MEDA - Case: 1.2MM Spotify songs

Assistant Professor Iván Mendivelso

Purpose

Exploratory Data Analysis (EDA) is an important stage in generating knowledge from data. This activity is thought of as applying some concepts learned in the first part of the course. You should be able to make summaries, plots, and interpret them correctly. Also, proposing original analysis is a plus.

About Dataset

Context

```
df_spotify <- data.table::fread('~/Downloads/tracks_features.csv',
                                stringsAsFactors = TRUE)
```

The Spotify API offers a rich source of data on its songs and their physical characteristics[1]. The dataset includes a wide range of variables, such as `id`, `name`, `album`, `album_id`, `artists`, `artist_ids`, `track_number`, `disc_number`, `explicit`, `danceability`, `energy`, `key`, `loudness`, `mode`, `speechiness`, `acousticness`, `instrumentalness`, `liveness`, `valence`, `tempo`, `duration_ms`, `time_signature`, `year`, and `release_date`. Each of these variables plays a unique role in shaping a song's overall composition and emotional impact.

The `id` variable provides a unique identifier for each song, while the `name` variable indicates the name of the song. The `album` variable provides the name of the album the song is from, and the `album_id` variable provides a unique identifier for that album. The `artists` variable lists the artist or artists that performed the song, while the `artist_ids` variable provides unique identifiers for those artists.

The `track_number` variable lists the order in which the song appears on its album, while the `disc_number` variable lists the disc number on which the song appears (in the case of multi-disc albums). The `explicit` variable indicates whether the song contains explicit content, which could be used to classify different types of audience.

The variables `danceability`, `energy`, `key`, `loudness`, `mode`, `speechiness`, `acousticness`, `instrumentalness`, `liveness`, `valence`, and `tempo` are all measures of the song's physical characteristics, including its rhythm, melody, and tonality. These variables can be used to gain insight into the overall mood and emotional impact of the song.

The `duration_ms` variable indicates the length of the song in milliseconds, while the `time_signature` variable provides information on the song's time signature (e.g., 4/4, 3/4). The `year` variable lists the year in which the song was released, while the `release_date` variable provides more detailed information on the song's release date.

While the dataset is rich in variables, it is important to note that it does not include each song's popularity. This variable is of particular interest to those looking to analyze a song's overall success or impact. As a result, careful cleaning and detailed exploratory data analysis are necessary to better understand the underlying relationships and structures within the dataset.

[1]The dataset is taken from Kaggle at https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs)

The data

The next arrange shows a glimpse of the dataset:

```
library(tidyverse)
library(lubridate)
library(patchwork)
library(ggridges)
glimpse(df_spotify)
```

```
## Rows: 1,204,025
## Columns: 24
## $ id               <fct> 7lmeHLHBe4nmXzuXc0HDjk, 1wsRitfRRtWyEapl0q22o8, 1hR0f~
## $ name             <fct> "Testify", "Guerrilla Radio", "Calm Like a Bomb", "Mi~
## $ album            <fct> "The Battle Of Los Angeles", "The Battle Of Los Angel~
## $ album_id         <fct> 2eia0myWFgoHuttJytCxgX, 2eia0myWFgoHuttJytCxgX, 2eia0~
## $ artists          <fct> "['Rage Against The Machine']", "['Rage Against The M~
## $ artist_ids       <fct> "['2d0hyoQ5ynDBnkvAbJKORj']", "['2d0hyoQ5ynDBnkvAbJKO~
## $ track_number     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5,~
## $ disc_number      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ explicit         <lgl> FALSE, TRUE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE,~
## $ danceability     <dbl> 0.470, 0.599, 0.315, 0.440, 0.426, 0.298, 0.417, 0.27~
## $ energy           <dbl> 0.978, 0.957, 0.970, 0.967, 0.929, 0.848, 0.976, 0.87~
## $ key              <int> 7, 11, 7, 11, 2, 2, 9, 11, 7, 9, 7, 6, 4, 7, 1, 7, 4,~
## $ loudness         <dbl> -5.399, -5.764, -5.424, -5.830, -6.729, -5.947, -6.03~
## $ mode             <int> 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,~
## $ speechiness      <dbl> 0.0727, 0.1880, 0.4830, 0.2370, 0.0701, 0.0727, 0.175~
## $ acousticness     <dbl> 0.026100, 0.012900, 0.023400, 0.163000, 0.001620, 0.0~
## $ instrumentalness <dbl> 1.09e-05, 7.06e-05, 2.03e-06, 3.64e-06, 1.05e-01, 1.5~
## $ liveness         <dbl> 0.3560, 0.1550, 0.1220, 0.1210, 0.0789, 0.2010, 0.107~
## $ valence          <dbl> 0.503, 0.489, 0.370, 0.574, 0.539, 0.194, 0.483, 0.61~
## $ tempo            <dbl> 117.906, 103.680, 149.749, 96.752, 127.059, 148.282, ~
## $ duration_ms      <int> 210133, 206200, 298893, 213640, 205600, 280960, 20204~
## $ time_signature   <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,~
## $ year             <int> 1999, 1999, 1999, 1999, 1999, 1999, 1999, 1999, 1999,~
## $ release_date     <fct> 1999-11-02, 1999-11-02, 1999-11-02, 1999-11-02, 1999-~
```

Challenge

Build a report and tell a story based on the given 1.2 million songs, utilizing the skills you have acquired through this course or your professional experience. Set hypothesis: Ask yourself questions that the available data could answer. Utilize summary statistics, plots, dashboards, web applications, or any reporting tool you consider appropriate. Encourage creativity in your approach. If you are unsure about how to define a target, label or response variable in a supervised learning algorithm, consider using `explicit` as the dependent variable for making predictions.

```
########################################
# Questions

# Which variables are correlated?
# Which ones influence the variable "explicit"?
# How is the distribution of the songs per album
# Does "explicit" relate to which variables? Does
# it have sense?
# How is "explicit" related to qualitative variables like
```

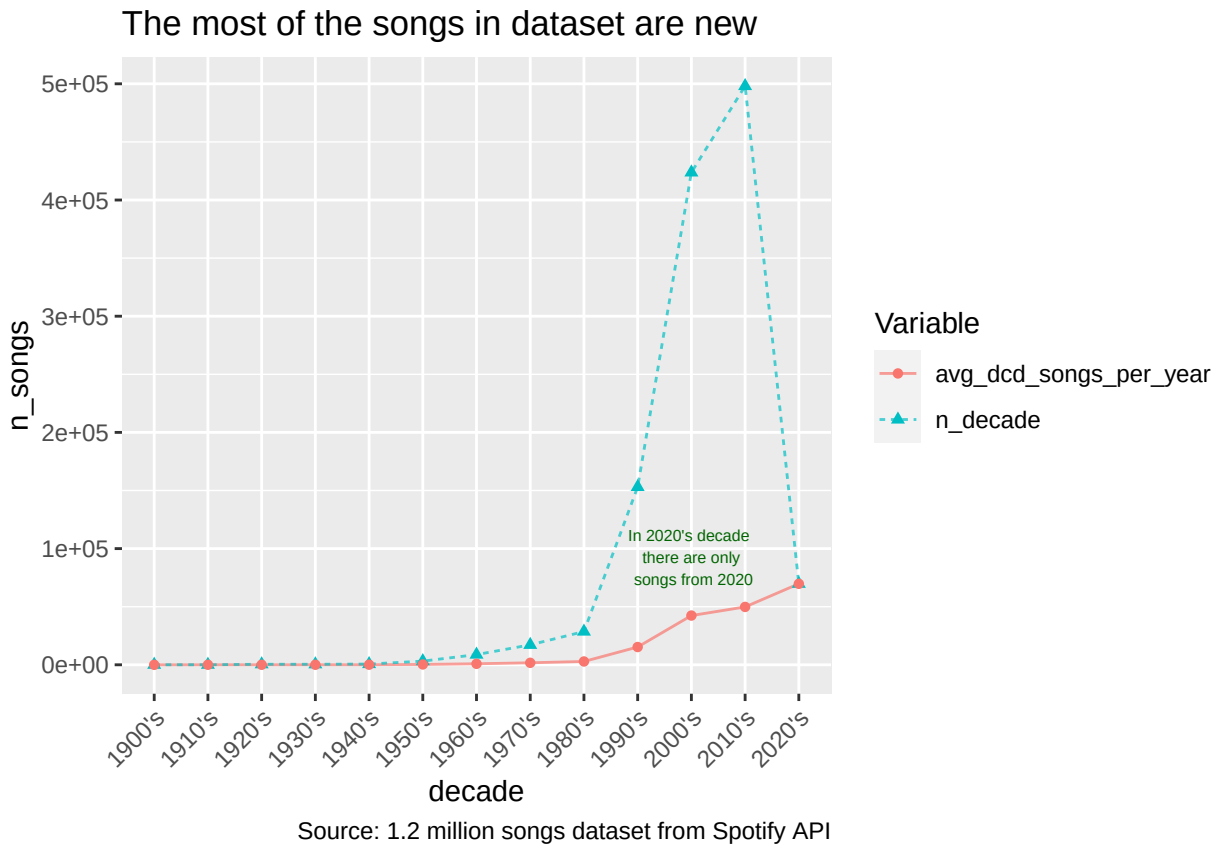
```r
# "key", "mode" and "time_signature"?

#######################################
```

```r
## Canciones por año
df_spotify_2 <- df_spotify %>%
  mutate(decade=case_when(
    year >= 1900 & year < 1910 ~ "1900's",
    year >= 1910 & year < 1920 ~ "1910's",
    year >= 1920 & year < 1930 ~ "1920's",
    year >= 1930 & year < 1940 ~ "1930's",
    year >= 1940 & year < 1950 ~ "1940's",
    year >= 1950 & year < 1960 ~ "1950's",
    year >= 1960 & year < 1970 ~ "1960's",
    year >= 1970 & year < 1980 ~ "1970's",
    year >= 1980 & year < 1990 ~ "1980's",
    year >= 1990 & year < 2000 ~ "1990's",
    year >= 2000 & year < 2010 ~ "2000's",
    year >= 2010 & year < 2020 ~ "2010's",
    year >= 2020  ~ "2020's",
    TRUE ~ NA_character_
  ),
  release = ymd(release_date))

tabla <- df_spotify_2 %>%
  group_by(decade)%>%
  summarise(n=n(),
            year_min=min(year),
            year_max=max(year))%>%
  mutate(decade=decade,
            n_decade=n,
            avg_dcd_songs_per_year=round(n/(year_max-year_min+1),1))%>%
  filter(!is.na(decade))

tabla%>%
  knitr::kable()
```

decade	n	year_min	year_max	n_decade	avg_dcd_songs_per_year
1900's	58	1900	1909	58	5.8
1910's	52	1917	1917	52	52.0
1920's	461	1920	1929	461	46.1
1930's	453	1930	1939	453	45.3
1940's	653	1942	1949	653	81.6
1950's	3159	1950	1959	3159	315.9
1960's	8784	1960	1969	8784	878.4
1970's	17183	1970	1979	17183	1718.3
1980's	28595	1980	1989	28595	2859.5
1990's	153049	1990	1999	153049	15304.9
2000's	423753	2000	2009	423753	42375.3
2010's	498089	2010	2019	498089	49808.9
2020's	69726	2020	2020	69726	69726.0

```
tabla_1 <- tabla %>%
  pivot_longer(c("n_decade", "avg_dcd_songs_per_year"),
               names_to = "Variable",
               values_to = "n_songs")
```

```
g1 <- ggplot(tabla_1, aes(decade,
                          n_songs,
                          group=Variable,
                          col=Variable,
                          linetype=Variable)) +
  geom_line(alpha=0.7) +
  geom_point(aes(decade, n_songs, shape=Variable, col=Variable)) +
  ggtitle("The most of the songs in dataset are new") +
  labs(caption="Source: 1.2 million songs dataset from Spotify API") +
  theme(axis.text.x = element_text(angle = 45,
                                   hjust = 1))+
  geom_text(data=tabla_1[22,], aes(decade,
                                   n_songs+5e4,
                                   label="In 2020's decade \n there are only \n songs from 2020"),
            col='darkgreen',
            size=2)

g1
```




```
#####
# So let's focus on 2020.
```

```
# Later we will do some trend graphics
# to compare with information we already have.

spotify_2015_plus <- df_spotify_2 %>%
  filter(year>=2015)
dim(spotify_2015_plus)
```

```
## [1] 338462     26
```

```
spotify_2015_plus %>%
  glimpse
```

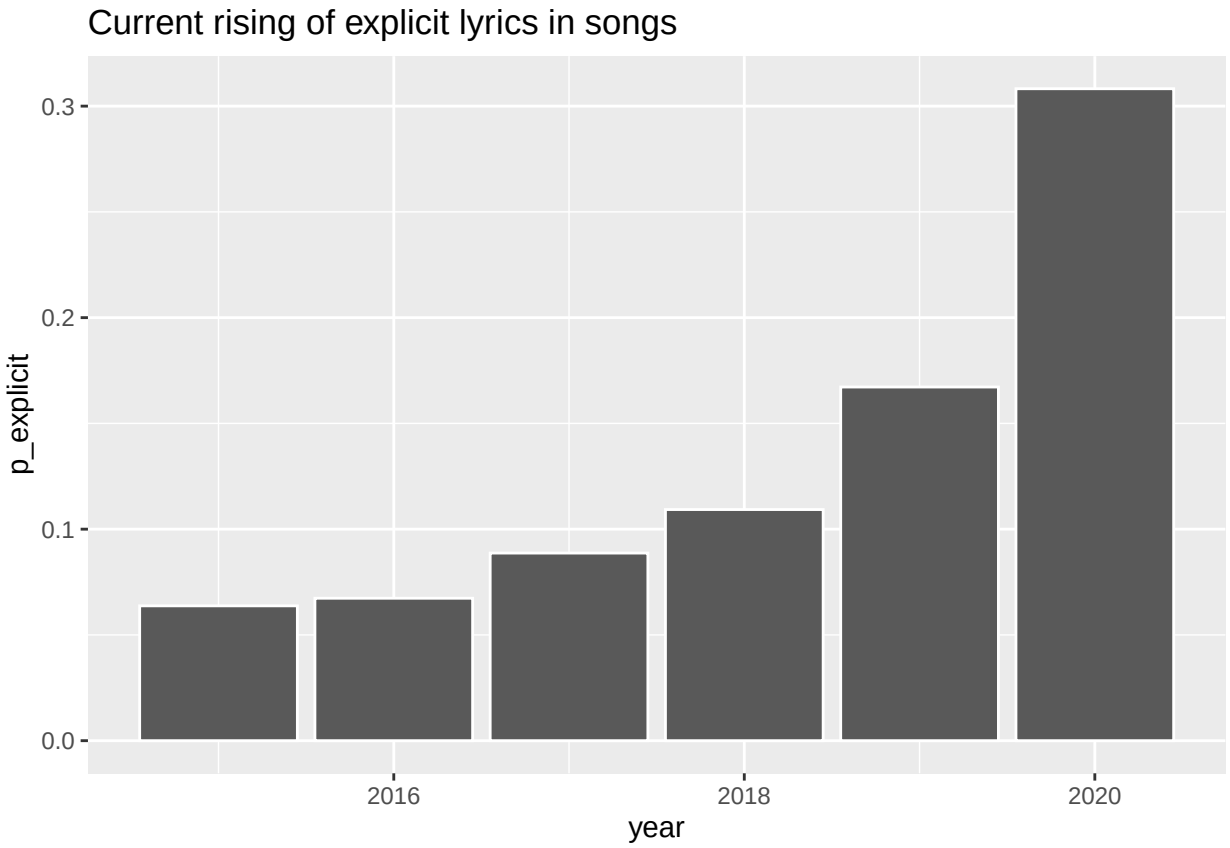
```
## Rows: 338,462
## Columns: 26
## $ id               <fct> 2SwgVZn9S4NGueAaEAryf1, 0QCQ1Isa0YPVyIbs6JwpO1, 3kIBE~
## $ name             <fct> "Man on a Mission", "Do It for Love", "Someday We'll ~
## $ album            <fct> "Do It for Love", "Do It for Love", "Do It for Love",~
## $ album_id         <fct> 4evw6IBex3N8x1oA2axMTH, 4evw6IBex3N8x1oA2axMTH, 4evw6~
## $ artists          <fct> "['Daryl Hall & John Oates']", "['Daryl Hall & John O~
## $ artist_ids       <fct> "['77tT1kLj6mCWtFNqiOmP9H']", "['77tT1kLj6mCWtFNqiOmP~
## $ track_number     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1, 2, ~
## $ disc_number      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ explicit         <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS~
## $ danceability     <dbl> 0.787, 0.587, 0.565, 0.651, 0.833, 0.782, 0.605, 0.67~
## $ energy           <dbl> 0.903, 0.958, 0.781, 0.567, 0.805, 0.619, 0.921, 0.89~
## $ key              <int> 0, 4, 1, 9, 0, 8, 9, 10, 10, 1, 7, 5, 10, 8, 1, 8, 6,~
## $ loudness         <dbl> -4.894, -5.149, -5.073, -6.417, -4.554, -5.759, -4.33~
## $ mode             <int> 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,~
## $ speechiness      <dbl> 0.0315, 0.0586, 0.0308, 0.0240, 0.0347, 0.0266, 0.040~
## $ acousticness     <dbl> 2.92e-01, 1.07e-01, 2.33e-02, 5.62e-01, 7.60e-02, 2.6~
## $ instrumentalness <dbl> 2.48e-05, 0.00e+00, 9.91e-06, 5.78e-06, 1.36e-02, 0.0~
## $ liveness         <dbl> 0.1010, 0.0574, 0.0819, 0.1860, 0.0731, 0.0607, 0.228~
## $ valence          <dbl> 0.962, 0.832, 0.461, 0.370, 0.974, 0.898, 0.705, 0.96~
## $ tempo            <dbl> 119.946, 87.976, 109.977, 97.030, 116.013, 110.004, 9~
## $ duration_ms      <int> 224307, 238000, 268013, 277813, 209960, 229253, 22117~
## $ time_signature   <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,~
## $ year             <int> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018,~
## $ release_date     <fct> 2018-04-10, 2018-04-10, 2018-04-10, 2018-04-10, 2018-~
## $ decade           <chr> "2010's", "2010's", "2010's", "2010's", "2010's", "20~
## $ release          <date> 2018-04-10, 2018-04-10, 2018-04-10, 2018-04-10, 2018~
```

```
spotify_2015_plus %>%
  count(explicit) %>%
  mutate(p=n/sum(n))
```

```
##    explicit      n         p
## 1:    FALSE 288871 0.8534813
## 2:     TRUE  49591 0.1465187
```

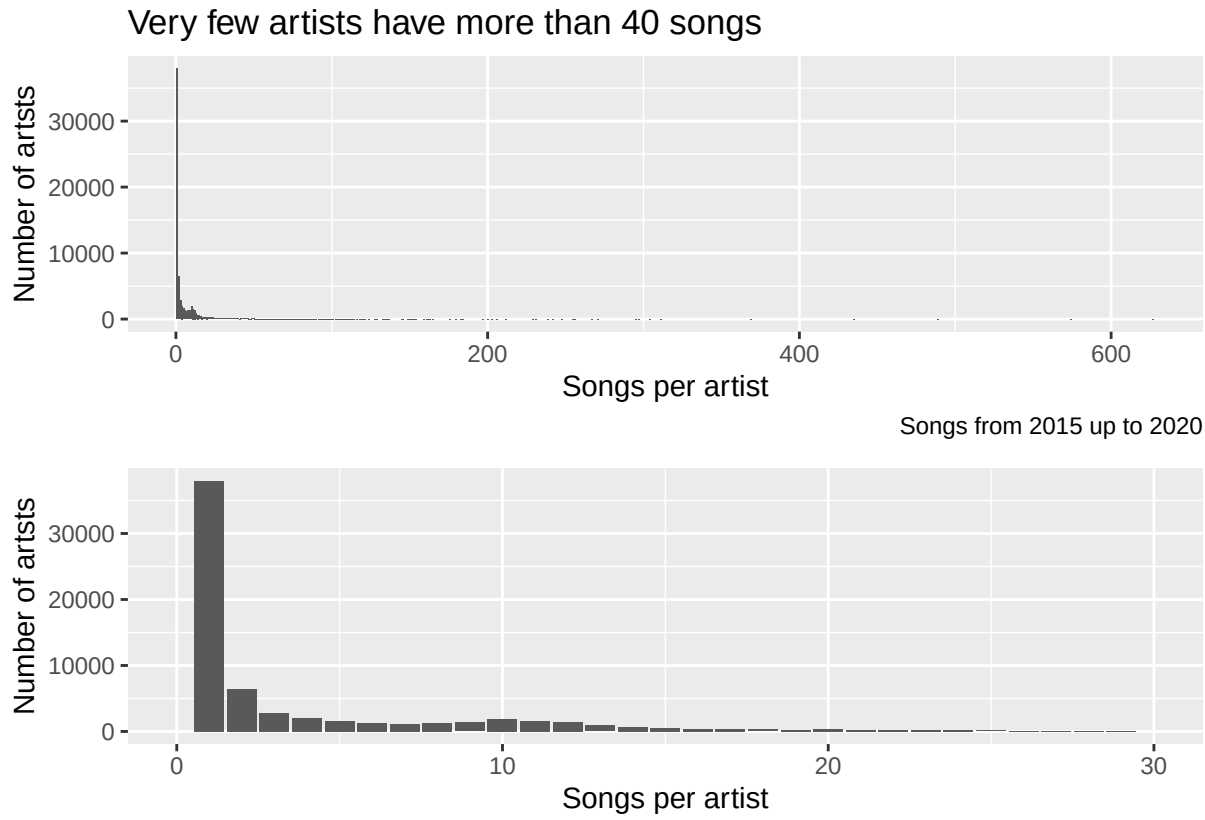
```
spotify_2015_plus %>%
  group_by(year) %>%
  summarise(n=n(),
            explicit=sum(explicit)) %>%
  ungroup() %>%
  mutate(p_explicit=explicit/n) %>%
  ggplot(aes(year, p_explicit)) +
```

```
  geom_col(col="white") +
  labs(title="Current rising of explicit lyrics in songs")
```




```
############################################################
############################################################
# Resumir variables a nivel artista
############################################################
############################################################

per_artist <- spotify_2015_plus %>%
  group_by(artists) %>%
  summarise(n=n())

a1 <- per_artist %>%
  ggplot(aes(n)) +
  geom_bar() +
  labs(title='Very few artists have more than 40 songs',
       x="Songs per artist",
       y="Number of artsts",
       caption="Songs from 2015 up to 2020")

a2 <- per_artist %>%
  ggplot(aes(n)) +
  geom_bar() +
  xlim(c(0, 30)) +
    labs(x="Songs per artist",
```

```
       y="Number of artsts")
a1/a2
```




```
per_artist %>%
  arrange(desc(n)) %>%
  glimpse
```

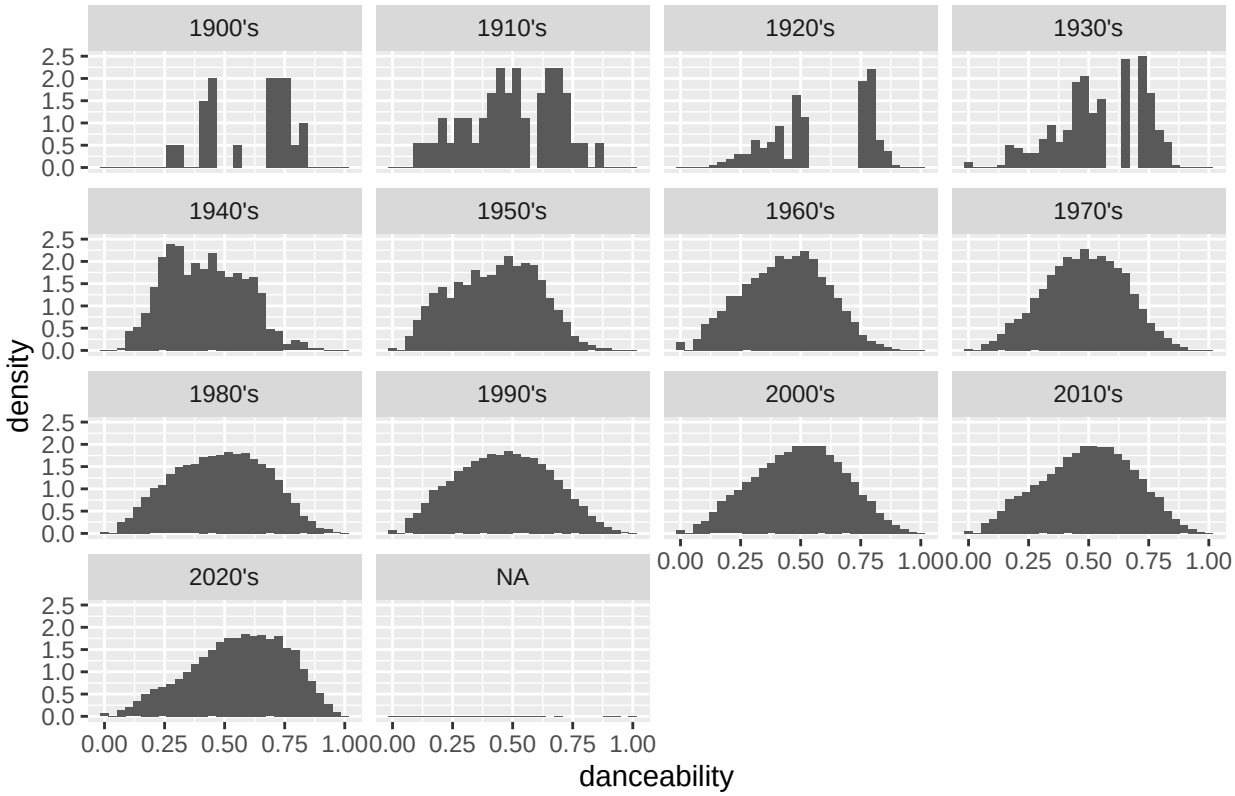
```
## Rows: 67,344
## Columns: 2
## $ artists <fct> "['Revolt Production Music']", "['Grant Macdonald']", "['Circl~
## $ n       <int> 627, 574, 489, 435, 369, 311, 304, 297, 295, 295, 271, 267, 25~
```

Análisis univariado por década

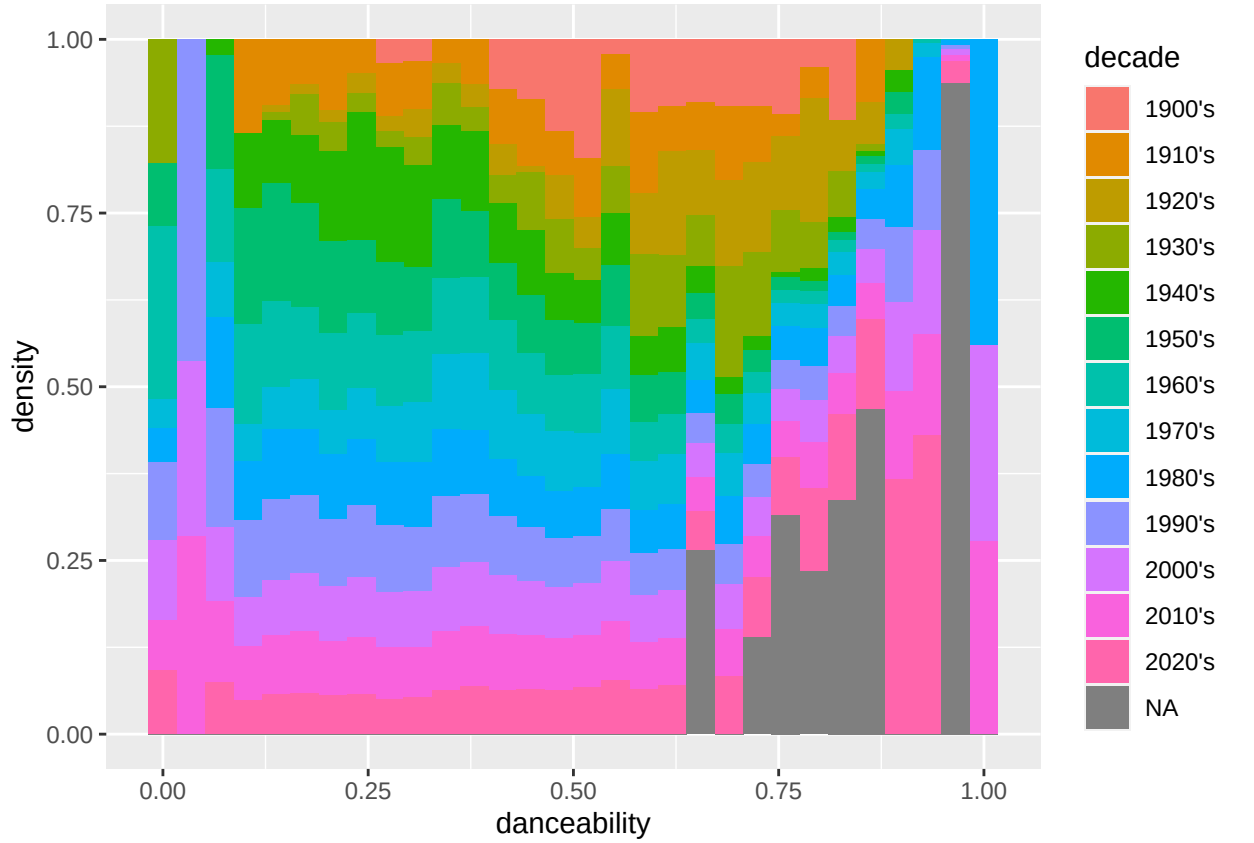
Danceability

```
df_spotify_2 %>%
  ggplot(aes(danceability, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 2.5)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

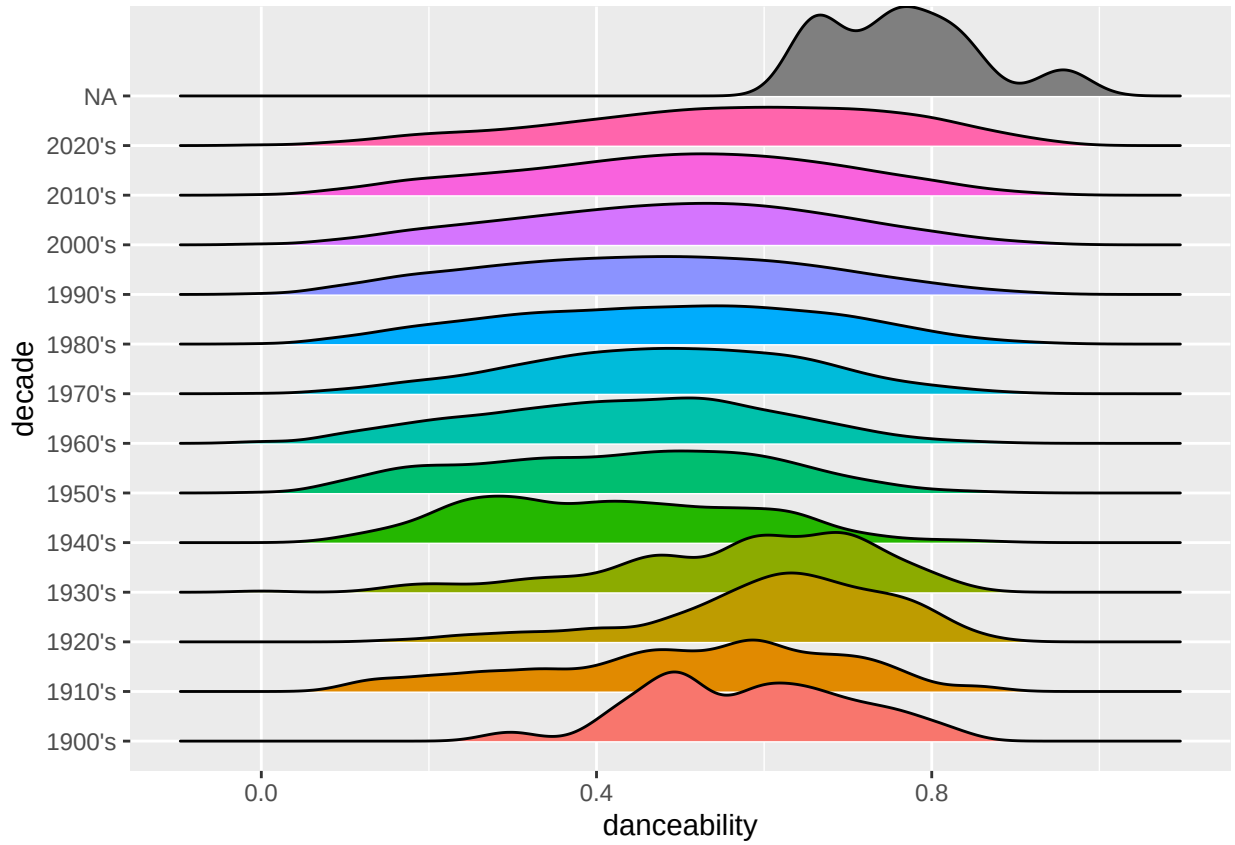
En 2020 hay mayor asimetría




```
g1 <- df_spotify_2 %>%
  ggplot(aes(danceability, y=..density.., fill=decade)) +
  geom_histogram(position="fill")
g2 <- df_spotify_2 %>%
  ggplot(aes(danceability, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")
g1
```


1.00
0.75
0.50
0.25
0.00
density
0.00
0.25
0.50
0.75
1.00
danceability
decade
1900's
1910's
1920's
1930's
1940's
1950's
1960's
1970's
1980's
1990's
2000's
2010's
2020's
NA

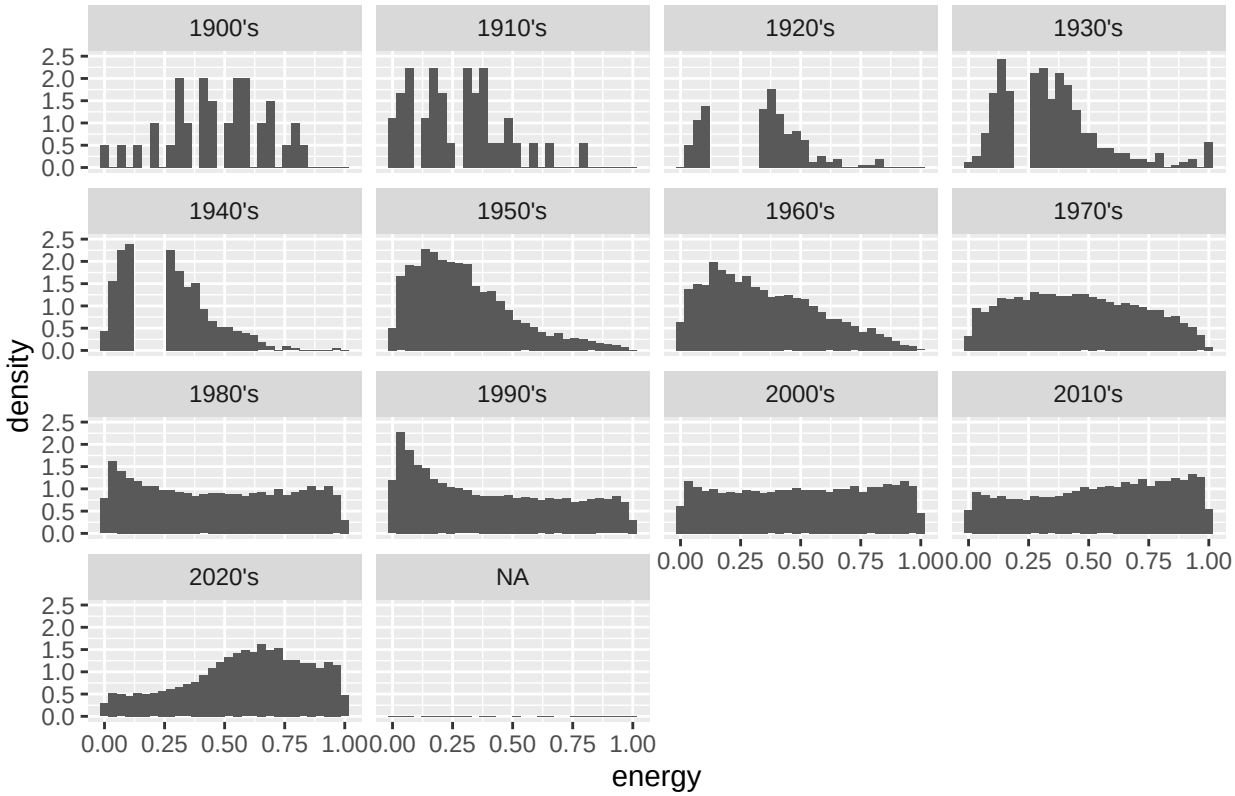
g2



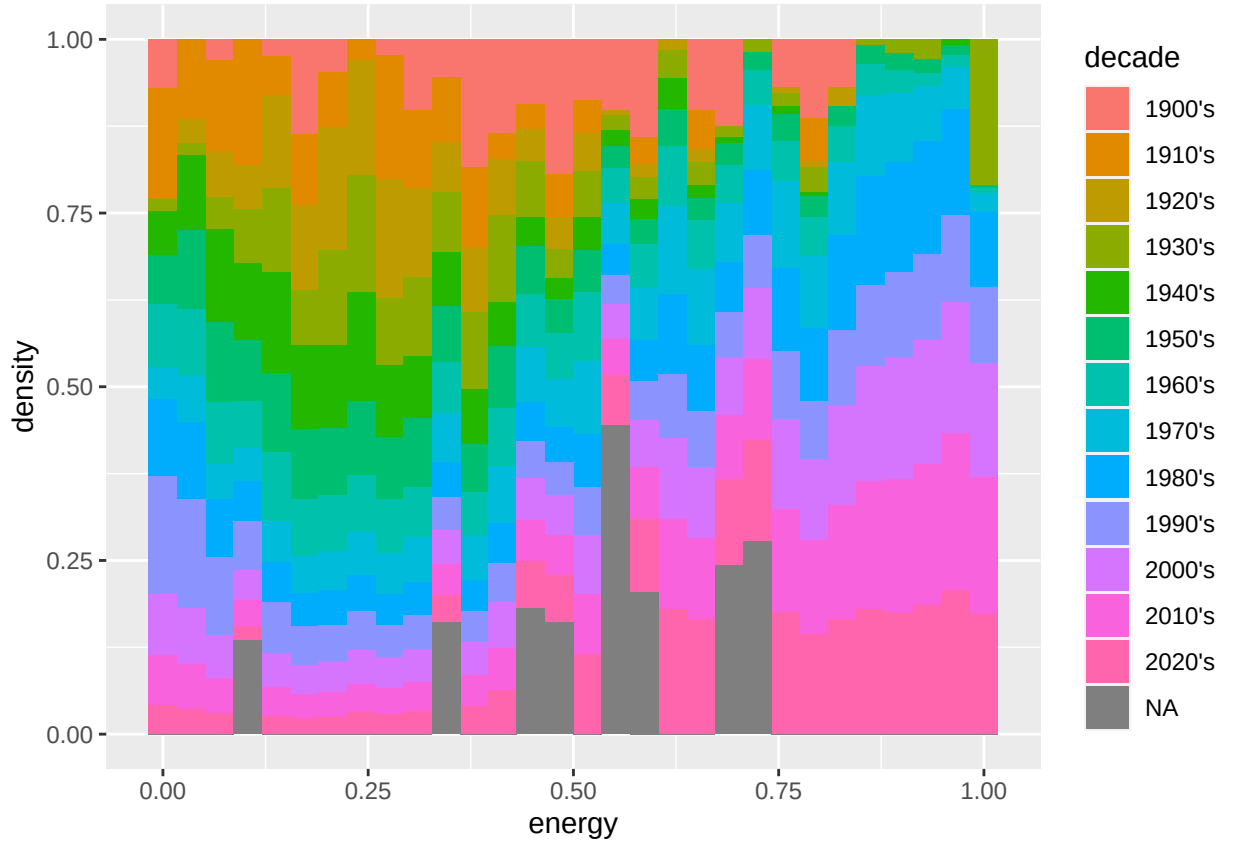
Energy

```
df_spotify_2 %>%
  ggplot(aes(energy, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 2.5)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

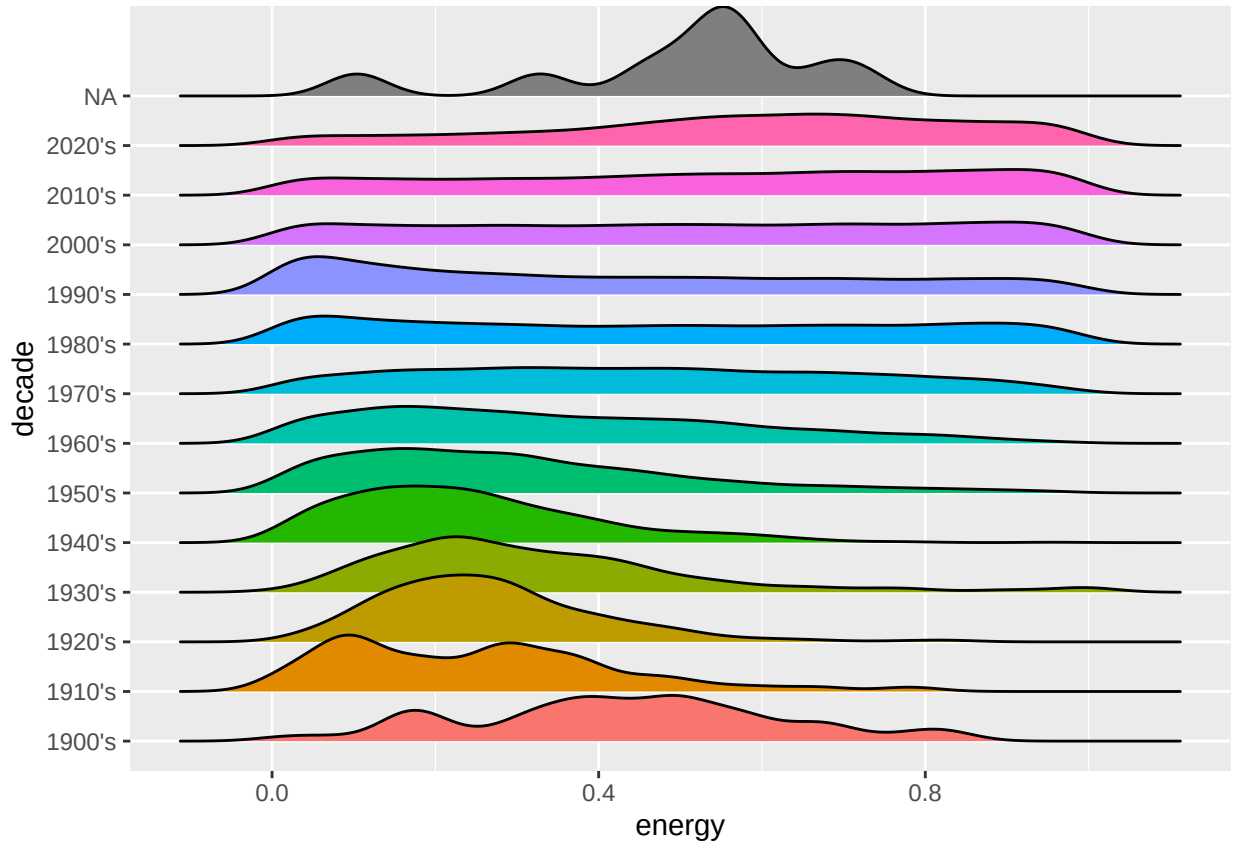
En 2020 hay mayor asimetría




```
g1 <- df_spotify_2 %>%
  ggplot(aes(energy, y=..density.., fill=decade)) +
  geom_histogram(position="fill")
g2 <- df_spotify_2 %>%
  ggplot(aes(energy, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")
g1
```


1.00
0.75
0.50
0.25
0.00
density
0.00
0.25
0.50
0.75
1.00
energy
decade
1900's
1910's
1920's
1930's
1940's
1950's
1960's
1970's
1980's
1990's
2000's
2010's
2020's
NA

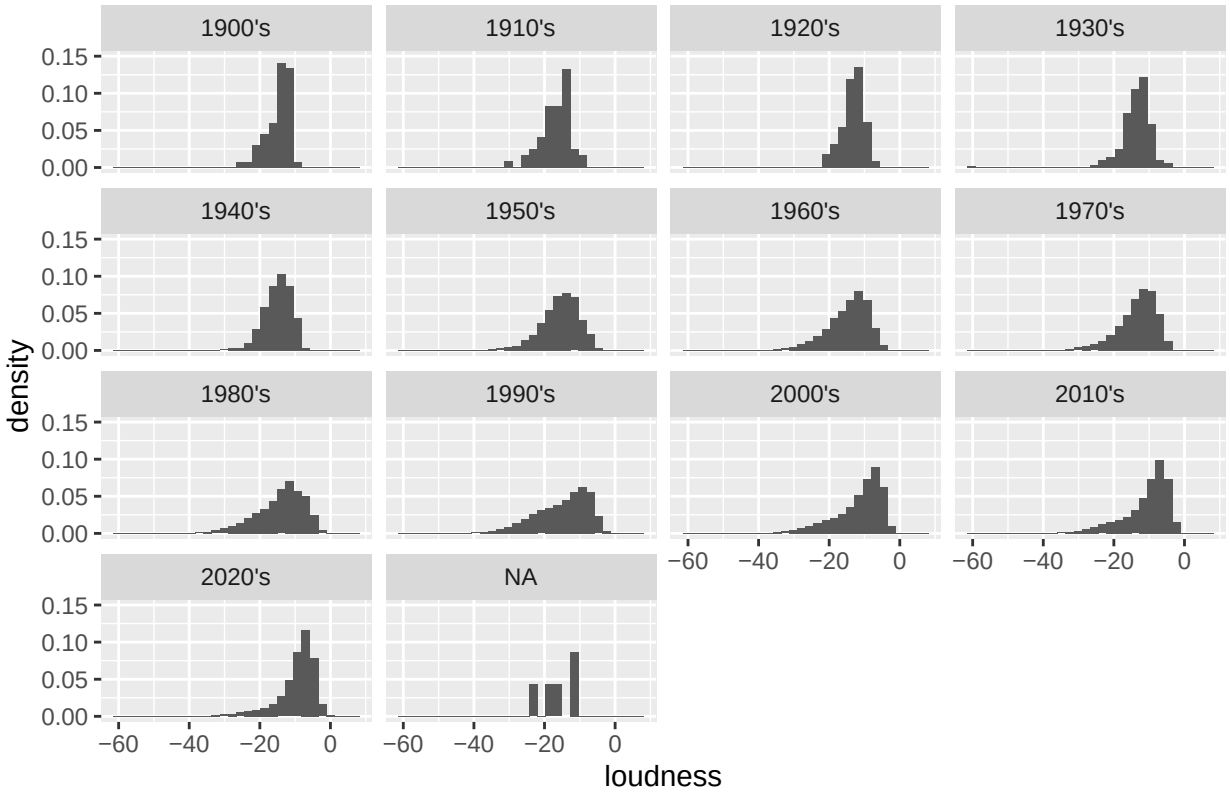
```
g2
```



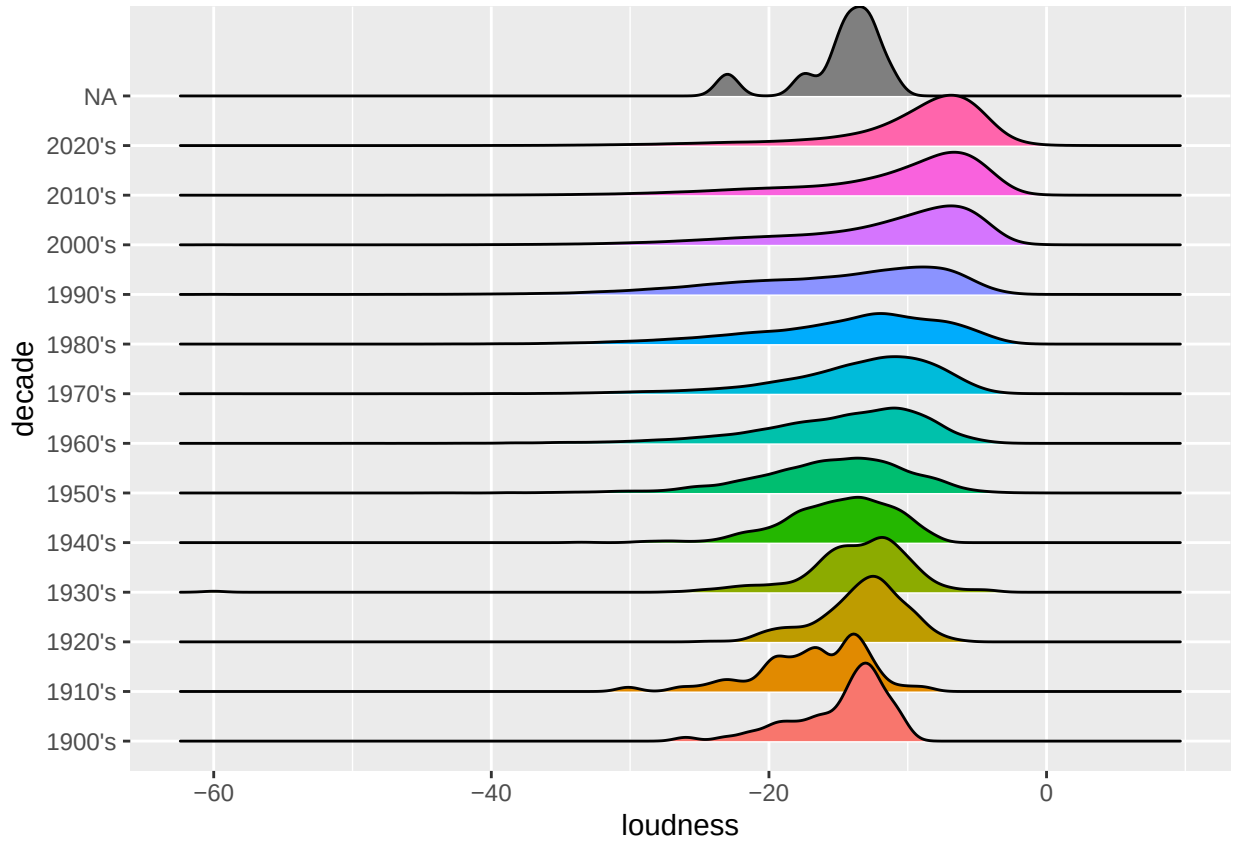
Loudness

```
df_spotify_2 %>%
  ggplot(aes(loudness, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 0.15)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

En 2020 hay mayor asimetría



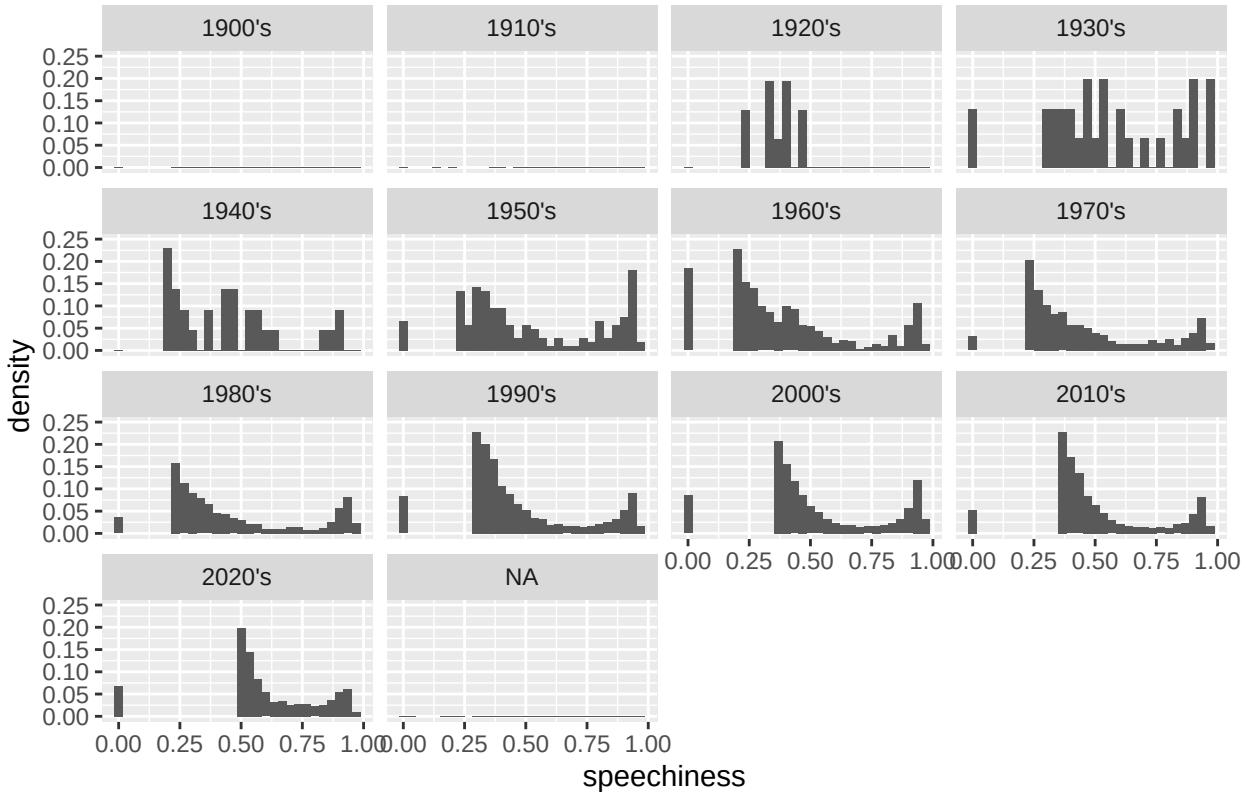

```
g2 <- df_spotify_2 %>%
  ggplot(aes(loudness, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")
g2
```



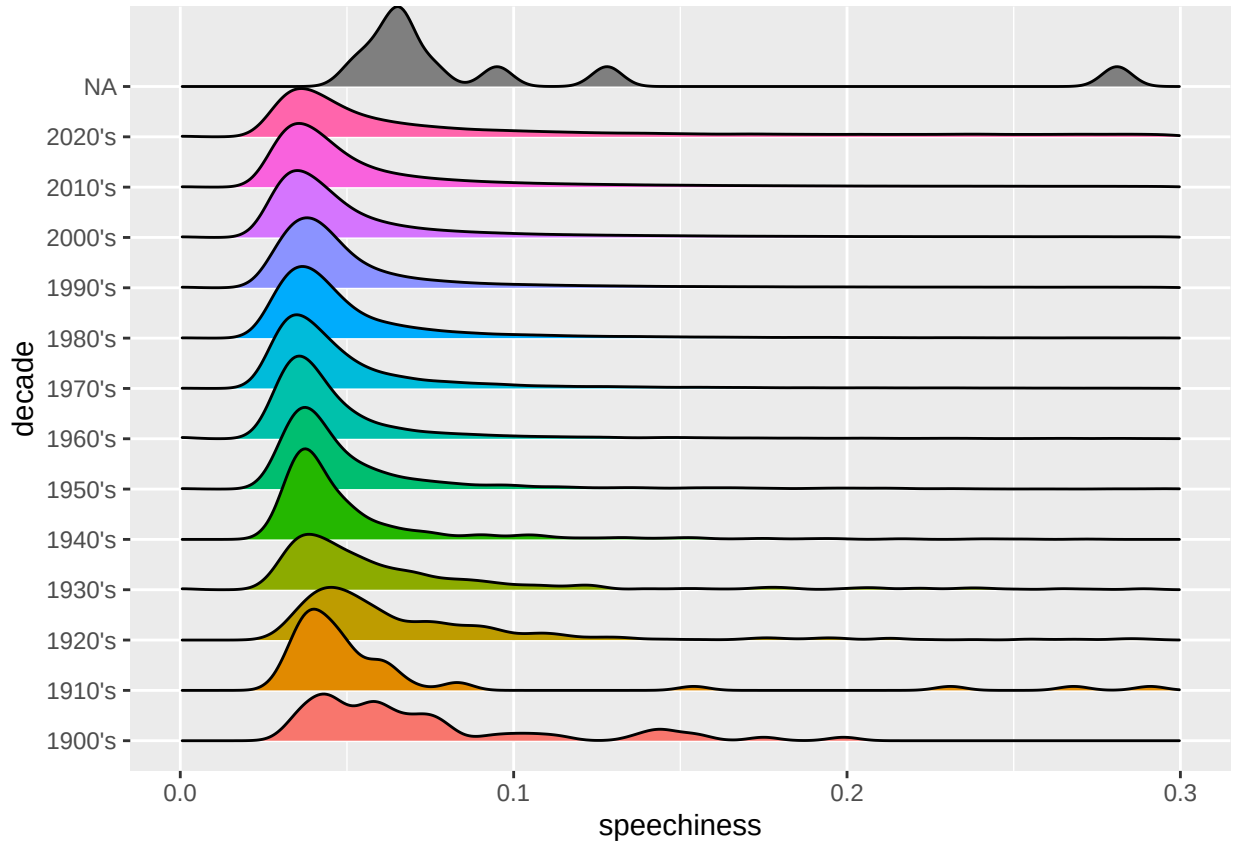
Speechiness

```
df_spotify_2 %>%
  ggplot(aes(speechiness, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 0.25)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

En 2020 hay mayor asimetría



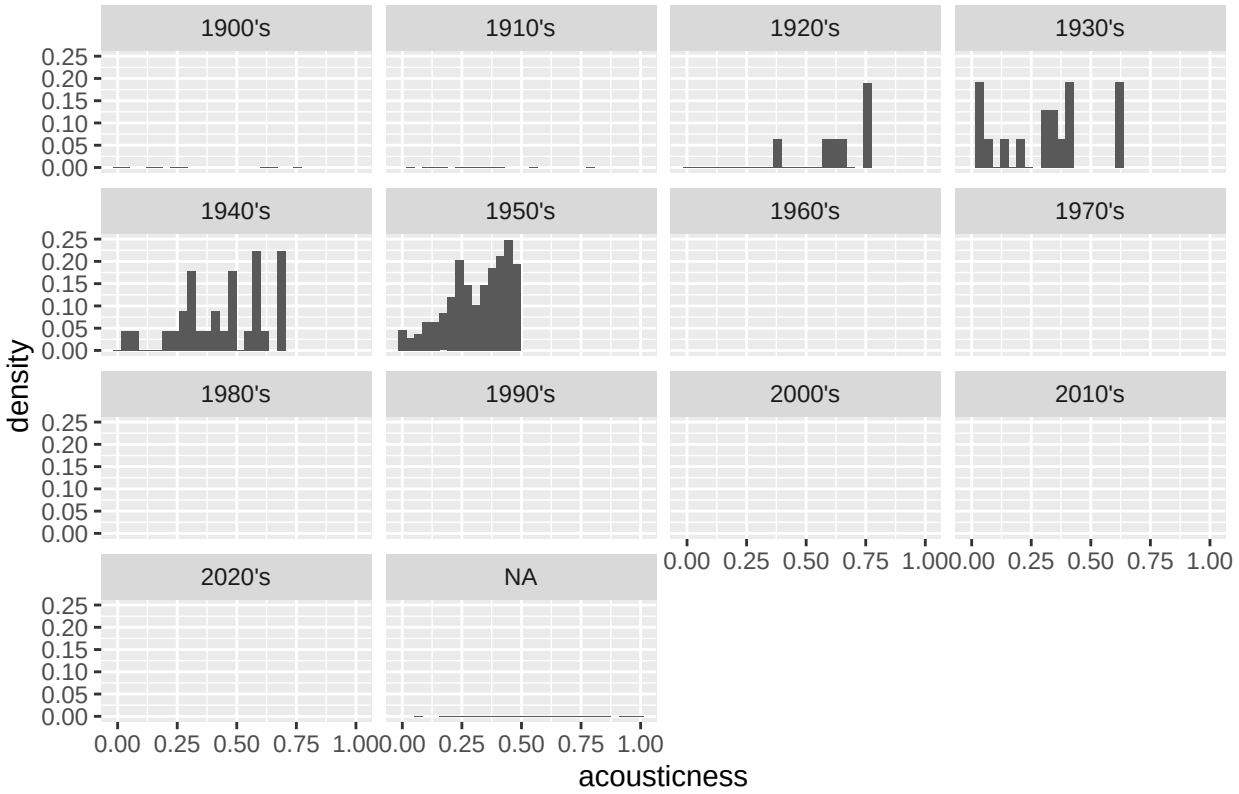

```
g2 <- df_spotify_2 %>%
  ggplot(aes(speechiness, y=decade, fill=decade)) +
  geom_density_ridges() +
    xlim(c(0, 0.3)) +
  theme(legend.position = "none")
g2
```



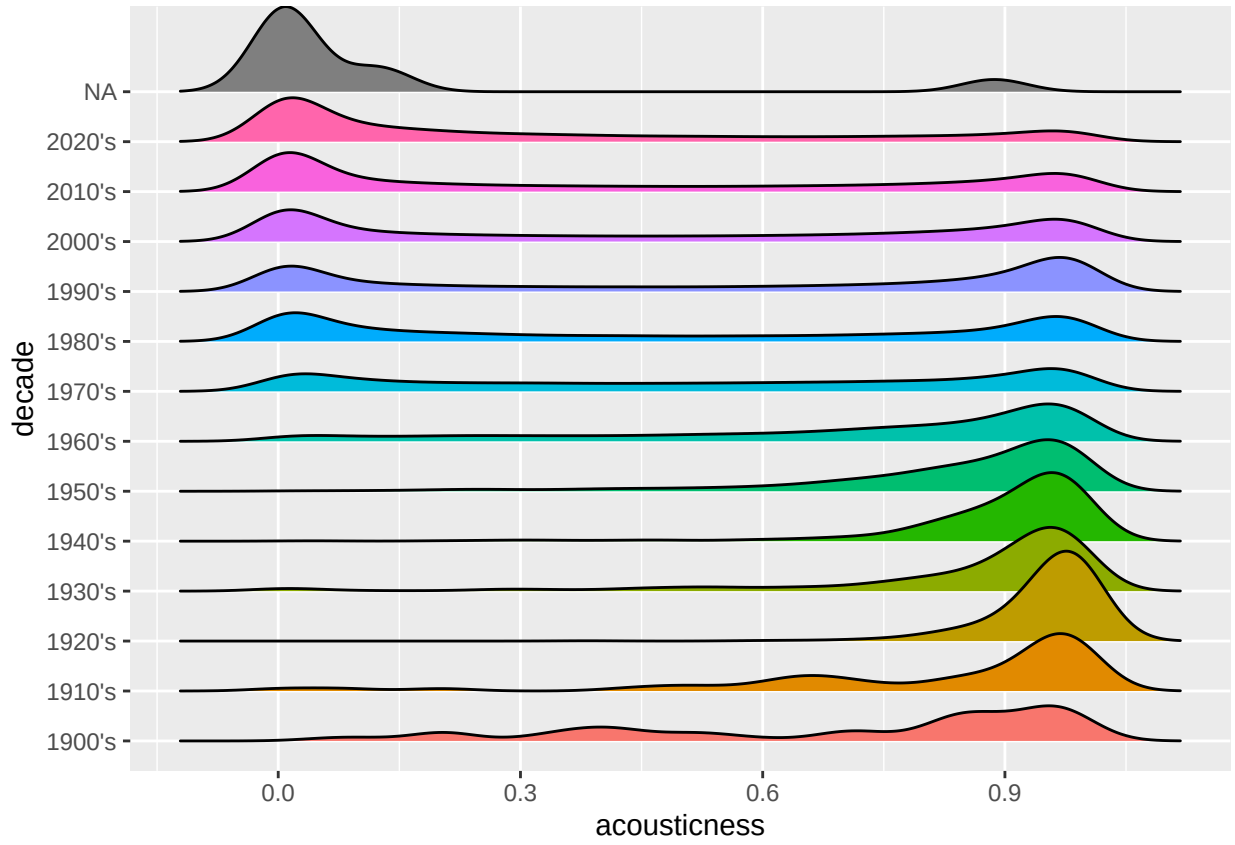
Acousticness

```
df_spotify_2 %>%
  ggplot(aes(acousticness, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 0.25)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

En 2020 hay mayor asimetría



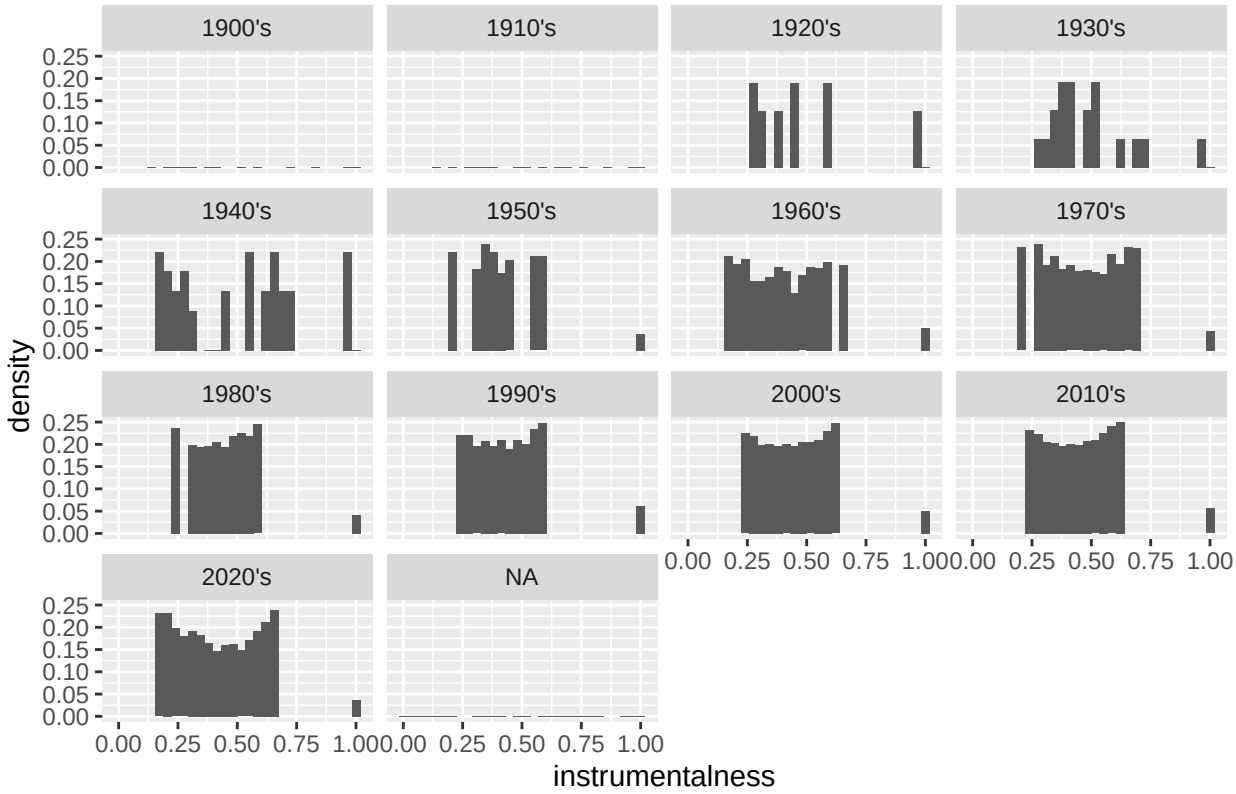

```
g2 <- df_spotify_2 %>%
  ggplot(aes(acousticness, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")
g2
```



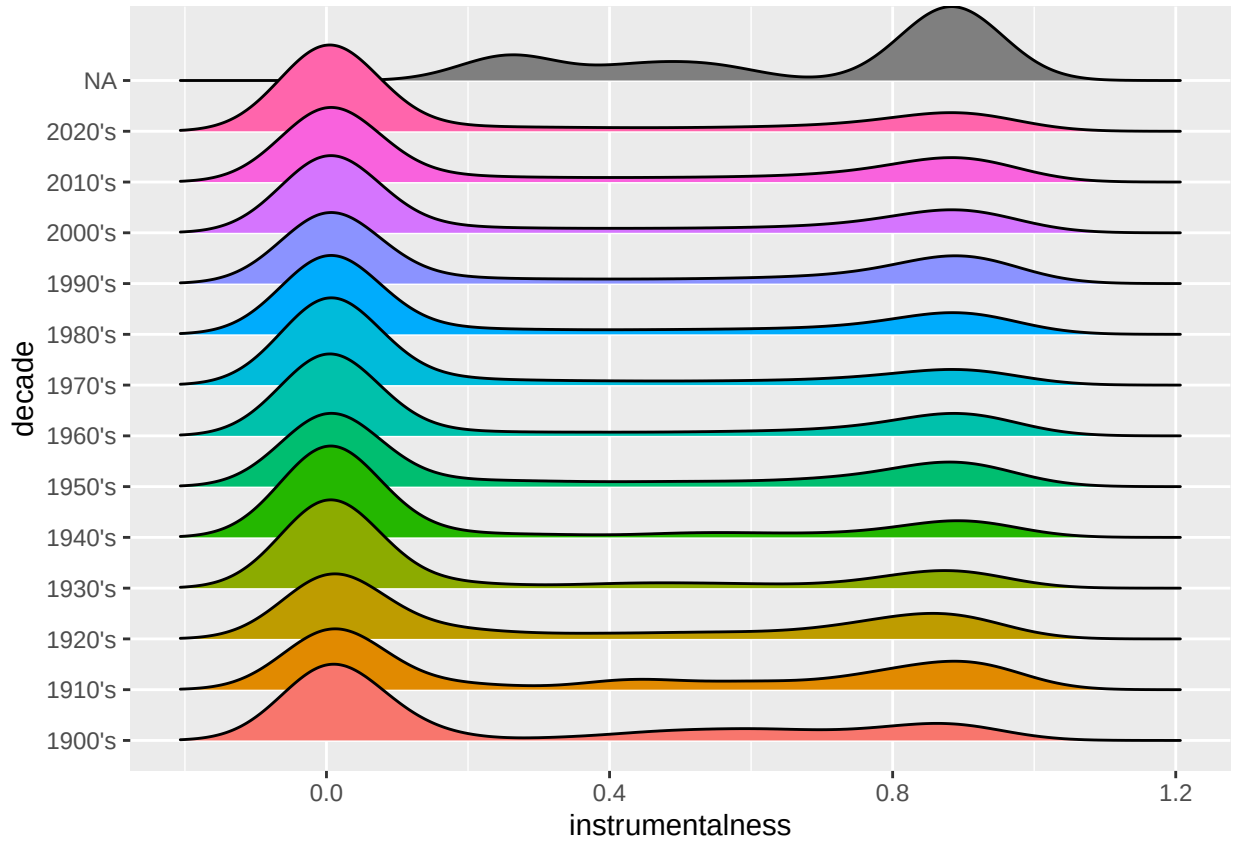
Instrumentalness

```
df_spotify_2 %>%
  ggplot(aes(instrumentalness, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 0.25)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

En 2020 hay mayor asimetría



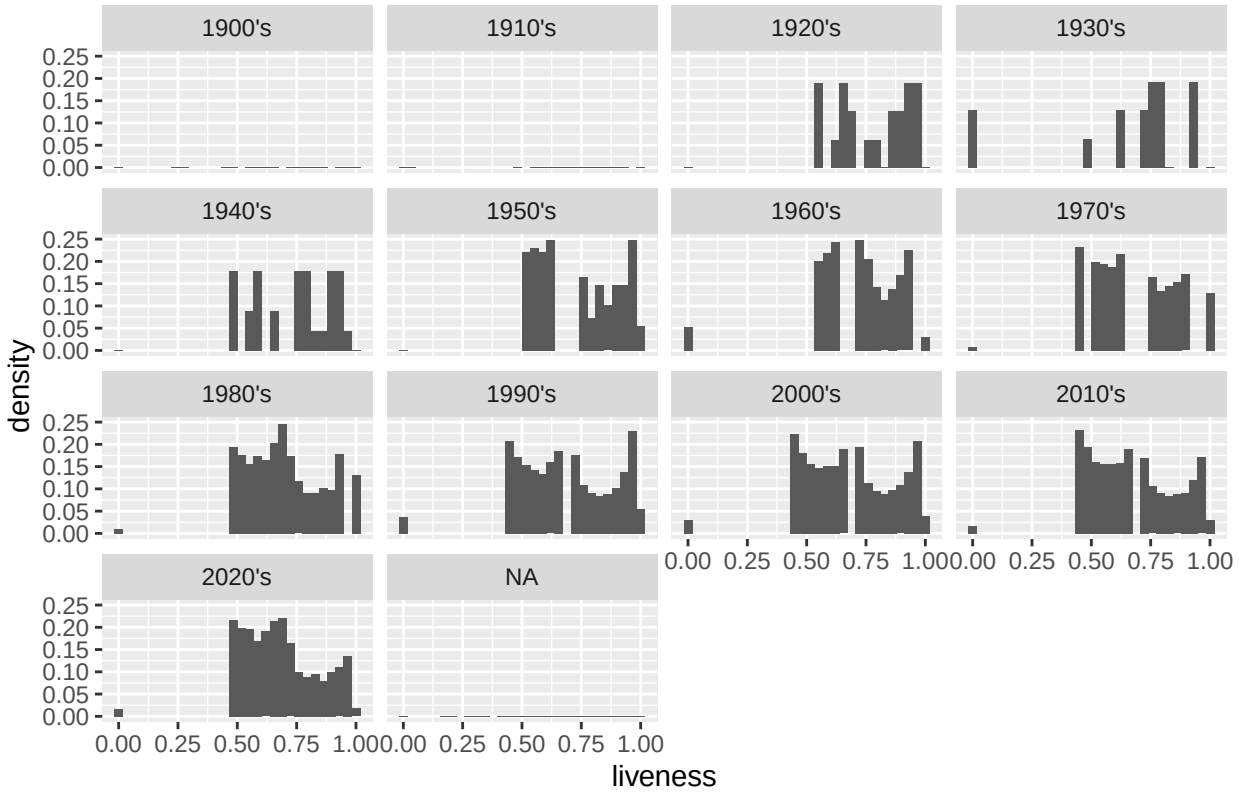

```
g2 <- df_spotify_2 %>%
  ggplot(aes(instrumentalness, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")
g2
```



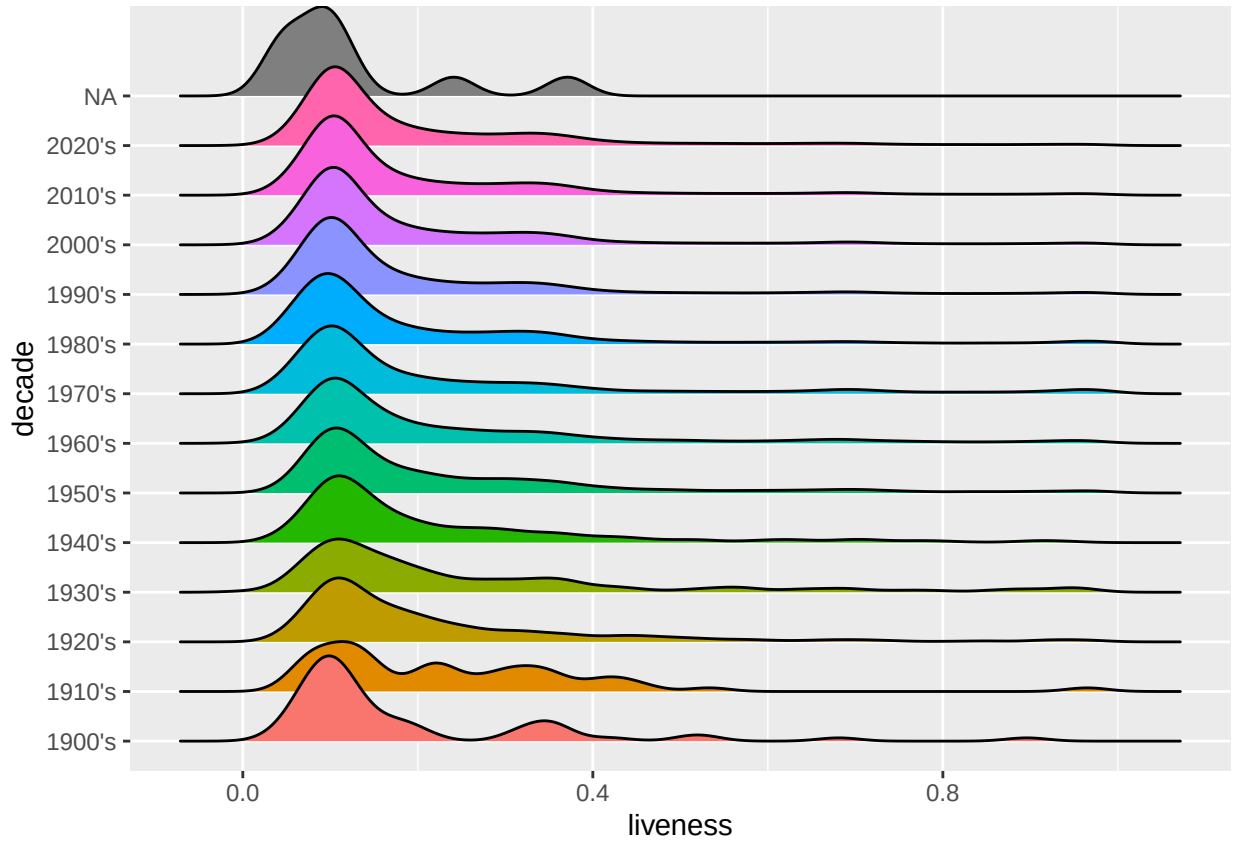
liveness

```
df_spotify_2 %>%
  ggplot(aes(liveness, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 0.25)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

En 2020 hay mayor asimetría



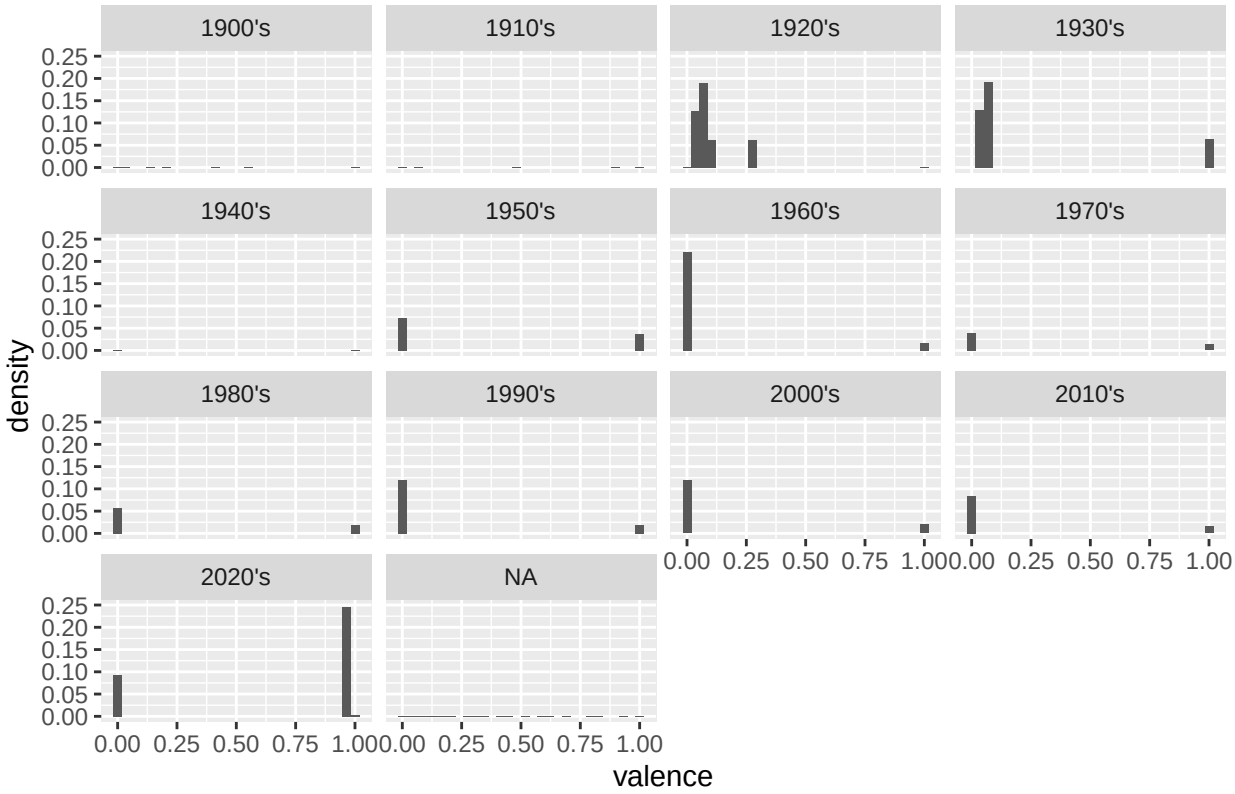

```
g2 <- df_spotify_2 %>%
  ggplot(aes(liveness, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")
g2
```



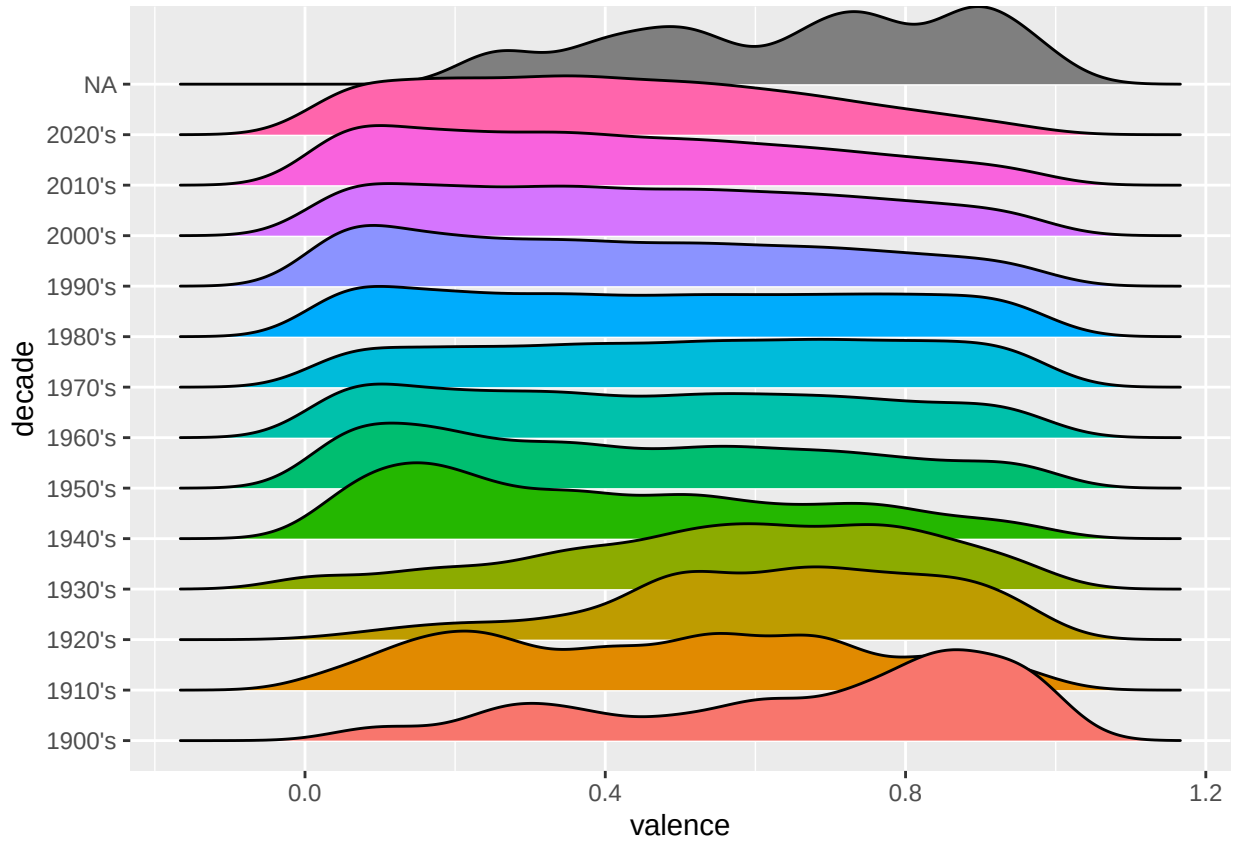
Valence

```
df_spotify_2 %>%
  ggplot(aes(valence, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 0.25)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

En 2020 hay mayor asimetría



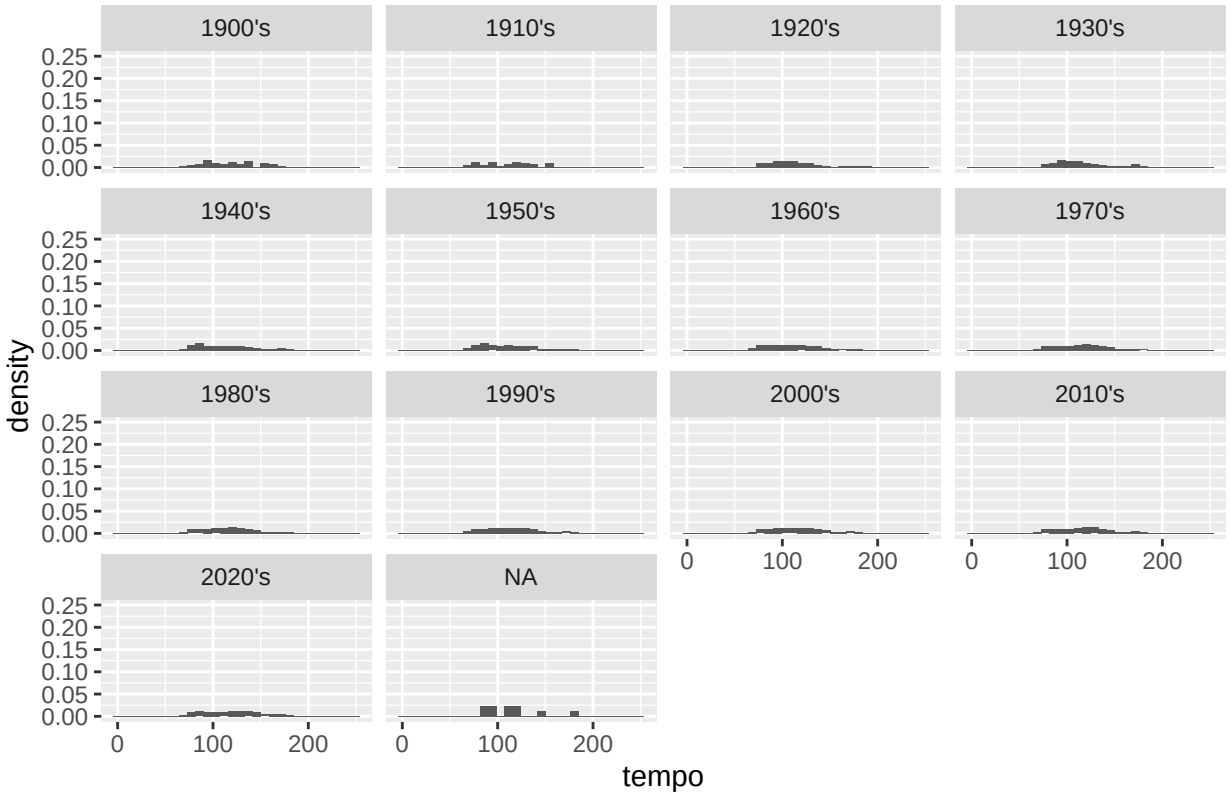

```
g2 <- df_spotify_2 %>%
  ggplot(aes(valence, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")
g2
```



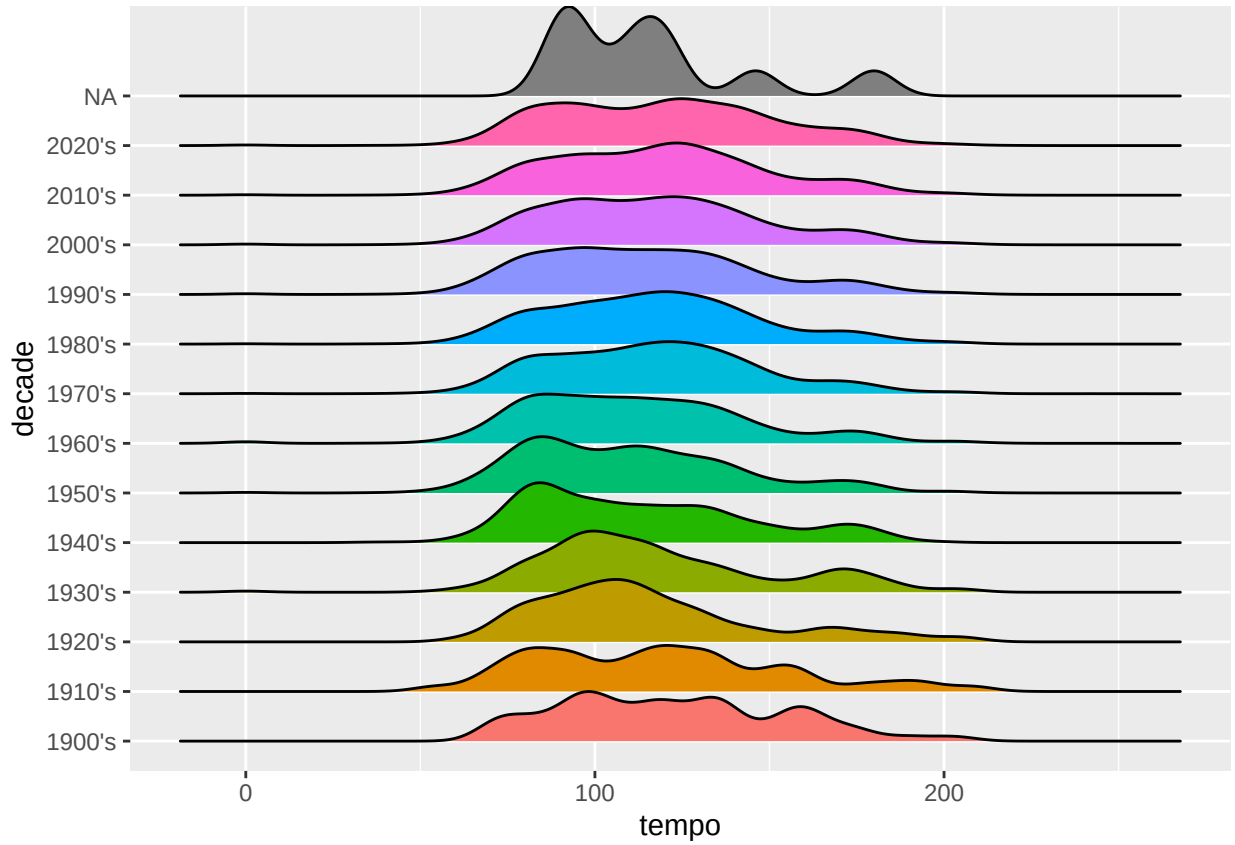
Tempo

```
df_spotify_2 %>%
  ggplot(aes(tempo, y=..density..)) +
  geom_histogram() +
  ylim(c(0, 0.25)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")
```

En 2020 hay mayor asimetría



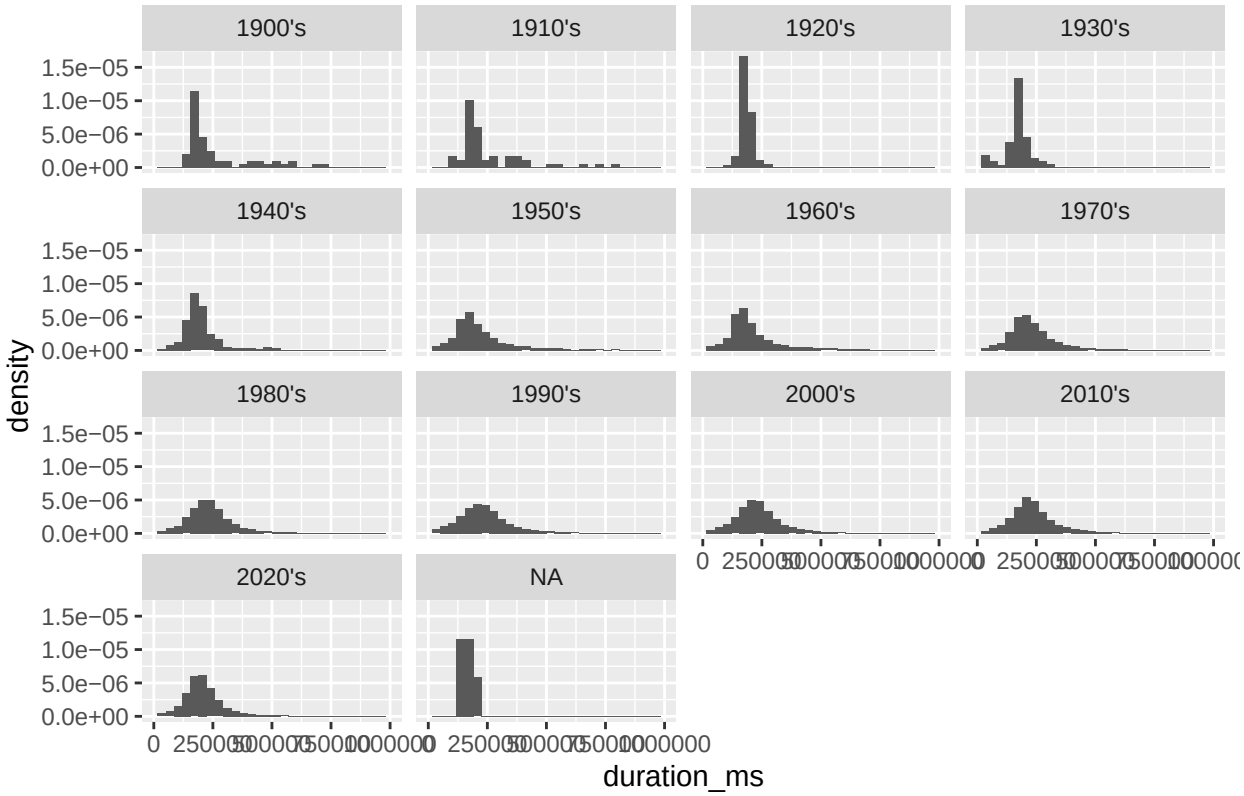

```
g2 <- df_spotify_2 %>%
  ggplot(aes(tempo, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")
g2
```



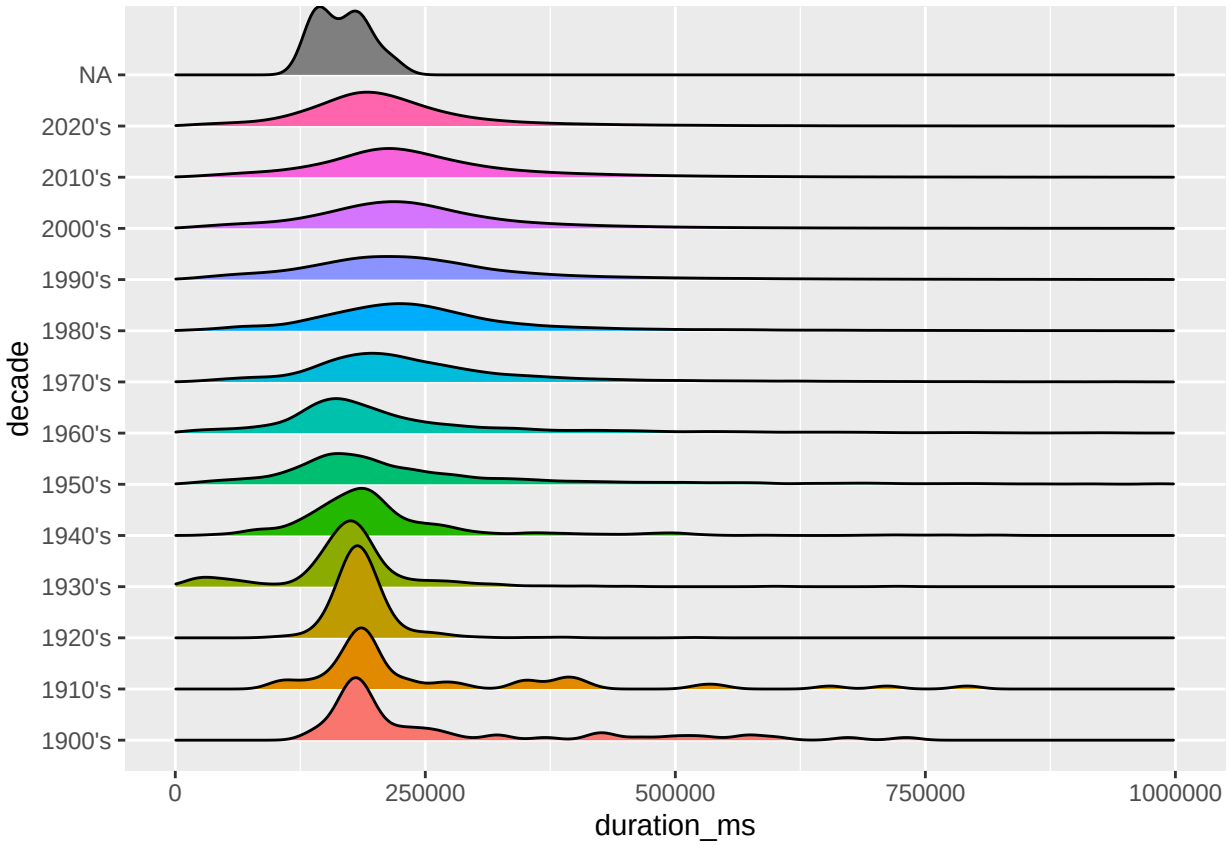
Duration

```
df_spotify_2 %>%
  ggplot(aes(duration_ms, y=..density..)) +
  geom_histogram() +
#  ylim(c(0, 0.25)) +
  facet_wrap(vars(decade)) +
  ggtitle("En 2020 hay mayor asimetría")+
  xlim(c(0,1e6))
```

En 2020 hay mayor asimetría




```
g2 <- df_spotify_2 %>%
  ggplot(aes(duration_ms, y=decade, fill=decade)) +
  geom_density_ridges() +
  theme(legend.position = "none")+
  xlim(c(0,1e6))
g2
```



Not available data

Not available data are located in the `release_date` variable, however as have been seen before, the missing data reflects a particular behavior in relation to variables like `danceability`, “