```
import os
print(os.listdir('/content'))
```

```
['.config', 'spotify.csv', 'sample_data']
```

```
import pandas as pd

# Load the file from the correct path
df_spotify = pd.read_csv('/content/spotify.csv')

# Display the first few rows
df_spotify.head()
```

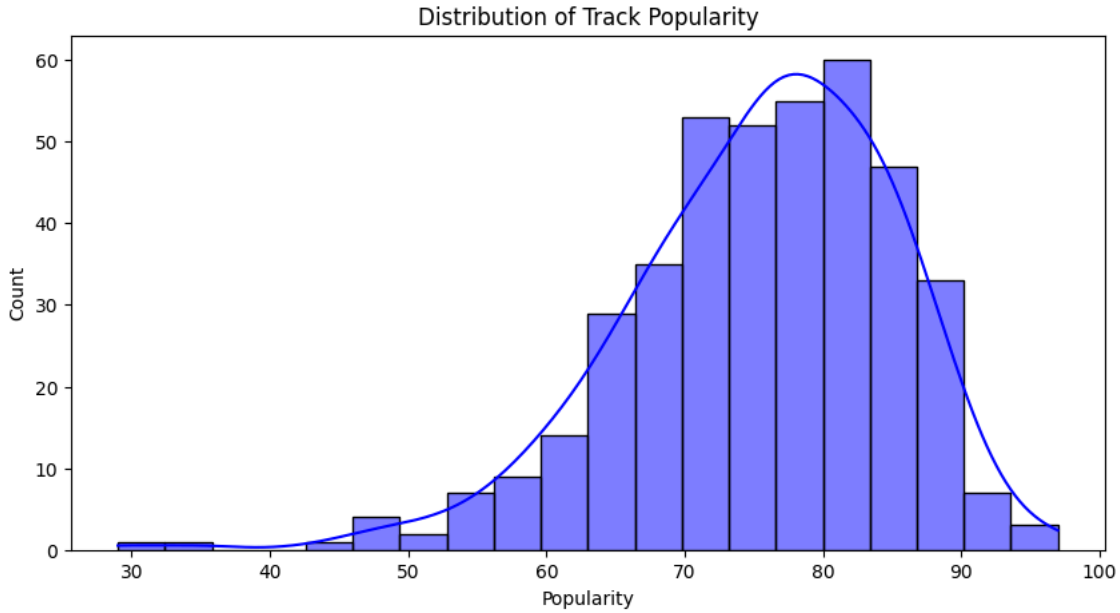
	Artist	Track Name	Popularity	Duration (ms)	Track ID
0	Drake	Rich Baby Daddy (feat. Sexyy Red & SZA)	92	319191	1yeB8MUNeLo9Ek1UEpsyz6
1	Drake	One Dance	91	173986	1zi7xx7UVEFkmKfv06H8x0
2	Drake	IDGAF (feat. Yeat)	90	260111	2YSzYUF3jWqb9YP9VXmpjE
3	Drake	First Person Shooter (feat. J. Cole)	88	247444	7aqfrAY2p9BUSiupwk3svU
4	Drake	Jimmy Cooks (feat. 21 Savage)	88	218364	3F5CqOj3wFlRv51JsHbxhe

Next steps: Generate code with df_spotify | View recommended plots | New interactive sheet

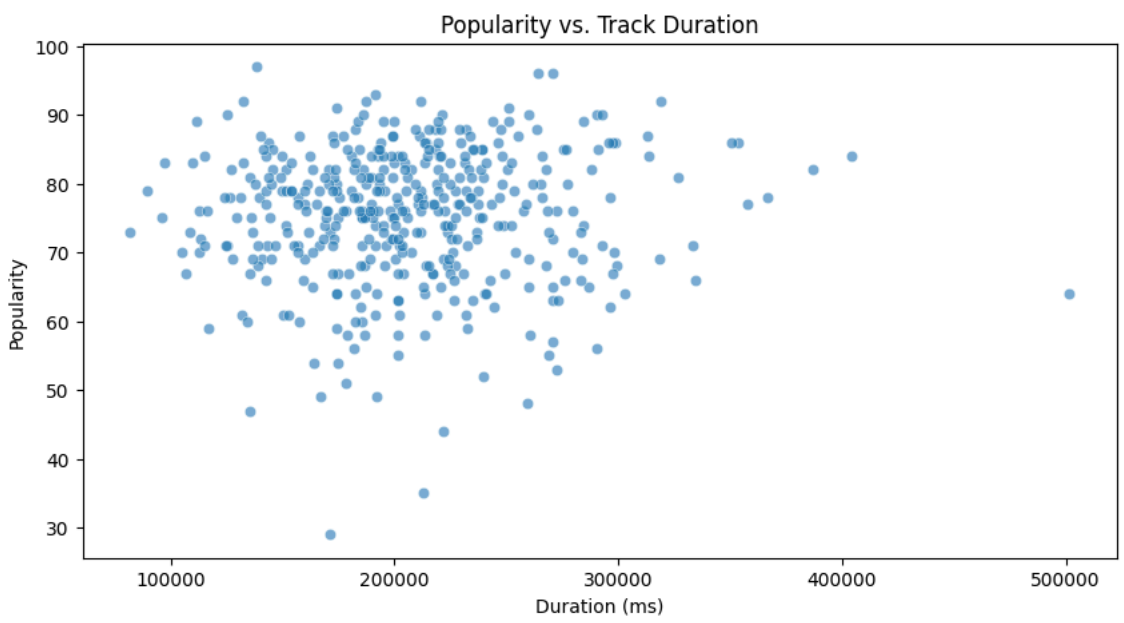
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


# 1. Check for missing values and duplicates
def clean_data(df):
    print("Missing Values:\n", df.isnull().sum())
    print("Duplicate Rows:", df.duplicated().sum())
    df.drop_duplicates(inplace=True)
    df.fillna(method='ffill', inplace=True)
    return df


df_spotify = clean_data(df_spotify)
```

```
Missing Values:
 Artist          0
Track Name       0
Popularity       0
Duration (ms)    0
Track ID         0
dtype: int64
Duplicate Rows: 27
<ipython-input-22-a6288894c827>:6: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. U
  df.fillna(method='ffill', inplace=True)
```

```
# 2. Distribution of popularity
def plot_popularity_distribution(df):
    plt.figure(figsize=(10,5))
    sns.histplot(df['Popularity'], bins=20, kde=True, color='blue')
    plt.title("Distribution of Track Popularity")
    plt.xlabel("Popularity")
    plt.ylabel("Count")
    plt.show()


plot_popularity_distribution(df_spotify)
```

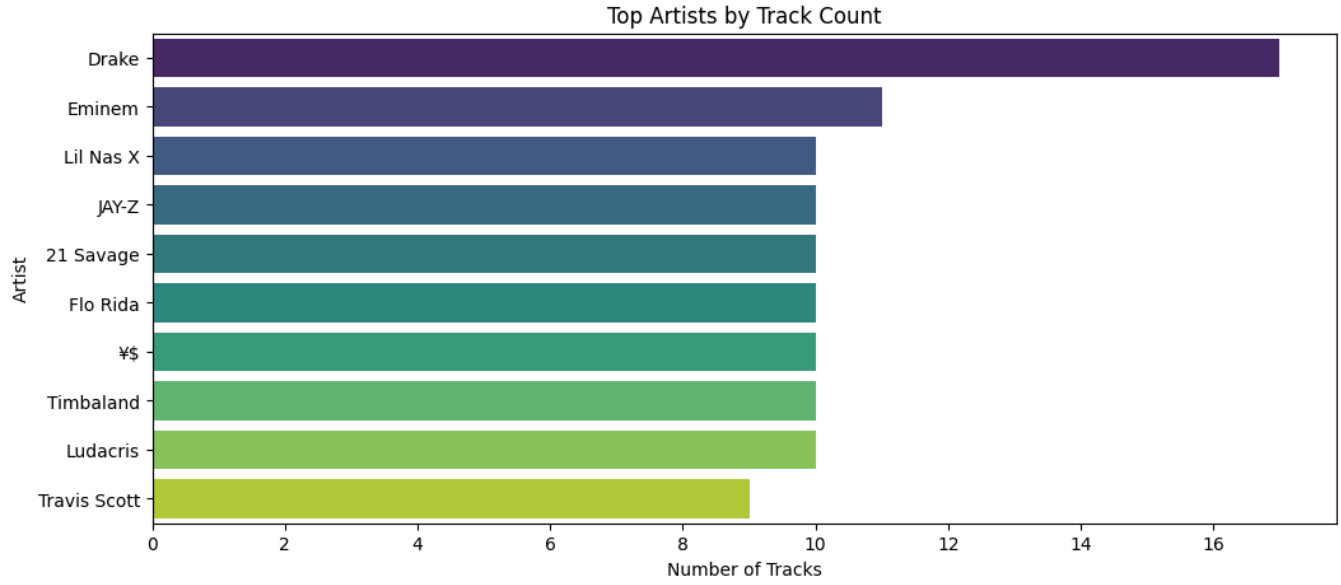



```
# 3. Relationship between Popularity and Duration
def plot_popularity_vs_duration(df):
    plt.figure(figsize=(10,5))
    sns.scatterplot(x=df['Duration (ms)'], y=df['Popularity'], alpha=0.6)
    plt.title("Popularity vs. Track Duration")
    plt.xlabel("Duration (ms)")
    plt.ylabel("Popularity")
    plt.show()


plot_popularity_vs_duration(df_spotify)
```




```
# 4. Artist with the highest number of tracks
def top_artists(df):
    plt.figure(figsize=(12,5))
    sns.countplot(y=df['Artist'], order=df['Artist'].value_counts().index[:10], palette='viridis')
    plt.title("Top Artists by Track Count")
    plt.xlabel("Number of Tracks")
    plt.ylabel("Artist")
    plt.show()


top_artists(df_spotify)
```

```
<ipython-input-28-89fcaa1ad994>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le

  sns.countplot(y=df['Artist'], order=df['Artist'].value_counts().index[:10], palette='viridis')
```




```
# 5. Least popular tracks
def least_popular_tracks(df):
    least_popular = df.nsmallest(5, 'Popularity')[['Artist', 'Track Name', 'Popularity']]
    print("Top 5 Least Popular Tracks:\n", least_popular)


least_popular_tracks(df_spotify)
```

```
Top 5 Least Popular Tracks:
                Artist                    Track Name  Popularity
207          Pressa  Attachments (feat. Coi Leray)          29
231   Justin Bieber                     Intentions          35
413  French Montana                Splash Brothers          44
225        Lil Baby                   On Me - Remix          47
407     Wyclef Jean      911 (feat. Mary J. Blige)          48
```

```
# 6. Top 5 most popular artists and their average popularity
def top_artists_avg_popularity(df):
    top_artists = df.groupby('Artist')['Popularity'].mean().nlargest(5)
    print("Top 5 Artists by Average Popularity:\n", top_artists)


top_artists_avg_popularity(df_spotify)
```

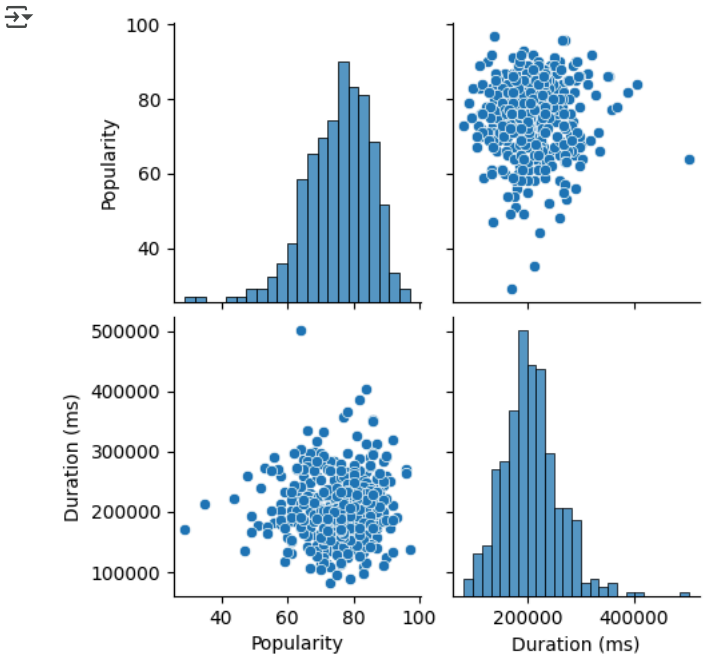
```
Top 5 Artists by Average Popularity:
 Artist
cassö           92.000000
Trueno          89.000000
David Guetta    87.000000
Travis Scott    86.555556
¥$              85.100000
Name: Popularity, dtype: float64
```

```
# 7. Most popular tracks of top 5 artists
def top_artists_top_tracks(df):
    top_artists = df.groupby('Artist')['Popularity'].mean().nlargest(5).index
    for artist in top_artists:
        top_track = df[df['Artist'] == artist].nlargest(1, 'Popularity')[['Track Name', 'Popularity']]
        print(f"Most popular track of {artist}:\n", top_track)


top_artists_top_tracks(df_spotify)
```

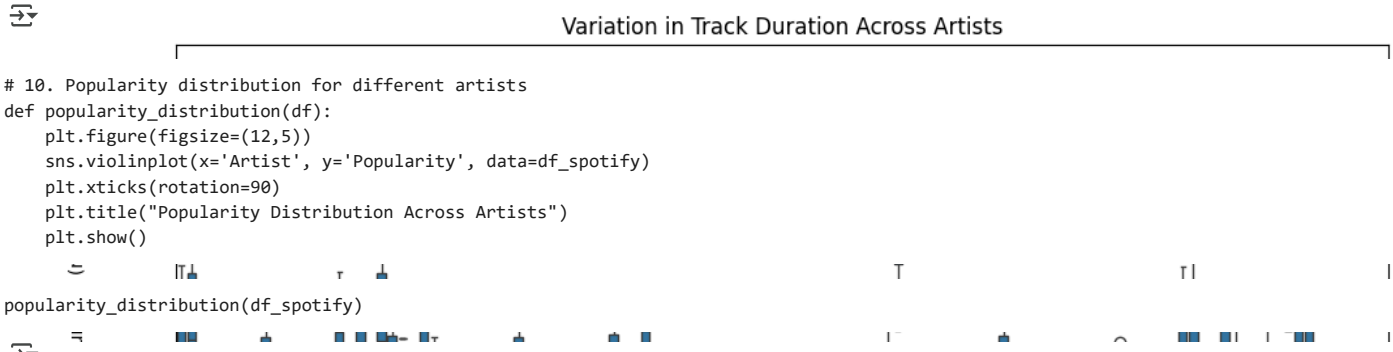
```
Most popular track of cassö:
     Track Name  Popularity
140      Prada          92
Most popular track of Trueno:
                        Track Name  Popularity
241  Mamichula - con Nicki Nicole          89
Most popular track of David Guetta:
               Track Name  Popularity
```

```
200  Baby Don't Hurt Me          87
Most popular track of Travis Scott:
                    Track Name  Popularity
30  FE!N (feat. Playboi Carti)          93
Most popular track of ¥$:
     Track Name  Popularity
260   CARNIVAL          96
```

```
# 8. Pairplot of numerical variables
sns.pairplot(df_spotify[['Popularity', 'Duration (ms)']])
plt.show()
```




```
# 9. Duration variation across artists
def duration_variation(df):
    plt.figure(figsize=(12,5))
    sns.boxplot(x='Artist', y='Duration (ms)', data=df_spotify)
    plt.xticks(rotation=90)
    plt.title("Variation in Track Duration Across Artists")
    plt.show()


duration_variation(df_spotify)
```




```
# 10. Popularity distribution for different artists
def popularity_distribution(df):
    plt.figure(figsize=(12,5))
    sns.violinplot(x='Artist', y='Popularity', data=df_spotify)
    plt.xticks(rotation=90)
    plt.title("Popularity Distribution Across Artists")
    plt.show()

popularity_distribution(df_spotify)
```