In [1]:
```
!pip install spotipy
!pip install pivottablejs
```

```
Requirement already satisfied: spotipy in /opt/anaconda3/lib/python3.9/site-packages (2.22.1)
Requirement already satisfied: redis>=3.5.3 in /opt/anaconda3/lib/python3.9/site-packages (from spotipy) (4.5.1)
Requirement already satisfied: six>=1.15.0 in /opt/anaconda3/lib/python3.9/site-packages (from spotipy) (1.16.0)
Requirement already satisfied: requests>=2.25.0 in /opt/anaconda3/lib/python3.9/site-packages (from spotipy) (2.27.1)
Requirement already satisfied: urllib3>=1.26.0 in /opt/anaconda3/lib/python3.9/site-packages (from spotipy) (1.26.9)
Requirement already satisfied: async-timeout>=4.0.2 in /opt/anaconda3/lib/python3.9/site-packages (from redis>=3.5.3->spotipy) (4.
0.2)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/lib/python3.9/site-packages (from requests>=2.25.0->spotipy)
(2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in /opt/anaconda3/lib/python3.9/site-packages (from requests>=2.25.0->spotipy) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/anaconda3/lib/python3.9/site-packages (from requests>=2.25.0->spo
tipy) (2.0.4)
Requirement already satisfied: pivottablejs in /opt/anaconda3/lib/python3.9/site-packages (0.9.0)
```

In [2]:
```
import spotipy
import requests
import spotipy.util as util
import pandas as pd
import matplotlib.pyplot as plt
from pivottablejs import pivot_ui
```

In [33]:
```
CLIENT_ID = 'Spotify Credential.ipynb/CLIENT_ID'
CLIENT_SECRET = 'Spotify Credential.ipynb/CLIENT_SECRET'


AUTH_URL = 'https://accounts.spotify.com/api/token'
```

In [34]:
```
from spotipy.oauth2 import SpotifyClientCredentials
client_credentials_manager = SpotifyClientCredentials(client_id=CLIENT_ID, client_secret=CLIENT_SECRET)
sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)
```

In [35]:
```
artist_name = []
track_name = []
track_popularity = []
artist_id = []
track_id = []
album_name = []
for i in range(0,1000,50):
    track_results = sp.search(q='year:2023', type='track', limit=50,offset=i)
    for i, t in enumerate(track_results['tracks']['items']):
        artist_name.append(t['artists'][0]['name'])
        artist_id.append(t['artists'][0]['id'])
        track_name.append(t['name'])
        track_id.append(t['id'])
        track_popularity.append(t['popularity'])
        album_name.append(t['album']['name'])
```

In [36]:
```
track_df = pd.DataFrame({'artist_name' : artist_name, 'track_name' : track_name, 'track_id' : track_id, 'track_popularity' : track_p
print(track_df.shape)
track_df.head()
```

```
(1000, 6)
```

Out[36]:

	artist_name	track_name	track_id	track_popularity	artist_id	album_name
0	PinkPantheress	Boy's a liar Pt. 2	6AQbmUe0Qwf5PZnt4HmTXv	95	78rUTD7y6Cy67W1RVzYs7t	Boy's a liar Pt. 2
1	Miley Cyrus	Flowers	0yLdNVWF3Srea0uzk55zFn	100	5YGY8feqx7naU7z4HrwZM6	Flowers
2	Morgan Wallen	Last Night	59uQI0PADDKeE6UZDTJEe8	87	4oUHIQIBe0LHzYfvXNW4QM	3 Songs At A Time Sampler
3	The Weeknd	Die For You - Remix	7oDd86yk8itslrA9HRP2ki	94	1Xyo4u8uXC1ZmMpatF05PJ	Die For You (Remix)
4	Morgan Wallen	Last Night	7K3BhSpAxZBznislvUMVtn	81	4oUHIQIBe0LHzYfvXNW4QM	One Thing At A Time

In [7]:
```
len(track_df)
```

Out[7]:
```
1000
```

In [8]:
```
artist_popularity = []
artist_genres = []
artist_followers = []
for a_id in track_df.artist_id:
  artist = sp.artist(a_id)
  artist_popularity.append(artist['popularity'])
  artist_genres.append(artist['genres'])
  artist_followers.append(artist['followers']['total'])
```

In [9]:
```
track_df = track_df.assign(artist_popularity=artist_popularity, artist_genres=artist_genres, artist_followers=artist_followers)
track_df.head()
```

Out[9]:

	artist_name	track_name	track_id	track_popularity	artist_id	album_name	artist_popularity	artist_genres	artist_followers
0	PinkPantheress	Boy's a liar Pt. 2	6AQbmUe0Qwf5PZnt4HmTXv	95	78rUTD7y6Cy67W1RVzYs7t	Boy's a liar Pt. 2	86	[]	1843782
1	Miley Cyrus	Flowers	0yLdNVWF3Srea0uzk55zFn	100	5YGY8feqx7naU7z4HrwZM6	Flowers	93	[pop]	20013675
2	Morgan Wallen	Last Night	59uQl0PADDKeE6UZDTJEe8	87	4oUHIQIBe0LHzYfvXNW4QM	3 Songs At A Time Sampler	94	[contemporary country]	5214795
3	The Weeknd	Die For You - Remix	7oDd86yk8itslrA9HRP2ki	94	1Xyo4u8uXC1ZmMpatF05PJ	Die For You (Remix)	99	[canadian contemporary r&b, canadian pop, pop]	60625943
4	Morgan Wallen	Last Night	7K3BhSpAxZBznislvUMVtn	81	4oUHIQIBe0LHzYfvXNW4QM	One Thing At A Time	94	[contemporary country]	5214795

In [10]:
```
track_features = []
for t_id in track_df['track_id']:
    af = sp.audio_features(t_id)
    track_features.append(af)
print(pd.DataFrame(track_features))
```

```
                                                      0
0     {'danceability': 0.696, 'energy': 0.809, 'key'...
1     {'danceability': 0.707, 'energy': 0.681, 'key'...
2     {'danceability': 0.517, 'energy': 0.675, 'key'...
3     {'danceability': 0.531, 'energy': 0.525, 'key'...
4     {'danceability': 0.492, 'energy': 0.675, 'key'...
..                                                  ...
995   {'danceability': 0.928, 'energy': 0.468, 'key'...
996   {'danceability': 0.758, 'energy': 0.786, 'key'...
997   {'danceability': 0.816, 'energy': 0.955, 'key'...
998   {'danceability': 0.622, 'energy': 0.257, 'key'...
999   {'danceability': 0.622, 'energy': 0.257, 'key'...

[1000 rows x 1 columns]
```

In [11]:
```
len(track_features)
```

Out[11]: 1000

In [12]:
```
print(track_features[0][0].keys())
```

```
dict_keys(['danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'va
lence', 'tempo', 'type', 'id', 'uri', 'track_href', 'analysis_url', 'duration_ms', 'time_signature'])
```

In [13]:
```
tf_df = pd.DataFrame(columns = ['danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalnes
```

In [14]:
```
for element in track_features:
  for feat in element:
    tf_df.loc[len(tf_df)] = feat
print(tf_df)
#tf_df = pd.concat([feat,tf_df], ignore_index= True)
```

```
     danceability  energy  key  loudness  mode  speechiness  acousticness  \
0           0.696   0.809    5    -8.254     1       0.0500        0.2520
1           0.707   0.681    0    -4.325     1       0.0668        0.0632
2           0.517   0.675    6    -5.382     1       0.0357        0.4590
3           0.531   0.525    1    -6.500     0       0.0671        0.2320
4           0.492   0.675    6    -5.456     1       0.0389        0.4670
..            ...     ...  ...       ...   ...          ...           ...
995         0.928   0.468    2    -8.196     1       0.3640        0.3280
996         0.758   0.786    0    -4.847     1       0.0928        0.1670
997         0.816   0.955    1    -1.732     1       0.3600        0.0828
998         0.622   0.257    0    -9.811     1       0.0293        0.4780
999         0.622   0.257    0    -9.811     1       0.0293        0.4780

     instrumentalness  liveness  valence    tempo            type  \
0            0.000128    0.2480    0.857  132.962  audio_features
1            0.000005    0.0322    0.646  117.999  audio_features
2            0.000000    0.1510    0.518  203.853  audio_features
3            0.000000    0.4410    0.502   66.900  audio_features
4            0.000000    0.1420    0.478  203.759  audio_features
..                ...       ...      ...      ...             ...
995          0.000004    0.0986    0.470   96.006  audio_features
996          0.000008    0.0705    0.592   92.984  audio_features
997          0.000375    0.3520    0.765   97.867  audio_features
998          0.000000    0.0954    0.174   79.924  audio_features
999          0.000000    0.0954    0.174   79.924  audio_features

                        id                                  uri  \
0    6AQbmUe0Qwf5PZnt4HmTXv  spotify:track:6AQbmUe0Qwf5PZnt4HmTXv
1    0yLdNVWF3Srea0uzk55zFn  spotify:track:0yLdNVWF3Srea0uzk55zFn
2    59uQI0PADDKeE6UZDTJEe8  spotify:track:59uQI0PADDKeE6UZDTJEe8
3    7oDd86yk8itslrA9HRP2ki  spotify:track:7oDd86yk8itslrA9HRP2ki
4    7K3BhSpAxZBznislvUMVtn  spotify:track:7K3BhSpAxZBznislvUMVtn
..                      ...                                 ...
995  55ZATsjPlTeSTNJOuW90pW  spotify:track:55ZATsjPlTeSTNJOuW90pW
996  3jUDqkHatwtprUmZ8epP8P  spotify:track:3jUDqkHatwtprUmZ8epP8P
997  09HafXIrcz1OofFYvROj4H  spotify:track:09HafXIrcz1OofFYvROj4H
998  2S7Y56VPjFgah09rGwuh4w  spotify:track:2S7Y56VPjFgah09rGwuh4w
999  0JPMwWrAbN5L6U36V3Cdvd  spotify:track:0JPMwWrAbN5L6U36V3Cdvd

                                     track_href  \
0    https://api.spotify.com/v1/tracks/6AQbmUe0Qwf5...
1    https://api.spotify.com/v1/tracks/0yLdNVWF3Sre...
2    https://api.spotify.com/v1/tracks/59uQI0PADDKe...
3    https://api.spotify.com/v1/tracks/7oDd86yk8its...
4    https://api.spotify.com/v1/tracks/7K3BhSpAxZBz...
..                                                 ...
995  https://api.spotify.com/v1/tracks/55ZATsjPlTeS...
996  https://api.spotify.com/v1/tracks/3jUDqkHatwtp...
997  https://api.spotify.com/v1/tracks/09HafXIrcz1O...
998  https://api.spotify.com/v1/tracks/2S7Y56VPjFga...
999  https://api.spotify.com/v1/tracks/0JPMwWrAbN5L...

                                    analysis_url  duration_ms  \
0    https://api.spotify.com/v1/audio-analysis/6AQb...       131013
1    https://api.spotify.com/v1/audio-analysis/0yLd...       200455
2    https://api.spotify.com/v1/audio-analysis/59uQ...       163855
3    https://api.spotify.com/v1/audio-analysis/7oDd...       232857
4    https://api.spotify.com/v1/audio-analysis/7K3B...       163855
..                                                 ...          ...
995  https://api.spotify.com/v1/audio-analysis/55ZA...       265000
996  https://api.spotify.com/v1/audio-analysis/3jUD...       174960
997  https://api.spotify.com/v1/audio-analysis/09Ha...       157932
998  https://api.spotify.com/v1/audio-analysis/2S7Y...       190893
999  https://api.spotify.com/v1/audio-analysis/0JPM...       190893

     time_signature
0                 4
1                 4
2                 4
3                 4
4                 4
..              ...
995               4
996               4
997               4
998               4
999               4

[1000 rows x 18 columns]
```

In [15]:
```
tf_df.head()
```

Out[15]:

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	type	id	
0	0.696	0.809	5	-8.254	1	0.0500	0.2520	0.000128	0.2480	0.857	132.962	audio_features	6AQbmUe0Qwf5PZnt4HmTXv	spotify:track:6AC
1	0.707	0.681	0	-4.325	1	0.0668	0.0632	0.000005	0.0322	0.646	117.999	audio_features	0yLdNVWF3Srea0uzk55zFn	spotify:track:0y
2	0.517	0.675	6	-5.382	1	0.0357	0.4590	0.000000	0.1510	0.518	203.853	audio_features	59uQI0PADDKeE6UZDTJEe8	spotify:track:59u
3	0.531	0.525	1	-6.500	0	0.0671	0.2320	0.000000	0.4410	0.502	66.900	audio_features	7oDd86yk8itslrA9HRP2ki	spotify:track
4	0.492	0.675	6	-5.456	1	0.0389	0.4670	0.000000	0.1420	0.478	203.759	audio_features	7K3BhSpAxZBznislvUMVtn	spotify:track:7

In [16]:
```
len(tf_df)
```

Out[16]:
```
1000
```

In [17]:
```
tf_df['duration_ms'] = pd.to_numeric(tf_df['duration_ms'])
#cols_to_drop1 = ['track_id','artist_id']
#cols_to_drop2 = ['key','mode','type', 'uri','track_href','analysis_url']
#track_df = track_df.drop(columns=cols_to_drop1)
#tf_df = tf_df.drop(columns=cols_to_drop2)
print(track_df.info())
print(tf_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   artist_name        1000 non-null   object
 1   track_name         1000 non-null   object
 2   track_id           1000 non-null   object
 3   track_popularity   1000 non-null   int64 
 4   artist_id          1000 non-null   object
 5   album_name         1000 non-null   object
 6   artist_popularity  1000 non-null   int64 
 7   artist_genres      1000 non-null   object
 8   artist_followers   1000 non-null   int64 
dtypes: int64(3), object(6)
memory usage: 70.4+ KB
None
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   danceability      1000 non-null   float64
 1   energy            1000 non-null   float64
 2   key               1000 non-null   int64  
 3   loudness          1000 non-null   float64
 4   mode              1000 non-null   int64  
 5   speechiness       1000 non-null   float64
 6   acousticness      1000 non-null   float64
 7   instrumentalness  1000 non-null   float64
 8   liveness          1000 non-null   float64
 9   valence           1000 non-null   float64
 10  tempo             1000 non-null   float64
 11  type              1000 non-null   object 
 12  id                1000 non-null   object 
 13  uri               1000 non-null   object 
 14  track_href        1000 non-null   object 
 15  analysis_url      1000 non-null   object 
 16  duration_ms       1000 non-null   int64  
 17  time_signature    1000 non-null   int64  
dtypes: float64(9), int64(4), object(5)
memory usage: 148.4+ KB
None
```

In [18]:
```
track_df['artist_name'] = track_df['artist_name'].astype("string")
track_df['track_name'] = track_df['track_name'].astype("string")
track_df['track_id'] = track_df['track_id'].astype("string")
track_df['artist_id'] = track_df['artist_id'].astype("string")
tf_df['instrumentalness'] = pd.to_numeric(tf_df['instrumentalness'])
tf_df['time_signature'] = tf_df['time_signature'].astype("category")
print(track_df.info())
print(tf_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   artist_name        1000 non-null   string
 1   track_name         1000 non-null   string
 2   track_id           1000 non-null   string
 3   track_popularity   1000 non-null   int64
 4   artist_id          1000 non-null   string
 5   album_name         1000 non-null   object
 6   artist_popularity  1000 non-null   int64
 7   artist_genres      1000 non-null   object
 8   artist_followers   1000 non-null   int64
dtypes: int64(3), object(2), string(4)
memory usage: 70.4+ KB
None
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   danceability      1000 non-null   float64
 1   energy            1000 non-null   float64
 2   key               1000 non-null   int64
 3   loudness          1000 non-null   float64
 4   mode              1000 non-null   int64
 5   speechiness       1000 non-null   float64
 6   acousticness      1000 non-null   float64
 7   instrumentalness  1000 non-null   float64
 8   liveness          1000 non-null   float64
 9   valence           1000 non-null   float64
 10  tempo             1000 non-null   float64
 11  type              1000 non-null   object
 12  id                1000 non-null   object
 13  uri               1000 non-null   object
 14  track_href        1000 non-null   object
 15  analysis_url      1000 non-null   object
 16  duration_ms       1000 non-null   int64
 17  time_signature    1000 non-null   category
dtypes: category(1), float64(9), int64(3), object(5)
memory usage: 141.8+ KB
None
```

MOST POPULAR ARTIST/TRACK/ALBUM

In [19]:
```
list(track_df)
```

Out[19]:
```
['artist_name',
 'track_name',
 'track_id',
 'track_popularity',
 'artist_id',
 'album_name',
 'artist_popularity',
 'artist_genres',
 'artist_followers']
```

In [20]:
```
#Top 20 popular songs/artist/album
track_df.sort_values(by=['track_popularity'], ascending=False)[['track_name', 'artist_name', 'album_name']].head(20)
```

Out[20]:

	track_name	artist_name	album_name
1	Flowers	Miley Cyrus	Flowers
16	Shakira: Bzrp Music Sessions, Vol. 53	Bizarrap	Shakira: Bzrp Music Sessions, Vol. 53
0	Boy's a liar Pt. 2	PinkPantheress	Boy's a liar Pt. 2
3	Die For You - Remix	The Weeknd	Die For You (Remix)
13	TQG	KAROL G	MAÑANA SERÁ BONITO
19	OMG	NewJeans	NewJeans 'OMG'
47	PRC	Peso Pluma	PRC
278	Tormenta (feat. Bad Bunny)	Gorillaz	Cracker Island
2	Last Night	Morgan Wallen	3 Songs At A Time Sampler
125	Fin de Semana	Oscar Maydon	Fin de Semana
283	X SI VOLVEMOS	KAROL G	X SI VOLVEMOS
45	Love Again	The Kid LAROI	Love Again
10	Watch This - ARIZONATEARS Pluggnb Remix	Lil Uzi Vert	Watch This (ARIZONATEARS Pluggnb Remix)
492	Yandel 150	Yandel	Resistencia
279	on the street (with J. Cole)	j-hope	on the street (with J. Cole)
138	El Gordo Trae El Mando	Chino Pacas	El Gordo Trae El Mando
122	Heaven	Niall Horan	Heaven
534	LLYLM	ROSALÍA	LLYLM
333	Unholy (feat. Kim Petras)	Sam Smith	Gloria
995	Shorty Party	Cartel De Santa	Shorty Party

In [21]:
```
by_art_fol = pd.DataFrame(track_df.sort_values(by=['artist_followers'], ascending=False)[['artist_followers','artist_popularity', 'a
by_art_fol.astype(str).drop_duplicates().head(20)
```

Out[21]:

	artist_followers	artist_popularity	artist_name	artist_genres	artist_id
352	110409920	93	Ed Sheeran	['pop', 'uk pop']	6eUKZXaKkcviH0Ku9w2n3V
850	77653248	90	Billie Eilish	['art pop', 'electropop', 'pop']	6qqNVTkY8uBg9cP3Jd7DAH
433	70542988	92	Justin Bieber	['canadian pop', 'pop']	1uNFoZAHBGtllmzznpCI3s
134	69634772	99	Taylor Swift	['pop']	06HL4z0CvFAxyc27GXpf02
60	65999740	100	Bad Bunny	['reggaeton', 'trap latino', 'urbano latino']	4q3ewBCX7sLwd24euuV69X
144	60625943	99	The Weeknd	['canadian contemporary r&b', 'canadian pop', ...	1Xyo4u8uXC1ZmMpatF05PJ
666	46911748	91	Bruno Mars	['dance pop', 'pop']	0du5cEVh5yTK9QJze8zA0C
617	46290965	91	Imagine Dragons	['modern rock', 'rock']	53XhwfbYqKCa1cC15pYq2q
756	42516193	91	Coldplay	['permanent wave', 'pop']	4gzpq5DPGxSnKTe4SA8HAU
991	40770901	89	XXXTENTACION	['emo rap', 'miami hip hop']	15UsOTVnJzReFVN1VCnxy4
793	39978000	90	Post Malone	['dfw rap', 'melodic rap', 'rap']	246dkjvS1zLTtiykXe5h60
915	39008765	90	Dua Lipa	['dance pop', 'pop', 'uk pop']	6M2wZ9GZgrQXHCFfjv46we
997	36807560	89	J Balvin	['reggaeton', 'reggaeton colombiano', 'urbano ...	1vyhD5VmyZ7KMfW5gqLgo5
531	34930453	85	Marshmello	['brostep', 'dance pop', 'edm', 'pop', 'pop da...	64KEffDW9EtZ1y2vBYgq8T
608	33908167	89	Beyoncé	['dance pop', 'pop', 'r&b']	6vWDO969PvNqNYHIOW5v0m
319	33164402	94	KAROL G	['reggaeton', 'reggaeton colombiano', 'urbano ...	790FomKkXshlbRYZFtlgla
897	31859806	90	Daddy Yankee	['latin hip hop', 'reggaeton', 'urbano latino']	4VMYDCV2IEDYJArk749S6m
522	28184382	92	Juice WRLD	['chicago rap', 'melodic rap']	4MCBfE4596Uoi2O4DtmEMz
989	27620208	88	Nicki Minaj	['dance pop', 'hip pop', 'pop', 'queens hip ho...	0hCNtLu0JehylgoiP8L4Gh
286	27504871	91	Anuel AA	['reggaeton', 'reggaeton flow', 'trap boricua'...	2R21vXR83lH98kGeO99Y66

In [22]:
```
pivot_ui(track_df)
```

Out[22]:

[pop out]

Heatmap | Average | artist_followers

track_name | track_id | track_popularity | artist_id | album_name | artist_genres | artist_followers

artist_name | artist_popularity

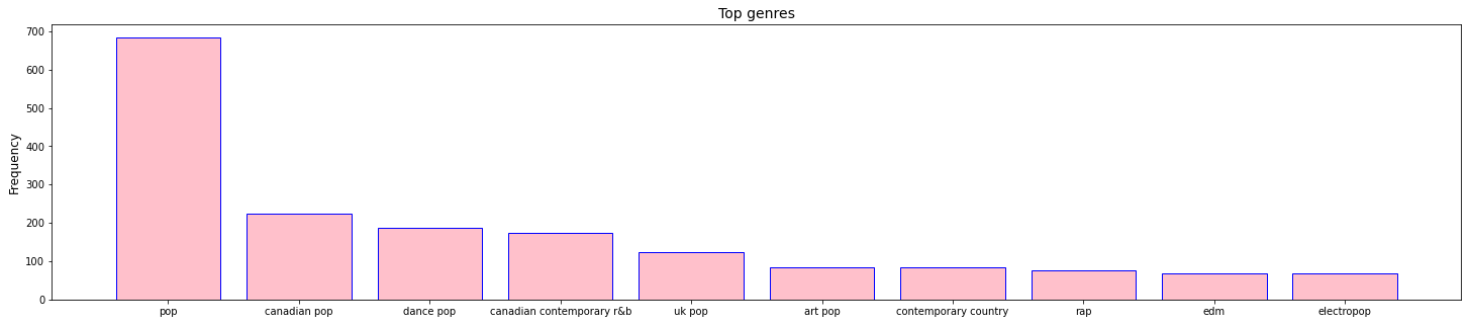
artist_name	artist_popularity	Totals
Ed Sheeran	93	110,409,920.00
Billie Eilish	90	77,653,248.00
Justin Bieber	92	70,542,988.00
Taylor Swift	99	69,634,772.00
Bad Bunny	100	65,999,740.00
The Weeknd	99	60,625,943.00
Bruno Mars	91	46,911,748.00
Imagine Dragons	91	46,290,965.00
Coldplay	91	42,516,193.00
XXXTENTACION	89	40,770,901.00
Post Malone	90	39,978,000.00
Dua Lipa	90	39,008,765.00
J Balvin	89	36,807,560.00
Marshmello	85	34,930,453.00
Beyoncé	89	33,908,167.00
KAROL G	94	33,164,402.00

In [23]:
```
def to_1D(series):
 return pd.Series([x for _list in series for x in _list])
to_1D(track_df['artist_genres']).value_counts().head(20)
```

Out[23]:
```
pop                         684
canadian pop                223
dance pop                   187
canadian contemporary r&b   173
uk pop                      124
art pop                      85
contemporary country         84
rap                          75
edm                          69
electropop                   67
pop dance                    67
big room                     65
trap                         63
melodic rap                  61
urbano latino                54
philly rap                   49
reggaeton                    42
reggaeton colombiano         39
modern rock                  37
rock                         36
dtype: int64
```

In [24]:
```
fig, ax = plt.subplots(figsize = (25,5))
ax.bar(to_1D(track_df['artist_genres']).value_counts().index[:10],
        to_1D(track_df['artist_genres']).value_counts().values[:10],
       color='pink', edgecolor='blue')
ax.set_ylabel("Frequency", size = 12)
ax.set_title("Top genres", size = 14)
```

Out[24]:
```
Text(0.5, 1.0, 'Top genres')
```



In [25]:
```
#Top 20 artists by followers for each Top 10 genres
top_10_genres = list(to_1D(track_df['artist_genres']).value_counts().index[:20])
top_artists_by_genre = []
for genre in top_10_genres:
    for index, row in by_art_fol.iterrows():
        if genre in row['artist_genres']:
            top_artists_by_genre.append({'artist_name': row['artist_name'], 'artist_genre':genre})
            break
pd.json_normalize(top_artists_by_genre)
```

Out[25]:

	artist_name	artist_genre
0	Ed Sheeran	pop
1	Justin Bieber	canadian pop
2	Bruno Mars	dance pop
3	The Weeknd	canadian contemporary r&b
4	Ed Sheeran	uk pop
5	Billie Eilish	art pop
6	Luke Combs	contemporary country
7	Post Malone	rap
8	Marshmello	edm
9	Billie Eilish	electropop
10	Marshmello	pop dance
11	David Guetta	big room
12	Lil Uzi Vert	trap
13	Post Malone	melodic rap
14	Bad Bunny	urbano latino
15	Lil Uzi Vert	philly rap
16	Bad Bunny	reggaeton
17	J Balvin	reggaeton colombiano
18	Imagine Dragons	modern rock
19	Imagine Dragons	rock

In [26]:
```
20 songs by top song by Genre

rack_pop = pd.DataFrame(track_df.sort_values(by=['track_popularity'], ascending=False)[['track_popularity','track_name', 'artist_name
rack_pop.astype(str).drop_duplicates().head(20)
songs_by_genre = []
genre in top_10_genres:
r index, row in by_track_pop.iterrows():
if genre in row['artist_genres']:
  top_songs_by_genre.append({'track_name':row['track_name'], 'track_popularity':row['track_popularity'],'artist_name':row['artist_nam
  break
son_normalize(top_songs_by_genre)
```

Out[26]:

	track_name	track_popularity	artist_name	artist_genre
0	Flowers	100	Miley Cyrus	pop
1	Die For You - Remix	94	The Weeknd	canadian pop
2	Heaven	85	Niall Horan	dance pop
3	Die For You - Remix	94	The Weeknd	canadian contemporary r&b
4	Unholy (feat. Kim Petras)	84	Sam Smith	uk pop
5	A&W	81	Lana Del Rey	art pop
6	Last Night	87	Morgan Wallen	contemporary country
7	Watch This - ARIZONATEARS Pluggnb Remix	86	Lil Uzi Vert	rap
8	El Merengue	80	Marshmello	edm
9	Die 4 Me	79	Halsey	electropop
10	El Merengue	80	Marshmello	pop dance
11	Get Together	1	David Guetta	big room
12	Watch This - ARIZONATEARS Pluggnb Remix	86	Lil Uzi Vert	trap
13	Watch This - ARIZONATEARS Pluggnb Remix	86	Lil Uzi Vert	melodic rap
14	Shakira: Bzrp Music Sessions, Vol. 53	96	Bizarrap	urbano latino
15	Watch This - ARIZONATEARS Pluggnb Remix	86	Lil Uzi Vert	philly rap
16	TQG	94	KAROL G	reggaeton
17	TQG	94	KAROL G	reggaeton colombiano
18	Watch The World Burn	80	Falling In Reverse	modern rock
19	Bones	5	Imagine Dragons	rock

In [27]:
```
pivot_ui(pd.DataFrame(top_songs_by_genre))
```

Out[27]:

Average ↑ ↔

track_popularity

track_popularity

track_name

track_name \ track_popularity	1	5	79	80	81	84	85	86	87	94	96	100	Totals
Flowers												100.00	**100.00**
Shakira: Bzrp Music Sessions, Vol. 53											96.00		**96.00**
Die For You - Remix										94.00			**94.00**
TQG										94.00			**94.00**
Last Night									87.00				**87.00**
Watch This - ARIZONATEARS Pluggnb Remix								86.00					**86.00**
Heaven							85.00						**85.00**
Unholy (feat. Kim Petras)						84.00							**84.00**
A&W					81.00								**81.00**
El Merengue				80.00									**80.00**
Watch The World Burn				80.00									**80.00**
Die 4 Me			79.00										**79.00**
Bones		5.00											**5.00**
Get Together	1.00												**1.00**
Totals	**1.00**	**5.00**	**79.00**	**80.00**	**81.00**	**84.00**	**85.00**	**86.00**	**87.00**	**94.00**	**96.00**	**100.00**	**78.90**

Track recommendations

In [28]:
```
#Get a list of genres available for the recommendation function
genre = sp.recommendation_genre_seeds()
print(genre)
```

```
{'genres': ['acoustic', 'afrobeat', 'alt-rock', 'alternative', 'ambient', 'anime', 'black-metal', 'bluegrass', 'blues', 'bossanov
a', 'brazil', 'breakbeat', 'british', 'cantopop', 'chicago-house', 'children', 'chill', 'classical', 'club', 'comedy', 'country',
'dance', 'dancehall', 'death-metal', 'deep-house', 'detroit-techno', 'disco', 'disney', 'drum-and-bass', 'dub', 'dubstep', 'edm',
'electro', 'electronic', 'emo', 'folk', 'forro', 'french', 'funk', 'garage', 'german', 'gospel', 'goth', 'grindcore', 'groove', 'g
runge', 'guitar', 'happy', 'hard-rock', 'hardcore', 'hardstyle', 'heavy-metal', 'hip-hop', 'holidays', 'honky-tonk', 'house', 'id
m', 'indian', 'indie', 'indie-pop', 'industrial', 'iranian', 'j-dance', 'j-idol', 'j-pop', 'j-rock', 'jazz', 'k-pop', 'kids', 'lat
in', 'latino', 'malay', 'mandopop', 'metal', 'metal-misc', 'metalcore', 'minimal-techno', 'movies', 'mpb', 'new-age', 'new-releas
e', 'opera', 'pagode', 'party', 'philippines-opm', 'piano', 'pop', 'pop-film', 'post-dubstep', 'power-pop', 'progressive-house',
'psych-rock', 'punk', 'punk-rock', 'r-n-b', 'rainy-day', 'reggae', 'reggaeton', 'road-trip', 'rock', 'rock-n-roll', 'rockabilly',
'romance', 'sad', 'salsa', 'samba', 'sertanejo', 'show-tunes', 'singer-songwriter', 'ska', 'sleep', 'songwriter', 'soul', 'soundtr
acks', 'spanish', 'study', 'summer', 'swedish', 'synth-pop', 'tango', 'techno', 'trance', 'trip-hop', 'turkish', 'work-out', 'worl
d-music']}
```

In [29]:
```
recommendations = sp.recommendations(seed_artists=[artist['id']], seed_genres=["pop"], seed_tracks=None, limit=10)
for track in recommendations['tracks']:
  print(track['artists'][0]['name'], track['name'])
```

```
Sofia Carson Love Is the Name
Dalex +Linda
J Balvin Dónde Estarás
Garmiani BARRACA (feat. MC Pikachu)
Matt Hunter Dicen
Leishman Read Descontrolao
Topo La Maskara Panda Pon
Scorpion Ta Malo
Maluma Borro Cassette
Marshmello El Merengue
```

Write to CSV files

In [30]:
```
#track_df.to_csv('spotify_track_df.csv', mode ='a', index=False, header= True)
```

In [31]:
```
#tf_df.to_csv('track_features_df.csv', mode ='a', index=False, header= True)
```

In [ ]:
```

```