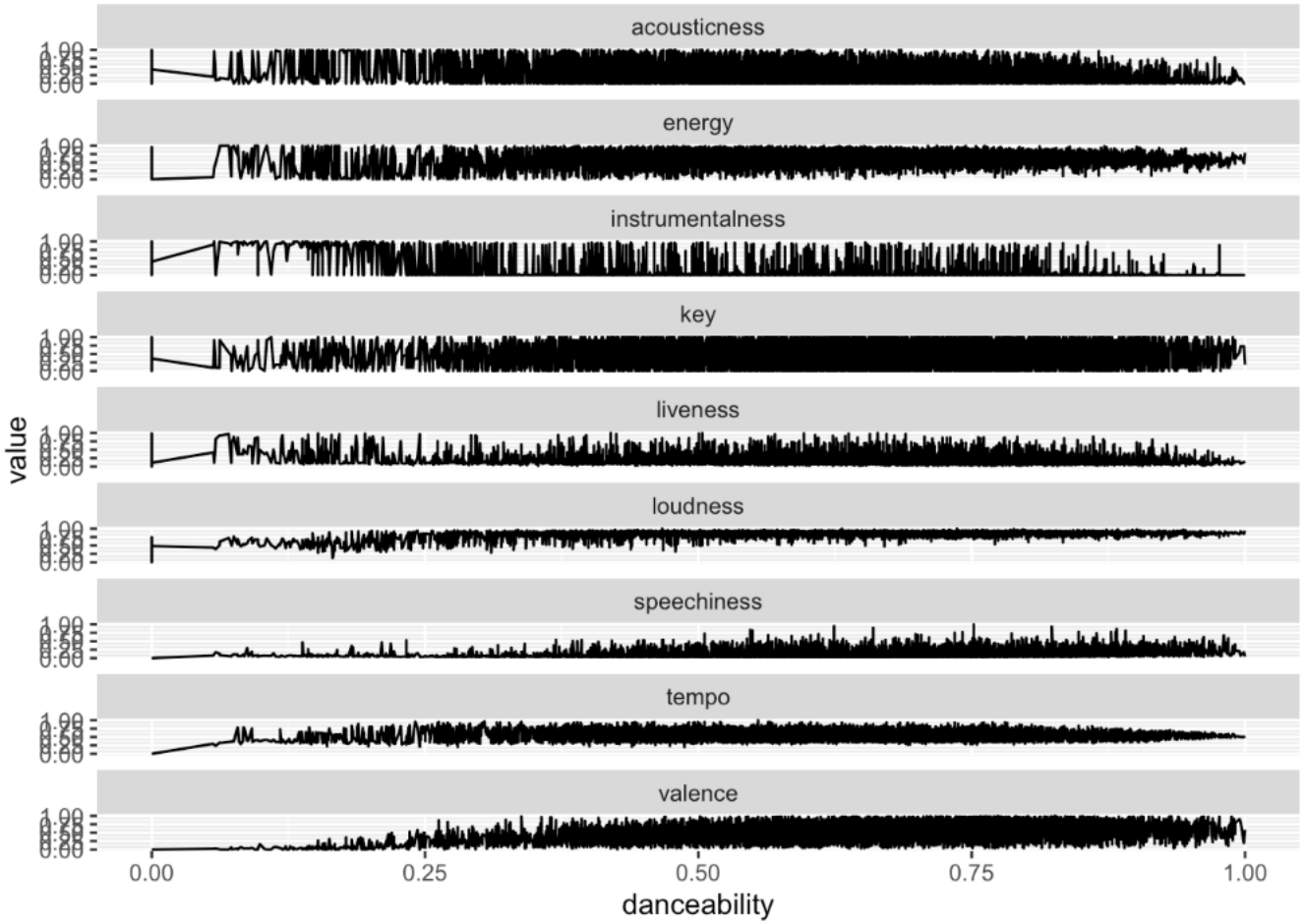
```
ggplot(music_top1000, aes(popularity, energy, color = popularity))
```

```
ggplot(music_top1000, aes(popularity, instrumentalness, color = popularity))
```

```
ggplot(music_top1000, aes(popularity, loudness, color = popularity))
```


Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

What Features are important for a song to be danceable?



Code Review

```r
# function to scale the values between 0 and 1
regularization <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

danceable_music <- music_top1000 %>%
  arrange(desc(popularity)) %>%
  select(-c(name, artists, year, decade, popularity, duration)) %>%
  mutate(across(everything(), regularization)) %>%
  pivot_longer(cols = -c(danceability), names_to = "variable", values_to = "value")

ggplot(danceable_music, aes(danceability, value)) +
  geom_line() +
  facet_wrap(~variable, scales = "free_y", ncol = 1)
```


Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Most common words in title with a WordCloud

in 21st Century:

Girl : 24

Bad : 14

Baby : 12

Don : 33

Code Review

```
#  avoid the brackets, they have the singer names and words like feat,etc.
music_top100_21stCentury <-
  music_top100 %>%
  filter(decade %in% c("2000s", "2010s"))

music_top100_21stCentury$name <-
  str_extract_all(music_top100_21stCentury$name, "[a-zA-Z ]+") %>%
  sapply(paste0, collapse = " ")

seg <- unlist(str_split(music_top100_21stCentury$name, "\\s+"))

# Remove common words (stop words)
stop_words <- c(stopwords("english"), "feat", "remix", "mix")
seg <- seg[!tolower(seg) %in% stop_words]

wordcloud2(table(seg), size = 1.5)
```


Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Most common words in title with a WordCloud

in late 20th Century:

Girl : 38

Baby : 12

Don : 77

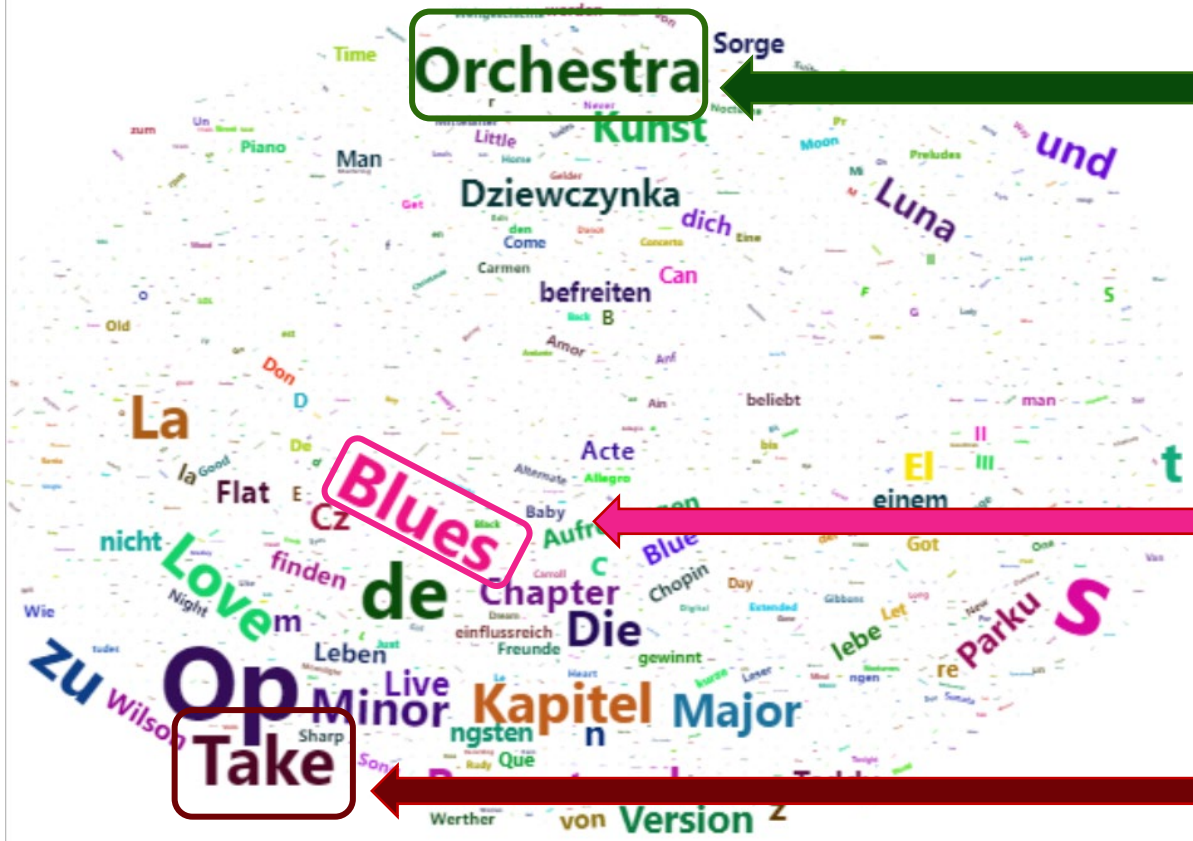
Code Review

```
music_top100_late20thCentury <-
  music_top100 %>%
  filter(decade %in% c("1960s", "1970s", "1980s", "1990s"))

music_top100_late20thCentury$name <-
  str_extract_all(music_top100_late20thCentury$name,
                  "[a-zA-Z ]+") %>%
  sapply(paste0, collapse = " ")

seg <- unlist(str_split(music_top100_late20thCentury$name, "\\s+"))

# Remove common words (stop words)
stop_words <- c(stopwords("english"), "feat", "remix", "mix")
seg <- seg[!tolower(seg) %in% stop_words]

wordcloud2(table(seg), size = 1.5)
```


Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Most common words in title with a WordCloud

in early 20th Century:



Code Review

```
music_top100_early20thCentury <-
  music_top100 %>%
  filter(decade %in% c("1920s", "1930s", "1940s", "1950s"))

music_top100_early20thCentury$name <-
  str_extract_all(music_top100_early20thCentury$name,
                  "[a-zA-Z ]+") %>%
  sapply(paste0, collapse = " ")

seg <- unlist(str_split(music_top100_early20thCentury$name, "\\s+"))

# Remove common words (stop words)
stop_words <- c(stopwords("english"), "feat", "remix", "mix")
seg <- seg[!tolower(seg) %in% stop_words]

wordcloud2(table(seg), size = 1.5)
```


Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Summary

Common Characteristics of Popular Songs

- lower acousticness
- higher energy
- less instrumentation
- higher loudness
- higher danceability

Significance of Project

- These findings can help understand the changing trends in music over the years and the key factors contributing to a song's popularity.
- However, it is essential to note that this analysis is based on a limited dataset and does not consider the complete landscape of the music industry.

Group 16 : Fangru Linghu , Yiming Tan , Siyu Chen , Arun Mishra

Thank you for Listening !