Dataset: dataset.csv

Context: Music Streaming Industry

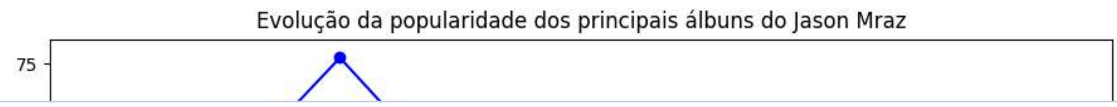
This project aims to create **visual analytical support** for non-technical stakeholders in the music industry.
I used **Pandas** and **Matplotlib** to generate a range of clear and informative exploratory charts that illustrate key trends in the Spotify dataset.
The notebook emphasizes how effective data visualization helps decision-makers interpret complex data intuitively.

Skills demonstrated: Data visualization, exploratory analysis, Pandas, Matplotlib, stakeholder communication

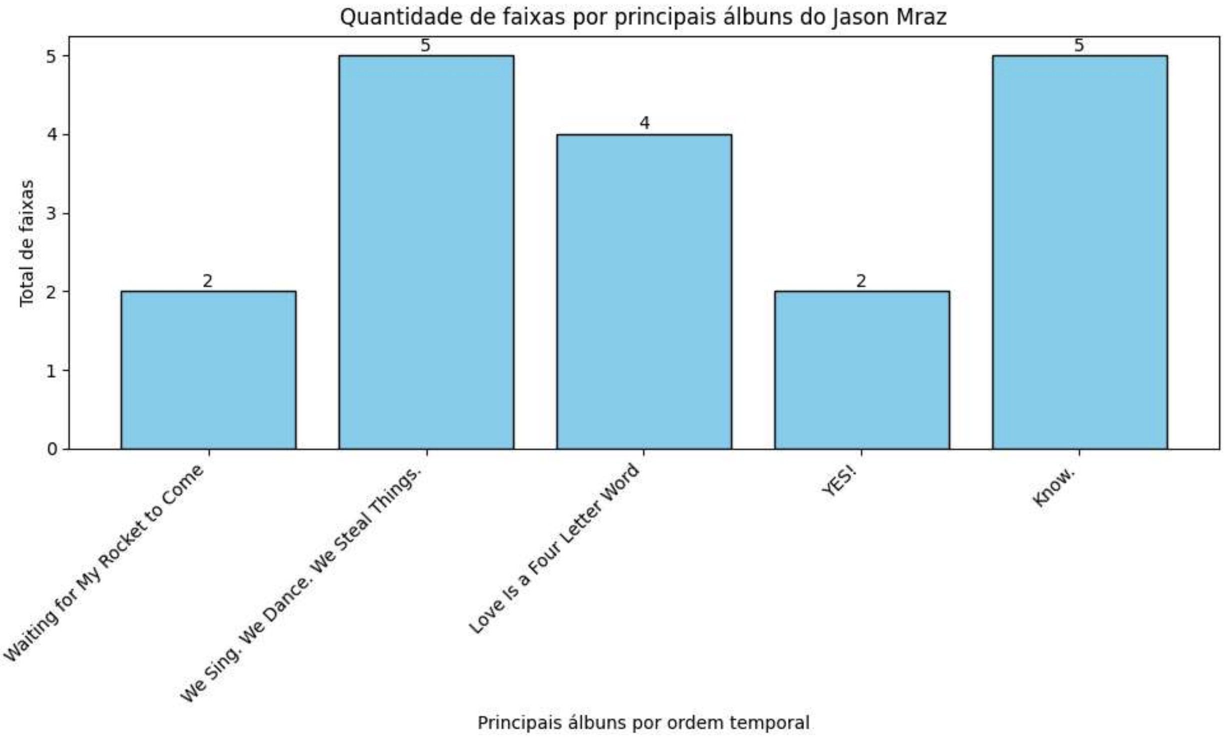
Question 1 - Basic charts with Matplotlib

```
import pandas as pd import matplotlib.pyplot as plt

df= pd.read_csv("dataset.csv")
jason_data= df[df['artists'].str.contains("Jason Mraz", case=False, na=False)]

unique_albums= jason_data['album_name'].unique() for album in unique_albums: print(album)

jason_mraz_albums_ordered= [ "Waiting for My Rocket to Come", "We Sing. We Dance. We Steal Things.", "Love Is a Four Lett

df['popularity'] = pd.to_numeric(df['popularity'], errors='coerce')

jason_data = df[df['album_name'].isin(jason_mraz_albums_ordered)]

popularity_by_album = jason_data.groupby('album_name')['popularity'].mean()

popularity_by_album= jason_data.groupby('album_name')['popularity'].mean()

popularity_by_album= popularity_by_album.reindex(jason_mraz_albums_ordered)

plt.figure(figsize=(10, 6))
plt.plot(popularity_by_album.index, popularity_by_album.values, marker='o', linestyle='-', color='b')
plt.title("Evolução da popularidade dos principais álbuns do Jason Mraz")
plt.xlabel("Álbuns (data de lançamento)")
plt.ylabel("Popularidade média")
plt.xticks(rotation=45, ha= 'right')

plt.tight_layout()

plt.show()
```

```
We Sing. We Dance. We Steal Things.
Love Is a Four Letter Word
Coffee Moment
Human - Best Adult Pop Tunes
Mellow Adult Pop
Holly Jolly Christmas
Feeling Good - Adult Pop Favorites
Christmas Time
Perfect Christmas Hits
Merry Christmas
Christmas Music - Holiday Hits
Know.
I Won't Give Up
Waiting for My Rocket to Come (Expanded Edition)
I'm Yours
YES!
Look For The Good
Waiting for My Rocket to Come
Have It All
Now That's What I Call Music 2012-13
Could I Love You Any More (feat. Jason Mraz)
Helpsters: Apple TV+ Original Series Soundtrack, Vol. 1
```



```
jason_mraz_albums_ordered= [
"Waiting for My Rocket to Come",
"We Sing. We Dance. We Steal Things.",
"Love Is a Four Letter Word",
"YES!",
"Know."
]
df['popularity'] = pd.to_numeric(df['popularity'], errors='coerce')
jason_data = df[df['album_name'].isin(jason_mraz_albums_ordered)]
tracks_per_album= jason_data['album_name'].value_counts()
tracks_per_album= tracks_per_album.reindex(jason_mraz_albums_ordered)

plt.figure(figsize=(10,6))
plt.bar(tracks_per_album.index, tracks_per_album.values, color='skyblue', edgecolor='black')

plt.title("Quantidade de faixas por principais álbuns do Jason Mraz")
plt.xlabel("Principais álbuns por ordem temporal")
plt.ylabel("Total de faixas")
plt.xticks(rotation=45, ha= 'right')

for i, v in enumerate(tracks_per_album.values):
    plt.text(i, v + 0.01, str(v), ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

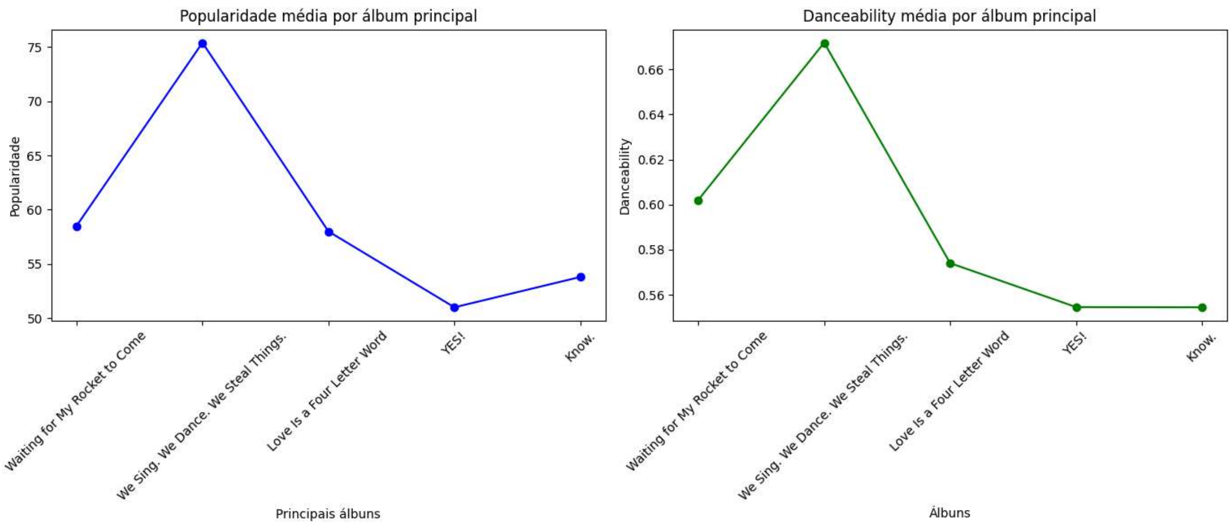


Line Chart It provides an understanding of the singer's own career, which albums have seen peaks in popularity, and the presence of a boom/bust curve.

Bar Graph It provides an overview of how many tracks there are on each of the artist's main albums and whether or not this number influences their popularity (if we cross-reference the information with the first graph).

Question 2 - Style and subplots

```
jason_mraz_albums_ordered= [
"Waiting for My Rocket to Come",
"We Sing. We Dance. We Steal Things.",
"Love Is a Four Letter Word",
"YES!",
"Know."
]

df['popularity']= pd.to_numeric(df['popularity'], errors='coerce')
df['danceability']= pd.to_numeric(df['danceability'], errors='coerce')

jason_data=df[df['album_name'].isin(jason_mraz_albums_ordered)]
metrics_by_album= jason_data.groupby('album_name')[['popularity', 'danceability']].mean()
metrics_by_album= metrics_by_album.reindex(jason_mraz_albums_ordered)

fig, axes= plt.subplots(1, 2, figsize=(14, 6), sharey= False)

axes[0].plot(metrics_by_album.index, metrics_by_album['popularity'], marker='o', linestyle='-', color='b')
axes[0].set_title("Popularidade média por álbum principal")
axes[0].set_xlabel("Principais álbuns")
axes[0].set_ylabel("Popularidade")
axes[0].tick_params(axis='x', rotation=45)

axes[1].plot(metrics_by_album.index, metrics_by_album['danceability'], marker='o', linestyle='-', color='g')
axes[1].set_title("Danceability média por álbum principal")
axes[1].set_xlabel("Álbuns")
axes[1].set_ylabel("Danceability")
axes[1].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```



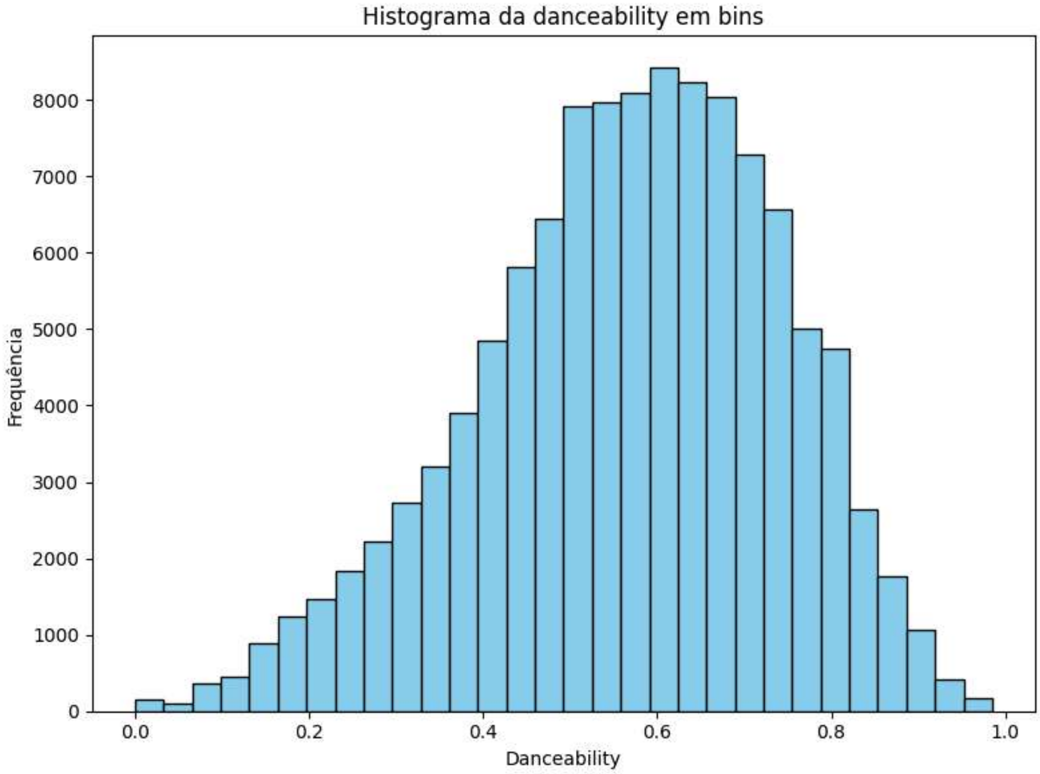
In the construction of the graphics, the following aesthetic choices were adopted:

- colors: blue for the line of the first graph and green for the line of the second, in order to differentiate them more quickly;
- markers that help identify albums;
- 45° angle on the x-axis to allow clear reading of album titles;
- "plt.tight_layout()" that allows spacing between chart elements, preventing information from overlapping.

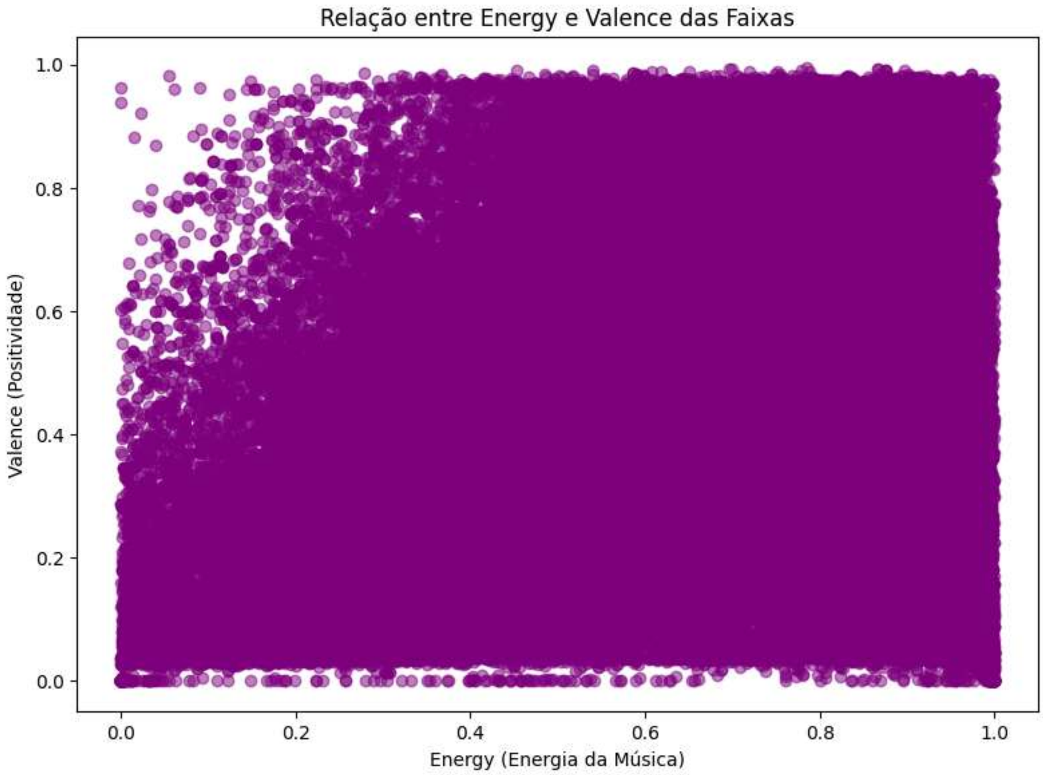
In short, aesthetic choices are fundamental in the visualization of graphs, as they allow a quick reading and understanding of the problem. When we think about the product, how we should deliver it to the customer, this is a characteristic that must be considered.

Question 3 - Distributions and patterns

```
df['danceability']= pd.to_numeric(df['danceability'], errors='coerce')

plt.figure(figsize=(8, 6))
plt.hist(df['danceability'].dropna(), bins=30, color='skyblue', edgecolor='black')

plt.title("Histograma da danceability em bins")
plt.xlabel("Danceability")
plt.ylabel("Frequência")
plt.tight_layout()
plt.show()
```



```
df['energy'] = pd.to_numeric(df['energy'], errors='coerce')
df['valence'] = pd.to_numeric(df['valence'], errors='coerce')

plt.figure(figsize=(8,6))
plt.scatter(df['energy'], df['valence'], alpha=0.5, color='purple')

plt.title("Relação entre Energy e Valence das Faixas")
plt.xlabel("Energy (Energia da Música)")
plt.ylabel("Valence (Positividade)")
plt.tight_layout()
plt.show()
```



Histogram

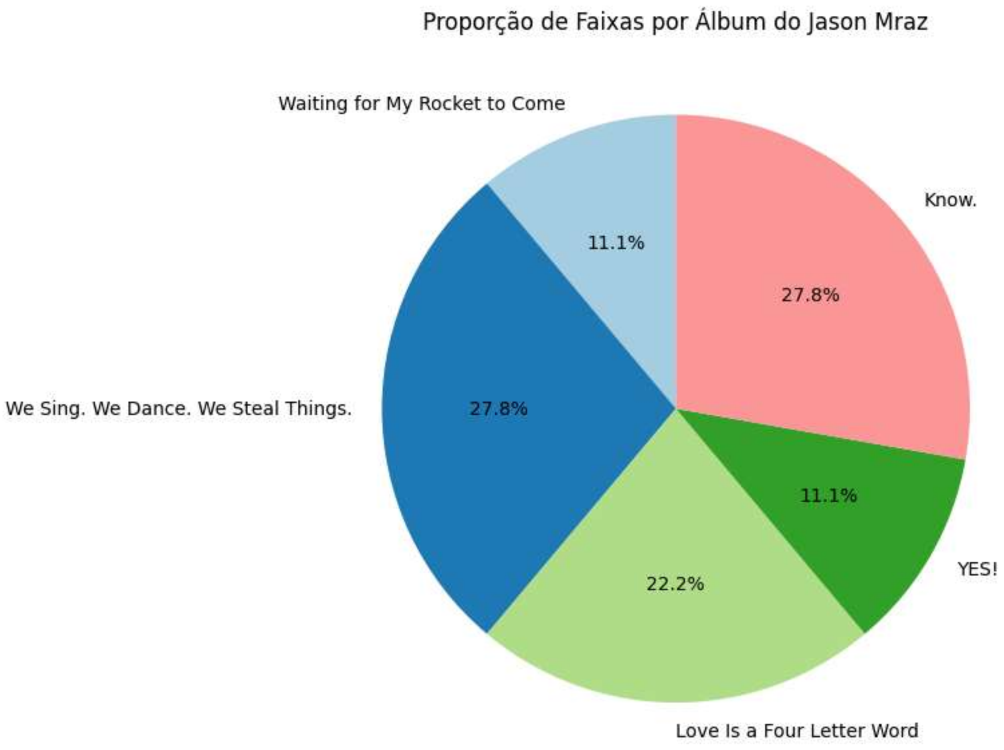
- Here we can see a higher frequency in terms of more danceable songs between 0.4 and 0.8. We can consider outliers the few values that represent the peaks of danceability and the lower values.

Scatter plot

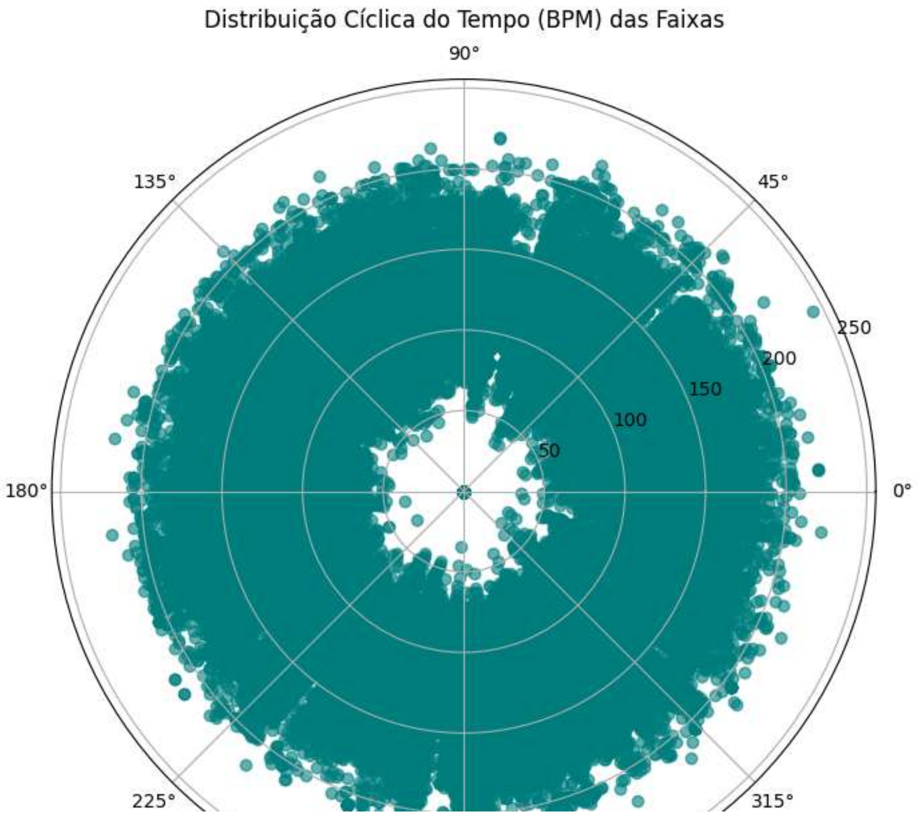
- In this graph the data is mostly concentrated between 0.4 and 1.0 of the x-axis, indicating a high compatibility between the energy of the music and its positivity.

Question 4 Proportions and angular data

```
jason_mraz_albums_ordered= [
"Waiting for My Rocket to Come",
"We Sing. We Dance. We Steal Things.",
"Love Is a Four Letter Word",
"YES!",
"Know."
]

jason_data=df[df['album_name'].isin(jason_mraz_albums_ordered)]
tracks_per_album= jason_data['album_name'].value_counts()
tracks_per_album= tracks_per_album.reindex(jason_mraz_albums_ordered)
plt.figure(figsize=(8,8))
plt.pie(tracks_per_album.dropna(), labels=tracks_per_album.index, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.

plt.title("Proporção de Faixas por Álbum do Jason Mraz")
plt.tight_layout()
plt.show()
```



```
import numpy as np
df['tempo']= pd.to_numeric(df['tempo'], errors= 'coerce')
tempo_data=df['tempo'].dropna()
angles= np.linspace(0, 2*np.pi, len(tempo_data), endpoint=False)

plt.figure(figsize=(8,8))
ax = plt.subplot(111, polar=True)

ax.scatter(angles, tempo_data, alpha=0.6, c='teal')

ax.set_title("Distribuição Cíclica do Tempo (BPM) das Faixas", va='bottom')
plt.show()
```

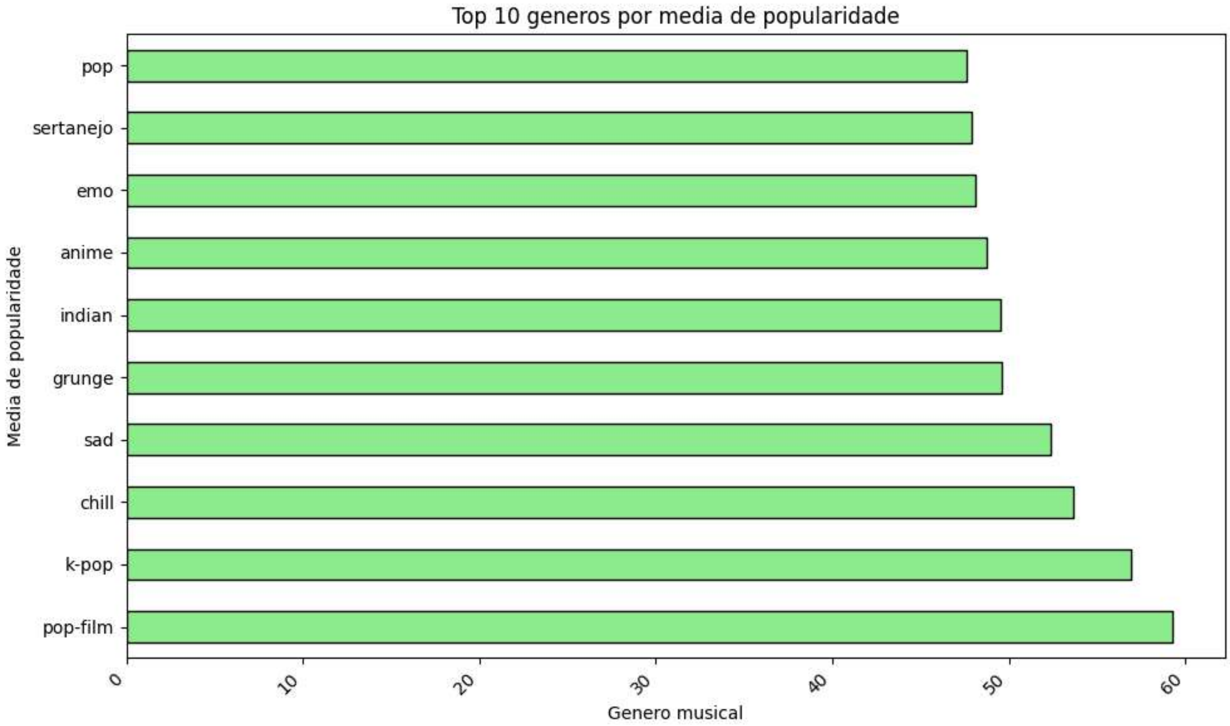


Pie Chart Advantages - it is a good option when we have few categories and we want to show the percentage difference of them. The visualization turns out to be more intuitive and clearer.
Disadvantages – it becomes confusing when there are too many categories to be shown.

Polar Chart Advantages - ideal for representing cyclical categories such as BPM, seasons and months. It helps to identify circular patterns.
Disadvantages - it is not so intuitive, and you may need an analysis of its operation to later analyze the data itself.

Question 5 - Exploration with Pandas + Matplotlib

```
df['popularity']= pd.to_numeric(df['popularity'], errors= 'coerce')
popularity_by_genre= df.groupby("track_genre")['popularity'].mean()
popularity_by_genre= popularity_by_genre.sort_values(ascending=False).head(10)

popularity_by_genre.plot(
    kind='barh',
    figsize=(10,6),
    color="lightgreen",
    edgecolor="black",
    title="Top 10 generos por media de popularidade"
)
plt.ylabel("Media de popularidade")
plt.xlabel("Genero musical")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Pandas and Matplotlib work well together, as one ends up complementing the other in exploratory analysis. With Pandas we can prepare and aggregate the data through ready-made aggregation functions such as sum(), mean() and value_counts() and with Matplotlib we can control the way in which the data will be visualized, creating figures with different colors, angles, shapes, legends and sizes. With Matplotlib we can create more detailed and customized graphics, quickly and efficiently.