

# 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>

track_id	bpm	key	mode
4002370	125	B	B
6247887	92	C#	F#
6974739	138	F	F
2362023	170	A	A

### 💡 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 :

```

25 COUNTIF(`in_deezer_charts` IS NULL)AS `in_deezer_charts`,
26 COUNTIF(`in_shazam_charts` IS NULL)AS `in_shazam_charts`,
27
28 FROM
29 `my-project-laboratoria.dadoslaboratoria.track_in_competition_competition`
30
31 SELECT
32
33 COUNTIF(`track_id` IS NULL)AS `track_id`,
34 COUNTIF(`bpm` IS NULL)AS `bpm`,
35 COUNTIF(`key` IS NULL)AS `key`,
36 COUNTIF(`mode` IS NULL)AS `mode`,
37 COUNTIF(`danceability` IS NULL)AS `danceability`,
38 COUNTIF(`valence` IS NULL)AS `valence`,
39 COUNTIF(`energy` IS NULL)AS `energy`,
40 COUNTIF(`acousticness` IS NULL)AS `acousticness`,
41 COUNTIF(`instrumentalness` IS NULL)AS `instrumentalness`,
42 COUNTIF(`liveness` IS NULL)AS `liveness`,
43 COUNTIF(`speechiness` IS NULL)AS `speechiness`,
44
45 FROM
46
47 Consulta concluída

```

Resultados da consulta

Informações do job	Resultados	Gráfico	JSON	Detalhes da execução	Gráfico de execução						
Unha	track_id	bpm	key	mode	danceability	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:

```

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
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
308
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
788
789
789
790
791
792
793
794
795
796
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2
```

```

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`

```

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 editor 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 results table contains one error message row:

track_id	in_spotify_playlists	in_spotify_charts	in_spotify_playlist	in_spotify_chart
				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

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_playlist IS NULL) AS in_spotify_playlist,
  COUNTIF(in_spotify_chart IS NULL) AS in_spotify_chart
FROM
  `my-project-laboratoria.dadoslaboratoria.track_in_spotify-spotify`

```

The screenshot shows the Google Cloud BigQuery interface. The results table shows all counts as zero, confirming no null values were present.

track_id	track_name	artist_a_name	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts	in_spotify_playlist	in_spotify_chart
1	0	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,
    track_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

```

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:

```

149 REGEXP_REPLACE(track_name, r'[^\\x00-\\x7F]+', '') AS track_name
150
151 FROM
152 `my-project-laboratoria.dadoslaboratoria.track_in_spotify` AS t
153
154 SELECT
155   track_id,
156   track_name,
157   SAFE_CONVERT_BYTES_TO_STRING(CAST(artist_s_name AS BYTES)) AS artist_s_name,
158   SAFE_CONVERT_BYTES_TO_STRING(CAST(track_name AS BYTES)) AS track_name
159
160
161 `my-project-laboratoria.dadoslaboratoria.track_in_spotify` AS t

```

Esta consulta vai processar 0 B quando executada.

**Resultados da consulta**

Linha	track_name	artist_s_name	artist_s_name_1	track_name_1
20	Like Crazy	Jimin	Jimin	Like Crazy
21	LADY GAGA	Gabito Ballesteros, Junior H. Pe..	Gabito Ballesteros, Junior H. Pe..	LADY GAGA
22	I Can See You (Taylor&#039;s Ve..	Taylor Swift	Taylor Swift	I Can See You (Taylor&#039;s Ve..
23	I Wanna Be Yours	Arctic Monkeys	Arctic Monkeys	I Wanna Be Yours
24	Green Glitter - Green Movie Queen	Brennan Bahr Glitter	Brennan Bahr Glitter	Green Glitter - Green Movie Queen

Resultados por página: 50 ▾ 1 – 50 de 953

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

```

139 `my-project-laboratoria.dadoslaboratoria.track_in_spotify` AS t
140
141 SELECT
142   track_id,
143   REGEXP_REPLACE(track_name, r'[^\\x00-\\x7F]+', '') AS track_name,
144   REGEXP_REPLACE(artist_s_name, r'[^\\x00-\\x7F]+', '') AS artist_s_name,
145   artist_count,
146   released_year,
147   released_month,
148   released_day,
149   in_spotify_charts,
150   FROM
151 `my-project-laboratoria.dadoslaboratoria.track_in_spotify` AS t

```

Consulta concluída.

**Resultados da consulta**

Linha	track_id	track_name	artist_s_name	artist_count	released_year	released_month	released
164	6735267	Enchanted	Taylor Swift	1	2010	1	
165	1961503	Save Your Tears	The Weeknd	1	2020	3	
166	3451143	Sure Thing	Miguel	1	2010	5	
167	6450217	Every Breath You Take - Demise	The Online	1	1993	1	

Resultados por página: 200 ▾ 1 – 200 de 953



### 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.

```

1 SELECT
2   COUNT(*),
3   max(streams),
4   avg(streams)
5 FROM
6   `my-project-laboratoria.dadoslaboratoria.track_in_spotify_corrigido`
7
8 SELECT
9   max(in_spotify_playlists),
10  max(in_spotify_playlists),
11  avg(in_spotify_playlists)
12 FROM
13  `my-project-laboratoria.dadoslaboratoria.track_in_spotify_corrigido`
14

```

Consulta concluída.

**Resultados da consulta**

Linha	1	2	3
1	2762	3703895074	513675818.7901..

Resultados por página: 50 ▾ 1 – 1 de 1

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 of data:

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', and 'Compartilhar'. 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 of data:

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 artist_s_name,
3        COUNT(*) AS total_tracks
4     FROM `my-project-laboratoria.dadoslaboratoria.view_unificada`
5     GROUP BY artist_s_name
6 )
7
8 SELECT artist_s_name,
9       teste.total_tracks
10  FROM `my-project-laboratoria.dadoslaboratoria.view_unificada` AS s
11  LEFT JOIN teste
12    ON s.artist_s_name = teste.artist_s_name

```

Esta consulta vai processar 0 B quando executada.

**Resultados da consulta**

Informações do job	Resultados	Gráfico	JSON	Detalhes de execução	Gráfico de execução
	Resultados por página: 50 1 - 50 de 945				
	Atualizar				

```

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

```

Esta consulta vai processar 0 B quando executada.

**Resultados da consulta**

Informações do job	Resultados	Gráfico	JSON	Detalhes de execução	Gráfico de execução
	Resultados por página: 50 1 - 50 de 945				
	Atualizar				

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

```

Esta consulta vai processar 0 B quando executada.

**Resultados da consulta**

Informações do job	Resultados	Gráfico	JSON	Detalhes de execução	Gráfico de execução
	Resultados por página: 50 1 - 50 de 945				
	Atualizar				

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

DataSource.Error [ODBC: ERROR [42000] [Microsoft][BigQuery] [100] Error interacting with REST API Access Denied: BigQueryBigQuery: Permission denied while getting Drive credentials.

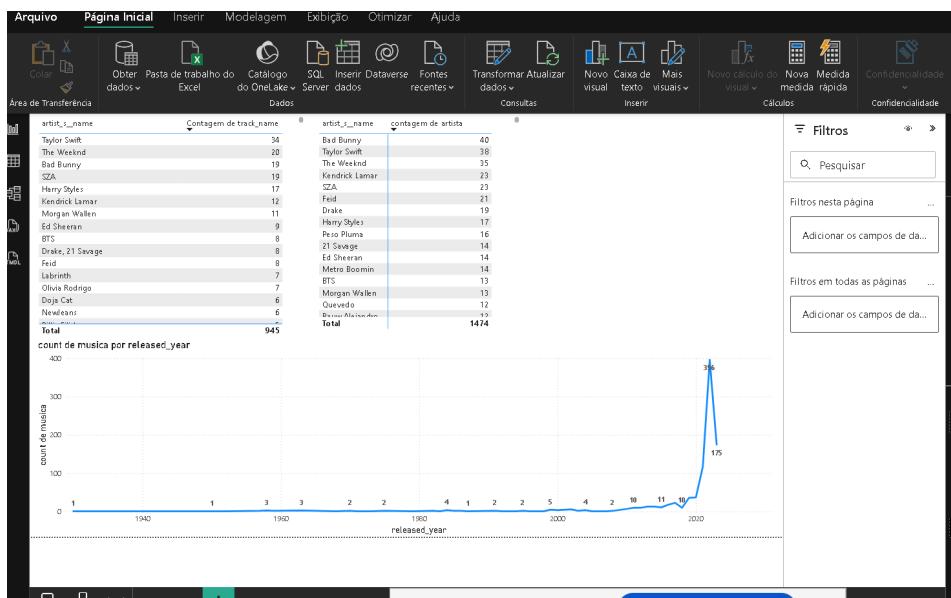
Details:

- DataSourceKind=GoogleBigQuery
- DataSourcePath=GoogleBigQuery
- DataSourceName=Table

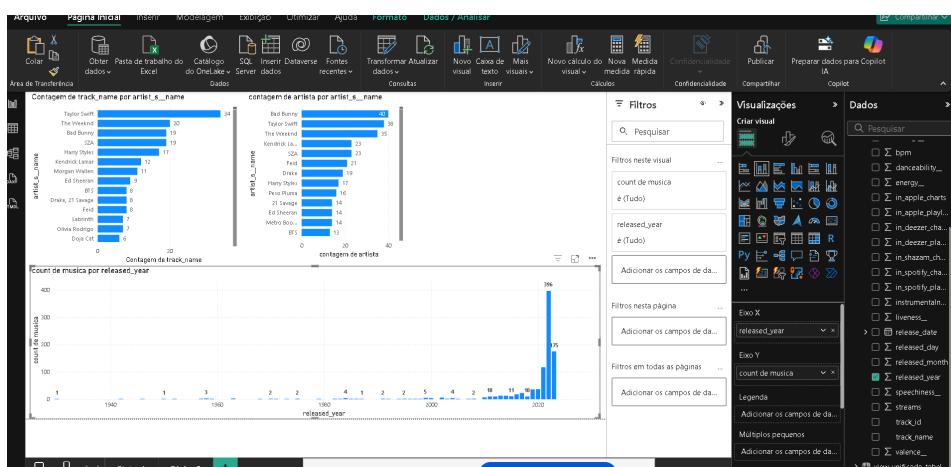
e resolvi criando uma versão tabela dessa view unificada.

track_id	track_name	artist_x_mi
1647815	Se La Ve	Arcangel, Ox
7239348	We Don't Talk About Bruno	Adassa, Mai
7870058	Cay La Noche (feat. Cris Cofun, Abhir Hathv, Bejo, El IMA)	Osevedo, Li
5350155	Nobody Like U - From "Turning Red"	Jordan Fisher
193046	Bastharan Rang (From "Pathaan")	Vishal Shekhar
6602768	Ihoomie to Pathaan	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
8337645	Rockin' Around The Christmas Tree	Andy Williams
8157749	Deck The Hall - Remastered 1999	Dean Martin
4227295	Let It Snow! Let It Snow! Let It Snow!	Andy Williams
5350503	It's the Most Wonderful Time of the Year	The Ronettes
6900052	Sleigh Ride	Dionne Warwick
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.

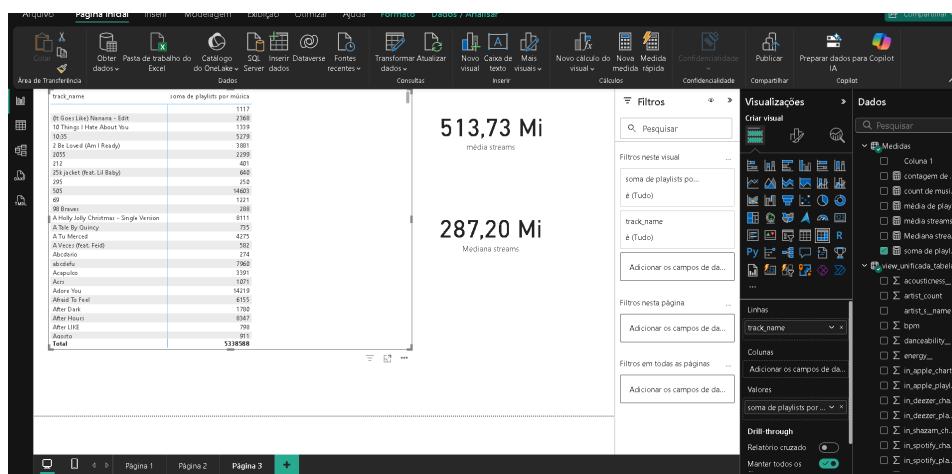


### 5.1.2 Ver variáveis categóricas

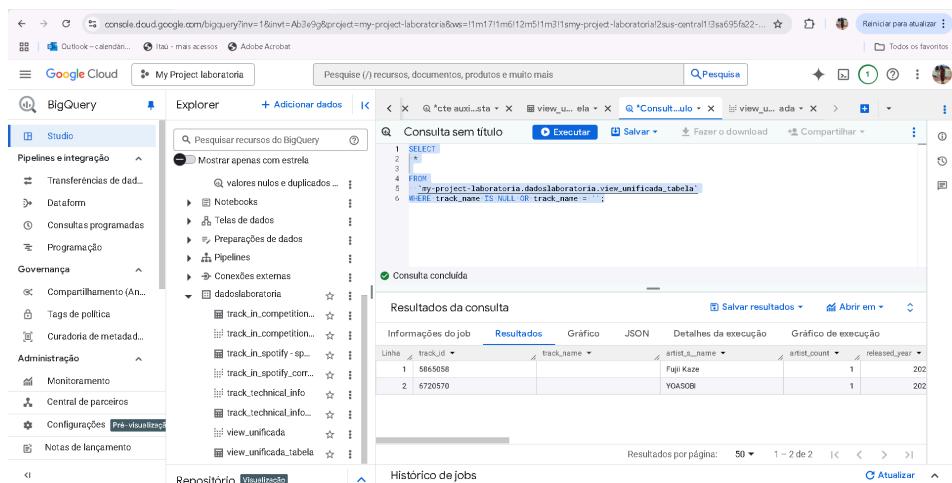


### 5.2.3 Aplicar medidas de tendência central

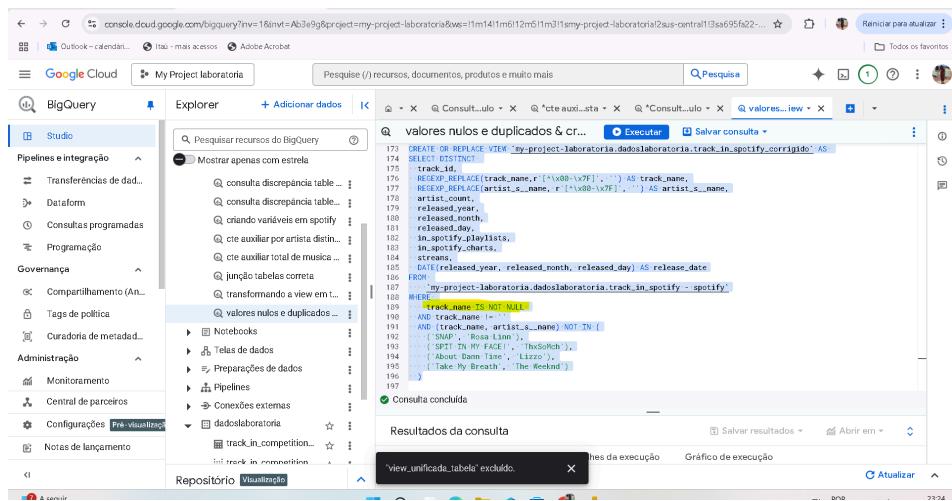
Ao tentar criar a soma de playlists(deezer,spotify, e apple music) verifiquei que existia duas musicas sem nome e consultei no big query para confirmar:



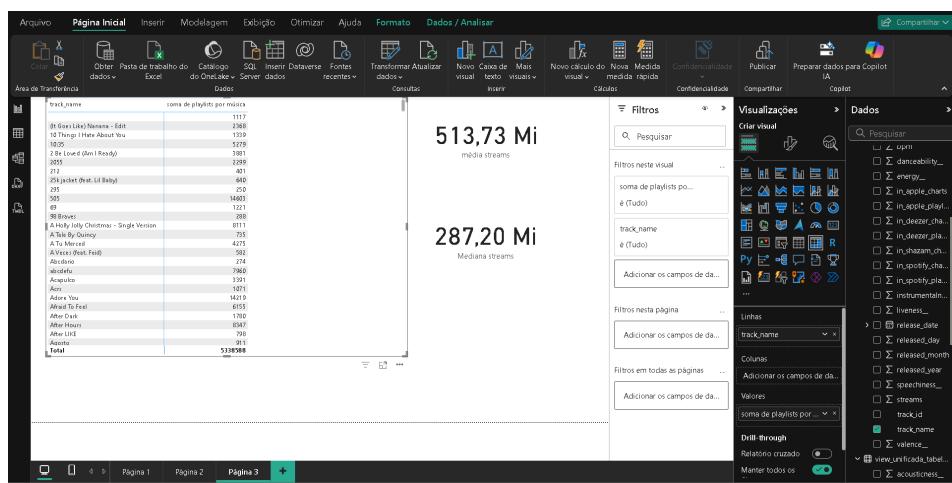
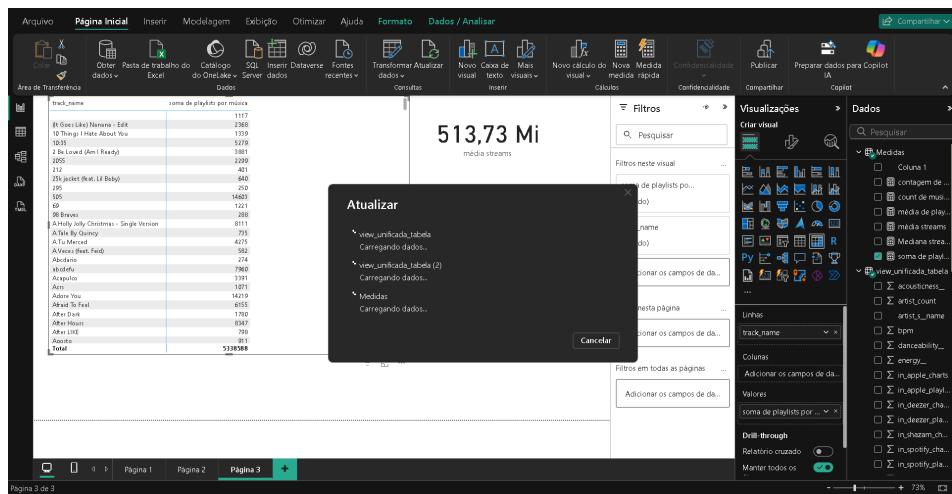
Quando usei o regex para alterar os caracteres não legíveis, terminei alterando por dados vazios.



Fiz o ajuste na própria view:



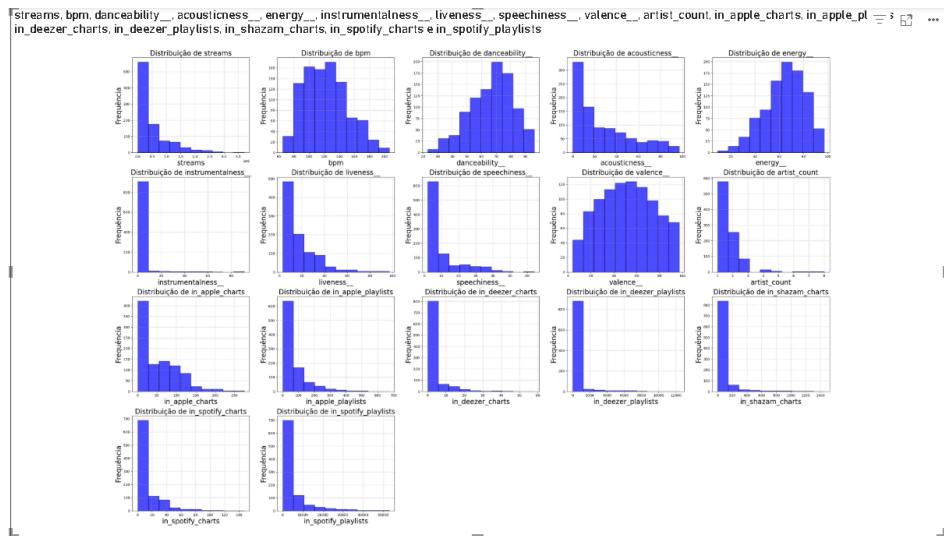
repiquei na view unificada e recriei a tabela unificada para leitura no bi



**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` ...

```

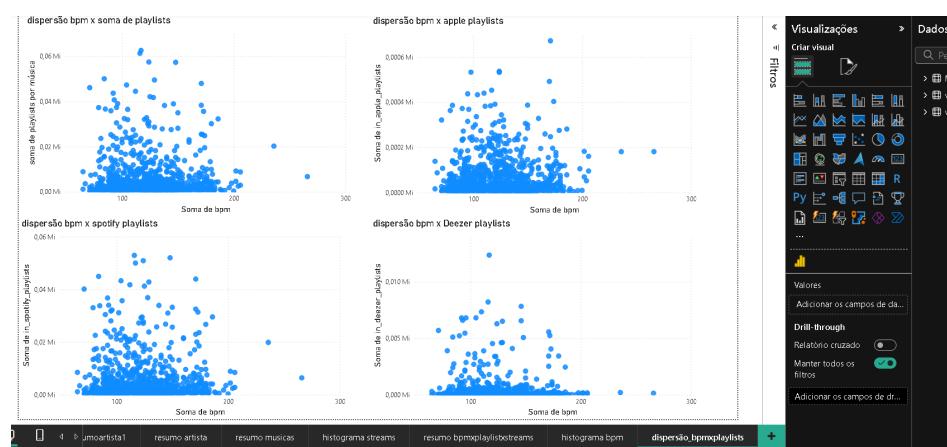
Consulta concluída

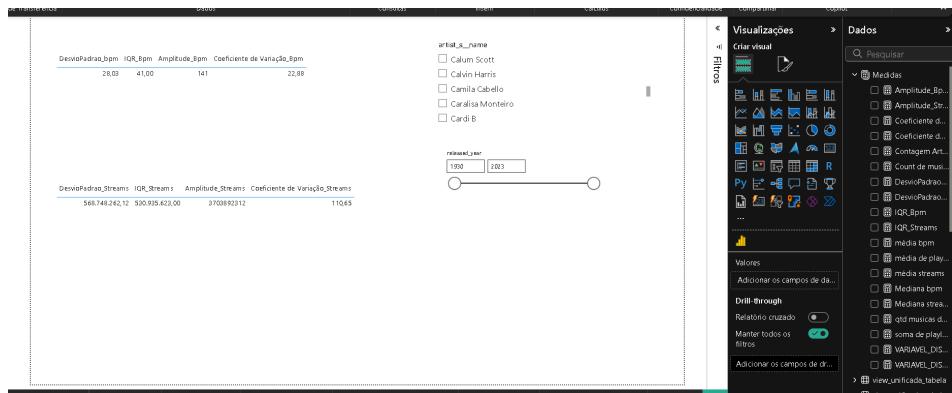
### Resultados da consulta

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%):**

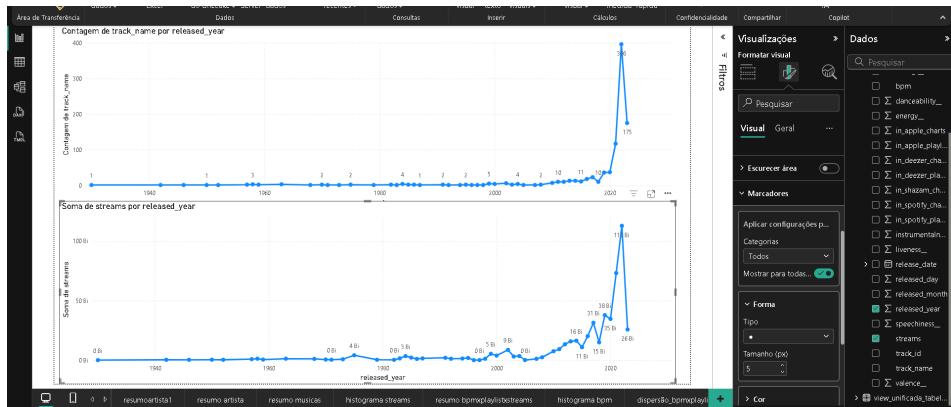
Um CV **acima de 100%** é raro e indica **alta dispersão relativa**. Isso significa que o desvio padrão é maior que a própria média, típico em dados com distribuição assimétrica e outliers.

### 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.

```

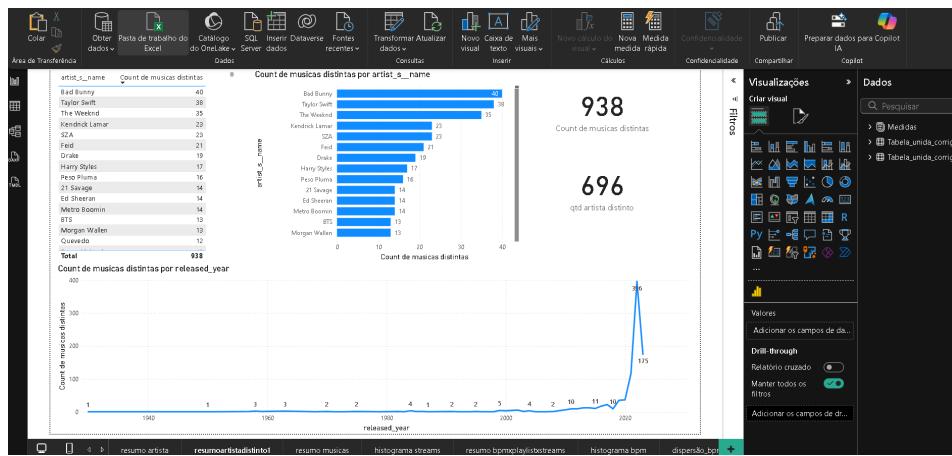
1 WITH quartiles AS (
2   SELECT
3     track_id,
4     streams,
5     NTILE(4) OVER (ORDER BY streams) AS quartiles_streams
6   FROM
7     `my-project-laboratoria.dadoslaboratoria.view.unificada_tabela`
8 )
9 SELECT
10   a.*,
11   q.quartiles_streams,
12   CASE
13     WHEN q.quartiles_streams = 1 THEN '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_unificados AS (
    -- 1. Unifica as três tabelas em uma base única
    SELECT
        -- Colunas da track_in_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,
        spot.cover_url,

        -- Colunas da track_in_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` AS spot
    LEFT JOIN
        `my-project-laboratoria.dadoslaboratoria.track_in_competition_view` AS comp
    ON spot.track_id = comp.track_id
)

```

```

LEFT JOIN
`my-project-laboratoria.dadoslaboratoria.track_technical_info` AS tech
-- CORREÇÃO: A junção deve ser feita com a tabela principal (spot) para garantir a consistência dos dados
ON spot.track_id = tech.track_id
),

dados_com_quartis AS (
-- 2. Calcula os quartis sobre a base unificada
SELECT
*, -- Seleciona todas as colunas da etapa anterior
NTILE(4) OVER (ORDER BY streams) AS quartil_streams,
NTILE(4) OVER (ORDER BY bpm) AS quartil_bpm,
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_unificados
)

-- 3. Seleciona todos os dados e adiciona as classificações textuais
SELECT
*,
CASE
WHEN quartil_bpm = 1 THEN 'Baixo'
WHEN quartil_bpm = 2 THEN 'Médio-Baixo'
WHEN quartil_bpm = 3 THEN 'Médio-Alto'
WHEN quartil_bpm = 4 THEN 'Alto'
END AS classificacao_bpm,

CASE
WHEN quartil_streams = 1 THEN 'Baixo'
WHEN quartil_streams = 2 THEN 'Médio-Baixo'
WHEN quartil_streams = 3 THEN 'Médio-Alto'
WHEN quartil_streams = 4 THEN 'Alto'
END AS classificacao_streams,

CASE
WHEN quartil_danceability = 1 THEN 'Baixo'
WHEN quartil_danceability = 2 THEN 'Médio-Baixo'
WHEN quartil_danceability = 3 THEN 'Médio-Alto'
WHEN quartil_danceability = 4 THEN 'Alto'
END AS classificacao_danceability,

CASE
WHEN quartil_valence = 1 THEN 'Baixo'
WHEN quartil_valence = 2 THEN 'Médio-Baixo'
WHEN quartil_valence = 3 THEN 'Médio-Alto'
WHEN quartil_valence = 4 THEN 'Alto'
END AS classificacao_valence,

CASE
WHEN quartil_energy = 1 THEN 'Baixo'
WHEN quartil_energy = 2 THEN 'Médio-Baixo'
WHEN quartil_energy = 3 THEN 'Médio-Alto'

```

```

WHEN quartil_energy = 4 THEN 'Alto'
END AS classificacao_energy,

CASE
WHEN quartil_acousticness = 1 THEN 'Baixo'
WHEN quartil_acousticness = 2 THEN 'Médio-Baixo'
WHEN quartil_acousticness = 3 THEN 'Médio-Alto'
WHEN quartil_acousticness = 4 THEN 'Alto'
END AS classificacao_acousticness,

CASE
WHEN quartil_instrumentalness = 1 THEN 'Baixo'
WHEN quartil_instrumentalness = 2 THEN 'Médio-Baixo'
WHEN quartil_instrumentalness = 3 THEN 'Médio-Alto'
WHEN quartil_instrumentalness = 4 THEN 'Alto'
END AS classificacao_instrumentalness,

CASE
WHEN quartil_liveness = 1 THEN 'Baixo'
WHEN quartil_liveness = 2 THEN 'Médio-Baixo'
WHEN quartil_liveness = 3 THEN 'Médio-Alto'
WHEN quartil_liveness = 4 THEN 'Alto'
END AS classificacao_liveness,

CASE
WHEN quartil_speechiness = 1 THEN 'Baixo'
WHEN quartil_speechiness = 2 THEN 'Médio-Baixo'
WHEN quartil_speechiness = 3 THEN 'Médio-Alto'
WHEN quartil_speechiness = 4 THEN 'Alto'
END AS classificacao_speechiness
FROM
dados_com_quartis;

```

	Esquema	Detalhes	Visualização	Buscador de tabelas	Visualização	Insights	Linhagem	Perfil de dados	Qualidade
	in_deezer_charts	INTEGER	NULLABLE	-	-	-	-	-	-
	in_shazam_charts	INTEGER	NULLABLE	-	-	-	-	-	-
	bpm	INTEGER	NULLABLE	-	-	-	-	-	-
	danceability_	INTEGER	NULLABLE	-	-	-	-	-	-
	valence_	INTEGER	NULLABLE	-	-	-	-	-	-
	energy_	INTEGER	NULLABLE	-	-	-	-	-	-
	acousticness_	INTEGER	NULLABLE	-	-	-	-	-	-
	instrumentalness_	INTEGER	NULLABLE	-	-	-	-	-	-
	liveness_	INTEGER	NULLABLE	-	-	-	-	-	-
	speechiness_	INTEGER	NULLABLE	-	-	-	-	-	-
	quantity_streams	INTEGER	NULLABLE	-	-	-	-	-	-
	classificacao_streams	STRING	NULLABLE	-	-	-	-	-	-

### 💡 5.2.8 Calcular correlação entre variáveis

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

```

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

```

```

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

```

## 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).

```

9
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
10 FROM `projeto02-laboratoria-musica.competition.aggregated_view` ;
1
2
3
4
5
6
7
```

Consulta concluída

Resultados da consulta

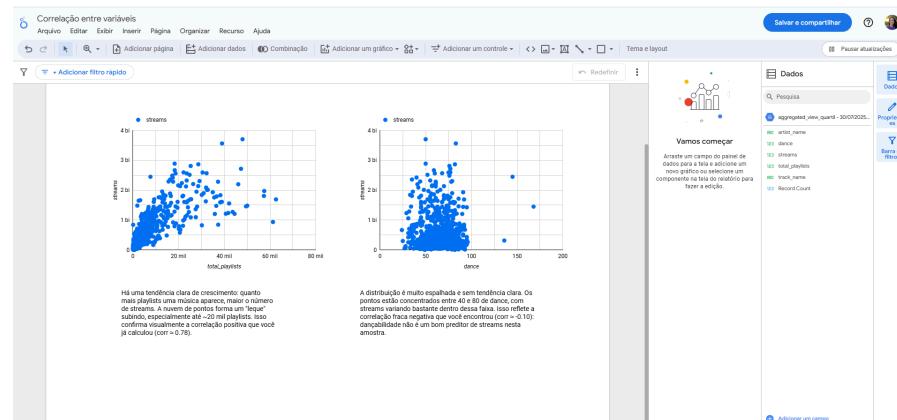
Informações do job    Resultados    Gráfico    JSON    Detalhes da execução    Gráfico de execução

	corr_value	corr_s_dance
1	0.784153628293...	-0.10545688369...

## 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).



Analisando os dados em gráficos através do Looker

**Resultados encontrados:**

**Gráfico 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 ( $\text{corr} \approx 0.78$ ).

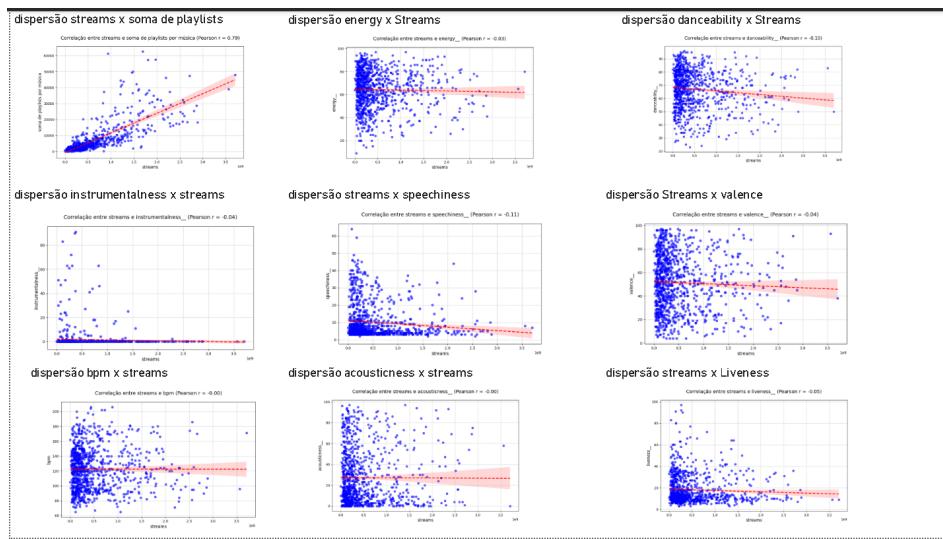
**Gráfico 2 -**

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 ( $\text{corr} \approx -0.10$ ): 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.

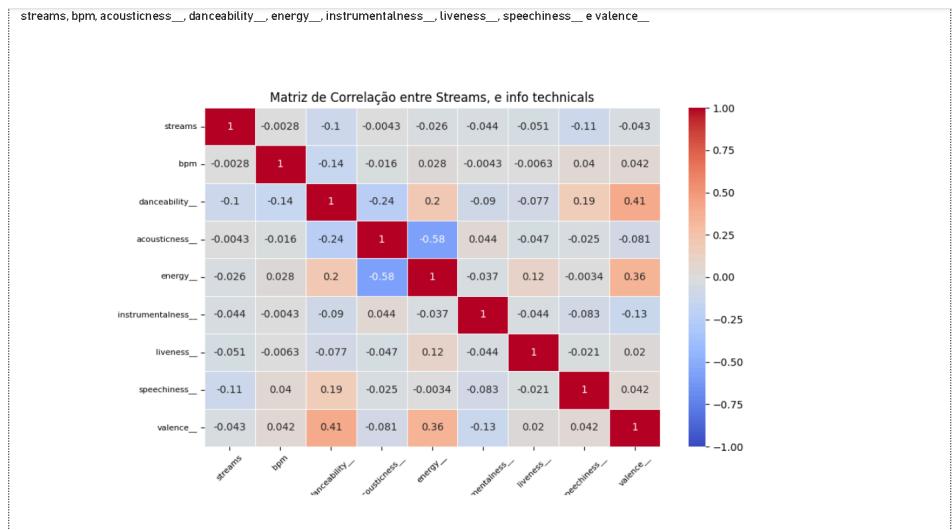
### 3. Diferença em Relação ao Anterior ( $r = -0.79$ ):

- 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.



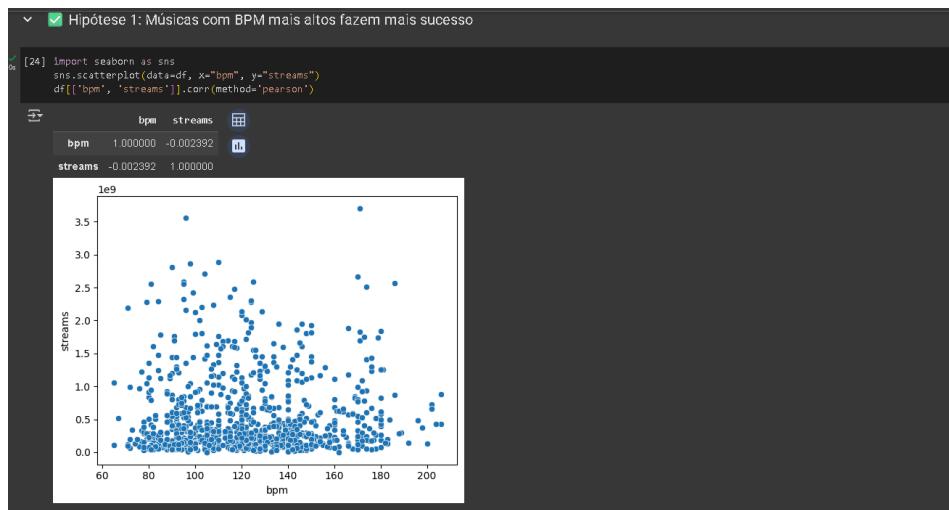
### 5.3 Aplicar técnica de análise

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).

```
[ ] alta_group = df[df['classificacao_bpm'].isin(categorias_altas)]['streams']  
# Lista de categorias que você considera como 'grupo alto'  
categorias_baixas = ['Médio-Baixo', 'Baixo']  
  
# O grupo 'Baixa_group' agora filtra pelas duas categorias usando .isin()  
# e seleciona a coluna 'streams'  
baixa_group = df[df['classificacao_bpm'].isin(categorias_baixas)]['streams']  
  
# Execute o teste de Mann-Whitney U  
estatistica, p_value = mannwhitneyu(alta_group, baixa_group, alternative='two-sided')  
  
# Imprima os resultados  
print("Mann-Whitney U statistic: {estatistica:.4f}")  
print("P-value: {p_value:.4f}")  
  
# Verifique se o p-value é significativo (por exemplo, menor que 0.05)  
if p_value < 0.05:  
    print("A diferença entre os grupos 'alto' e 'baixo' da característica bpm é estatisticamente significativa.")  
else:  
    print("Não há diferença estatisticamente significativa entre os grupos 'alto' e 'baixo' da característica bpm.")  
  
Mann-Whitney U statistic: 50471.5000  
P-value: 0.0462  
A diferença entre os grupos 'alto' e 'baixo' da característica bpm é estatisticamente significativa.
```

```

--- RESUMO DA REGRESSÃO ---
OLS Regression Results
=====
Dep. Variable: streams R-squared: 0.000
Model: OLS Adj. R-squared: -0.001
Method: Least Squares F-statistic: 0.01388
Date: Fri, 15 Aug 2025 Prob (F-statistic): 0.906
Time: 19:04:05 Log-Likelihood: -20348.
No. Observations: 943 AIC: 4.070e+04
Df Residuals: 941 BIC: 4.071e+04
Df Model: 1
Covariance Type: nonrobust
=====
      coef  std err      t      P>|t|      [0.025      0.975]
const  5.24e+08  8.32e+07   6.301      0.000   3.61e+08  6.87e+08
bpm    -7.798e+04  6.62e-05  -0.118      0.906  -1.38e+06  1.22e+06
=====
Omnibus: 381.912 Durbin-Watson: 1.099
Prob(Omnibus): 0.000 Jarque-Bera (JB): 1345.567
Skew: 1.991 Prob(JB): 6.52e-293
Kurtosis: 7.289 Cond. No. 564.
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```



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;

```

? Hipótese 2: Popularidade no Spotify é semelhante em outras plataformas
[25]: df[['in_apple_charts', 'in_deezer_charts', 'in_shazam_charts', 'in_spotify_charts']].corr(method='spearman')

```

	in_apple_charts	in_deezer_charts	in_shazam_charts	in_spotify_charts
in_apple_charts	1.000000	0.409615	0.497791	0.524224
in_deezer_charts	0.409615	1.000000	0.420882	0.592444
in_shazam_charts	0.497791	0.420882	1.000000	0.544350
in_spotify_charts	0.524224	0.592444	0.544350	1.000000

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.

```
[ ] # CORRELAÇÃO PARA A HIPÓTESE 2

import pandas as pd

X1 = df['in_spotify_charts']
X2 = df['in_apple_charts']
y = df['in_deezer_charts']

# Calculando a correlação entre in_spotify_charts e in_deezer_charts
correlacao_spotify_deezer = X1.corr(y)

# Calculando a correlação entre in_spotify_charts e in_apple_charts
correlacao_spotify_apple = X1.corr(X2)

print("Correlação entre Spotify e Deezer:", correlacao_spotify_deezer)
print("Correlação entre Spotify e Apple:", correlacao_spotify_apple)

➡ Correlação entre Spotify e Deezer: 0.6096106053237494
Correlação entre Spotify e Apple: 0.5524439549036637
```

```
[ ] 
    print("Não há evidências suficientes para rejeitar a hipótese nula. As distribuições são semelhantes.")
    print("\n" + "="*80 + "\n")

# --- Teste de Mann-Whitney para APPLE ---
print("### Teste: Músicas em Spotify Charts vs. Demais Charts (para Apple) ###")

# A variável a ser comparada é a mesma nos dois grupos: 'in_apple_charts'
stat_apple, p_valor_apple = stats.mannwhitneyu(grupo_spotify_charts['in_apple_charts'], grupo_demais_charts['in_apple_charts'])

print("Estatística de teste de Mann-Whitney:", stat_apple)
print("Valor p:", p_valor_apple)

if p_valor_apple < 0.05:
    print("Há evidências suficientes para rejeitar a hipótese nula. As distribuições de 'in_apple_charts' são diferentes entre os grupos.")
else:
    print("Não há evidências suficientes para rejeitar a hipótese nula. As distribuições são semelhantes.")

➡ ### Teste: Músicas em Spotify Charts vs. Demais Charts (para Apple) ###
Estatística de teste de Mann-Whitney: 1315.0
Valor p: 0.0215827350231086
Há evidências suficientes para rejeitar a hipótese nula. As distribuições de 'in_apple_charts' são diferentes entre os grupos.

-----
### Teste: Músicas em Spotify Charts vs. Demais Charts (para Deezer) ###
Estatística de teste de Mann-Whitney: 1090.0
Valor p: 0.720829792312712
Não há evidências suficientes para rejeitar a hipótese nula. As distribuições são semelhantes.
```

```
OLS Regression Results
=====
Dep. Variable:      in_deezer_charts   R-squared:                  0.054
Model:                          OLS   Adj. R-squared:                0.044
Method:                         Least Squares   F-statistic:                 5.542
Date:                 Fri, 15 Aug 2025   Prob (F-statistic):        0.0206
Time:                      17:01:39   Log-Likelihood:            -69.809
No. Observations:          100   AIC:                         143.6
Df Residuals:                 98   BIC:                         148.8
Df Model:                      1
Covariance Type:            nonrobust
=====
            coef    std err      t    P>|t|      [0.025    0.975]
-----
const       0.4143     0.059    7.055    0.000      0.298     0.531
in_spotify_charts   0.2524     0.107    2.354    0.021      0.040     0.465
-----
Omnibus:             1165.890   Durbin-Watson:           1.827
Prob(Omnibus):        0.000    Jarque-Bera (JB):        13.214
Skew:                  0.060    Prob(JB):                  0.00135
Kurtosis:                 1.223    Cond. No.                   2.42
-----
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

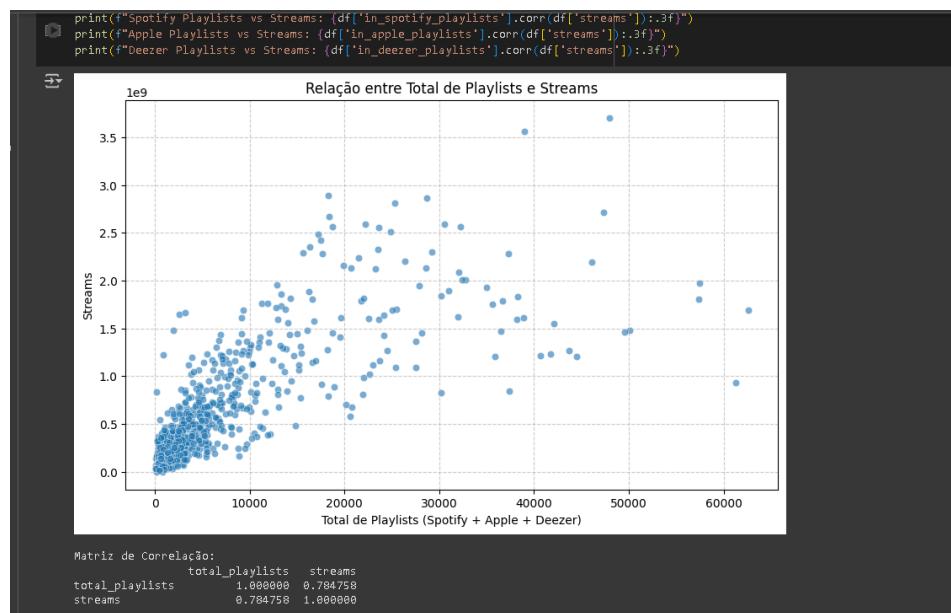
Gráfico de Dispersão - Spotify charts vs. Deezer charts

Spotify charts	Deezer charts
1.0	1.0

✓ 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;



```
[ ] # TESTE DE MANN-WHITNEY PARA TESTAR A HIPÓTESE 3 (MÚSICA EM PLAYLISTS X STREAMS)

import scipy.stats as stats
import pandas as pd

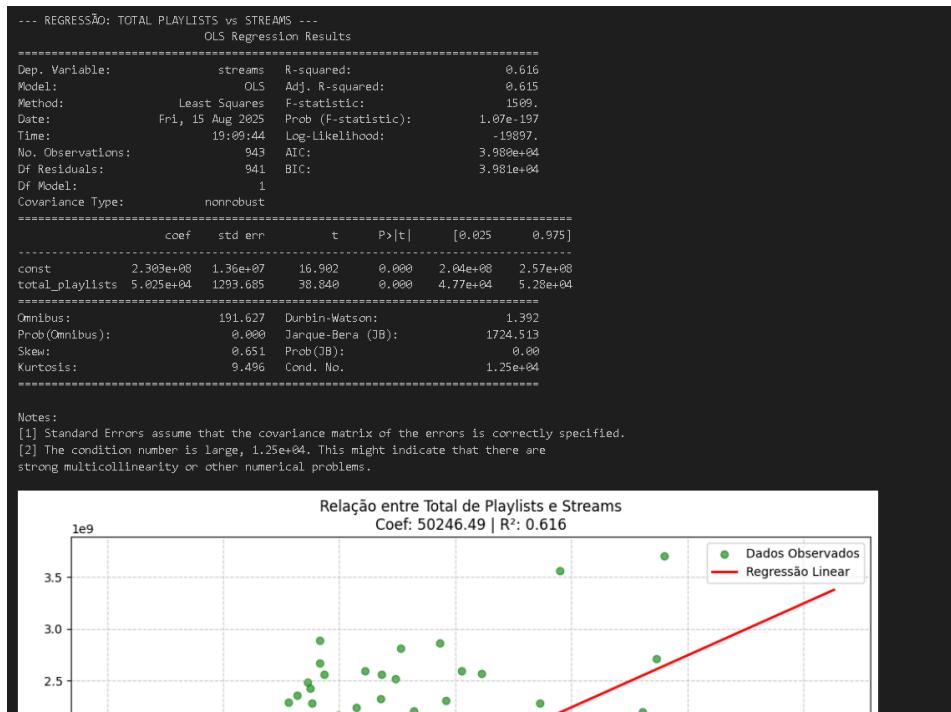
# Separando os dados em dois grupos com base na variável soma_playlists
grupo_maior_playlists = df[df['total_playlists'] > df['total_playlists'].median()]
grupo_menor_playlists = df[df['total_playlists'] <= df['total_playlists'].median()]

# Realizando o teste de Mann-Whitney
stat, p_valor = stats.mannwhitneyu(grupo_maior_playlists['streams'], grupo_menor_playlists['streams'])

# Interpretando os resultados
print("Estatística de teste de Mann-Whitney:", stat)
print("Valor p:", p_valor)

if p_valor < 0.05:
    print("Há evidências suficientes para rejeitar a hipótese nula, ou seja, há diferença significativa entre os grupos.")
else:
    print("Não há evidências suficientes para rejeitar a hipótese nula, ou seja, não há diferença significativa entre os grupos.")

Estatística de teste de Mann-Whitney: 206546.0
Valor p: 3.6552893778775378e-115
Há evidências suficientes para rejeitar a hipótese nula, ou seja, há diferença significativa entre os grupos.
```



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	Styx	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
```

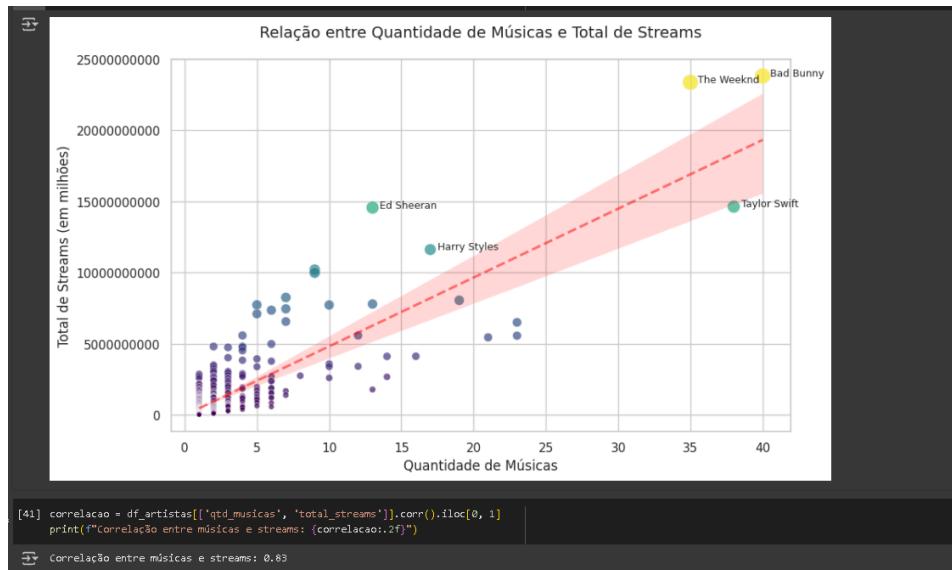
- Calcular o indicadores da hipótese

```
[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:



```
[ ] # TESTE DE MANN-WHITNEY PARA TESTAR A HIPÓTESE 4 (track_id, artist_name x streams)

import pandas as pd
from scipy.stats import mannwhitneyu

# Agrupar os dados pela variável "artist_s_name" e calcular o número de músicas e a soma dos streams
grouped_data = df.groupby('artist_s_name').agg({'track_id': 'count', 'streams': 'sum'})

# Dividir os dados em dois grupos com base no número de músicas
median_tracks = grouped_data['track_id'].median()
artists_with_more_tracks = grouped_data[grouped_data['track_id'] > median_tracks]
artists_with_less_tracks = grouped_data[grouped_data['track_id'] <= median_tracks]

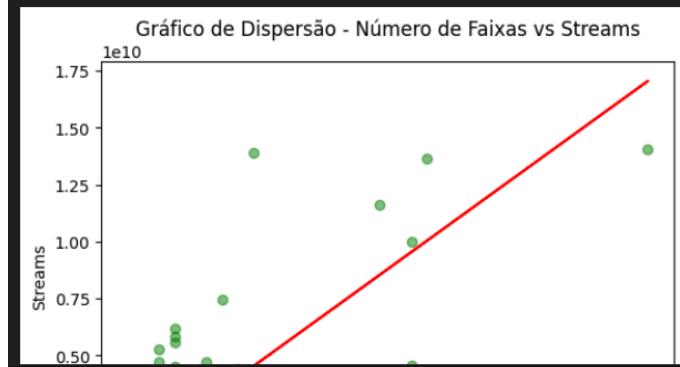
# Aplicar o teste de Mann-Whitney
statistic, p_value = mannwhitneyu(artists_with_more_tracks['streams'], artists_with_less_tracks['streams'])
print("Estatística de teste de Mann-Whitney:", statistic)
print("Valor p:", p_value)

# Interpretando os resultados
alpha = 0.05
if p_value < alpha:
    print("Há uma diferença significativa nos streams entre os dois grupos.")
else:
    print("Não há evidências suficientes para concluir que há uma diferença significativa nos streams entre os dois grupos.")

→ Estatística de teste de Mann-Whitney: 46088.0
Valor p: 3.196434552693207e-28
Há uma diferença significativa nos streams entre os dois grupos.
```

OLS Regression Results						
Dep. Variable:	streams	R-squared:	0.603			
Model:	OLS	Adj. R-squared:	0.602			
Method:	Least Squares	F-statistic:	969.7			
Date:	Fri, 15 Aug 2025	Prob (F-statistic):	3.14e-130			
Time:	19:15:10	Log-Likelihood:	-14110.			
No. Observations:	641	AIC:	2.822e+04			
Df Residuals:	639	BIC:	2.823e+04			
Df Model:	1					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
const	2.019e+07	4.2e+07	0.480	0.631	-6.24e+07	1.03e+08
track_id	5.007e+08	1.61e+07	31.140	0.000	4.69e+08	5.32e+08
<hr/>						
Omnibus:	469.260	Durbin-Watson:	1.944			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	17813.827			
Skew:	2.782	Prob(JB):	0.00			
Kurtosis:	28.220	Cond. No.	3.32			
<hr/>						

Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



📊 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))

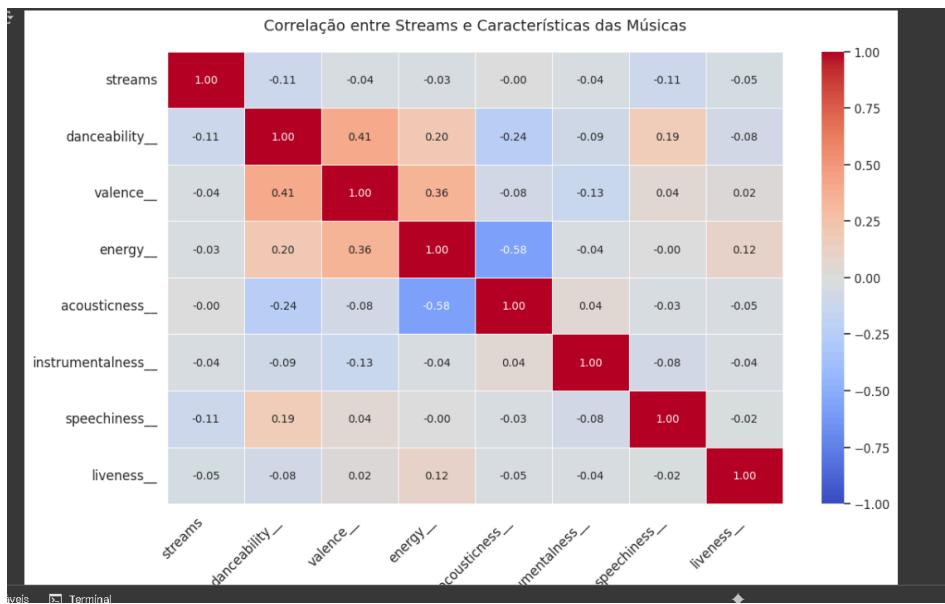
# Heatmap com anotações e mapa de cores
sns.heatmap(
    corr_matrix,
    annot=True,           # Mostra valores dentro dos quadrados
    cmap='coolwarm',      # Mapa de cores (quente/frio)
    vmin=-1, vmax=1,      # Limites da escala de cores (-1 a 1 para correlação)
    center=0,             # Centraliza 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()
```



```
[48] # Separe os dados em categorias 'alta' e 'baixa' para cada variável
alta_group = df[df['classificacao_(var)'].isin(categorias_altas)]['streams']
baixa_group = df[df['classificacao_(var)'].isin(categorias_baixas)]['streams']

# Teste de Mann-Whitney U
estatística, p_value = mannwhitneyu(alta_group, baixa_group, alternative='two-sided')

# Armazenando o p-valor no dicionário
p_values[var] = p_value

# p-valor para cada variável
for var, p_value in p_values.items():
    print(f"p-valor para {var}: ({p_value:.4f})")

# Condições para cada p-valor
if p_value < 0.05:
    print(f"A diferença nas medianas de streams entre os grupos 'alto' e 'baixo' da característica {var} é estatisticamente significativa.")
else:
    print(f"Não há diferença estatisticamente significativa nas medianas de streams entre os grupos 'alto' e 'baixo' da característica {var}.")
```

P-value para danceability: 0.6118  
 Não há diferença estatisticamente significativa nas medianas de streams entre os grupos 'alto' e 'baixo' da característica danceability.  
 P-value para valence: 0.5620  
 Não há diferença estatisticamente significativa nas medianas de streams entre os grupos 'alto' e 'baixo' da característica valence.  
 P-value para energy: 0.9684  
 Não há diferença estatisticamente significativa nas medianas de streams entre os grupos 'alto' e 'baixo' da característica energy.  
 P-value para acousticness: 0.0995  
 Não há diferença estatisticamente significativa nas medianas de streams entre os grupos 'alto' e 'baixo' da característica acousticness.  
 P-value para instrumentalness: 0.1092  
 Não há diferença estatisticamente significativa nas medianas de streams entre os grupos 'alto' e 'baixo' da característica instrumentalness.  
 P-value para speechiness: 0.1723  
 Não há diferença estatisticamente significativa nas medianas de streams entre os grupos 'alto' e 'baixo' da característica speechiness.  
 P-value para liveness: 0.7672  
 Não há diferença estatisticamente significativa nas medianas de streams entre os grupos 'alto' e 'baixo' da característica liveness.

```

OLS Regression Results
-----
Dep. Variable: streams R-squared:  0.029
Model: OLS Adj. R-squared:  0.022
Method: Least Squares F-statistic:  4.048
Date: Fri, 15 Aug 2025 Prob (F-statistic):  0.00026
Time: 19:19:50 Log-Likelihood: -20334.
No. Observations: 943 AIC: 4.060e+04
Df Residuals: 935 BIC: 4.072e+04
Df Model: 7
Covariance Type: nonrobust
-----
            coef std err      t      P>|t|    [0.025  0.975]
const      0.039e+00  1.4e+00   7.125  0.000   7.2e+00  1.27e-09
danceability_- -0.002e+00  1.44e+00  -2.830  0.005  -6.91e+00 -1.25e-06
valence_     0.137e+00  9.31e-05  0.154  0.877  -1.68e+00  1.97e-06
energy_       -1.097e+00  1.408e+00  -0.771  0.459  -1.87e+00  1.01e-06
acousticness_- -0.002e+00  2.21e+00  -0.125  0.905  -2.57e+00  6.51e-05
instrumentalness_- -4.259e+00  2.21e+00  -1.928  0.054  -8.50e+00  7.09e-04
liveness_     -2.605e+00  1.36e+00  -1.919  0.055  -5.27e+00  5.81e-04
speechiness_- -5.004e+00  1.88e+00  -3.084  0.002  -9.5e+00 -2.11e-06
-----
Omnibus: 372.341 Durbin-Watson:  1.127
Prob(Omnibus): 0.000 Jarque-Bera (JB): 1386.047
Skew: 1.941 Prob(JB): 1.59e-284
Kurtosis: 7.265 Cond. No. 863.
-----
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

● Diagnóstico dos Dados:
    Músicas analisadas: 943/943
    Músicas com streams:
        streams: 1.000
        acousticness_-: -0.005
        energy_-: -0.026
        valence_-: -0.042
        instrumentalness_-: -0.044
        liveness_-: -0.051

```

## 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

```

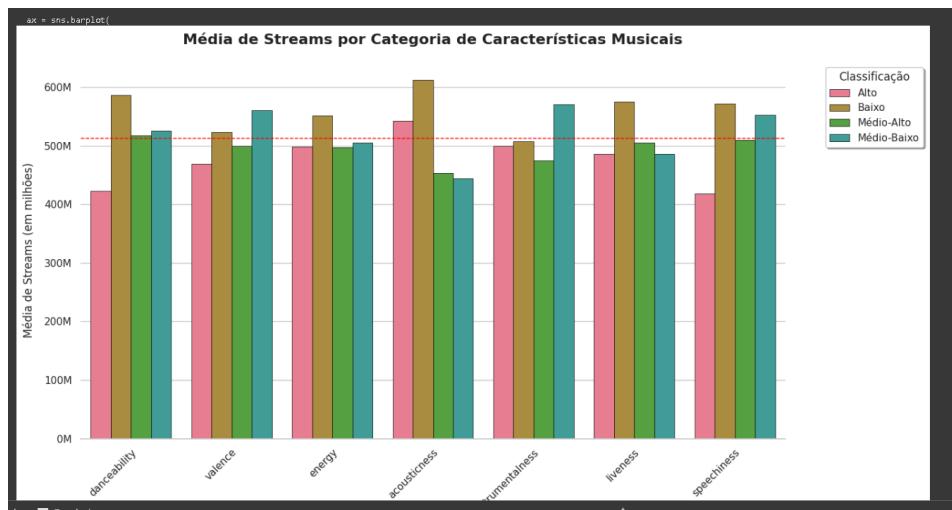
[50] caracteristicas = [
    'classificacao_danceability', 'classificacao_valence', 'classificacao_energy',
    'classificacao_acousticness', 'classificacao_instrumentalness',
    'classificacao_liveness', 'classificacao_speechiness'
]

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.

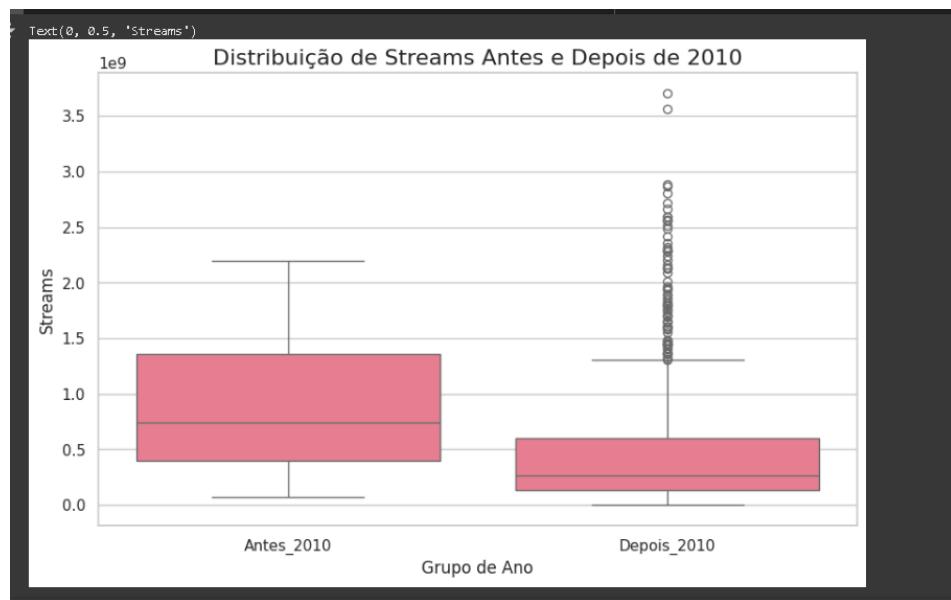
```
[44] características = ['danceability_', 'valence_', 'energy_', 'acousticness_',
    'instrumentalness_', 'liveness_', 'speechiness_']
df.groupby('grupo_ano')[características].mean()
```

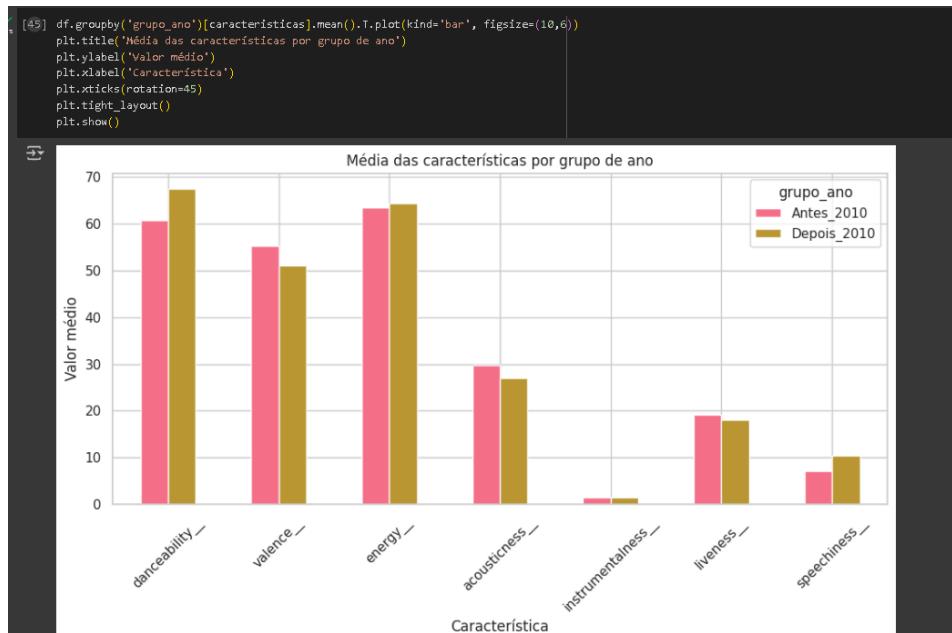
grupo_ano	danceability_	valence_	energy_	acousticness_	instrumentalness_	liveness_	speechiness_
Antes_2010	60.73913	55.333333	63.42029	29.782609	1.565217	19.144928	7.072464
Depois_2010	67.413714	51.018266	64.284571	27.033143	1.573714	18.041143	10.416

```
grupo_anter = df[df['grupo_ano'] == 'Antes_2010']['streams']
grupo_depois = df[df['grupo_ano'] == 'Depois_2010']['streams']

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

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\_anter 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=evJrljoiNjq5NmQ4NDctM2NJS00MdC3LWFhOGEtNDk0NzMoNihjYTnKliwidCl6ljhjZTBkOTY1LWE1NTktNDYyNC1iNT

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

Colab validação hipóteses: <https://colab.research.google.com/drive/19k12DbT36730i4KpUx1zMKqI8AXI-QIH#scrollTo=vSTLqLo7R6>

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

Repositório github: [https://github.com/jmxavier-1993/Spotify\\_hipotesis](https://github.com/jmxavier-1993/Spotify_hipotesis)