

# Documentação projeto 02 laboratoria

## CONTEXTO

Num mundo onde a **indústria musical** é extremamente competitiva e em constante evolução, a capacidade de tomar decisões baseadas em dados tornou-se um ativo inestimável.

Neste contexto, uma gravadora enfrenta o emocionante desafio de lançar um novo artista no cenário musical global. Felizmente, ela tem uma ferramenta poderosa em seu arsenal: um extenso conjunto de dados do Spotify com informações sobre as músicas mais ouvidas em 2023.

- A gravadora levantou uma série de hipóteses sobre o que faz uma música seja mais ouvida. Essas hipóteses incluem:
- Músicas com BPM (Batidas Por Minuto) mais altos fazem mais sucesso em termos de número de streams no Spotify.
- As músicas mais populares no ranking do Spotify também possuem um comportamento semelhante em outras plataformas, como a Deezer.
- A presença de uma música em um maior número de playlists está correlacionada com um maior número de streams.
- Artistas com um maior número de músicas no Spotify têm mais streams.
- As características da música influenciam o sucesso em termos de número de streams no Spotify.

Você deve validar (refutar ou confirmar) essas hipóteses através da análise de dados e fornecer recomendações estratégicas com base em suas descobertas. O objetivo principal desta análise é que a gravadora e o novo artista possam tomar decisões informadas que aumentem suas chances de alcançar o "sucesso".

## 1.3 Insumos

Este conjunto de dados contém dados sobre as músicas mais populares reproduzidas no Spotify em 2023. Os dados são divididos em 3 tabelas, a primeira com a performance de cada música no Spotify, a segunda com o seu desempenho em outras plataformas, como Deezer ou Apple Music, e a terceira com as características dessas músicas.

O conjunto de dados (dataset) está disponível para download neste link [dataset](#). Observe que é um arquivo compactado, portanto você terá que descompactá-lo para acessar os arquivos com os dados.

Abaixo, você pode consultar a descrição das variáveis que compõem as tabelas deste conjunto de dados:

### Trackinspotify

- **track\_id**: Identificador exclusivo da música. É um número inteiro de 7 dígitos que não se repete.
- **track\_name**: Nome da música.
- **\*artist(s)\_name\*\***: Nome do(s) artista(s) da música.

- **artist\_count**: Número de artistas que contribuíram na música.
- **released\_year**: Ano em que a música foi lançada.
- **released\_month**: Mês em que a música foi lançada.
- **released\_day**: Dia do mês em que a música foi lançada.
- **inspotifyplaylists**: Número de listas de reprodução do Spotify em que a música está incluída
- **inspotifycharts**: Presença e posição da música nas paradas do Spotify
- **streams**: Número total de streams no Spotify. Representa o número de vezes que a música foi ouvida.

### **Trackincompetition**

- **track\_id**: Identificador exclusivo da música. É um número inteiro de 7 dígitos que não se repete.
- **inappleplaylists**: número de listas de reprodução da Apple Music em que a música está incluída.
- **inapplecharts**: Presença e classificação da música nas paradas da Apple Music.
- **indeezerplaylists**: Número de playlists do Deezer em que a música está incluída.
- **indeezercharts**: Presença e posição da música nas paradas da Deezer.
- **inshazamcharts**: Presença e classificação da música nas paradas da Shazam.

### **Tracktechnicalinfo**

- **track\_id**: Identificador exclusivo da música. É um número inteiro de 7 dígitos que não se repete.
- **bpm**: Batidas por minuto, uma medida do tempo da música.
- **key**: Tom musical da música.
- **mode**: Modo de música (maior ou menor).
- **danceability\_%**: Porcentagem que indica o quanto apropriado a canção é para dançar
- **valence\_%**: Positividade do conteúdo musical da música.
- **energy\_%**: Nível de energia percebido da música.
- **acousticness\_%**: Quantidade de som acústico na música.
- **instrumentality\_%**: Quantidade de conteúdo instrumental na música.
- **liveness\_%**: Presença de elementos de performance ao vivo.
- **speechiness\_%**: Quantidade de palavras faladas na música.



5.1.1 Conectar/importar dados para outras ferramentas

Subi os arquivos em csv para minha pasta do drive e as converti para google planilhas:

Criei meu projeto no big query, seguindo a orientação dos videos da mirela, entretanto não conseguia fazer o big query reconhecer o cabeçalho, dai então achei um video no youtube e consegui finalizar essa parte de conexão inicial.

<https://www.youtube.com/watch?v=grPxdUmLnUc>

### 💡 5.1.2 🔎 Identificar e tratar valores nulos

A próxima orientação, informa para encontrar os valores nulos:

Na tabela track\_technical\_info-technical\_info fui consultando variável por variável e achei estes nulos :

```

    COUNTIF(`in_deezer_charts` IS NULL) AS `in_deezer_charts`,
    COUNTIF(`in_shazam_charts` IS NULL) AS `in_shazam_charts`,

  28
  FROM
  29   `my-project-laboratoria.dadoslaboratoria.track_in_competition_competition`*
  30
  31  SELECT
  32    COUNTIF(`track_id` IS NULL) AS `track_id`,
  33    COUNTIF(`bpm` IS NULL) AS `bpm`,
  34    COUNTIF(`key` IS NULL) AS `key`,
  35    COUNTIF(`mode` IS NULL) AS `mode`,
  36    COUNTIF(`dancability` IS NULL) AS `dancability`,
  37    COUNTIF(`valence` IS NULL) AS `valence`,
  38    COUNTIF(`energy` IS NULL) AS `energy`,
  39    COUNTIF(`acousticness` IS NULL) AS `acousticness`,
  40    COUNTIF(`instrumentalness` IS NULL) AS `instrumentalness`,
  41    COUNTIF(`liveness` IS NULL) AS `liveness`,
  42    COUNTIF(`speechiness` IS NULL) AS `speechiness`,
  43
  44  FROM
  45
  
```

**Consulta concluída**

**Resultados da consulta**

Informações do job	Resultados	Gráfico	JSON	Detalhes da execução	Gráfico de execução						
Unha	track_id	bpm	key	mode	dancability	valence	energy	acousticness	instrumentalness	liveness	speechiness
1	0	0	95	0	0	0	0	0	0	0	0

Resultados por página: 50 | 1 - 1 de 1 | < > |

Substitui os nulos em key por não informado:

```

    `valores_nulos_e_duplicados`*
  40
  41
  42
  43
  44
  45
  46
  47
  48
  49
  50
  51
  52
  53
  54
  55
  56
  57
  58
  59
  60
  61
  62
  63
  64
  65
  66
  67
  68
  69
  70
  71
  72
  73
  74
  75
  76
  77
  78
  79
  80
  81
  82
  83
  84
  85
  86
  87
  88
  89
  90
  91
  92
  93
  94
  95
  96
  97
  98
  99
  
```

**Consulta concluída**

**Resultados da consulta**

Informações do job	Resultados	Gráfico	JSON	Detalhes da execução	Gráfico de execução		
Unha	track_id	bpm	key	mode	dancability	valence	energy
1	0	0	95	0	0	0	0

Resultados por página: 50 | 1 - 50 de 953 | < > |

na tabela track\_in\_competition - competition também encontrei na coluna in\_shazam\_charts valores nulos:

```

    COUNTIF(`in_apple_playlists` IS NULL) AS `in_apple_playlists`,
    COUNTIF(`in_apple_charts` IS NULL) AS `in_apple_charts`,
    COUNTIF(`in_deezer_playlists` IS NULL) AS `in_deezer_playlists`,
    COUNTIF(`in_deezer_charts` IS NULL) AS `in_deezer_charts`,
    COUNTIF(`in_shazam_charts` IS NULL) AS `in_shazam_charts`,

  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34
  35
  36
  37
  38
  39
  40
  
```

**Consulta concluída**

**Resultados da consulta**

Informações do job	Resultados	Gráfico	JSON	Detalhes da execução	Gráfico de execução	
Unha	track_id	in_apple_playlists	in_apple_charts	in_deezer_playlists	in_deezer_charts	in_shazam_charts
1	0	0	0	0	0	50

Resultados por página: 50 | 1 - 1 de 1 | < > |

Para os 50 nulos, substitui por 0

```

    SELECT
        track_id,
        in_apple_playlists,
        in_apple_charts,
        in_deezer_playlists,
        in_deezer_charts,
        in_shazam_charts
    FROM
        `my-project-laboratoria.dadoslaboratoria.track_in_competition_competition`
    WHERE
        track_id IS NOT NULL

```

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has a tree view of projects and datasets. The main area shows a query editor with the above SQL code. Below the code is a results table with three rows of data:

track_id	in_apple_playlists	in_apple_charts	in_deezer_playlists	in_deezer_charts	in_shazam_charts
198	7948655	20	46	21	8
199	5279142		28	125	1
200	7451979	34	0	5108	6

Ao tentar verificar as variáveis nulas da tabela **track\_in\_spotify-spotify** encontrei um erro na linha 575

```

    SELECT
    *
    FROM
        `my-project-laboratoria.dadoslaboratoria.track_in_spotify-spotify`

```

The screenshot shows the Google Cloud BigQuery interface. The main area shows a query editor with the above SQL code. A red error message is displayed: "Error while reading table: my-project-laboratoria.dadoslaboratoria.track\_in\_spotify-spotify, error message: Could not convert value to integer: Row 575; Col 9. File: 1ZOBbdJAavtJfID-CiId:jAKG.VGIZ5RkmJcJ3mA". Below the code is a results table with one row of data:

track_id	in_spotify_playlists	in_spotify_charts	in_spotify_tracks
1	0	0	0

por se tratar de apenas uma célula com erro, optei por excluir direto na base, e o erro foi corrigido, confirmando que não havia nulos nesta tabela.

```

    SELECT
        COUNTIF(track_id IS NULL) AS track_id,
        COUNTIF(track_name IS NULL) AS track_name,
        COUNTIF(artist_a_name IS NULL) AS artist_a_name,
        COUNTIF(artist_count IS NULL) AS artist_count,
        COUNTIF(released_year IS NULL) AS released_year,
        COUNTIF(released_month IS NULL) AS released_month,
        COUNTIF(released_day IS NULL) AS released_day,
        COUNTIF(in_spotify_playlists IS NULL) AS in_spotify_playlists,
        COUNTIF(in_spotify_charts IS NULL) AS in_spotify_charts,
        COUNTIF(in_spotify_tracks IS NULL) AS in_spotify_tracks
    FROM
        `my-project-laboratoria.dadoslaboratoria.track_in_spotify-spotify`

```

The screenshot shows the Google Cloud BigQuery interface. The main area shows a query editor with the above SQL code. Below the code is a results table with one row of data:

track_id	track_name	artist_a_name	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts	in_spotify_tracks
1	0	0	0	0	0	0	0	0	0

Feita estas correções criei as views de technical info e competition-competition

```

SELECT
  track_id,
  COUNT(*) AS quantidade
FROM
  `my-project-laboratoria.dadoslaboratoria.track_in_spotify - spotify`
GROUP BY
  track_id
HAVING COUNT(*) > 1;
-- Para a tabela Trackincompetition
SELECT
  track_id,
  COUNT(*) AS quantidade
FROM
  `my-project-laboratoria.dadoslaboratoria.track_in_competition_competition`
GROUP BY
  track_id
HAVING COUNT(*) > 1;
-- Para a tabela Tracktechnicalinfo
SELECT
  track_id,
  COUNT(*) AS quantidade
FROM
  `my-project-laboratoria.dadoslaboratoria.track_technical_info`
GROUP BY
  track_id
HAVING COUNT(*) > 1;

```



### 5.1.3 Identificar e tratar valores duplicados

Fiz a consulta nas 3 tabelas se havia track\_id duplicados e nenhuma resultou em valor duplicado

```

SELECT
  track_id,
  COUNT(*) AS quantidade
FROM
  `my-project-laboratoria.dadoslaboratoria.track_in_spotify - spotify`
GROUP BY
  track_id
HAVING COUNT(*) > 1;
-- Para a tabela Trackincompetition
SELECT
  track_id,
  COUNT(*) AS quantidade
FROM
  `my-project-laboratoria.dadoslaboratoria.track_in_competition_competition`
GROUP BY
  track_id
HAVING COUNT(*) > 1;
-- Para a tabela Tracktechnicalinfo
SELECT
  track_id,
  COUNT(*) AS quantidade
FROM
  `my-project-laboratoria.dadoslaboratoria.track_technical_info`
GROUP BY
  track_id
HAVING COUNT(*) > 1;

```

posteriormente fiz a consulta para valores duplicados de acordo com a orientação do projeto e encontrei 4 valores duplicados para track\_name&artist\_name na tabela spotify:

```

SELECT
  track_name,
  artist_s_name,
  COUNT(*) AS quantidade
FROM
  `my-project-laboratoria.dadoslaboratoria.track_in_spotify - spotify`
GROUP BY
  track_name,
  artist_s_name
HAVING COUNT(*) > 1;

```

track_name	artist_s_name	quantidade
SNAP	Rose Line	2
SPIT IN MY FACE!	ThySoMch	2
About Damn Time	Lizzo	2
Take My Breath	The Weeknd	2

Optamos por "excluir" já dentro da view de spotify as 8 linhas, pois os nome de musica e artista se repetem para ids diferentes .

```

    CREATE OR REPLACE VIEW `my-project-laboratoria.dadoslaboratoria.track_in_spotify_corrigido` AS
    SELECT DISTINCT
        track_id,
        artist_s_name,
        released_year,
        released_month,
        streams
    FROM
        `my-project-laboratoria.dadoslaboratoria.track_in_spotify` AS
    WHERE
        (track_name, artist_s_name) NOT IN (
            ('SNAP', 'Bass Line'),
            ('SPTT IN MY FACE', 'The Weeknd'),
            ('About Damn Time', 'Lizzo'),
            ('Take My Breath', 'The Weeknd')
        )
  
```

Esta consulta vai processar 0 B quando executada.



#### 5.1.4 Identificar e tratar dados fora do escopo de análise

Identifiquei conforme orientação que na tabela technical info as colunas key e mode seriam irrelevantes para analise:

```

    CREATE OR REPLACE VIEW `my-project-laboratoria.dadoslaboratoria.track_in_spotify - spotify` AS
    SELECT
        * EXCEPT(key, mode)
    FROM
        `my-project-laboratoria.dadoslaboratoria.track_technical_info_technical_info` AS
    WHERE
        track_name = 'Take My Breath'
  
```

Esta consulta vai processar 0 B quando executada.



#### 5.1.5 Identificar e tratar dados discrepantes em variáveis categóricas

Identifiquei que as colunas : track\_name, artist\_s\_name, continham valores com caracteres inválidos, o que não deixava a informação clara, e fiz uma tentativa de corrigir estes valores:

```

    SELECT
        REGEXP_REPLACE(track_name, r'[^x00-x7f]', '') AS track_name,
        artist_s_name,
        track_id,
        track_name,
        artist_s_name,
        track_name,
        artist_s_name
    FROM
        `my-project-laboratoria.dadoslaboratoria.track_in_spotify` - spotify
    WHERE
        track_name IS NULL OR artist_s_name IS NULL
    GROUP BY
        track_name,
        artist_s_name
    HAVING
        COUNT(*) > 1
    ORDER BY
        track_name,
        artist_s_name

```

Resultados da consulta

track_name	artist_s_name	track_name_1
Like Crazy	Jimin	Like Crazy
LADY GAGA	Gabito Ballesteros, Junior H. Pe..	LADY GAGA
I Can See You (Taylor&#039;s Ve..	Taylor Swift	I Can See You (Taylor&#039;s Ve..
I Wanna Be Yours	Arctic Monkeys	I Wanna Be Yours
Green Glitter - Slim Music Queen	Brennan Bahr Glitter	Green Glitter - Slim Music Queen

Como não fez a mudança esperada usei o regexp\_replace nas duas colunas e foi corrigido corretamente:

```

    SELECT
        REGEXP_REPLACE(track_name, r'[^x00-x7f]', '') AS track_name,
        artist_s_name,
        track_id,
        track_name,
        artist_s_name,
        track_name,
        artist_s_name
    FROM
        `my-project-laboratoria.dadoslaboratoria.track_in_spotify` - spotify
    WHERE
        track_name IS NULL OR artist_s_name IS NULL
    GROUP BY
        track_name,
        artist_s_name
    HAVING
        COUNT(*) > 1
    ORDER BY
        track_name,
        artist_s_name

```

Resultados da consulta

track_id	trackname	artist_s_name	artist_count	released_year	released_month	released
6735267	Enchanted	Taylor Swift	1	2010	1	2010-01-01
1961503	Save Your Tears	The Weeknd	1	2020	3	2020-03-01
3451143	Sure Thing	Miguel	1	2010	5	2010-05-01
6450517	Every Breath You Take - Demise	The Police	1	1983	1	1983-01-01



### 5.1.6 Identificar e tratar dados discrepantes em variáveis numéricas

Eu não tive problema com valor em string na coluna citada como exemplo para ajuste, pois meu shema da tabela spotify e competition conseguiu compreender que se tratavam de valores inteiros.

```

    SELECT
        COUNT(*)
    FROM
        `my-project-laboratoria.dadoslaboratoria.track_in_spotify.corrigido`
    WHERE
        track_id IN (
            SELECT
                track_id
            FROM
                `my-project-laboratoria.dadoslaboratoria.track_in_spotify`
            WHERE
                track_id NOT IN (
                    SELECT
                        track_id
                    FROM
                        `my-project-laboratoria.dadoslaboratoria.track_in_spotify.corrigido`
                )
        )

```

Resultados da consulta

1	2	3
2762	3703895074	513675818.7901

de todas as consultas achei apenas streams muito discrepante e pedi auxilio ao deep seek.

## Análise dos Resultados de Streams no Spotify

Os resultados da sua consulta mostram estatísticas básicas sobre a coluna `streams` da tabela do Spotify:

### Valores Obtidos:

- Mínimo (min):** 2,762 streams
- Máximo (max):** 3,703,895,074 streams (aproximadamente 3.7 bilhões)
- Média (avg):** 513,734,548.1619... streams (aproximadamente 513.7 milhões)

### Interpretação:

#### 1. Discrepância extrema:

- A diferença entre o mínimo (2,762) e o máximo (3.7 bilhões) é enorme
- Isso indica uma distribuição altamente assimétrica dos dados

#### 2. Média elevada:

- A média de ~513 milhões é muito próxima do valor máximo
- Isso sugere que:
  - A maioria das músicas tem poucos streams
  - Algumas poucas músicas têm números extremamente altos (possivelmente hits virais)

#### 3. Possíveis conclusões:

- A plataforma Spotify tem alguns "super hits" com bilhões de streams
- A grande maioria das músicas tem desempenho muito mais modesto
- A média não representa bem a experiência típica (seria melhor analisar a mediana)

Para confirmar as informações fiz consultas na variável `streams` e a maioria das musicas tem mais de 10 milhões de visualizações

The screenshot shows the Google Cloud BigQuery interface. In the top navigation bar, there are tabs for 'My Project laboratoria' and 'Pesquisar (/) recursos, documentos, produtos e muito mais'. A search bar contains the text 'Pesquisa'. Below the navigation, there's a toolbar with icons for 'Pesquisar', 'Salvar consulta', 'Fazer o download', and 'Compartilhar'. The main area displays a query titled 'consulta discrepância table\_sp...' with the following code:

```

19
20 SELECT
21   CASE
22     WHEN streams < 10000 THEN '0-10K'
23     WHEN streams < 100000 THEN '10K-100K'
24     WHEN streams < 1000000 THEN '100K-1M'
25     WHEN streams < 10000000 THEN '1M-10M'
26     ELSE '>10M'
27   END AS faixa_streams,
28   COUNT(*) AS quantidade_musicas
29 FROM

```

The results table shows three rows:

Linha	faixa_streams	quantidade_musicas
1	0-10K	1
2	1M-10M	1
3	>10M	943

At the bottom right of the results table, there are buttons for 'Salvar resultados', 'Abrir em', and 'Atualizar'.



### 5.1.7 🔎 Verificar e alterar o tipo de dados

The screenshot shows the Google Cloud BigQuery interface. In the top navigation bar, there are tabs for 'My Project laboratoria' and 'Pesquisar (/) recursos, documentos, produtos e muito mais'. A search bar contains the text 'Pesquisa'. Below the navigation, there's a toolbar with icons for 'Pesquisar', 'Salvar', 'Fazer o download', 'Compartilhar', and 'Programação'. The main area displays a query titled 'Consulta sem título' with the following code:

```

1 SELECT
2   SAFE_CAST(streams AS INT64)
3   FROM
4   `my-project-laboratoria.dadoslaboratoria.track_in_spotify.corrigido`
5 WHERE streams IS NULL

```

A message below the code states: 'Esta consulta vai processar 0 B quando executada.'

The results table shows one row:

Informações do job	Resultados	Gráfico	JSON	Detalhes da execução	Gráfico de execução
	<b>●</b> Não há dados para exibir.				

At the bottom right of the results table, there are buttons for 'Salvar resultados', 'Abrir em', and 'Atualizar'.

Como eu já havia feito a manejo direto na linha da tabela, minha variável streams foi classificada com integer e não sofri problemas para uso do cast.



### 5.1.8 🔎 Criar novas variáveis

Conforme orientação criei a variável de data e já atualizei na minha view de spotify.

```

6 streams IS NULL
7
8 CREATE OR REPLACE VIEW `my-project-laboratoria.dadoslaboratoria.track_in_spotify_corrigido` AS
9
10 SELECT DISTINCT(
11   track_id,
12   REGEXP_REPLACE(track_name,r'[\x00-\x7F]', '') AS track_name,
13   artist_s_name, r'[\x00-\x7F]' AS artist_s_name,
14   artist_count,
15   released_year,
16   released_month,
17   released_day,
18   in_spotify_playlists,
19   in_spotify_charts,
20   DATE(released_year, released_month, released_day) AS release_date
21 )
22 FROM `my-project-laboratoria.dadoslaboratoria.track_in_spotify` spot
23 WHERE (
24   (track_name, artist_s_name) NOT IN (
25     ('SNL', 'Saturday Night Live'),
26     ('TV Trop Rock', 'TV Trop Rock'),
27     ('About Damn Time', 'Lizzo'),
28     ('Take My Breath', 'The Weeknd')
29   )
30 );

```

Consulta concluída

Resultados da consulta

Informações do job Consulta atualizada

Histórico de jobs Consulta atualizada

### 5.1.9 Unir tabelas

Fiz uma consulta unindo as views e salvei os dados dessa consulta que me resultou 945 linhas.

```

19
20 CREATE OR REPLACE VIEW `dadoslaboratoria.view_unificada` AS
21
22 SELECT
23   ... Colunas da track_in_spotify -> spotify
24   ...
25   ... Colunas da track_in_competition.competition
26   ...
27   ... Colunas da track_technical_info
28   ...
29   ...
30   ...
31   ...
32   ...
33
34 FROM
35   `my-project-laboratoria.dadoslaboratoria.track_in_spotify_corrigido` spot
36   LEFT JOIN `my-project-laboratoria.dadoslaboratoria.track_in_competition_view` comp
37   ON spot.track_id = comp.track_id -- Substitui pela chave correta
38   LEFT JOIN `my-project-laboratoria.dadoslaboratoria.track_technical_info` tech
39   ON comp.track_id = tech.track_id; -- Substitui pela chave correta
40
41 Esta consulta vai processar 0B quando executada.

```

Resultados da consulta

Historico de jobs Consulta atualizada

### 5.1.10 Construir tabelas auxiliares

Fiz uma consulta do total de musicas por artista e percebi que a coluna de artista precisava de uma contagem única.

```

1 WITH teste AS (
2     SELECT
3         artist_s_name,
4         COUNT(*) AS total_tracks
5     FROM
6         `my-project-laboratoria.dadoslaboratoria.view_unificada`
7     GROUP BY
8         artist_s_name
9 )
10
11 SELECT
12     artist_s_name,
13     teste.total_tracks
14 FROM
15     `my-project-laboratoria.dadoslaboratoria.view_unificada` AS s
16 LEFT JOIN
17     teste
18 ON
19     s.artist_s_name = teste.artist_s_name

```

Esta consulta vai processar 0 B quando executada.

artist_s_name	total_tracks
Yahitzka Y Su Escena, Grupo Fr...	1
Taylor Swift	34
Taylor Swift	34
Fuerza Regida	2
Junior H, Peso Pluma	2

Notei que as musicas por mais de 1 artista tinham o separador de virgula na coluna de `artist_name` e fiz uma cte separando esta coluna por valor distinto.

```

1 WITH artistas_separados AS (
2     SELECT
3         track_name,
4         TRIM(artist) AS artista_individual
5     FROM
6         `my-project-laboratoria.dadoslaboratoria.view_unificada`
7     UNNEST(SPLIT(artist_s_name, ',')) AS artist
8 ),
9
10     contagem_por_artista AS (
11     SELECT
12         artista.individual AS artist_name,
13         COUNT(DISTINCT track_name) AS total_musicas_distintas
14     FROM
15         artistas_separados
16     GROUP BY
17         artista.individual
18     )
19
20     SELECT
21         artist_name,
22         total_musicas_distintas
23     FROM
24         contagem_por_artista
25     ORDER BY
26         total_musicas_distintas DESC

```

Esta consulta vai processar 0 B quando executada.

Linha	artist_name	total_musicas_dj
1	Bad Bunny	40
2	Taylor Swift	38
3	The Weeknd	35
4	SZA	23
5	Kendrick Lamar	23
6	Fleid	21
7	Drake	19
8	Harry Styles	17
9	Peso Pluma	16



### 5.2.1 🔮 Agrupar dados de acordo com variáveis categóricas

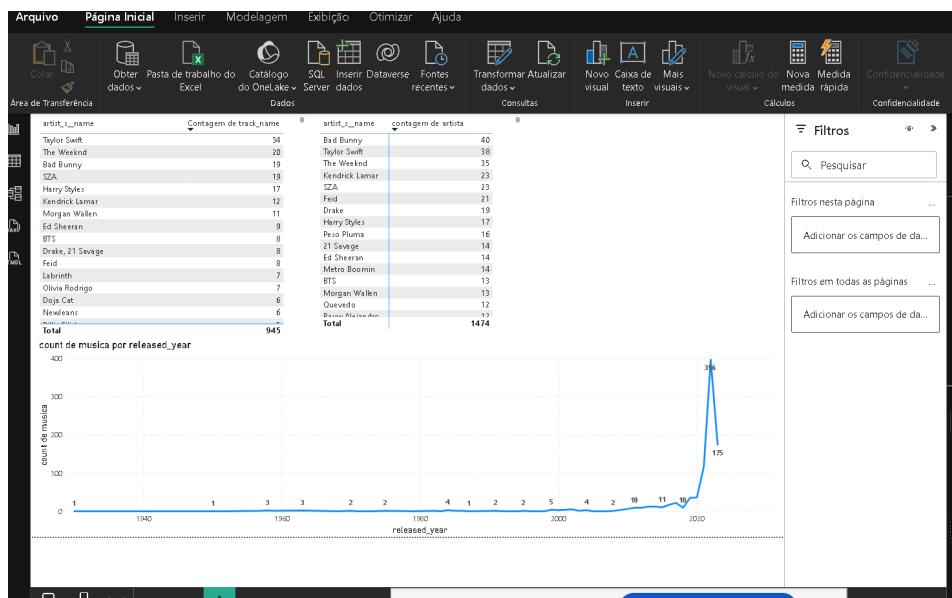
Obtive um erro de acesso ao tentar conectar minha view com o power bi

A DataSourceError ODBC: ERROR [42000] [Microsoft][BigQuery] [100] Error interacting with REST API Access Denied: BigQueryBigQuery; Permission denied while getting Drive credentials.  
Details:  
DataSourceKind=GoogleBigQuery  
DataSourcePath=GoogleBigQuery  
OdbcDrivers=[Table]

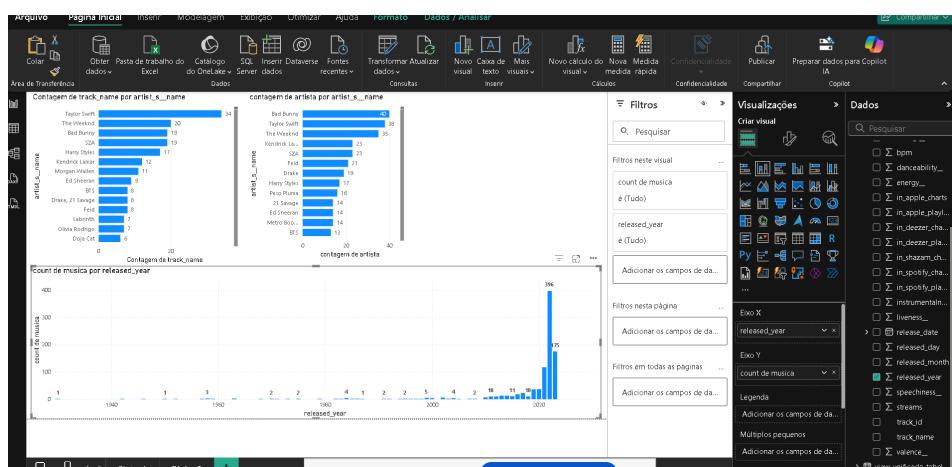
e resolvi criando uma versão tabela dessa view unificada.

track_id	track_name	artist_x_mi
1647815	Se La Ve	Arcaegel, Ox
7239348	We Don't Talk About Bruno	Adassa, Mai
7870058	Cay La Noche (feat. Cruz Cofun, Abhir Hathv, Bejo, El IMA)	Oseveido, Li
5350155	Nobody Like U - From "Turning Red"	Jordan Fisher
193046	Besbarra Rang (From "Patahan")	Vishal Shekhar
6602768	Izhoomo le Patahan	Ankit Singh
4801316	Los del Espacio	Big One, Dul
3087104	The Christmas Song (Merry Christmas To You) - Remastered 1999	Neil King Col
4002890	A Holly Jolly Christmas - Single Version	Burt Liles
3092002	Jingle Bells - Remastered 1999	Frank Sinatra
6373009	Jingle Bell Rock	Bobby Helms
8517749	Rockin' Around The Christmas Tree	Chubby Checker
8336945	Rockin' Around The Christmas Tree	Janet Jackson
8157749	Deck The Hall - Remastered 1999	Neil King Col
4227295	Let It Snow! Let It Snow! Let It Snow!	Dean Martin
5350503	It's the Most Wonderful Time of the Year	Andy Williams
6900052	Sleigh Ride	The Ronettes
3867590	Christmas (Baby Please Come Home)	Darlene Love
6250958	Have You Ever Seen The Rain?	Creedence Clearwater Revival
4061483	Love Grows (Where My Rosemary Grows)	Edision Light

Após corrigir o problema de visualização conseguimos criar as variáveis categóricas e expor em visuais no power bi.

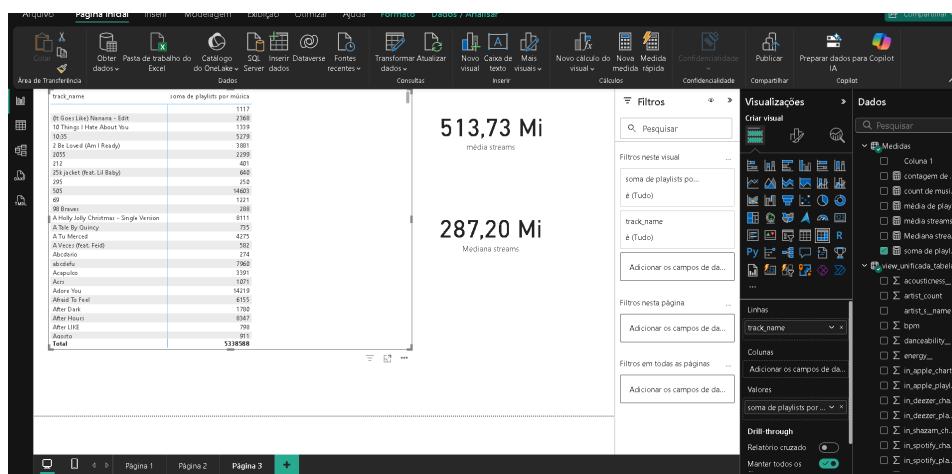


### 💡 5.1.2 Ver variáveis categóricas

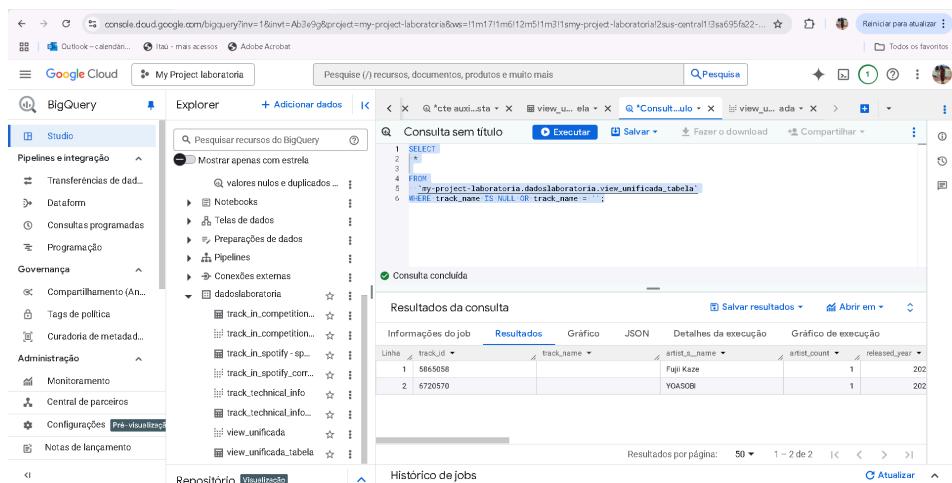


### 💡 5.2.3 Aplicar medidas de tendência central

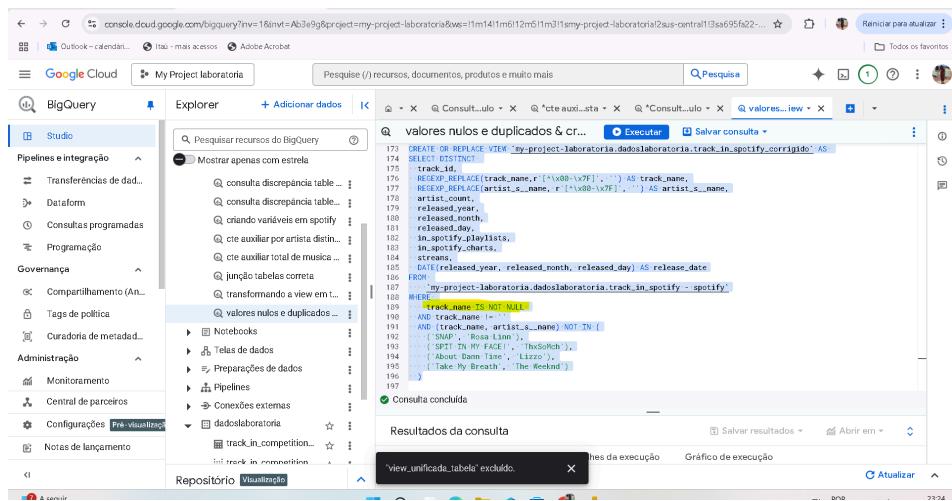
Ao tentar criar a soma de playlists(deezer,spotify, e apple music) verifiquei que existia duas musicas sem nome e consultei no big query para confirmar:



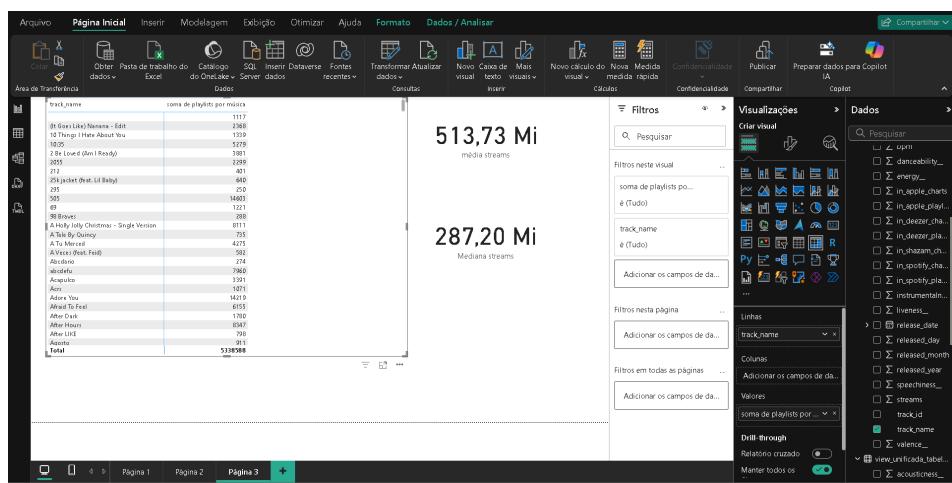
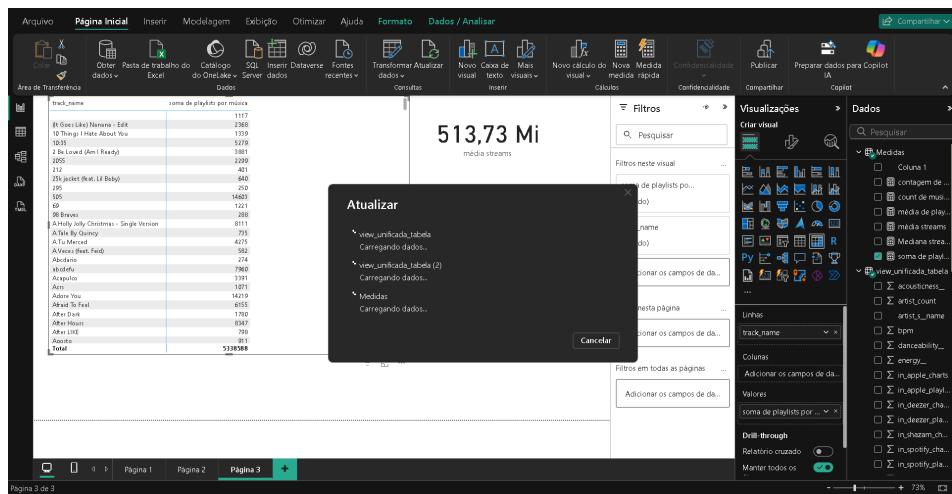
Quando usei o regex para alterar os caracteres não legíveis, terminei alterando por dados vazios.



Fiz o ajuste na própria view:



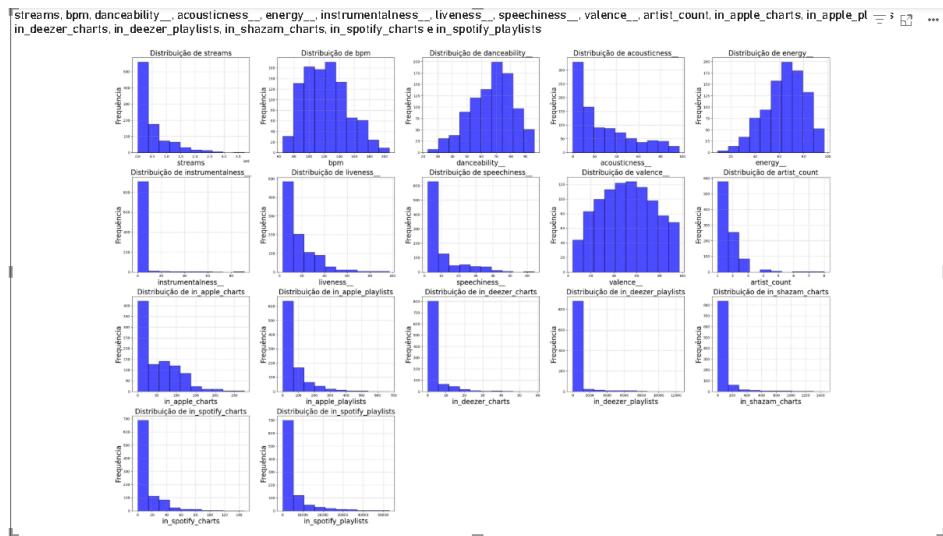
repiquei na view unificada e recriei a tabela unificada para leitura no bi



**Não resolveu o problema, optei por excluir linhas vazias nesta coluna e mesmo assim não consegui resolver, vou deixar em aberto pra resolver posteriormente. Obs: ajustei retirando-as no filtro de track\_name.**

#### 💡 5.2.4 🌐 Ver distribuição

Verificando a distribuição de streams no histograma com código em python identifiquei uma concentração muito próxima dos 513 mi, aproveitei e fiz uma consulta mais refinada no bigquery para confirmar esse histograma.



Consulta sem título

Executar Salvar Fazer o download

```

1 SELECT
2   AVG(streams)as media,
3   min(streams)as minimo,
4   max(streams) as maximo
5 FROM
6   `my-project-laboratoria.dadoslaboratoria.view_unificada_tabela`
7 LIMIT
8   1000
9
10 SELECT
11   APPROX_QUANTILES(streams, 100)[OFFSET(50)] AS mediana,
12   COUNTIF(streams < 1000000) AS qtd_abaiixo_1M,
13   COUNTIF(streams BETWEEN 1000000 AND 10000000) AS qtd_1M_a_100M,
14   COUNTIF(streams > 10000000) AS qtd_acima_100M
15 FROM `my-project-laboratoria.dadoslaboratoria.view_unificada_tabela` ...

```

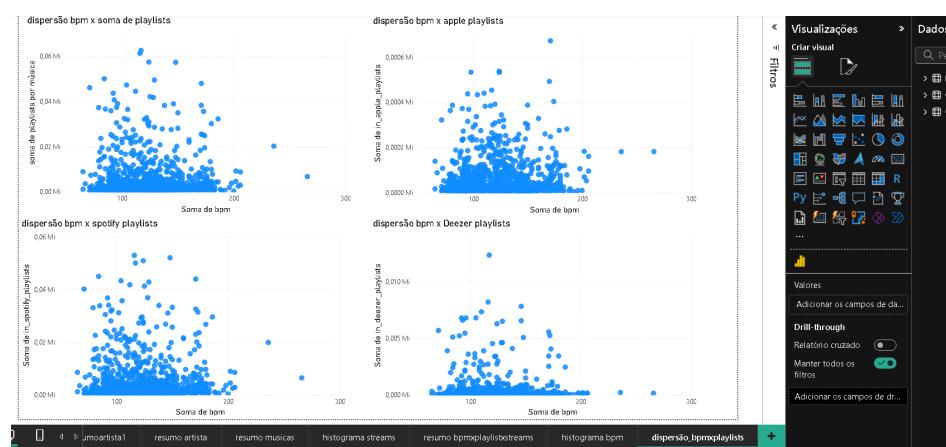
Consulta concluída

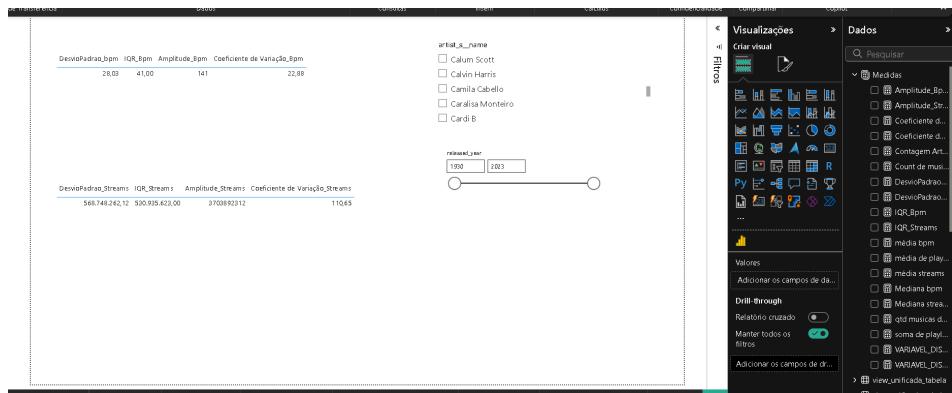
### Resultados da consulta

Informações do job	Resultados	Gráfico	JSON	Detalhes da execução	Gr
Linha // mediana ▾	mediana	qtd_abaiixo_1M	qtd_1M_a_100M	qtd_acima_100M	
1	287201015	1	152	792	



### 5.2.5 Aplicar medidas de dispersão





## 1. Análise do BPM (Batimentos Por Minuto)

- **Desvio Padrão (28,03):**

Indica que os BPMs das músicas variam, em média, **±28 BPM** em torno da média. Isso sugere uma **dispersão moderada**, comum em bases com múltiplos gêneros (ex.: pop = 100-130 BPM, hip-hop = 60-100 BPM).

- **IQR (41,00):**

O intervalo entre o 3º e 1º quartil abrange **41 BPM**, mostrando que os **50% centrais** das músicas estão em uma faixa razoavelmente ampla (ex.: se Q1=80 e Q3=121, há desde baladas até músicas dançantes).

- **Amplitude (141):**

A diferença entre o BPM máximo e mínimo é de **141 BPM**, confirmando a presença de estilos variados (ex.: uma música lenta com 60 BPM e uma eletrônica com 201 BPM).

- **Coeficiente de Variação (22,88%):**

Um CV de **22,88%** indica uma **variabilidade relativa moderada**. Como BPM é uma escala limitada (geralmente 60-200), esse valor é esperado.

### Conclusão para BPM:

A dispersão é **condizente com uma base diversificada em gêneros musicais**, sem outliers extremos. O IQR e o CV sugerem que os dados são relativamente equilibrados.

## 2. Análise de Streams

- **Desvio Padrão (568,7 milhões):**

Um desvio padrão altíssimo (**±568 milhões de streams**) confirma a **enorme desigualdade** na popularidade das músicas. A média é altamente influenciada por outliers (ex.: hits com bilhões de streams).

- **IQR (530,9 milhões):**

Os **50% centrais** das músicas têm uma diferença de **530 milhões de streams** entre si. Isso reforça que mesmo dentro da "faixa mediana", há músicas com popularidade muito variável.

- **Amplitude (3,7 bilhões):**

A diferença entre a música mais e menos popular é de **3.703.892.312 streams**, um sinal claro de **valores extremos** (ex.: uma música com 2.762 streams vs. outra com 3,7 bilhões).

- **Coeficiente de Variação (110,65%):**

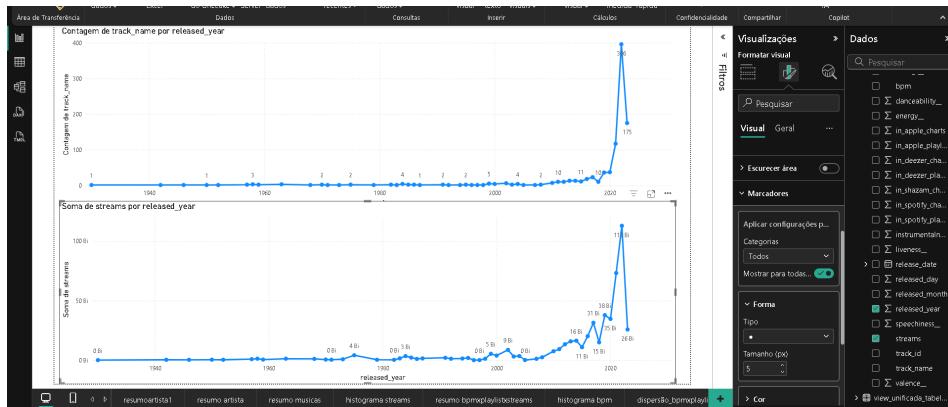
Um CV **acima de 100%** é raro e indica **alta dispersão relativa**. Isso significa que o desvio padrão é maior que a própria média, típico em dados com distribuição assimétrica e outliers.

### Conclusão para Streams:

A distribuição é **extremamente desigual**, com uma **minoria de músicas dominando a maioria dos streams**. A média é enganosa — a mediana (287 milhões, da análise anterior) é mais representativa do "meio" da distribuição.



## 5.2.6 Visualizar o comportamento dos dados ao longo do tempo



## 5.2.7 Calcular quartis, decis ou percentis

Calculei os quartis de streams, que é a variável principal da base e depois criei uma variável classificando pelo valor do quartil.

```

1 WITH quartiles AS (
2     SELECT
3         track_id,
4         streams,
5         NTILE(4) OVER (ORDER BY streams) AS quartiles_streams
6     FROM
7         `my-project-laboratoria.dadoslaboratoria.view.unificada_tabela`
8 )
9 SELECT
10     a.*,
11     q.quartiles_streams,
12     CASE
13         WHEN q.quartiles_streams = 1 THEN 'Baixo'
14         WHEN q.quartiles_streams = 2 THEN 'Medio-Baixo'
15         WHEN q.quartiles_streams = 3 THEN 'Medio-Alto'
16         ELSE 'Alto'
17     END AS classificacao_streams
18 END AS classificacao_streams
19 FROM
20     `my-project-laboratoria.dadoslaboratoria.view.unificada_tabela` AS a
21 LEFT JOIN quartiles q
22 ON a.track_id = q.track_id AND a.streams = q.streams
23

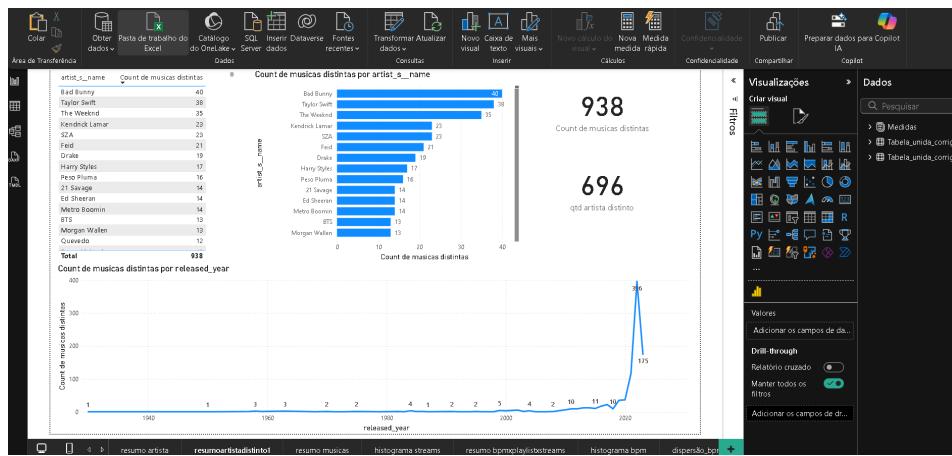
```

A partir disto decidi exportar os dados como uma nova tabela já com as 2 colunas novas.

Subi os dados corrigidos para o power bi, e corri a fonte de todas as medidas

Voltei para corrigir a coluna de artista distinto, por orientação do deep seek dupliquei a tabela e separei por delimitador a coluna de artist\_name.

Com esta correção identifiquei que temos 696 artistas distintos e 938 musicas distintas



Por orientação de Mirela pra que os valores se atualizem automaticamente caso eu precise acrescentar alguma coluna preferi corrigir direto em uma view e transformá-la em tabela para conexão com o power bi com o código abaixo:

```

CREATE OR REPLACE VIEW `dadoslaboratoria.view_unificada_com_quartis` AS
WITH dados_completos AS (
    SELECT
        -- Colunas da track_in_spotify - spotify
        spot.track_id,
        spot.track_name,
        spot.artist_s__name,
        spot.artist_count,
        spot.released_year,
        spot.released_month,
        spot.released_day,
        spot.in_spotify_playlists,
        spot.in_spotify_charts,
        spot.streams,
        spot.release_date,

        -- Colunas da track_in_competition_competition
        comp.in_apple_playlists,
        comp.in_apple_charts,
        comp.in_deezer_playlists,
        comp.in_deezer_charts,
        comp.in_shazam_charts,

        -- Colunas da track_technical_info
        tech.bpm,
        tech.danceability__,
        tech.valence__,
        tech.energy__,
        tech.acousticness__,
        tech.instrumentalness__,
        tech.liveness__,
        tech.speechiness__

    FROM
        `my-project-laboratoria.dadoslaboratoria.track_in_spotify_corrigido` spot
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratoria.track_in_competition_view` comp
    ON spot.track_id = comp.track_id
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratoria.track_technical_info` tech

```

```

    ON comp.track_id = tech.track_id
),

quartis_calculados AS (
SELECT
track_id,
streams,
danceability__,
valence__,
energy__,
acousticness__,
instrumentalness__,
liveness__,
speechiness__,
-- Cálculo dos quartis para cada métrica
NTILE(4) OVER (ORDER BY streams) AS quartil_streams,
NTILE(4) OVER (ORDER BY danceability__) AS quartil_danceability,
NTILE(4) OVER (ORDER BY valence__) AS quartil_valence,
NTILE(4) OVER (ORDER BY energy__) AS quartil_energy,
NTILE(4) OVER (ORDER BY acousticness__) AS quartil_acousticness,
NTILE(4) OVER (ORDER BY instrumentalness__) AS quartil_instrumentalness,
NTILE(4) OVER (ORDER BY liveness__) AS quartil_liveness,
NTILE(4) OVER (ORDER BY speechiness__) AS quartil_speechiness
FROM
dados_completos
)

SELECT
d.*,
-- Quartis numéricos
q.quartil_streams,
q.quartil_danceability,
q.quartil_valence,
q.quartil_energy,
q.quartil_acousticness,
q.quartil_instrumentalness,
q.quartil_liveness,
q.quartil_speechiness,

-- Classificações por quartil para streams
CASE
WHEN q.quartil_streams = 1 THEN 'Baixo'
WHEN q.quartil_streams = 2 THEN 'Médio-Baixo'
WHEN q.quartil_streams = 3 THEN 'Médio-Alto'
WHEN q.quartil_streams = 4 THEN 'Alto'
ELSE 'Fora da Classificação'
END AS classificacao_streams,

-- Classificações por quartil para danceability
CASE
WHEN q.quartil_danceability = 1 THEN 'Baixo'
WHEN q.quartil_danceability = 2 THEN 'Médio-Baixo'
WHEN q.quartil_danceability = 3 THEN 'Médio-Alto'
WHEN q.quartil_danceability = 4 THEN 'Alto'
ELSE 'Fora da Classificação'
END AS classificacao_danceability,

-- Classificações por quartil para valence

```

```

CASE
WHEN q.quartil_valence = 1 THEN 'Baixo'
WHEN q.quartil_valence = 2 THEN 'Médio-Baixo'
WHEN q.quartil_valence = 3 THEN 'Médio-Alto'
WHEN q.quartil_valence = 4 THEN 'Alto'
ELSE 'Fora da Classificação'
END AS classificacao_valence,

-- Classificações por quartil para energy
CASE
WHEN q.quartil_energy = 1 THEN 'Baixo'
WHEN q.quartil_energy = 2 THEN 'Médio-Baixo'
WHEN q.quartil_energy = 3 THEN 'Médio-Alto'
WHEN q.quartil_energy = 4 THEN 'Alto'
ELSE 'Fora da Classificação'
END AS classificacao_energy,

-- Classificações por quartil para acousticness
CASE
WHEN q.quartil_acousticness = 1 THEN 'Baixo'
WHEN q.quartil_acousticness = 2 THEN 'Médio-Baixo'
WHEN q.quartil_acousticness = 3 THEN 'Médio-Alto'
WHEN q.quartil_acousticness = 4 THEN 'Alto'
ELSE 'Fora da Classificação'
END AS classificacao_acousticness,

-- Classificações por quartil para instrumentalness
CASE
WHEN q.quartil_instrumentalness = 1 THEN 'Baixo'
WHEN q.quartil_instrumentalness = 2 THEN 'Médio-Baixo'
WHEN q.quartil_instrumentalness = 3 THEN 'Médio-Alto'
WHEN q.quartil_instrumentalness = 4 THEN 'Alto'
ELSE 'Fora da Classificação'
END AS classificacao_instrumentalness,

-- Classificações por quartil para liveness
CASE
WHEN q.quartil_liveness = 1 THEN 'Baixo'
WHEN q.quartil_liveness = 2 THEN 'Médio-Baixo'
WHEN q.quartil_liveness = 3 THEN 'Médio-Alto'
WHEN q.quartil_liveness = 4 THEN 'Alto'
ELSE 'Fora da Classificação'
END AS classificacao_liveness,

-- Classificações por quartil para speechiness
CASE
WHEN q.quartil_speechiness = 1 THEN 'Baixo'
WHEN q.quartil_speechiness = 2 THEN 'Médio-Baixo'
WHEN q.quartil_speechiness = 3 THEN 'Médio-Alto'
WHEN q.quartil_speechiness = 4 THEN 'Alto'
ELSE 'Fora da Classificação'
END AS classificacao_speechiness

```

```

FROM
dados_completos d
LEFT JOIN
quartis_calculados q

```

ON

d.track\_id = q.track\_id;

The screenshot shows the Google Cloud BigQuery interface. On the left, the 'Explorer' sidebar lists various datasets and tables, including 'Tabela\_unidada'. The main area displays the schema of the 'Tabela\_unidada' table, which contains 14 columns: 'in\_deezer\_charts', 'in\_shazam\_charts', 'bpm', 'danceability\_\_', 'valence\_\_', 'energy\_\_', 'acousticness\_\_', 'instrumentalness\_\_', 'liveness\_\_', 'speechiness\_\_', 'quartiles\_streams', 'classificacao\_streams', and 'view\_unificada'. Each column is defined with its data type (e.g., INTEGER, STRING) and whether it is nullable or not.



### 5.2.8 Calcular correlação entre variáveis

Calcular correlação entre variáveis streams / playlists / streams e danceability via comando CORR para verificar se existe entre as duas variáveis relação entre si.

The screenshot shows the Google Cloud BigQuery interface with a query editor. The query is as follows:

```
1 SELECT
2   CORR(streams.in_apple_playlists) AS apple_corr,
3   CORR(streams.in_deezer_playlists) AS deezer_corr,
4   CORR(streams.in_spotify_playlists) AS spotify_corr
5 FROM
6   `my-project-laboratoria.dadoslaboratoria.view_unificada_tabela`
7 LIMIT
8   1000
9
10 SELECT
11   CORR(streams.(in_apple_playlists + in_deezer_playlists + in_spotify_playlists)) AS correlation
12 FROM
13   `my-project-laboratoria.dadoslaboratoria.view_unificada_tabela`
14 LIMIT
15   1000
```

The results show a correlation value of 0.784759928109.

The screenshot shows the Google Cloud BigQuery interface with a query editor. The query is as follows:

```
1 SELECT
2   CORR(streams.in_apple_playlists) AS apple_corr,
3   CORR(streams.in_deezer_playlists) AS deezer_corr,
4   CORR(streams.in_spotify_playlists) AS spotify_corr
5 FROM
6   `my-project-laboratoria.dadoslaboratoria.view_unificada_tabela`
7 LIMIT
8   1000
9
10 SELECT
11   CORR(streams.(in_apple_playlists + in_deezer_playlists + in_spotify_playlists)) AS correlation
12 FROM
13   `my-project-laboratoria.dadoslaboratoria.view_unificada_tabela`
14 WHERE
15   streams.TS NOT NULL
16   AND in_apple_playlists.TS NOT NULL
17   AND in_deezer_playlists.TS NOT NULL
18   AND in_spotify_playlists.TS NOT NULL
```

The results show a correlation value of 0.784759928109.

## Interpretação do Resultado (0.785):

### 1. Forte Correlação Positiva:

- Valores de correlação variam de -1 a 1.
- **0.785 está próximo de 1**, sugerindo que há uma relação direta e forte entre o número de streams e a presença em playlists combinadas das três plataformas.

### 2. Implicações:

- Quanto mais uma música aparece em playlists (Apple, Deezer e Spotify), maior tende a ser seu número de streams.
- Isso reforça a importância de estratégias de colocação em playlists para impulsionar reproduções.

### 3. Limitações:

- Correlação não implica causalidade: Outros fatores (como popularidade do artista ou promoções externas) podem influenciar tanto as playlists quanto os streams.
- Verifique se há outliers distorcendo o resultado (ex.: uma música com poucas playlists mas muitos streams por viralização).

```
6 ---Calcular correlação entre variáveis streams e playlists / streams e danceability via comando CORR
7
8 select corr(streams,(in_apple_playlists + in_deezer_playlists + spotify_playlists)) AS corr_value,
9 corr(streams, danceability...) AS corr_s_dance
0 FROM `projeto2-laboratoria-musica.competition.aggregated_view`;
```

Consulta concluída

Resultados da consulta

Informações do job Resultados Gráfico JSON Detalhes da execução Gráfico de execução

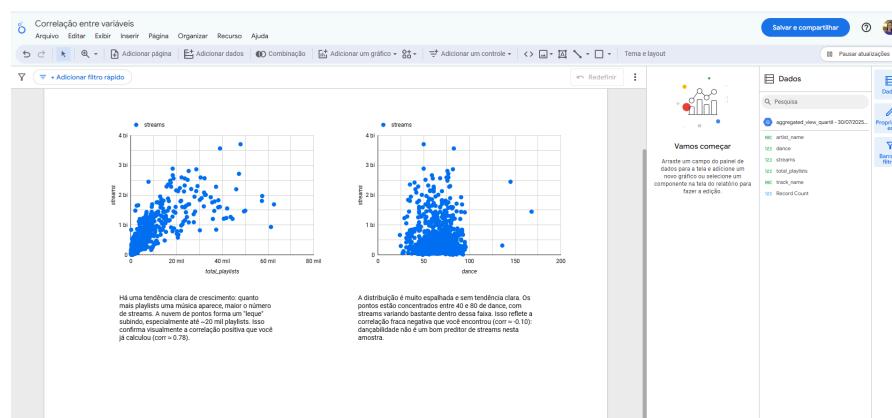
	corr_value	corr_s_dance
1	0.784153628293...	-0.1054568369...

Salvar

## Interpretação do Resultado (-0.10): Correlação negativa fraca

Praticamente não há relação entre o nível de "danceabilidade" e o número de "streams".

**Conclusão:** Músicas com mais "danceability" **não necessariamente** têm mais streams (pelo menos não de forma linear e direta).



Analizando os dados em gráficos através do Looker

## Resultados encontrados:

### Grafico 1 -

Há uma tendência clara de crescimento: quanto mais playlists uma música aparece, maior o número de streams.

A nuvem de pontos forma um "leque" subindo, especialmente até ~20 mil playlists.

Isso confirma visualmente a correlação positiva que você já calculou (corr  $\approx 0.78$ ).

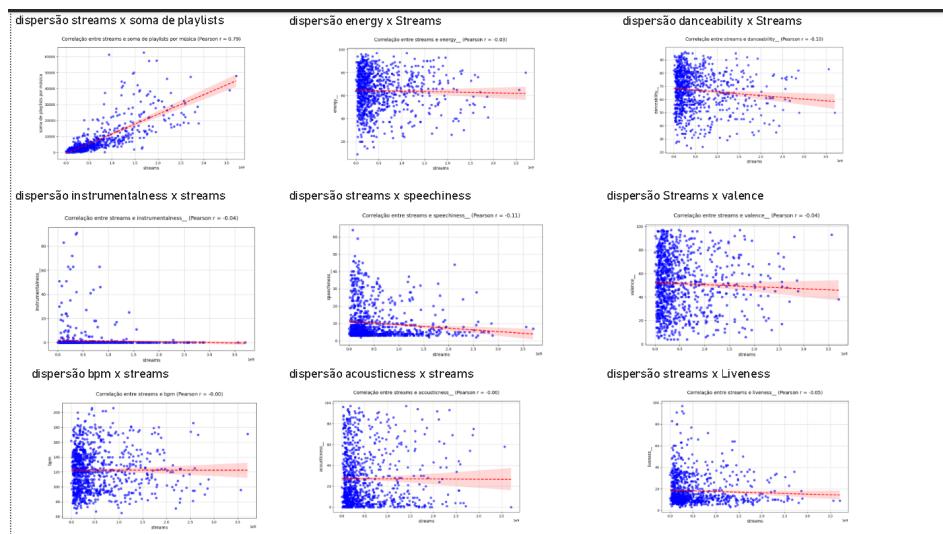
### Gráfico 2 -

A distribuição é muito espalhada e sem tendência clara.

Os pontos estão concentrados entre 40 e 80 de dance, com streams variando bastante dentro dessa faixa.

Isso reflete a correlação fraca negativa que encontramos (corr  $\approx -0.10$ ): dançabilidade não é um bom preditor de streams nesta amostra.

## Analizando com visual em python no power bi para confirmar vemos que :



## Interpretação do Resultado

### 1. Relação Direta:

- Músicas incluídas em **mais playlists tendem a ter mais streams**.
- Isso faz sentido, pois playlists são um dos principais mecanismos de descoberta em plataformas como Spotify.

### 2. Cenário Típico:

- Playlists curadas (ex.: "Today's Top Hits") expõem músicas a um público maior, impulsionando streams.
- Artistas com presença em playlists populares geralmente têm maior alcance orgânico.

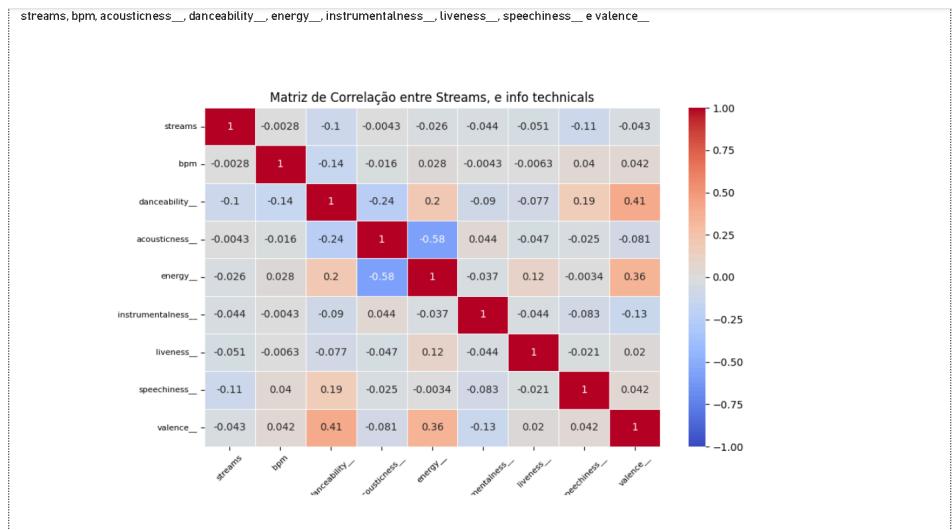
### 3. Diferença em Relação ao Anterior ( $r = -0.79$ ):

- O sinal positivo corrige a interpretação: a primeira análise sugeria um paradoxo (mais playlists = menos streams), o que é incomum. Agora, o resultado alinha-se ao esperado na indústria musical.

## 2. Correlações Fracas ou Próximas de Zero

As demais variáveis mostraram correlações insignificantes com **streams**:

- energy** ( $r = -0.03$ ), **instrumentalness** ( $r = -0.04$ ), **danceability** ( $r = -0.10$ ), **speechiness** ( $r = -0.11$ ), **valence** ( $r = -0.04$ ), **acousticness** ( $r = -0.00$ ), **liveness** ( $r = -0.05$ ).
- Interpretação:** Essas características musicais **não influenciam significativamente** o número de streams.
- Implicação:** O sucesso (em termos de streams) não está diretamente ligado a esses atributos técnicos. Fatores como **marketing, artista, tendências culturais ou algoritmos de plataformas** podem ser mais relevantes.



## 1. Correlações com streams (1ª linha/coluna)

Todos os valores são próximos de zero, mas vale destacar:

- **Maior correlação negativa:**
  - `speechiness__` ( $r = -0.11$ ): Músicas com mais falas/letras faladas tendem a ter *ligeiramente menos streams*.
  - `danceability__` ( $r = -0.10$ ): Surpreendentemente, músicas mais dançáveis têm *menos streams* (contrariando o senso comum).
- **Demais variáveis:**
  - `energy__` ( $r = -0.026$ ), `valence__` ( $r = -0.043$ ), etc.: Impacto insignificante.

**Conclusão:** Nenhum atributo técnico analisado explica significativamente o volume de streams.

## 2. Correlações entre Atributos Musicais

Algumas relações interessantes entre variáveis técnicas:

- **Forte correlação negativa:**
  - `acousticness__` VS `energy__` ( $r = -0.58$ ): Músicas acústicas tendem a ser menos energéticas.
- **Forte correlação positiva:**
  - `danceability__` VS `valence__` ( $r = 0.41$ ): Músicas dançáveis são mais "positivas" (valence).
  - `energy__` VS `valence__` ( $r = 0.36$ ): Músicas energéticas também são mais positivas.

**Implicação:** Essas relações eram esperadas e validam a consistência dos dados.

## 3. Padrão Geral

- **Correlações fracas:** A maioria dos valores está entre -0.1 e 0.1.
- **Exceções:** As relações entre atributos técnicos (não com streams) mostram padrões mais claros.

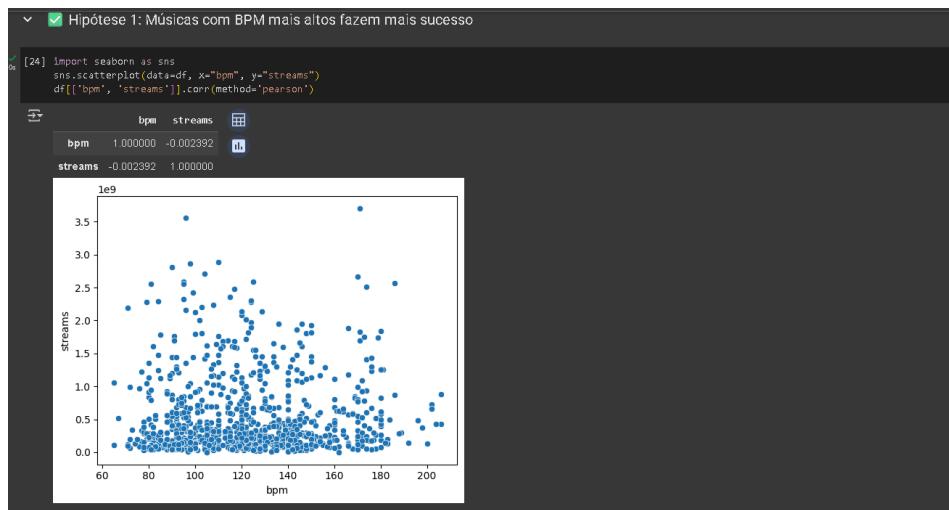


### 5.3 Aplicar técnica de análise

Neste marco, procuramos responder às hipóteses levantadas pela gravadora:

Fizemos todas as validações das hipóteses via colab com python.

- ❓ • Músicas com BPM (Batidas Por Minuto) mais altos fazem mais sucesso em termos de streams no Spotify;



#### 📊 O que o gráfico mostra

O gráfico de dispersão (scatter plot) mostra cada música como um ponto.

O eixo x representa o BPM (batidas por minuto).

O eixo y representa a quantidade de streams no Spotify.

Visualmente, os pontos estão dispersos sem uma tendência clara — ou seja, não há uma linha inclinada ascendente ou descendente que sugira uma relação forte entre BPM e streams.

✗ O que a correlação mostra A correlação de Pearson entre bpm e streams foi de -0.0035 .

#### ◆ Interpretação:

Uma correlação de +1.0 indica uma relação positiva perfeita.

Uma correlação de 0.0 indica nenhuma relação linear.

- 0.0035 é uma correlação muito fraca (quase nula).

✓ Conclusão da hipótese: Não há evidência suficiente de que BPM esteja fortemente relacionado ao sucesso (streams) no Spotify.

- ❓ • As músicas mais populares no ranking do Spotify também possuem um comportamento semelhante em outras plataformas como Deezer;



✗ Interpretação: As correlações com o Spotify variam de 0.52 a 0.59.

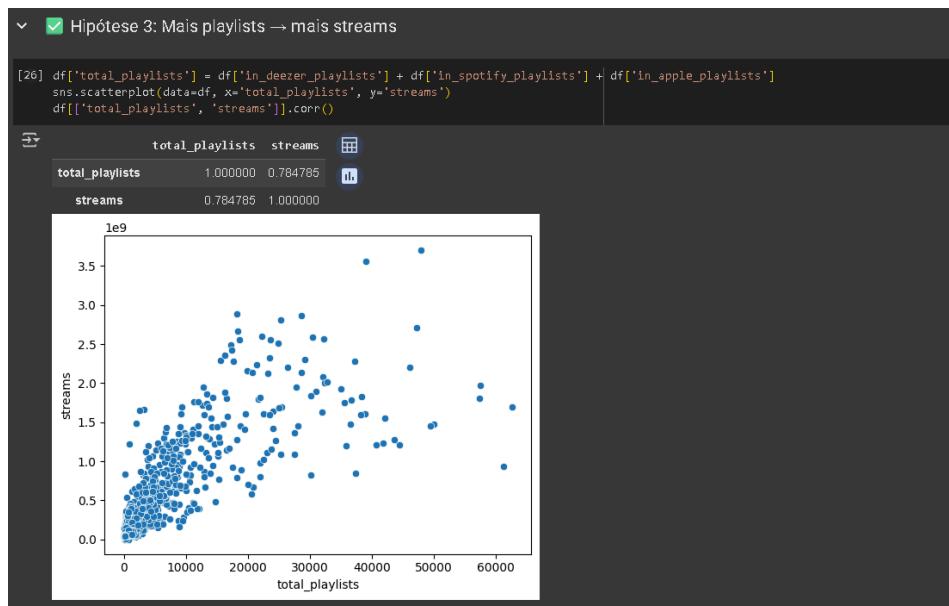
São todas correlações moderadas e positivas, o que indica que:

Músicas populares no Spotify tendem a ser populares também nas outras plataformas,

Mas não de forma perfeita ou automática — há variações relevantes entre plataformas.

✓ Conclusão: Hipótese confirmada parcialmente. Existe uma relação moderada entre o desempenho das músicas no Spotify e nas demais plataformas (Deezer, Apple e Shazam), especialmente com Deezer (0.59) e Shazam (0.58). Isso sugere que a popularidade é geralmente compartilhada entre as plataformas, mas há diferenças nos rankings de cada uma.

- ? • A presença de uma música em um maior número de playlists é relacionada a um maior número de streams;



Resultados obtidos: Correlação de 0.78 entre total\_playlists e streams.

Gráfico de dispersão mostra uma tendência clara crescente:

Quanto mais playlists uma música aparece, mais streams ela tende a ter.

Há variação, mas a nuvem de pontos é fortemente inclinada para cima.

✓ Interpretação: Uma correlação de 0.784 é alta e positiva.

Isso indica que a inserção em playlists tem forte associação com o sucesso da música no Spotify.

Algumas exceções (outliers) aparecem com muitas playlists ou muitos streams de forma isolada — o que é comum.

✓ Conclusão da hipótese: Hipótese confirmada. Há uma forte correlação positiva entre o número de playlists em que a música está presente e o total de streams no Spotify. Isso reforça a importância da visibilidade via playlists para o desempenho comercial das faixas.

- ? • Artistas com maior número de músicas no Spotify têm mais streams;

Para esta validação, tivemos que fazer alguns processos para localizar os artistas distintos:

- Separar a coluna de artist\_name

```
[31] df['artist_split'] = df['artist_s__name'].apply(
    lambda x: [a.strip() for a in x.split(',')]
    if isinstance(x, str) and ',' in x
    else [x.strip()] if isinstance(x, str)
    else []
)

df_explode = df.explode('artist_split', ignore_index=True).rename(
    columns={'artist_split': 'artist_individual'}
)
```

```
[32] df_explode[['artist_individual', 'artist_s__name']]
```

	artist_individual	artist_s__name
0	Styrx	Styx, utku INC, Thezth
1	utku INC	Styx, utku INC, Thezth
2	Thezth	Styx, utku INC, Thezth
3	The Ronettes	The Ronettes
4	Jos Felic	Jos Felic
...	...	...
1467	Peso Pluma	Junior H, Peso Pluma
1468	Nicki Minaj	Nicki Minaj, Aqua, Ice Spice
1469	Aqua	Nicki Minaj, Aqua, Ice Spice
1470	Ice Spice	Nicki Minaj, Aqua, Ice Spice
1471	Taylor Swift	Taylor Swift

1472 rows × 2 columns

- Conferir os valores únicos:

```
[33] print("Original:", df.shape[0])
print("Explodido:", df_explode.shape[0])

Original: 944
Explodido: 1472

[35] df_filtrado = df_explode[df_explode['artist_individual'].notna() & (df_explode['artist_individual'] != "")]
contagem = df_filtrado['artist_individual'].nunique()
print("Quantidade de artistas únicos:", contagem)

Quantidade de artistas únicos: 695
```

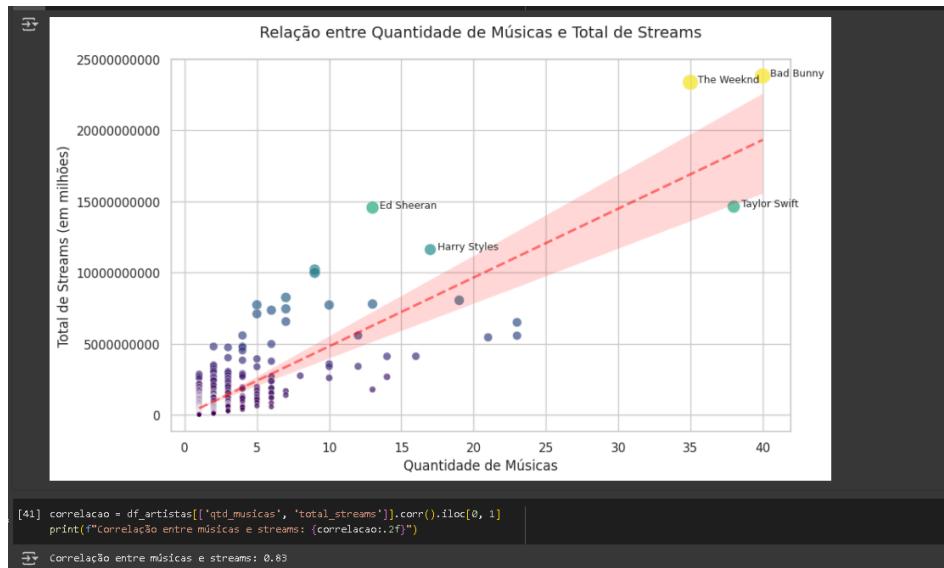
- Calcular o indicadores da hipótese

```
[36] df_artistas = (
    df_explode
    .groupby('artist_individual', as_index=False) # Evita que 'artist_individual' vire índice
    .agg(
        qtd_musicas=('track_id', 'nunique'), # Conta músicas únicas por artista
        total_streams=('streams', 'sum') # Soma todos os streams por artista
    )
    .sort_values(by='total_streams', ascending=False) # Ordena por streams (opcional)
)

[37] df_artistas.head()
```

	artist_individual	qtd_musicas	total_streams
68	Bad Bunny	40	23813527270
627	The Weeknd	35	23366402620
607	Taylor Swift	38	14630378183
186	Ed Sheeran	13	14559679731
246	Harry Styles	17	11608645649

- E plotar o gráfico conferindo a correlação:



📊 Resultado da correlação: Correlação de 0.83 entre qtd\_musicas e total\_streams.

↗️ Interpretação: Uma correlação de 0.83 é forte e positiva.

Isso indica que artistas que têm mais músicas publicadas tendem a acumular mais streams no total.

O comportamento faz sentido: mais músicas = mais oportunidades de ser ouvido.

✓ Conclusão da hipótese: Hipótese confirmada. Existe uma forte correlação positiva entre o número de músicas por artista e a quantidade total de streams. Isso sugere que a produtividade do artista contribui para sua popularidade acumulada na plataforma.



- As características da música influenciam no sucesso em termos de streams no Spotify.

```
[47] import seaborn as sns
import matplotlib.pyplot as plt

# Lista de características (features) analisadas
características = ['danceability_', 'valence_', 'energy_', 'acousticness_',
'instrumentalness_', 'speechiness_', 'liveness_']

# Calcula a matriz de correlação
corr_matrix = df[['streams'] + características].corr()

# Configura o tamanho do gráfico (largura, altura)
plt.figure(figsize=(12, 8))

# Heatmap com anotações e mapa de cores
sns.heatmap(
    corr_matrix,
    annot=True,           # Mostra valores dentro dos quadrados
    cmap='coolwarm',      # Mapa de cores (quente/frio)
    vmin=-1, vmax=1,      # Limites da escala de cores (-1 a 1 para correlação)
    center=0,             # Centraliza o branco em 0
    linewidths=0.5,        # Espaçamento entre células
    annot_kws={'size': 10}, # Tamanho da fonte dos valores
    fmt=".2f"              # Formato dos números (2 casas decimais)
)

# Ajusta os rótulos do eixo X (rotação de 45 graus)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.yticks(fontsize=12)

# Título do gráfico
plt.title("Correlação entre Streams e Características das Músicas", fontsize=14, pad=20)

# Melhora o layout para evitar cortes
plt.tight_layout()

# Mostra o gráfico
plt.show()
```



### ✓ Análise Detalhada:

Todas as correlações são fracas (entre -0.11 e 0.00):

- Nenhuma característica técnica tem impacto significativo (positivo ou negativo) nos streams.
- O valor mais alto em módulo é danceability\_ (-0.11), mas ainda é considerado irrelevante estatisticamente.

**Padrão geral negativo (mas insignificante):**

- As correlações negativas sugerem, de forma não conclusiva, que músicas com:

Maior dançabilidade (danceability\_),

Maior presença de voz (speechiness\_) tendem a ter ligeiramente menos streams, mas isso pode ser ruído nos dados.

**Relações entre outras variáveis (não diretamente com streams):**

energy\_ e acousticness\_ têm correlação forte e negativa (-0.58):

Músicas mais acústicas tendem a ser menos energéticas (esperado).

danceability\_ e valence\_ têm correlação moderada (0.41):

Músicas mais dançáveis tendem a ser mais "positivas" (valência).

Conclusão da Hipótese:  Hipótese refutada. As características técnicas analisadas não explicam a variação no número de streams.

Por quê? Fatores externos não capturados nos dados (ex: promoção, algoritmos de plataformas, viralidade em redes sociais) provavelmente dominam a popularidade.

Características como gênero musical, artista principal ou presença em playlists podem ser mais relevantes (não incluídas na análise).



5.3.1  Aplicar segmentação

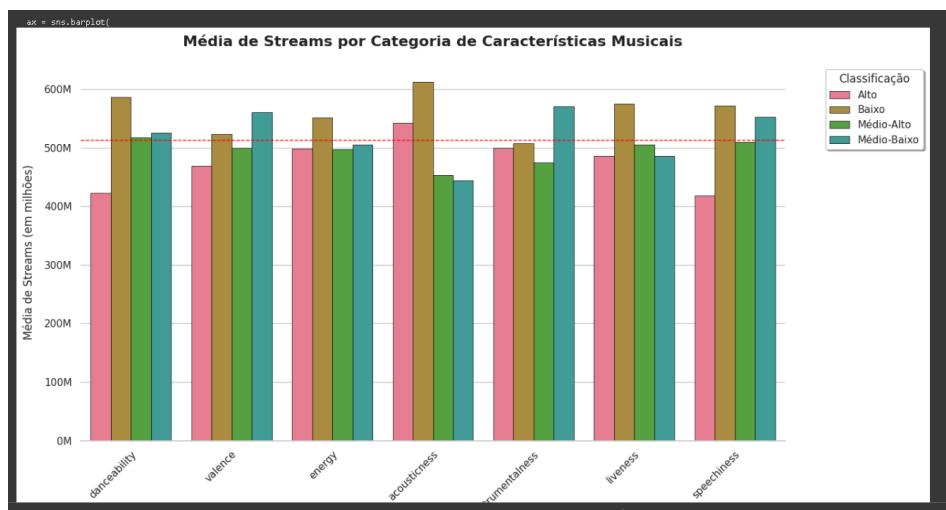
```

`[50] caracteristicas = [
    'classificacao_danceability', 'classificacao_valence', 'classificacao_energy',
    'classificacao_acousticness', 'classificacao_instrumentalness',
    'classificacao_liveness', 'classificacao_speechiness'
]
tabelas = []

for col in caracteristicas:
    media = df.groupby(col)['streams'].mean().reset_index()
    media['caracteristica'] = col.replace('classificacao_','')
    media.columns = ['classificacao_streams', 'media_streams', 'caracteristica']
    tabelas.append(media)

tabela_final = pd.concat(tabelas, ignore_index=True)

```



#### 📊 Análise Baseada nos Resultados do Gráfico:

##### 1. Speechiness (Fala/Vocalização) Padrão Claro:

Nível "Alto" tem a menor média de streams (~100M)

Nível "Baixo" tem a maior média (~500M)

Insight:

Músicas com muita fala/rap (ex.: podcasts, rap denso) têm desempenho inferior.

O público geral parece preferir músicas com menos conteúdo falado e mais melódico.

##### 1. Acousticness (Acústica) Padrão Claro:

Níveis "Baixo" e "Médio-Baixo" dominam (~400-600M streams)

Níveis "Alto" têm performance significativamente pior (~200M)

Insight:

Músicas eletrônicas ou com produção digital (baixa acústica) são mais populares.

Versões acústicas ou instrumentais orgânicas têm alcance limitado.

##### 1. Danceability, Valence, Energy (Dançabilidade, Positividade, Energia) Padrão Inconclusivo:

Níveis "Baixo" e "Moderado" performam bem, mas sem diferença significativa entre categorias.

Exemplo:

Danceability: "Médio-Baixo"(550M) vs. "Alto" (450M)

Valence: "Médio-Alto" (500M) vs. "Baixo" (480M)

Insight:

Não há uma preferência clara por músicas extremamente dançáveis, energéticas ou positivas.

Sugere que outros fatores (ex.: artista, gênero) são mais decisivos que essas características.

👉 Conclusões Estratégicas: Evite speechiness alto se o objetivo é maximizar streams.

Priorize produção não-acústica (ex.: sintetizadores, batidas eletrônicas).

Danceability/Energy/Valence: Flexibilidade na criação, pois não impactam drasticamente a popularidade.



### 5.3.2 🚩 Validar hipótese

**Objetivo:** Validar as hipóteses levantadas através de correlação e gráfico de dispersão

**Objetivo individual:**

Cada uma deve calcular a correlação das variáveis de uma hipótese e visualizar esses dados através de um gráfico de dispersão e discutir os resultados se existe ou não correlação e se a hipótese é verdadeira.

```
[44] características = ['danceability__', 'valence__', 'energy__', 'acousticness__',
    'instrumentalness__', 'liveness__', 'speechiness__']
    df.groupby('grupo_ano')[características].mean()

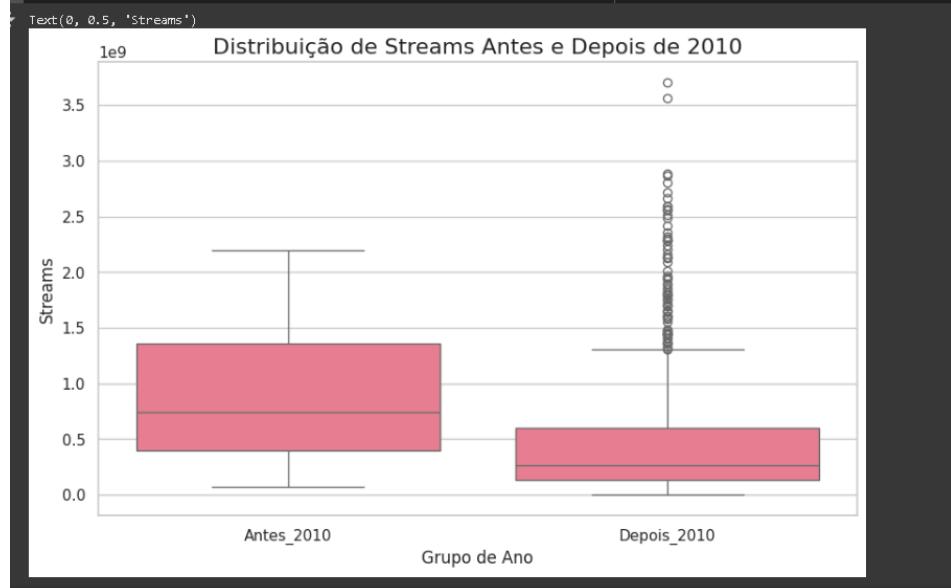
    danceability__ valence__ energy__ acousticness__ instrumentalness__ liveness__ speechiness__
    grupo_ano
    Antes_2010      60.73913  55.333333  63.42029     29.782609      1.565217  19.144928    7.072464
    Depois_2010     67.413714  51.018266  64.284571    27.033143      1.573714  18.041143    10.416

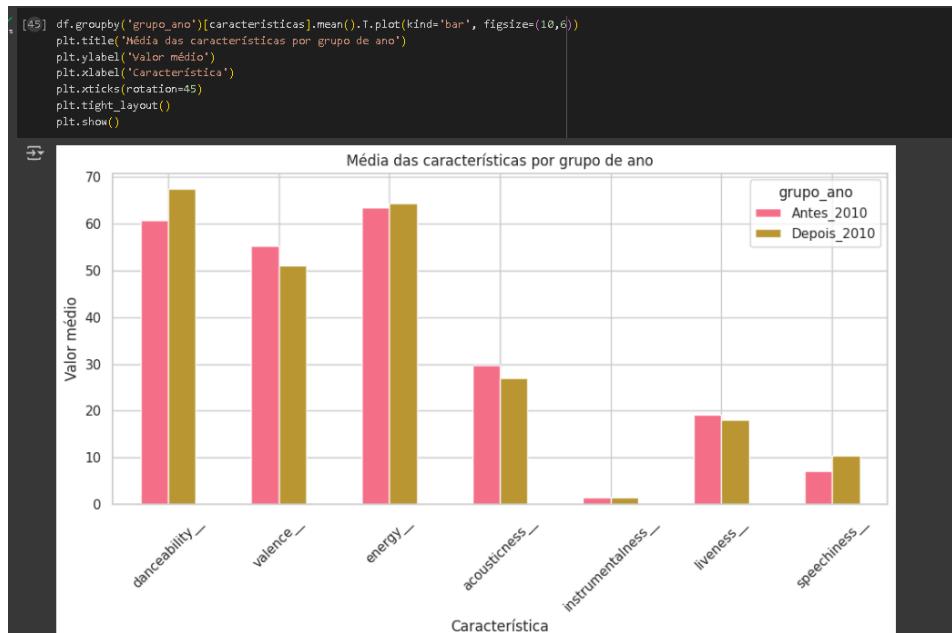
    grupo_antes = df[df['grupo_ano'] == 'Antes_2010']['streams']
    grupo_depois = df[df['grupo_ano'] == 'Depois_2010']['streams']

    stat, p_valor = mannwhitneyu(grupo_antes, grupo_depois, alternative='two-sided')
    print(f" Mann-Whitney U: {stat:.2f}, p-valor: {p_valor:.4f}")

Mann-Whitney U: 44202.00, p-valor: 0.0000

Conclusão:
O teste indica que há uma diferença estatisticamente significativa entre as distribuições do número de streams do grupo "antes de 2010" e do grupo "depois de 2010".
Em termos práticos, a evidência sugere fortemente que o número de streams mudou de forma significativa após o ano de 2010. Para saber se a média (ou mediana) de streams aumentou ou diminuiu, você precisaria calcular e comparar estatísticas descritivas (como a mediana) para cada um dos grupos (grupo_antes e grupo_depois).
```





## 5.4 Resumir informações em um dashboard ou relatório





#### 5.4.1 🍊 Representar dados por meio de tabela resumo ou scorecards

```
# Para manipulação de dados
import pandas as pd

# Para visualização (scorecard)
import plotly.graph_objects as go

# Indicadores gerais (exemplos)
total_musicas = df.shape[0]
media_streams = df['streams'].mean()
media_danceability = df['danceability_'].mean()
musicas_anteriores_2010 = df[df['grupo_ano'] == 'Antes_2010'].shape[0]
musicas_depois_2010 = df[df['grupo_ano'] == 'Depois_2010'].shape[0]

# Scorecard com Plotly
fig = go.Figure()

fig.add_trace(go.Indicator(
    mode="number",
    value=total_musicas,
    title={"text": "Total de Músicas"},
    domain={'row': 0, 'column': 0}))

fig.add_trace(go.Indicator(
    mode="number",
    value=media_streams,
    number={"prefix": "", "valueformat": ".0f"},
    title={"text": "Média de Streams"},
    domain={'row': 0, 'column': 1}))

fig.add_trace(go.Indicator(
    mode="number",
    value=media_danceability,
    number={"suffix": "%"},
    title={"text": "Média de Danceability"},
    domain={'row': 0, 'column': 2}))
```

```

fig.add_trace(go.Indicator(
    mode="number",
    value=musicas_anteriores_2010,
    title={"text": "Músicas < 2010"},
    domain={'row': 1, 'column': 0}))

fig.add_trace(go.Indicator(
    mode="number",
    value=musicas_depois_2010,
    title={"text": "Músicas ≥ 2010"},
    domain={'row': 1, 'column': 1}))

fig.update_layout(
    grid={'rows': 2, 'columns': 3, 'pattern': "independent"},
    height=500,
    title="Scorecard da Base de Dados Musical"
)

fig.show()

```



#### Links:

Relatório Power bi :[https://app.powerbi.com/view?r=eyJrIjoiNjg5NmQ4NDctM2NjNS00MDc3LWFhOGEtNDk0NzM0NjhiYTNkliwidCl6ljhiZTBkOTY1LWE1NTktNDYyNC1iNTJH#scrollTo=vS\\_TLqLO7R6](https://app.powerbi.com/view?r=eyJrIjoiNjg5NmQ4NDctM2NjNS00MDc3LWFhOGEtNDk0NzM0NjhiYTNkliwidCl6ljhiZTBkOTY1LWE1NTktNDYyNC1iNTJH#scrollTo=vS_TLqLO7R6)

Looker: <https://lookerstudio.google.com/s/k88vut58MzE>

Colab validação hipóteses: [https://colab.research.google.com/drive/19k12DbT36730i4KpUx1zMKql8AXI-QIH#scrollTo=vS\\_TLqLO7R6](https://colab.research.google.com/drive/19k12DbT36730i4KpUx1zMKql8AXI-QIH#scrollTo=vS_TLqLO7R6)

Apresentação: <https://docs.google.com/presentation/d/1W9NTyppgAdk5coQleATGt0dnyPsXGh-idZWPkovr0Hw/edit?slide=id.p#slide=id.p>