

Padrões na preferência musical dos(as) brasileiros(as) sob a ótica do Spotify

Não é incomum ouvir de pessoas ligadas ao mundo da música, que o que se produz hoje em dia, no meio musical, segue um padrão de indústria, como uma montagem perfeitamente arquitetada, sem possibilidade de grandes mudanças. O propósito desta análise é, através da análise dos dados das músicas mais ouvidas no Spotify Brasil, ajudar na compreensão do gosto musical do brasileiro nos últimos anos (2017-2021), permitindo-nos identificar se há, realmente, padrões muito evidentes em cada uma das características musicais (audio features) destas canções.

```
#Caso ainda não esteja instalado
```

```
#install.packages("dplyr")
#install.packages("tidyverse")
```

Importando as libraries que utilizaremos na análise:

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr    0.3.4
## v tibble   3.1.4     v dplyr    1.0.7
## v tidyr    1.1.3     v stringr  1.4.0
## v readr    2.0.1     vforcats  0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
```

```
library(here)
```

```
## here() starts at C:/Users/kelvin.c.custodio/OneDrive - Accenture/Documents/Private
```

```
library(readr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##       date, intersect, setdiff, union
```

```

library(plotly)

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout

theme_set(theme_bw())

```

Inicialmente, importaremos o primeiro dataset obtido. Ele contém as informações referentes às músicas que chegaram nos rankings das mais ouvidas em cada um dos países listados. Usaremos nesta análise os seguintes campos: title: o título da música; date: a data em que a música esteve em uma certa posição do ranking; artist: o nome do(s) intérprete(s) da música; e url: endpoint do Spotify no qual poderemos ouvir a música e buscar informações adicionais, através da API do Spotify. Neste caso, iremos usá-la para recuperar os dados das características musicais (audio features); e region: o país do ranking.

```

all_charts = read_csv(
  here("~/Private/charts.csv"),
  col_types = cols(
    .default = col_double(),
    title = col_character(),
    date = col_date(format = ""),
    artist = col_character(),
    url = col_character(),
    region = col_character()
  )
)

```

Para obtermos as características musicais de cada canção presente no dataset acima, iremos selecionar as urls de cada uma. Como há músicas que aparecem mais de uma vez, em diferentes datas e em diferentes países, selecionaremos usando a função “distinct”. Desta forma, não recuperaremos os dados de uma mesma música mais de uma vez.

```

url_charts = distinct(all_charts %>%
  select(url))

```

```

url_charts

```

```

## # A tibble: 203,788 x 1
##       url
##   <chr>

```

```

## 1 https://open.spotify.com/track/4un0svr9C2NqI9aJXrxpCy
## 2 https://open.spotify.com/track/6RUKPb4LETWmmr3iAEQktW
## 3 https://open.spotify.com/track/0ERbK7qVqveCaBWIiYCr13
## 4 https://open.spotify.com/track/0s6yB0ZF11LvYHAnGhXFga
## 5 https://open.spotify.com/track/2LawezPeJhN4AWuSBOGtAU
## 6 https://open.spotify.com/track/0eVborSuxUeSg0meWYd9dZ
## 7 https://open.spotify.com/track/0c1gHntWjKD7QShC8s99sq
## 8 https://open.spotify.com/track/4kf1IGfjdZJW4ot2ioixTB
## 9 https://open.spotify.com/track/4kfm1uI9QGaoN9qm6CkAMn
## 10 https://open.spotify.com/track/4BP3uhOhFLFRb5cjsgLqDh
## # ... with 203,778 more rows

```

Após selecionarmos apenas as urls de cada música presente nos rankings de mais ouvidas, iremos gravá-las em um arquivo csv. Este arquivo será lido por um script desenvolvido em Python e, com o auxílio da library Spotipy, realizaremos chamadas à API do Spotify para recuperarmos as “audio features” associadas à cada canção. Essas informações recuperadas serão gravadas em um arquivo csv que será unido com um subconjunto do dataset que contém as informações dos rankings (nem todas as colunas serão aproveitadas para esta análise). Como há músicas que aparecem mais de uma vez, em diferentes datas e em diferentes países, selecionaremos usando a função “distinct”. Desta forma, não recuperaremos os dados de uma mesma música mais de uma vez.

```
write.csv(url_charts, 'music_urls.csv')
```

Neste ponto, o script mencionado já recuperou as informações desejadas (vale salientar que não foi possível recuperar as informações de 19 músicas, por não estarem mais presentes na plataforma. Por esse ser um número bastante inexpressivo, diante das milhares de músicas obtidas, isso não terá nenhum efeito negativo nesta análise). Adiante, para facilitar a compreensão, eis algumas informações sobre o que significa cada uma das características musicais, no contexto desta análise:

Danceability: valor numérico que descreve o quanto adequada uma música é para dançar, com base em uma combinação de elementos musicais, incluindo tempo, estabilidade do ritmo, força da batida e regularidade geral. Um valor de 0.0 é o menos “dançável” e 1.0 é o mais;

Instrumentalness: valor numérico que indica se uma faixa não contém vocais. Os sons “ooh”, “aah” e semelhantes são tratados como instrumentais neste contexto. Faixas de rap, por exemplo, são claramente “vocais”, enquanto músicas eruditas costumam ser instrumentais. Quanto mais próximo o valor de 1.0, maior será a probabilidade da faixa não conter conteúdo vocal. Valores a partir de 0.5 passam a indicar que a faixa tem mais características instrumentais, e quanto mais próximos de 1.0, maior é a confiança de que a faixa é vocal;

Loudness: valor numérico que representa o volume geral de uma música em decibéis (dB). Os valores de sonoridade são calculados em toda a trilha. Seus valores típicos variam entre -60 e 0;

Speechiness: valor numérico que representa a presença de palavras faladas em uma faixa. Quanto mais exclusivamente falada for a gravação, mais próximo de 1.0 será o valor do atributo. Valores acima de 0.66 indicam faixas que provavelmente são compostas completamente por palavras faladas. Valores entre 0.33 e 0.66 descrevem faixas que podem conter música e fala, incluindo casos como música do gênero rap. Valores abaixo de 0.33 provavelmente representam música e outras faixas não semelhantes à fala;

Valence: medida na faixa de 0.0 a 1.0 que descreve a positividade musical transmitida por uma faixa. Faixas com alta valência passam a sensação de positividade, alegria, euforia; enquanto faixas com baixa valência soam mais negativas, tristes, melancólicas ou, até mesmo, raivas;

duration_ms: a duração da faixa em milissegundos;

track_href: a url utilizada para identificar a música.

Dito isso, sigamos com o import dos dados:

```
audio_features = read_csv(  
    here("~/Private/audio_features.csv"),  
    col_types = cols(  
        danceability = col_double(),  
        loudness = col_double(),  
        speechiness = col_double(),  
        instrumentalness = col_double(),  
        valence = col_double(),  
        track_href = col_character(),  
        duration_ms = col_integer()  
    ))
```

Agora, iremos unir os datasets de interesse:

```
complete_data = (join.dplyr %>% inner_join(all_charts, audio_features, by = c("url" = "track_href")))

## Warning: One or more parsing issues, see 'problems()' for details
```

Filtramos os dados desejados para esta análise pela região/país (Brasil) e arranjamos os dados de modo que os dados sejam ordenados, em modo decrescente, a partir da data.

```
brazil_music = complete_data %>%
  select(title, date, artist, region, danceability, loudness, speechiness, instrumentalness, valence, d)
  arrange(desc(date)) %>%
  filter(region == "Brazil")
```

Ao termos uma visão geral dos dados, vemos que o ranking que foi coletado mais recentemente data do dia 31 de julho de 2021 e que a duração das músicas está em um formato não muito intuitivo (milissegundos), assim como instrumentalness, com valores em formato exponencial.

```
glimpse(brazil_music)
```

```
## Rows: 417,152
## Columns: 10
## $ title          <chr> "Beggin'", "Smells Like Teen Spirit", "Se Joga no Pas~
## $ date           <date> 2021-07-31, 2021-07-31, 2021-07-31, 2021-07-31, 2021-
## $ artist          <chr> "Måneskin", "Malia J", "Brisa Star, Thiago Jhonathan ~
## $ region          <chr> "Brazil", "Brazil", "Brazil", "Brazil", "Brazil", "Br~
## $ danceability    <dbl> 0.714, 0.193, 0.654, 0.736, 0.708, 0.931, 0.668, 0.79-
## $ loudness         <dbl> -4.808, -9.343, -3.052, -3.325, -7.082, -6.487, -4.40-
## $ speechiness      <dbl> 0.0504, 0.0382, 0.0467, 0.0323, 0.0434, 0.0851, 0.056-
## $ instrumentalness <dbl> 0.00e+00, 0.00e+00, 1.37e-05, 0.00e+00, 1.06e-02, 0.0-
## $ valence          <dbl> 0.5890, 0.0956, 0.8660, 0.4800, 0.9620, 0.9390, 0.907-
## $ duration_ms       <int> 211560, 239000, 200000, 189127, 196371, 120231, 15717-
```

Agora, iniciaremos efetivamente a nossa análise.

Não é incomum ouvir de pessoas ligadas ao mundo da música, que o que se produz hoje em dia, no meio musical, segue um padrão de indústria, como uma montagem perfeitamente arquitetada, sem possibilidade de mudanças. Nesta análise, buscamos compreender do gosto musical dos(as) brasileiro(as) nos últimos anos (2017-2021), e identificar se há, realmente, padrões muito evidentes em cada uma das características musicais (audio features) presentes nas canções que fizeram parte dos rankings do Spotify.

No princípio desta pesquisa, estávamos curiosos para saber se poderíamos extrair informações que nos permitissem comparar os dados das características musicais entre os países, mas devido à complexidade aliada à falta de poder de processamento da máquina utilizada para a análise, diminuímos o escopo para o Brasil. Portanto, passamos a desejar conhecer as características musicais em comum das músicas que chegaram ao ranking das mais ouvidas no Brasil nos anos de 2017 até 2021.

Abaixo, obtemos apenas os dados referentes ao Brasil e arranjamos os dados para que possamos visualizar qual foi a data do ranking mais recente:

```
brazil_top_charts = brazil_music %>%
  select(title, date, artist, danceability, loudness, speechiness, instrumentalness, valence, duration)
  mutate(date = ymd(date))
```

Vamos visualizar uma prévia para termos ideia de como os dados que iremos lidar estão. Podemos observar, por exemplo, que o ranking mais recente data de 31 de julho de 2021 e que a duração das músicas foi medida em milissegundos.

```
glimpse(brazil_top_charts)
```

```
## Rows: 417,152
## Columns: 9
## $ title           <chr> "Beggin'", "Smells Like Teen Spirit", "Se Joga no Pas-
## $ date            <date> 2021-07-31, 2021-07-31, 2021-07-31, 2021-07-31, 2021-
## $ artist          <chr> "Måneskin", "Malia J", "Brisa Star, Thiago Jhonathan ~
## $ danceability    <dbl> 0.714, 0.193, 0.654, 0.736, 0.708, 0.931, 0.668, 0.79-
## $ loudness        <dbl> -4.808, -9.343, -3.052, -3.325, -7.082, -6.487, -4.40-
## $ speechiness     <dbl> 0.0504, 0.0382, 0.0467, 0.0323, 0.0434, 0.0851, 0.056-
## $ instrumentalness <dbl> 0.00e+00, 0.00e+00, 1.37e-05, 0.00e+00, 1.06e-02, 0.0-
## $ valence          <dbl> 0.5890, 0.0956, 0.8660, 0.4800, 0.9620, 0.9390, 0.907-
## $ duration_ms      <int> 211560, 239000, 200000, 189127, 196371, 120231, 15717-
```

A princípio, investigamos a duração das músicas e verificamos que não há nenhum dado nulo, logo podemos prosseguir sem maiores preocupações.

```
sum(is.na(brazil_top_charts$duration_ms))
```

[1] 0

Como os dados originais são em milissegundos, poderíamos ter dificuldade em avaliá-los. Neste caso, efetuaremos uma mutação nos dados, passando a duração das músicas para uma unidade mais comum no nosso contexto: a dos minutos. Esses dados ficarão em uma nova coluna chamada “duration”:

```
brazil_top_charts <- brazil_top_charts %>%  
  mutate(duration = duration ms / 60000)
```

Agora, temos os dados em minutos:

```
glimpse(brazil_top_charts$duration)
```

```
## num [1:417152] 3.53 3.98 3.33 3.15 3.27 ...
```

Também visualizaremos os valores máximo, mínimo e mediano para saber se há valores muito distantes do padrão. Pode-se notar que o valor máximo está bastante distante do valor mínimo, o que pode indicar grande espalhamento dos dados.

```
glimpse(max(brazil_top_charts$duration))

##   num 22.6

glimpse(min(brazil_top_charts$duration))

##   num 0.607

glimpse(median(brazil_top_charts$duration))

##   num 3.08
```

Para não ter que continuar usando a as funções month() e year() sempre que precisarmos realizar uma análise temporal, adicionaremos as colunas correspondentes nos datasets:

```
brazil_top_charts <- brazil_top_charts %>% mutate(month = case_when(month(date) == 1 ~ 1, month(date) == 12 ~ 12))

brazil_top_charts <- brazil_top_charts %>% mutate(year = case_when(year(date) == 2017 ~ 2017, year(date) == 2018 ~ 2018))
```

Como temos uma quantidade gigantesca de observações, devido ao caráter contínuo dos dados de duração das músicas, iremos agrupar os dados em 5 faixas de valores, baseando-nos também nos valores máximo e mínimo há pouco exibidos: (0.5, 4:00), [4:00, 7:00); [7:00, 10:00), [10:00, 13:00), [13:00, 22.6]. Essas faixas poderão ser interpretadas como: “muito curta”, “normal”, “acima do normal”, “longa” e “muito longa”, respectivamente.

```
brazil_top_charts <- brazil_top_charts %>% mutate(duration_group = case_when(duration >= 0 & duration < 4 ~ "muito curta", duration >= 4 & duration < 7 ~ "curta", duration >= 7 & duration < 10 ~ "normal", duration >= 10 & duration < 13 ~ "longa", duration >= 13 ~ "muito longa"))
```

É evidente que, das mais de 7800 músicas que chegaram em rankings de mais ouvidas, a grande maioria se encontra na faixa de valores de músicas mais curtas. Dado o que obtemos até aqui, não esperamos que hajam grandes mudanças na duração das faixas ao longo dos anos, logo, limitamo-nos a considerar que as canções, em geral, possuem durações mais curtas (com mediana 3.08).

```
duration_groups <- group_by(brazil_top_charts, duration_group)

a = distinct((duration_groups %>%
  select(title, artist, duration_group)))

nrow(a)

## [1] 7806

count(a)
```

```
## # A tibble: 5 x 2
## # Groups: duration_group [5]
##   duration_group     n
##   <chr>             <int>
## 1 1                 6399
## 2 2                 1315
## 3 3                  76
## 4 4                  12
## 5 5                  4
```

Agora, estamos interessados em saber se existe um padrão nítido para a característica “instrumentalness”.

Primeiro, ao visualizarmos os dados, verificamos que existem valores em formato exponencial, o que, do ponto de vista interpretativo, não é tão intuitivo. Apesar disso, temos um ponto positivo: verificamos que não existe nenhum valor nulo que possa prejudicar a análise.

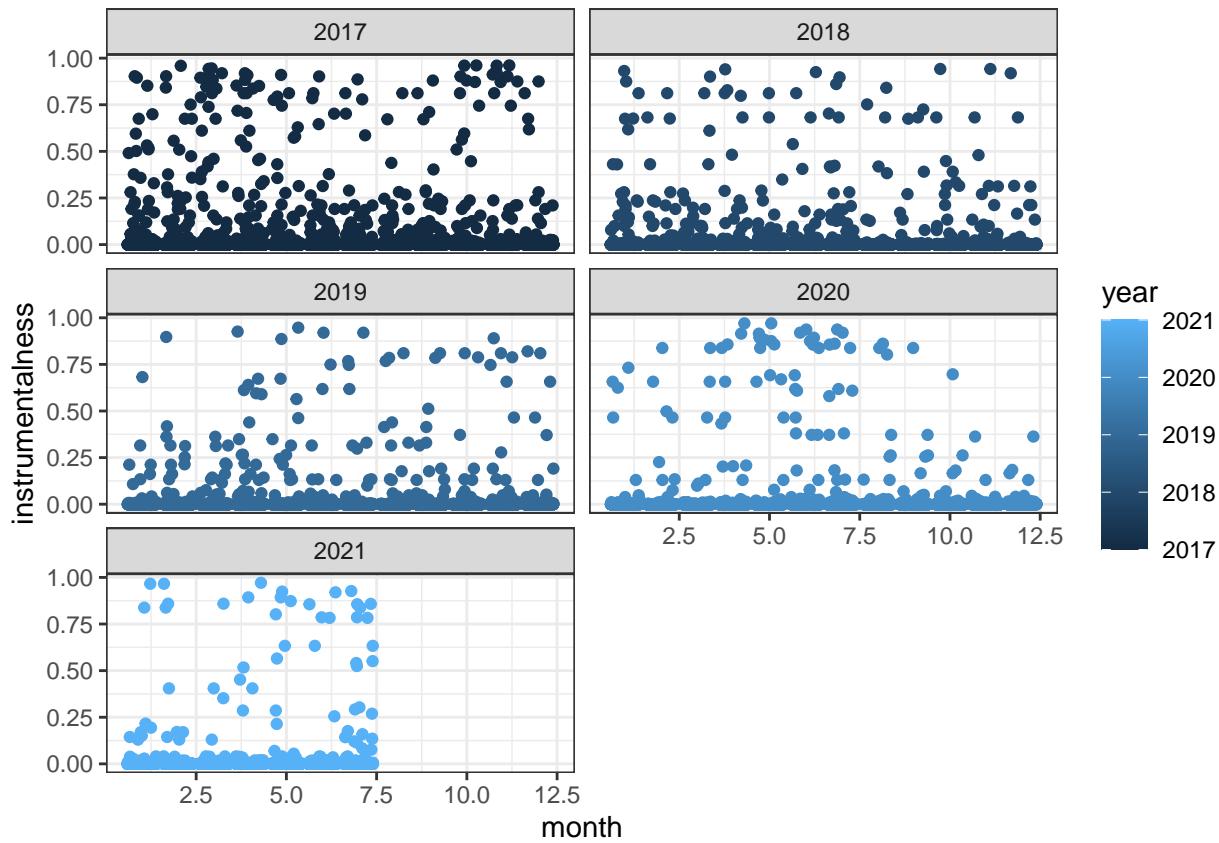
```
glimpse(brazil_top_charts %>% select(instrumentalness))
```

```
## Rows: 417,152
## Columns: 1
## $ instrumentalness <dbl> 0.00e+00, 0.00e+00, 1.37e-05, 0.00e+00, 1.06e-02, 0.0~  
  
sum(is.na(brazil_top_charts$instrumentalness))
```

[1] 0

Ao visualizarmos os dados de instrumentalness nos gráficos abaixo, vemos que apesar de haver uma grande quantidade de dados espalhados, muitos, inclusive, na faixa que indica uma grande possibilidade da música ser instrumental, parece haver, em todos os anos, uma concentração de valores no intervalo de 0 a 0.135, ou seja, na faixa que indica baixa presença de características instrumentais.

```
distinct_instrum <- distinct((brazil_top_charts %>%
  select(title, artist, instrumentalness, month, year)))  
  
distinct_instrum %>%
  ggplot(aes(x = month, y = instrumentalness, color = year)) +
  facet_wrap(~year, ncol = 2) +
  geom_jitter()
```

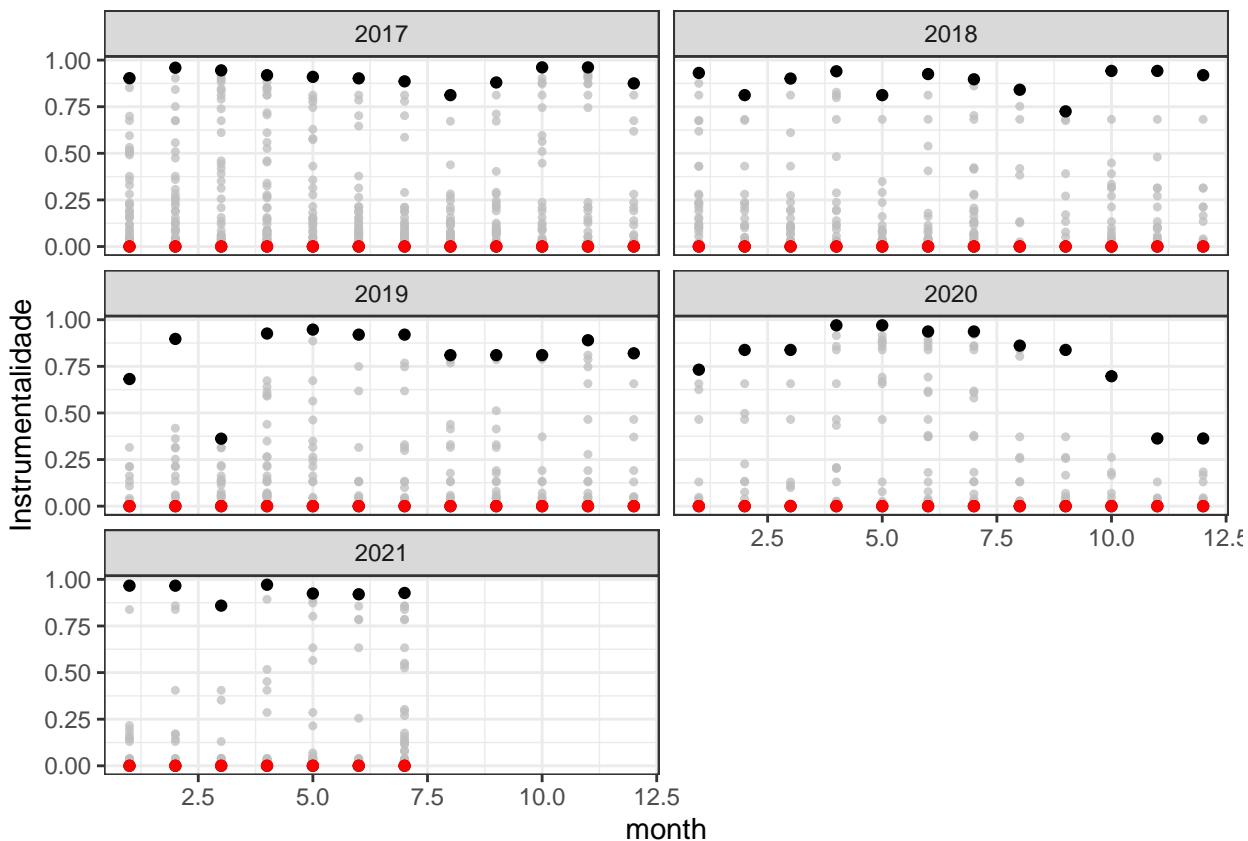


Aqui, com a primeira instrução, passaremos a exibir os valores da característica de estudo em forma decimal. Na segunda instrução, agrupamos os dados por ano e mês e sumarizamos os dados, para que possamos obter os valores máximo, mínimo e mediano dos dados da característica ao longo dos anos.

A partir de agora, exibiremos como essa característica esteve presente nas músicas ao longo dos anos. Com os sumários, apesar de perdemos informações, ganhamos em escalabilidade. Isso nos permite observar que os valores da mediana, em todos os anos, está bastante concentrada na faixa de valores igual ou muito próxima a 0, ou seja, pelo menos 50% das observações possuem poucas características instrumentais. Aparentemente, as músicas “mais intrumentais” que chegaram no ranking das mais ouvidas são mais raras. As faixas acima

de 0.25 possuem uma dispersão muito grande. O gráfico também indica que, ao longo dos anos, surgiram lacunas bem grandes entre uma concentração e outra.

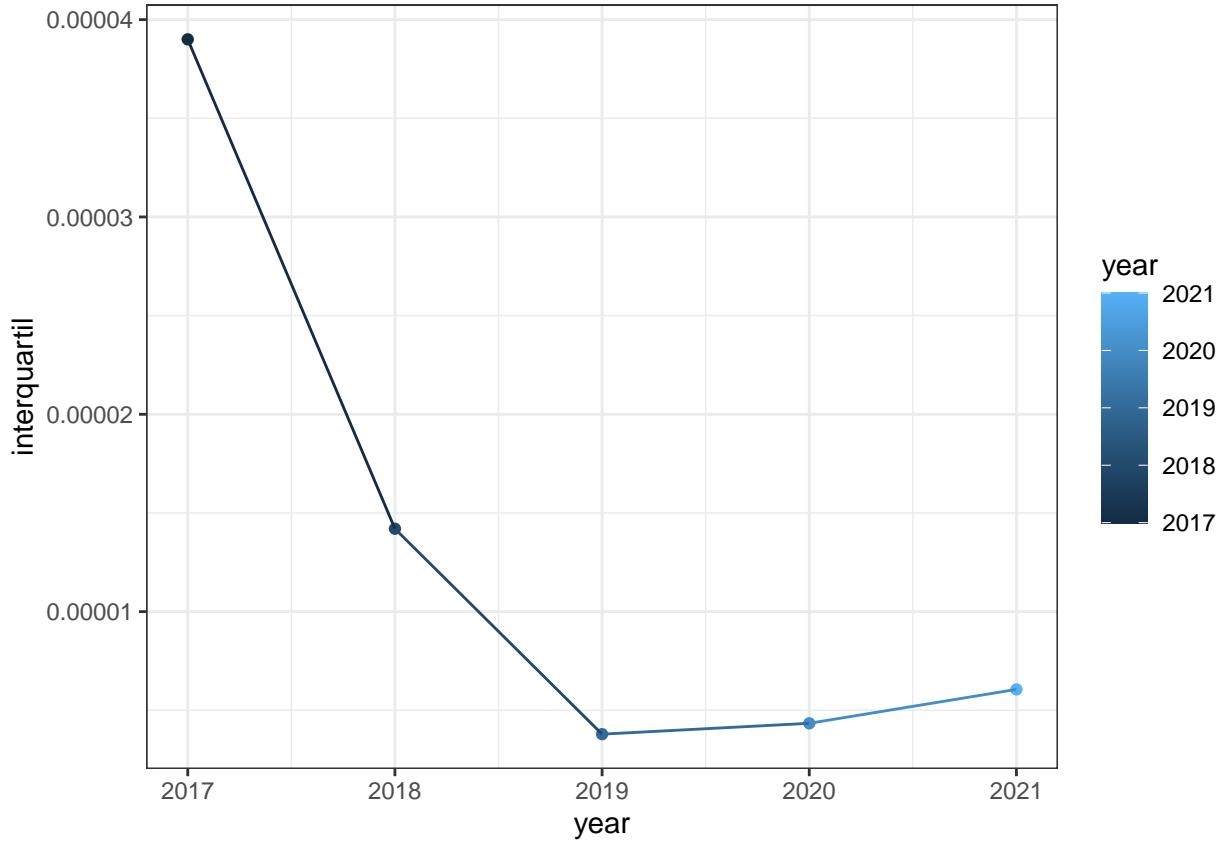
```
distinct_instrum %>%
  ggplot(aes(x = month, y = instrumentalness)) +
  facet_wrap(~year, ncol = 2) +
  geom_point(alpha = .75, size = .9, color = "grey") +
  geom_point(data = sumarios, aes(y = instrumentalness_max_anual)) +
  geom_point(data = sumarios, aes(y = instrumentalness_min_anual)) +
  geom_point(data = sumarios, aes(y = instrumentalness_median), color = "red") +
  labs(y = "Instrumentalidade")
```



Para observarmos o espalhamento/variação dos dados, usaremos a distância interquartil(distância entre o 25-percentil e o 75-percentil), que é menos afetada por pontos extremos. Através do gráfico, podemos afirmar que, em um quadro geral, não houve mudança. A grande maioria das músicas mais ouvidas é formada por músicas pouco instrumentais.

```
variacao = distinct_instrum %>%
  group_by(year) %>%
  summarise(amplitude = max(instrumentalness, na.rm = TRUE) - min(instrumentalness, na.rm = TRUE), interquartil = quantile(instrumentalness, na.rm = TRUE)[3] - quantile(instrumentalness, na.rm = TRUE)[1])

variacao %>%
  ggplot(aes(x = year, y = interquartil, color = year)) +
  geom_point() +
  geom_line()
```



A partir de agora, ao obtermos uma visão geral da característica “valence”/valência, seguindo o mesmo padrão da análise anterior, temos que os dados estão em formato decimal e que também não há dados nulos.

```
glimpse(brazil_top_charts %>% select(valence))

## # Rows: 417,152
## # Columns: 1
## $ valence <dbl> 0.5890, 0.0956, 0.8660, 0.4800, 0.9620, 0.9390, 0.9070, 0.8170~

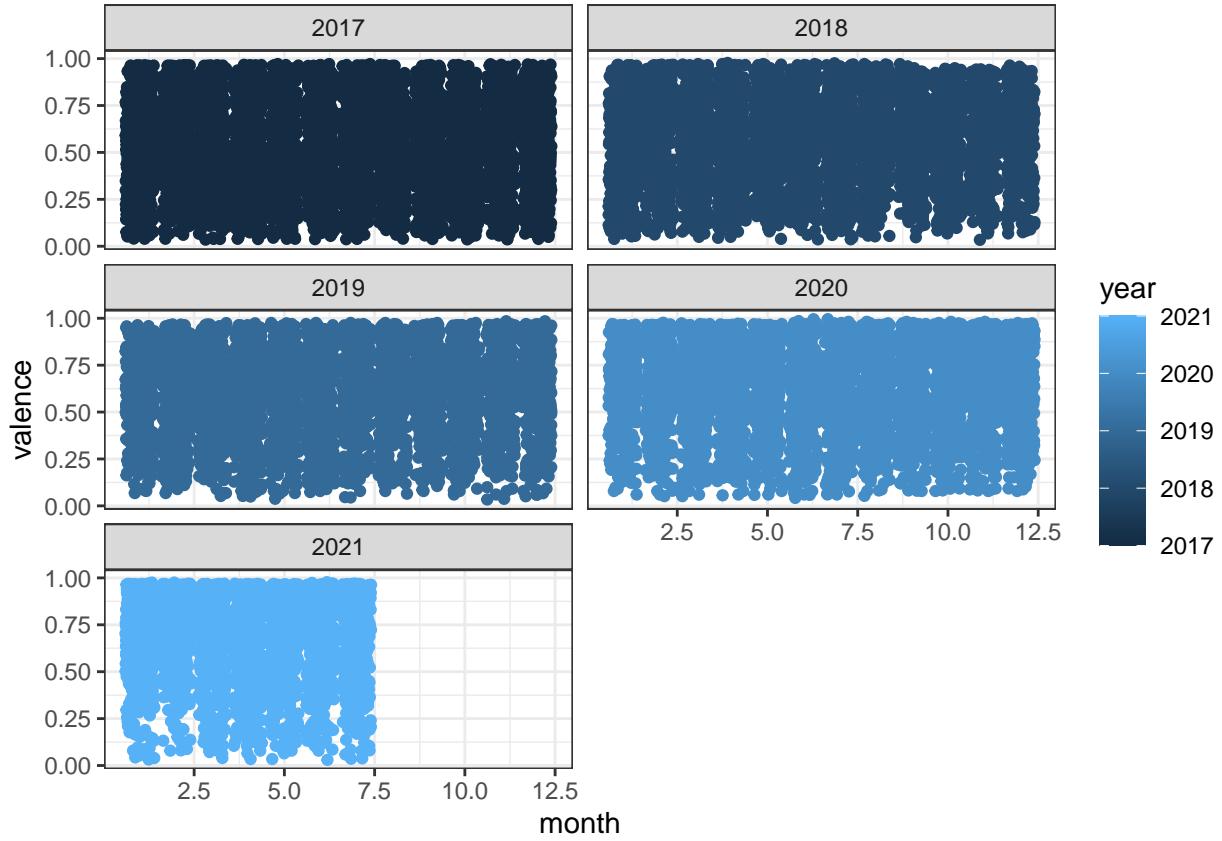
sum(is.na(brazil_top_charts$valence))

## [1] 0
```

A princípio, não é possível afirmarmos algo ao visualizar os pontos do gráfico dessa característica, já que praticamente todo o intervalo de valores parece ter alguma concentração.

```
distinct_valence <- distinct((brazil_top_charts %>%
  select(title, artist, valence, month, year)))

distinct_valence %>%
  ggplot(aes(x = month, y = valence, color = year)) +
  facet_wrap(~year, ncol = 2) +
  geom_jitter()
```



Aqui, agrupamos os dados por ano e mês e sumarizamos os dados, para que possamos obter os valores máximo, mínimo e mediano dos dados da característica ao longo dos anos.

```
sumarios = distinct_valence %>%
  group_by(year, month) %>%
  summarise(valence_max_anual = max(valence),
            valence_min_anual = min(valence),
            valence_median = median(valence),
            .groups = "drop")
```

Aqui, confirmamos o que vimos no gráfico acima: não há um padrão específico indicando se existe preferência tão nítida por músicas com baixa valência, ou seja, as que passam sentimentos negativos - tristeza, melancolia, raiva, etc -; ou por músicas com alta valência, ou seja, que passam a sensação de alegria, positividade, etc. Também não existe grande variabilidade entre os valores máximos, nem entre os valores mínimos.

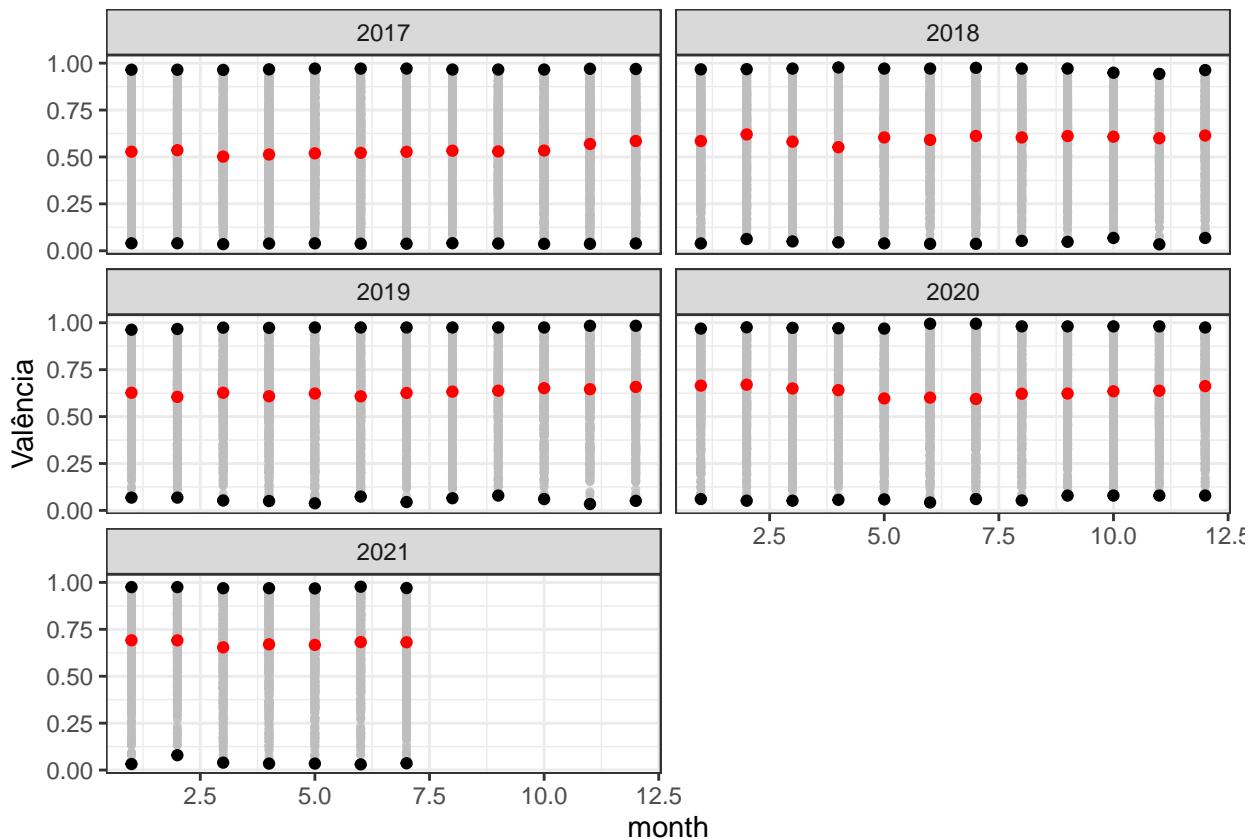
Nessa amostra, que considera um intervalo aproximado de 5 anos, o que se pode observar é que a variância aumentou levemente, passando da faixa de valores próxima logo acima de 0.5 para a faixa de valores acima de 0.62; além disso, o valor da variância de fevereiro até dezembro segue a mesma faixa de valores de janeiro, com mudanças mínimas. Apesar disso, não podemos considerar uma faixa de valores específica como uma tendência dos hits ouvidos no Brasil.

```
distinct_valence %>%
  ggplot(aes(x = month, y = valence)) +
  facet_wrap(~year, ncol = 2) +
  geom_point(alpha = .75, size = .9, color = "gray") +
  geom_point(data = sumarios, aes(y = valence_max_anual)) +
```

```

geom_point(data = sumarios, aes(y = valence_min_anual)) +
geom_point(data = sumarios, aes(y = valence_median), color = "red") +
labs(y = "Valência")

```



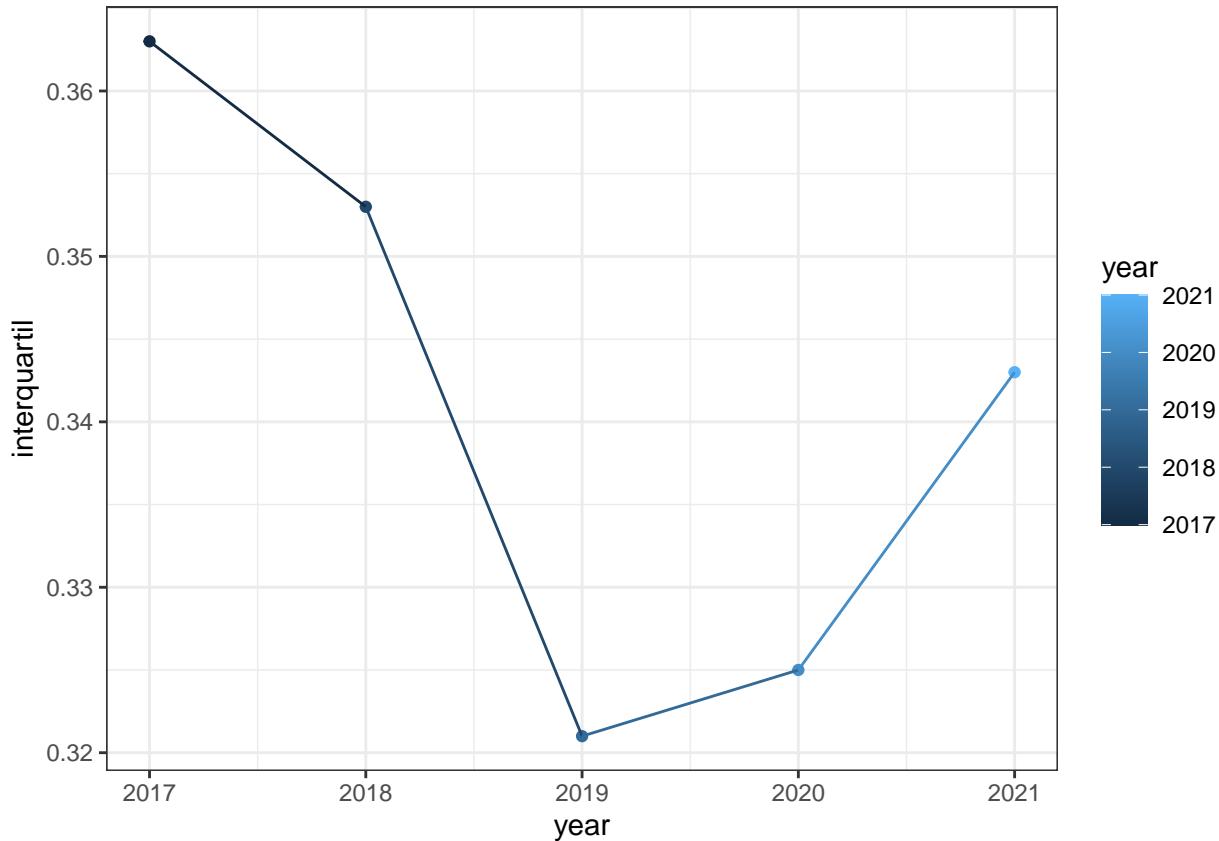
Por fim, a distância interquartil só confirma a alta variabilidade dos dados e ausência de padrão nítido.

```

variacao = distinct_valence %>%
  group_by(year) %>%
  summarise(amplitude = max(valence, na.rm = TRUE) - min(valence, na.rm = TRUE), interquartil = IQR(valence))

variacao %>%
  ggplot(aes(x= year, y = interquartil, color = year)) +
  geom_point() +
  geom_line()

```



Passamos para a próxima característica investigada: “danceability”, ou seja, o valor que representa se as músicas são dançáveis. Seguimos o mesmo padrão da análise das características anteriores, inicialmente, tendo uma visão geral dos valores dos dados e verificando se não existem valores nulos. Como não há valores nulos, seguimos sem nenhuma alteração.

```
glimpse(brazil_top_charts %>% select(danceability))
```

```
## Rows: 417,152  
## Columns: 1  
## $ danceability <dbl> 0.714, 0.193, 0.654, 0.736, 0.708, 0.931, 0.668, 0.797, 0~
```

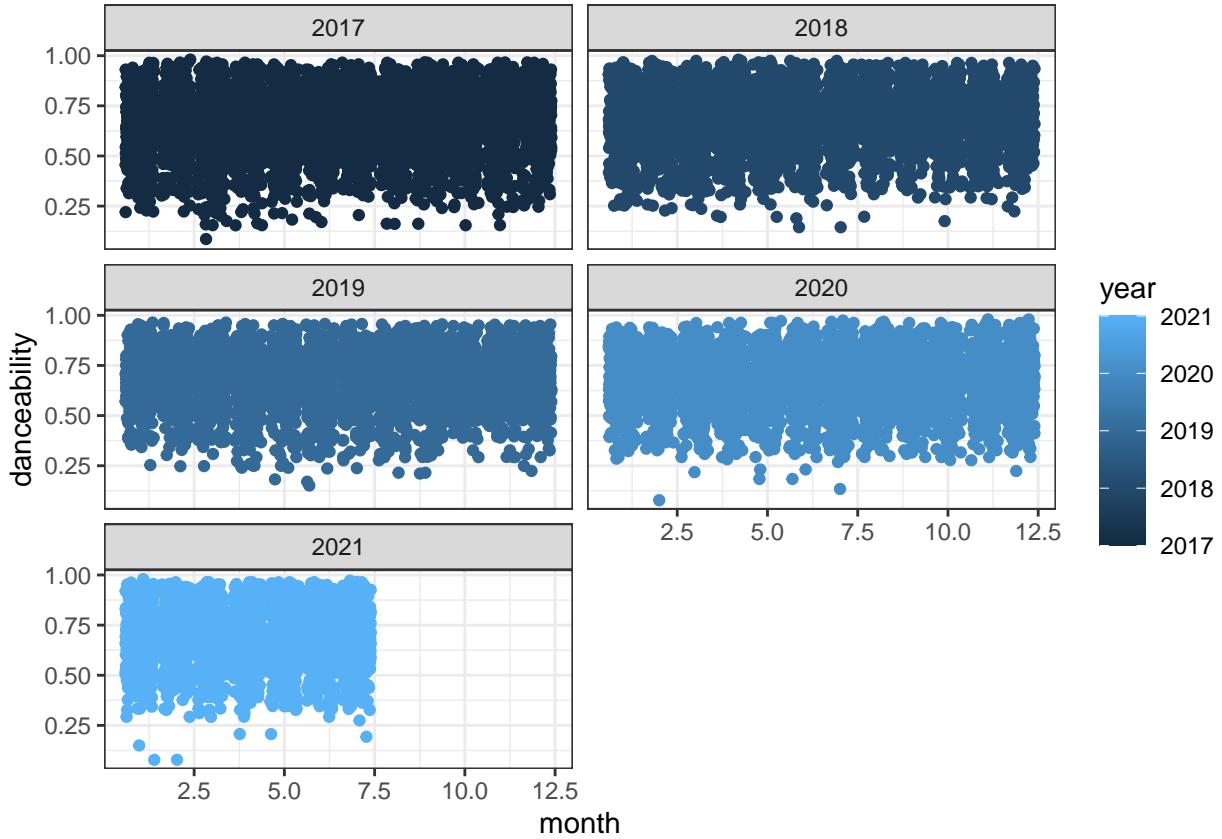
```
sum(is.na(brazil_top_charts$danceability))
```

[1] 0

A princípio, podemos afirmar que a faixa de valores abaixo de 0.375 possui menor concentração de observações.

```
distinct_danceability <- distinct((brazil_top_charts %>%
  select(title, artist, danceability, month, year))

distinct_danceability %>%
  ggplot(aes(x = month, y = danceability, color = year)) +
  facet_wrap(~year, ncol = 2) +
  geom_jitter()
```



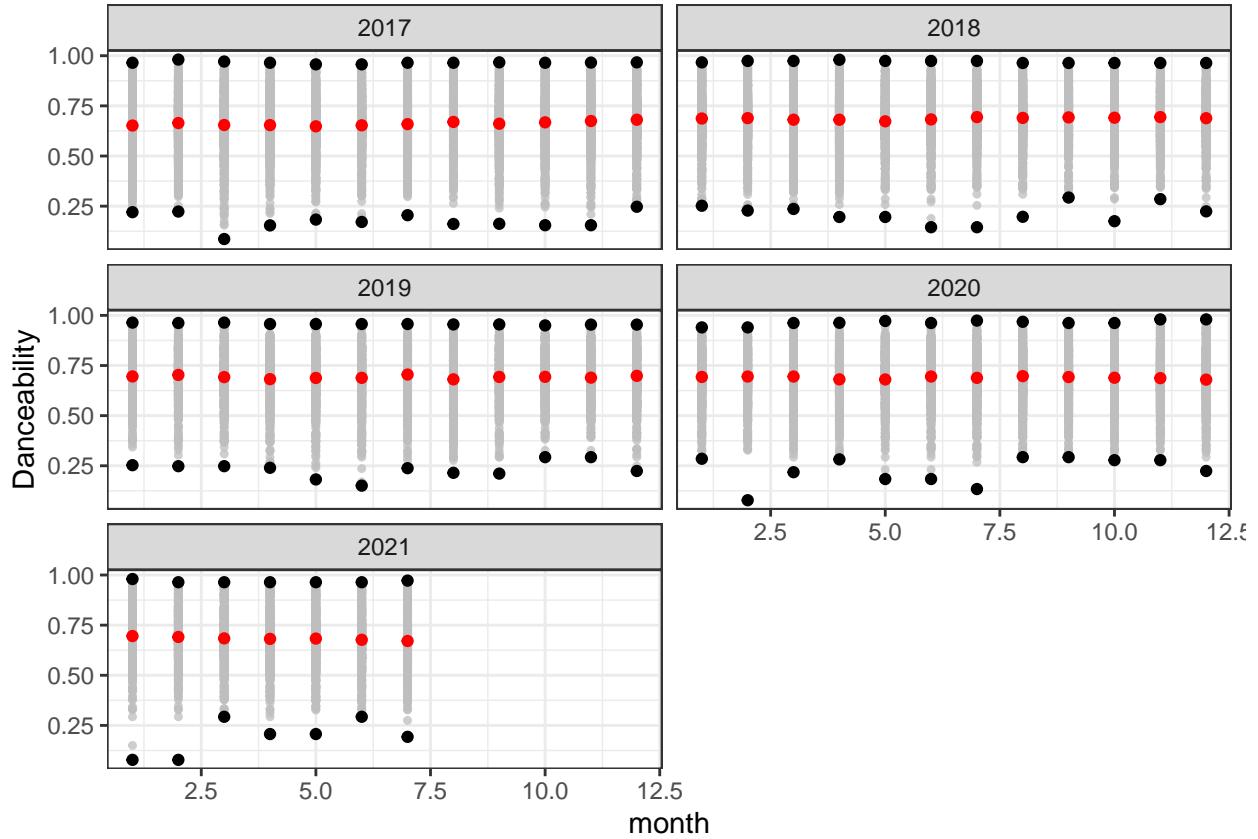
Aqui, agrupamos os dados por ano e mês e sumarizamos os dados, para que possamos obter os valores máximo, mínimo e mediano dos dados da característica ao longo dos anos.

```
sumarios = distinct_danceability %>%
  group_by(year, month) %>%
  summarise(danceability_max_anual = max(danceability),
            danceability_min_anual = min(danceability),
            danceability_median = median(danceability),
            .groups = "drop")
```

Aqui, existe uma impressão de que há uma preferência por faixas mais dançantes, já que a mediana, na maioria dos meses, se aproxima bastante do valor 0.75, porém é preciso ter cuidado com esta análise. Precisaremos, neste caso, analisar a distância interquartil para observarmos a variabilidade dos dados.

Também podemos verificar que a variabilidade entre os valores máximos é muito pequena, enquanto os valores mínimos variam bastante e, em várias ocasiões, com curvas muito acentuadas.

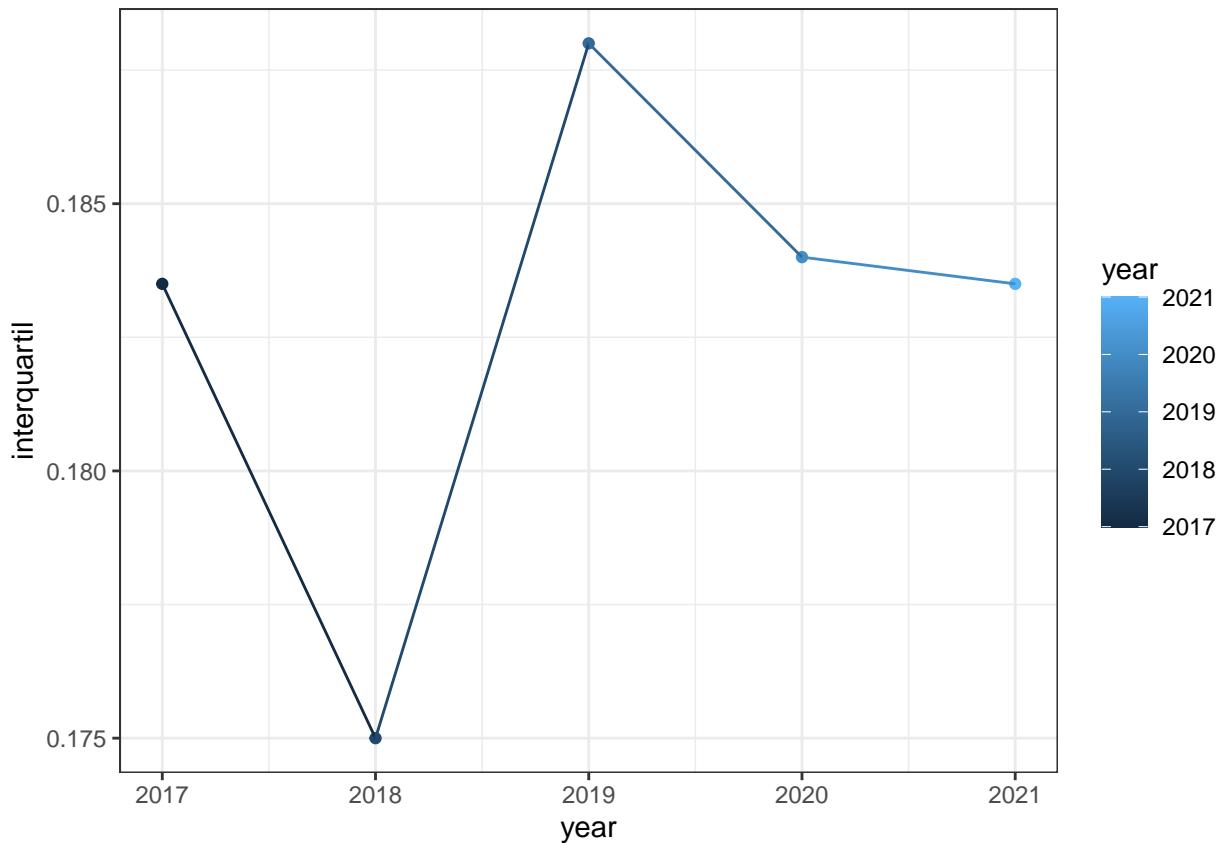
```
distinct_danceability %>%
  ggplot(aes(x = month, y = danceability)) +
  facet_wrap(~year, ncol = 2) +
  geom_point(alpha = .75, size = .9, color = "grey") +
  geom_point(data = sumarios, aes(y = danceability_max_anual)) +
  geom_point(data = sumarios, aes(y = danceability_min_anual)) +
  geom_point(data = sumarios, aes(y = danceability_median), color = "red") +
  labs(y = "Danceability")
```



Observando a distância interquartil, vemos que os dados variam bastante, dessa forma, não há uma tendência muito clara para danceability.

```
variacao = distinct_danceability %>%
  group_by(year) %>%
  summarise(amplitude = max(danceability, na.rm = TRUE) - min(danceability, na.rm = TRUE), interquartil =
  IQR(danceability))

variacao %>%
  ggplot(aes(x= year, y = interquartil, color = year)) +
  geom_point() +
  geom_line()
```



Passando para a próxima característica, nos deparamos com “speechiness”, ou seja, o valor que representa a presença de palavras nas músicas.

Abaixo, a visão geral dos valores dos dados que, assim como nas características anteriores, possuem formato decimal.

“Speechiness” também não possui valores nulos.

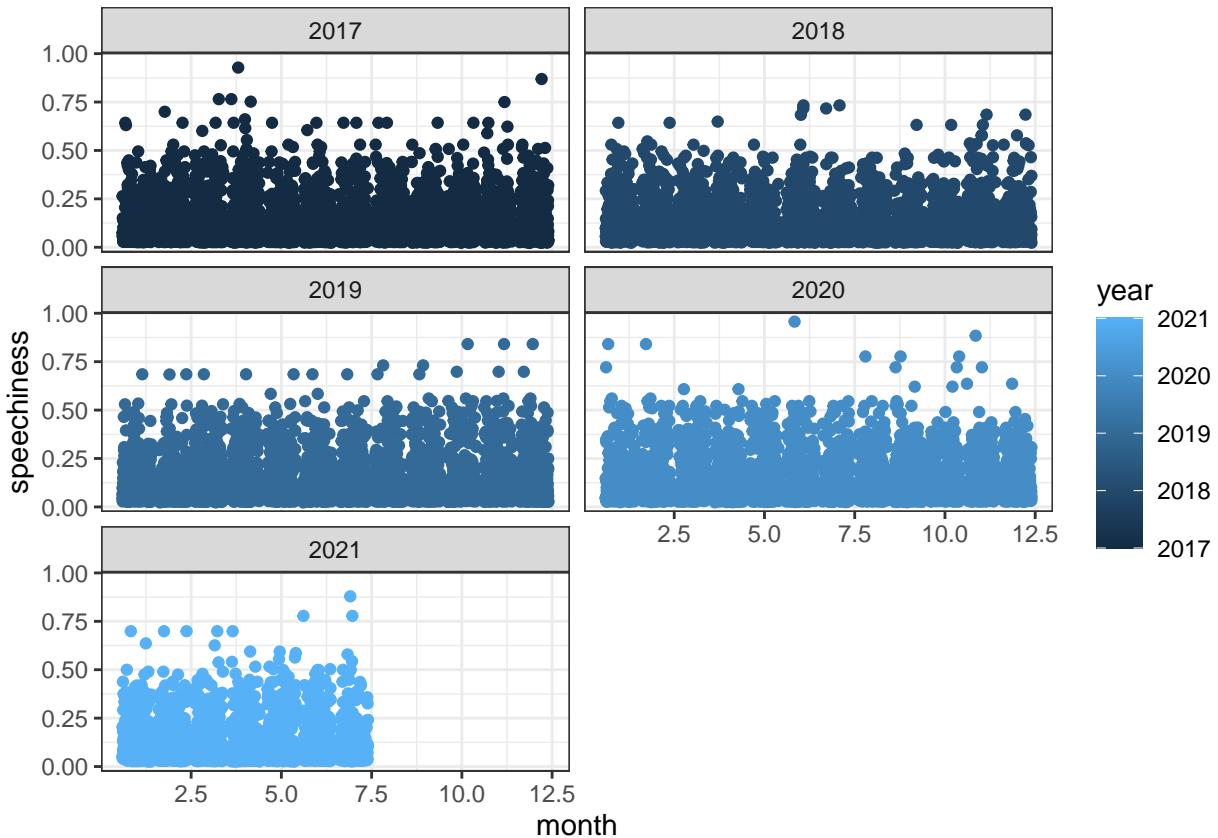
```
glimpse(brazil_top_charts %>% select(speechiness))
```

```
## Rows: 417,152
## Columns: 1
## $ speechiness <dbl> 0.0504, 0.0382, 0.0467, 0.0323, 0.0434, 0.0851, 0.0564, 0.~  
sum(is.na(brazil_top_charts$speechiness))  
## [1] 0
```

Observando o gráfico abaixo, temos o indício de que grande parte dos valores está concentrada na faixa entre 0 e 0.375, o que pode nos indicar que as músicas que chegam nos rankings de mais ouvidas possuem um vocabulário não tão diverso.

```
distinct_speechiness <- distinct((brazil_top_charts %>%  
  select(title, artist, speechiness, month, year)))
```

```
distinct_speechiness %>%
  ggplot(aes(x = month, y = speechiness, color = year)) +
  facet_wrap(~year, ncol = 2) +
  geom_jitter()
```

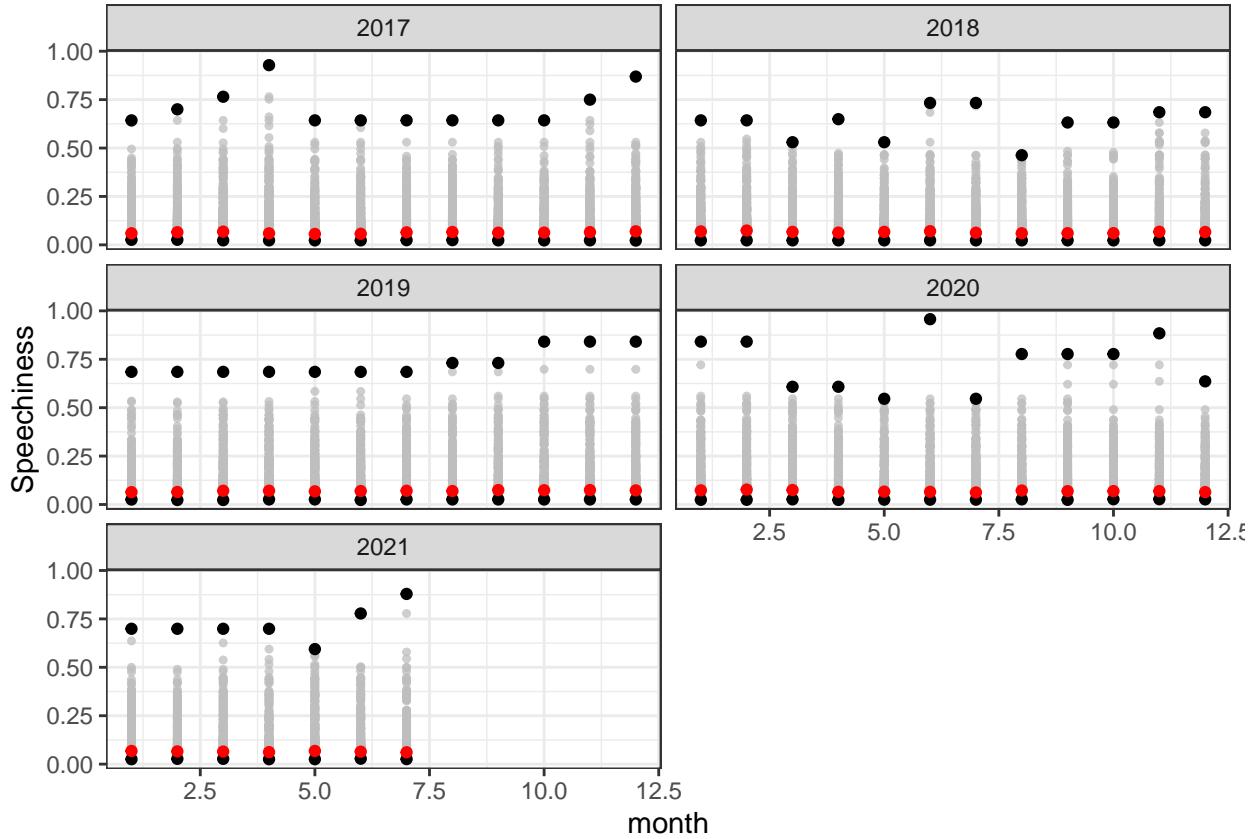


Sumarizamos os valores para podermos compará-los ao longo do tempo:

```
sumarios = distinct_speechiness %>%
  group_by(year, month) %>%
  summarise(speechiness_max_anual = max(speechiness),
            speechiness_min_anual = min(speechiness),
            speechiness_median = median(speechiness),
            .groups = "drop")
```

Observando os dados ao longo dos anos, vemos que 50% dos dados encontra-se na faixa de valores entre 0 e 0.125, o que confirma a suspeita dessas músicas apresentarem um vocabulário limitado.

```
distinct_speechiness %>%
  ggplot(aes(x = month, y = speechiness))+
  facet_wrap(~year, ncol = 2) +
  geom_point(alpha = .75, size = .9, color = "grey") +
  geom_point(data = sumarios, aes(y = speechiness_max_anual)) +
  geom_point(data = sumarios, aes(y = speechiness_min_anual)) +
  geom_point(data = sumarios, aes(y = speechiness_median), color = "red") +
  labs(y = "Speechiness")
```

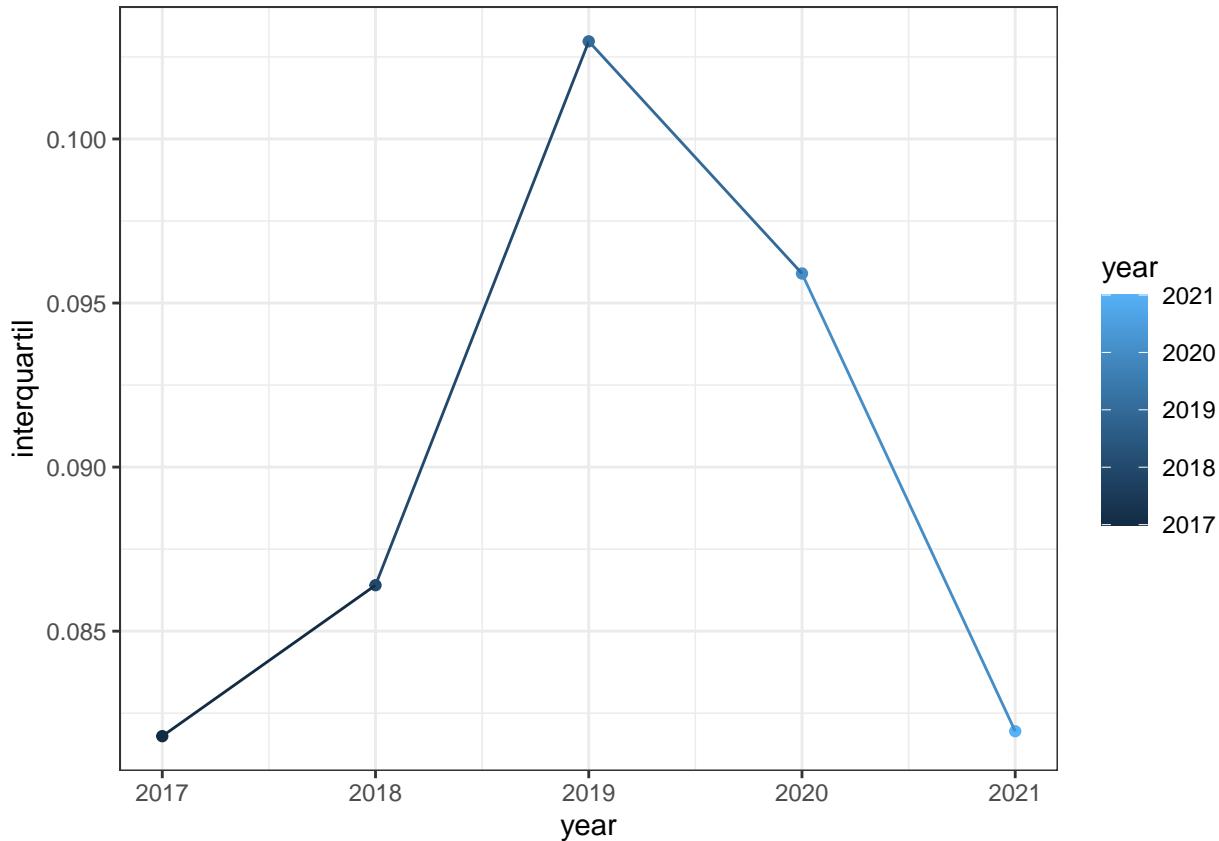


Ao observarmos a dispersão dos dados, vemos que há pouca variação nos dados entre o 25-percentil e o 75-percentil. Assim, temos a certeza da tendência das músicas populares usarem poucas palavras.

```

variacao = distinct_speechiness %>%
  group_by(year) %>%
  summarise(amplitude = max(speechiness, na.rm = TRUE) - min(speechiness, na.rm = TRUE), interquartil = iqr(speechiness))

variacao %>%
  ggplot(aes(x= year, y = interquartil, color = year)) +
  geom_point() +
  geom_line()
  
```



Por fim, investigaremos “loudness”, ou seja, o valor que representa o volume das músicas. Vale lembrar o intervalo de valores dessa característica fica entre -60 e 0, ou seja, quanto mais próximo de 0, maior é o volume em decibéis.

Neste caso, também temos dados não nulos.

```
glimpse(brazil_top_charts %>% select(loudness))
```

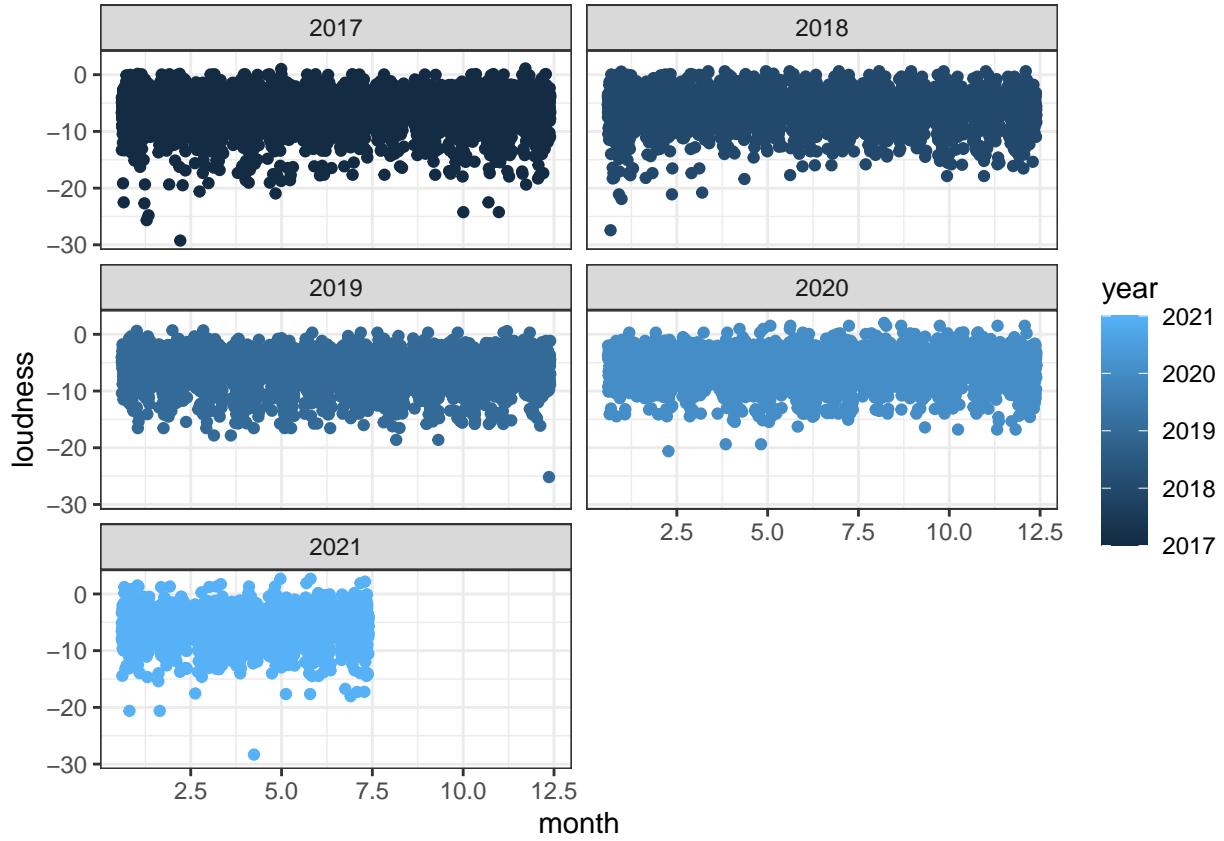
```
## Rows: 417,152  
## Columns: 1  
## $ loudness <dbl> -4.808, -9.343, -3.052, -3.325, -7.082, -6.487, -4.404, -6.57~
```

```
sum(is.na(brazil_top_charts$loudness))
```

```
## [1] 0
```

Podemos observar que os valores entre -10 e -5 apresentam uma grande concentração de valores;

```
distinct_loudness <- distinct((brazil_top_charts %>%
  select(title, artist, loudness, month, year)))  
  
distinct_loudness %>%
  ggplot(aes(x = month, y = loudness, color = year)) +
  facet_wrap(~year, ncol = 2) +
  geom_jitter()
```

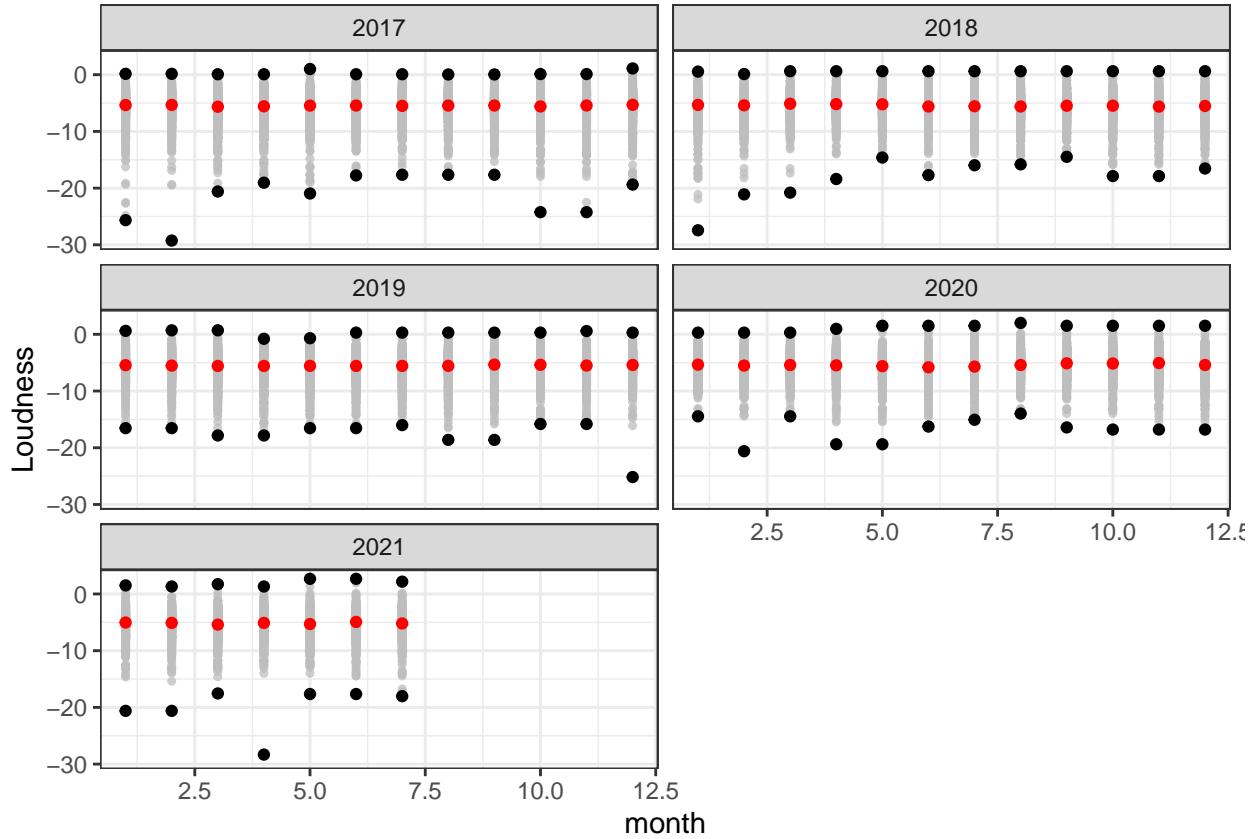


Seguindo o padrão já utilizado, sumarizamos os dados:

```
sumarios = distinct_loudness %>%
  group_by(year, month) %>%
  summarise(loudness_max_anual = max(loudness),
            loudness_min_anual = min(loudness),
            loudness_median = median(loudness),
            .groups = "drop")
```

Em todos os anos, 50% dos dados encontram-se concentrados na faixa de valores próximos a -5 e 0, o que aponta uma tendência de que as músicas sejam altas.

```
distinct_loudness %>%
  ggplot(aes(x = month, y = loudness)) +
  facet_wrap(~year, ncol = 2) +
  geom_point(alpha = .75, size = .9, color = "grey") +
  geom_point(data = sumarios, aes(y = loudness_max_anual)) +
  geom_point(data = sumarios, aes(y = loudness_min_anual)) +
  geom_point(data = sumarios, aes(y = loudness_median), color = "red") +
  labs(y = "Loudness")
```

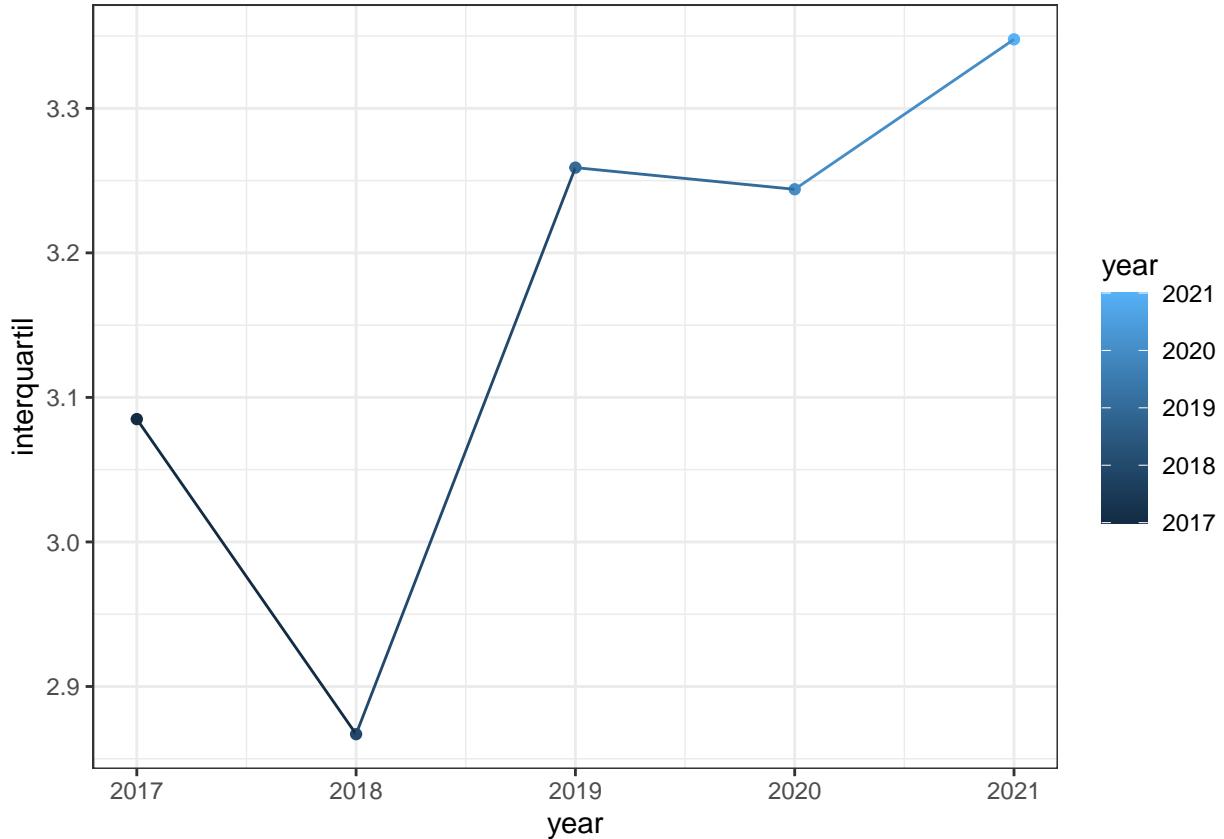


Verificando o intervalo interquartil, pouco maior que 3, em sua maioria, observamos que não há muito espalhamento nos dados, ou seja, se juntarmos essa informação ao que obtemos nos gráficos das medianas, podemos afirmar que as músicas mais ouvidas possuem a tendência de ter um valor alto em decibéis.

```

variacao = distinct_loudness %>%
  group_by(year) %>%
  summarise(amplitude = max(loudness, na.rm = TRUE) - min(loudness, na.rm = TRUE), interquartil = IQR(loudness))

variacao %>%
  ggplot(aes(x= year, y = interquartil, color = year)) +
  geom_point() +
  geom_line()
  
```



Finalizamos a nossa análise com a seguinte conclusão: as músicas mais ouvidas no Brasil, durante o intervalo de 2017 até julho de 2021, apresentaram a tendência a serem músicas com duração curta, com pouco uso de palavras, poucas características musicais e volume em decibéis alto. Por uma questão de cronograma para o trabalho, nem todas as características foram analisadas, mas outras, como “mode” e “key”, a princípio consideradas para a análise, não nos forneciam nenhum padrão muito evidente, e, como requeriam um aprofundamento em conceitos de teoria musical, foram eliminadas da análise. Sugestões de próximos trabalhos são: 1 - considerar intervalos de tempo maiores para realizar essa mesma análise ao longo dos anos (uma década, por exemplo); 2 - investigar correlação entre primeiras posições do ranking e padrões de características musicais; 3 - investigar padrões de características em gêneros musicais específicos.