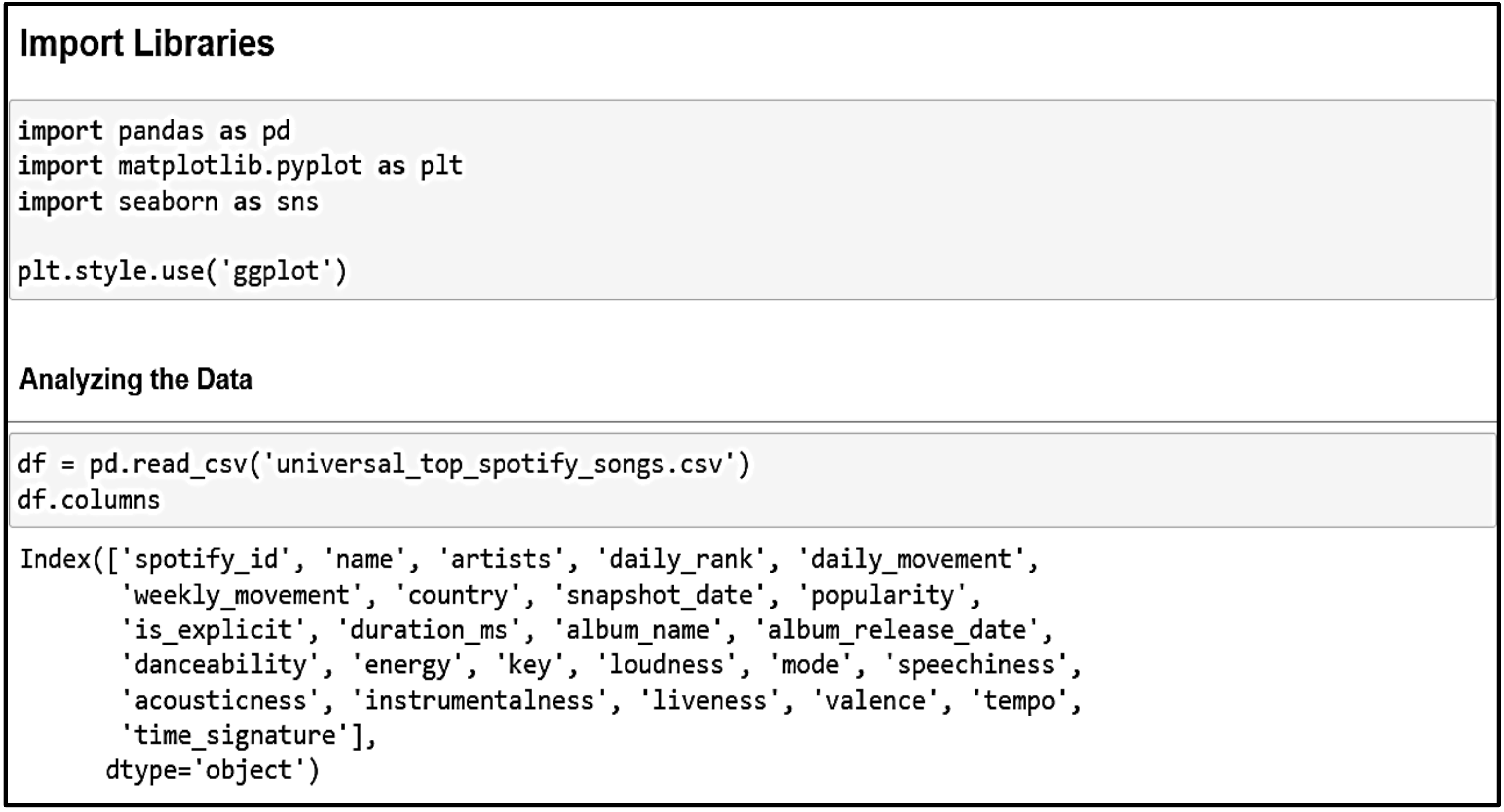


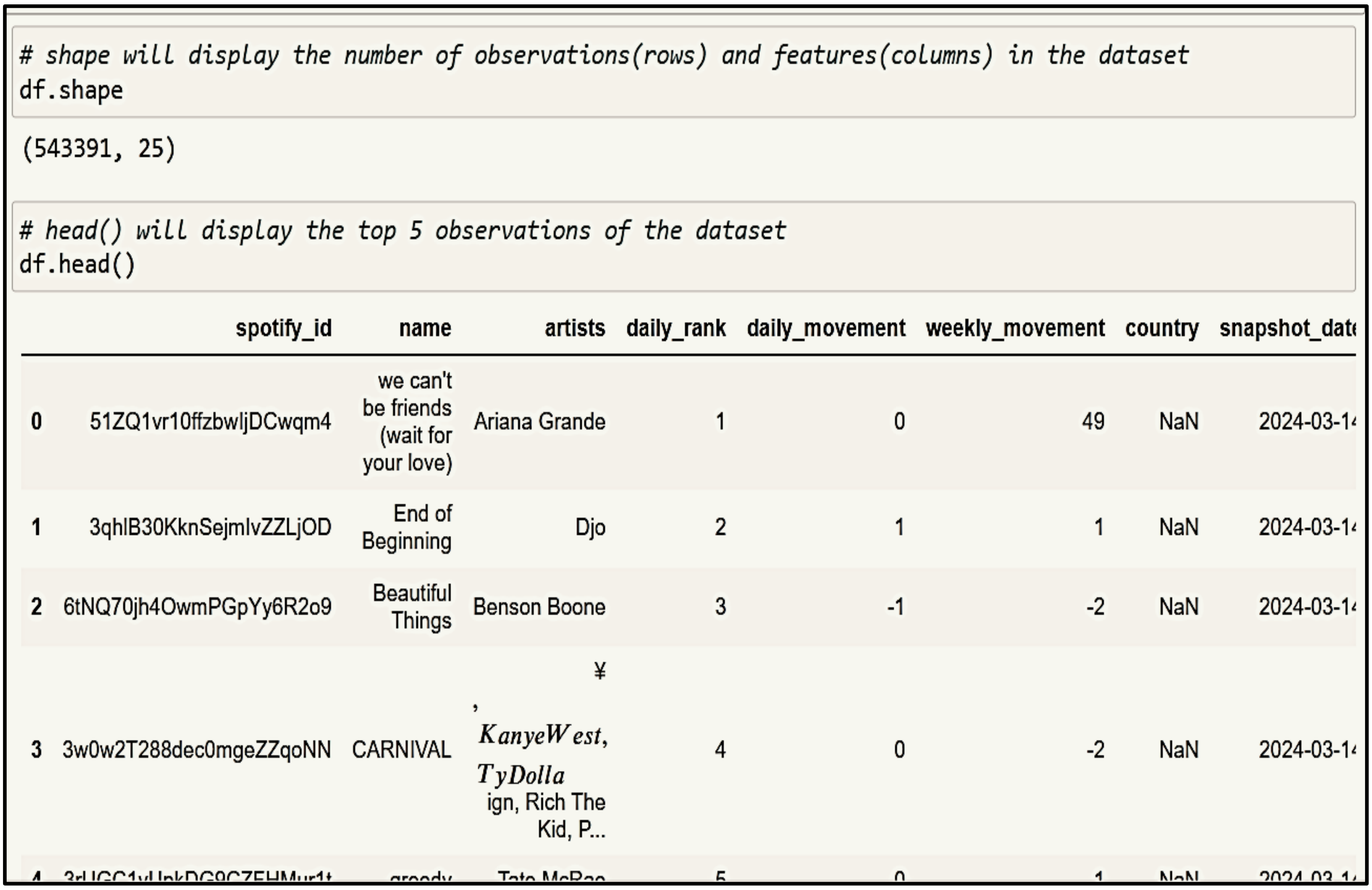
## Import Libraries

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

plt.style.use('ggplot')
```

### Analyzing the Data

```python
df = pd.read_csv('universal_top_spotify_songs.csv')
df.columns
```

```
Index(['spotify_id', 'name', 'artists', 'daily_rank', 'daily_movement',
       'weekly_movement', 'country', 'snapshot_date', 'popularity',
       'is_explicit', 'duration_ms', 'album_name', 'album_release_date',
       'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
       'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo',
       'time_signature'],
      dtype='object')
```

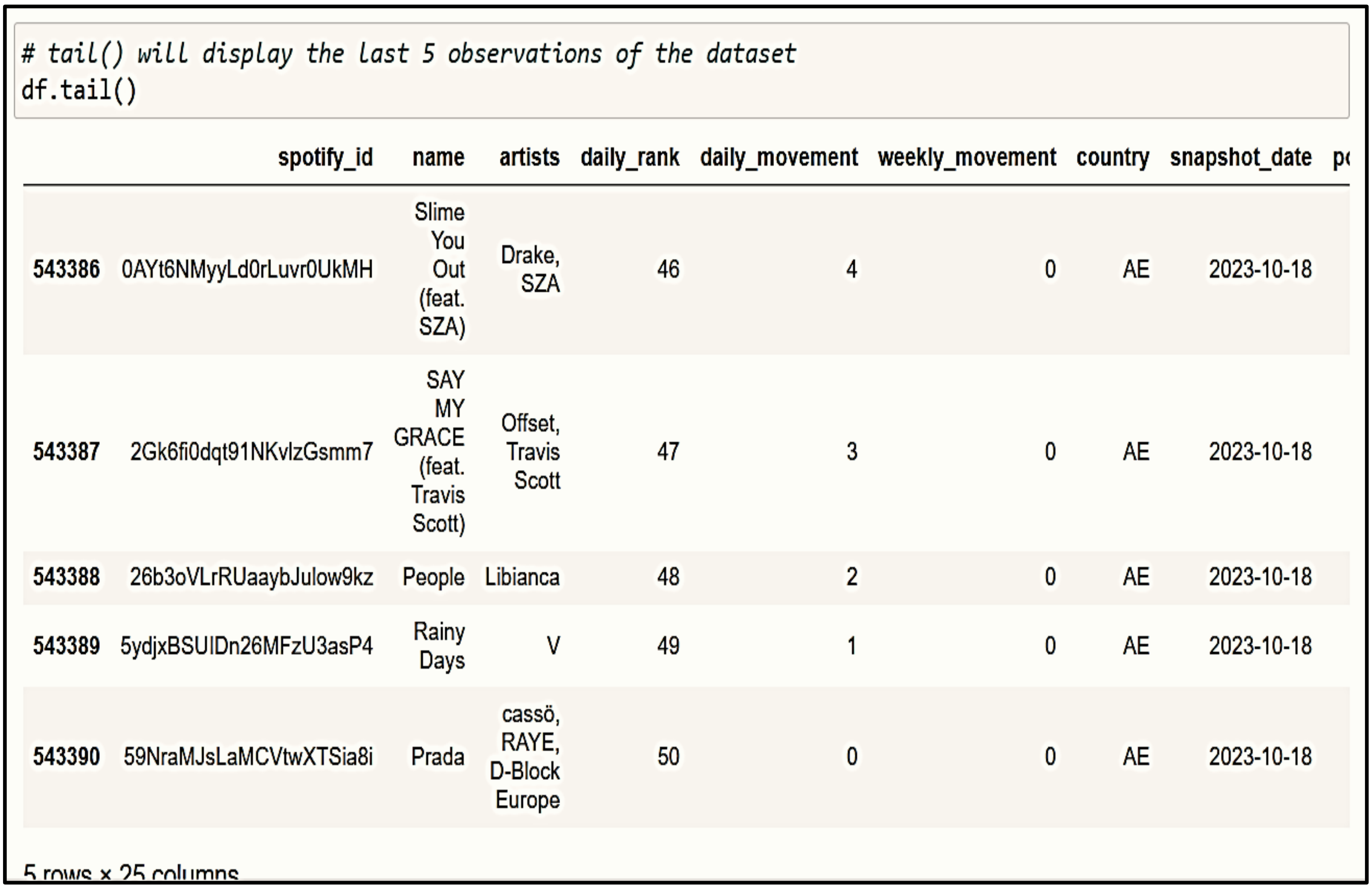


```python
# shape will display the number of observations(rows) and features(columns) in the dataset
df.shape
```

```
(543391, 25)
```

```python
# head() will display the top 5 observations of the dataset
df.head()
```

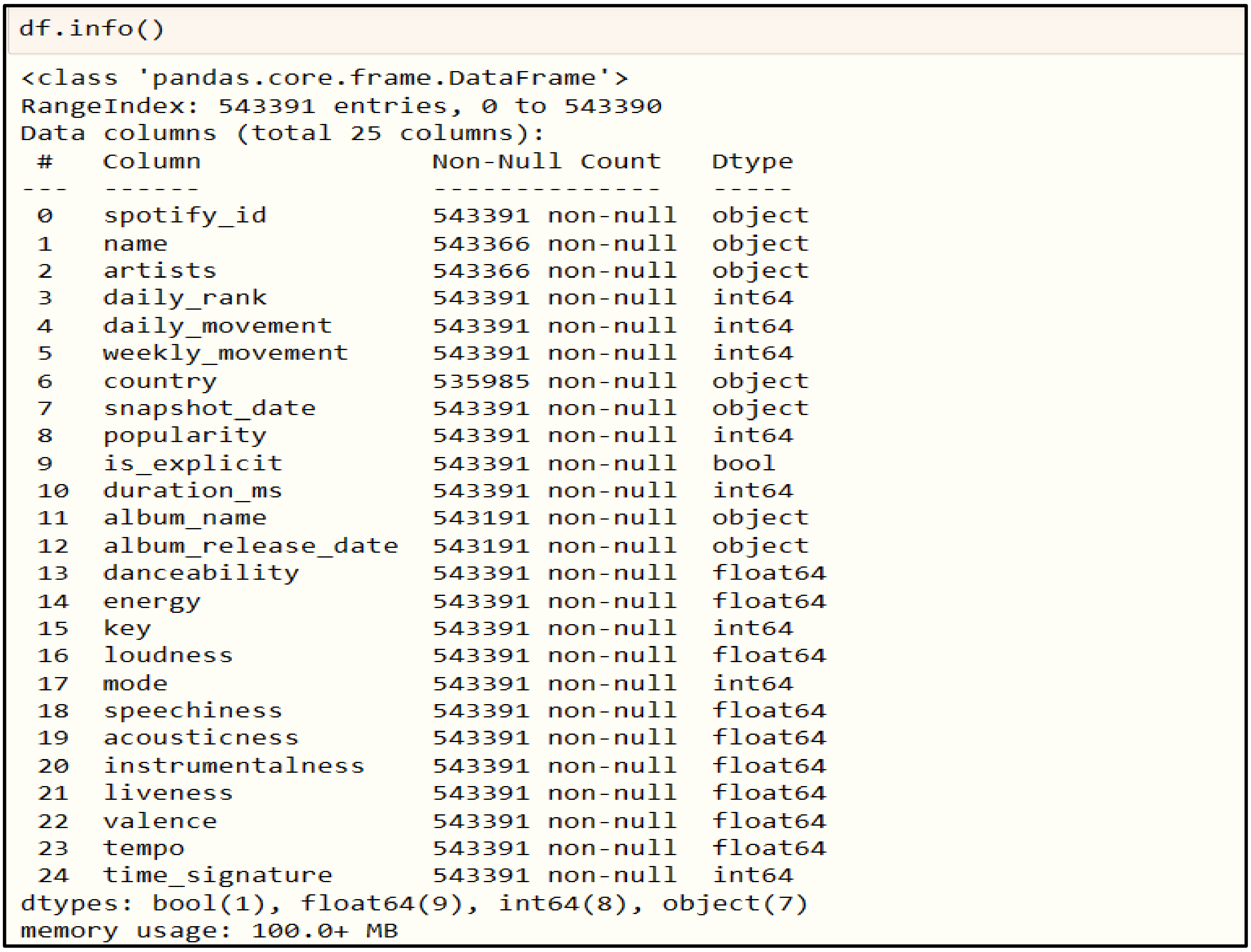
| | spotify_id | name | artists | daily_rank | daily_movement | weekly_movement | country | snapshot_dat |
|---|---|---|---|---|---|---|---|---|
| 0 | 51ZQ1vr10ffzbwljDCwqm4 | we can't be friends (wait for your love) | Ariana Grande | 1 | 0 | 49 | NaN | 2024-03-1 |
| 1 | 3qhlB30KknSejmlvZZLjOD | End of Beginning | Djo | 2 | 1 | 1 | NaN | 2024-03-1 |
| 2 | 6tNQ70jh4OwmPGpYy6R2o9 | Beautiful Things | Benson Boone | 3 | -1 | -2 | NaN | 2024-03-1 |
| 3 | 3w0w2T288dec0mgeZZqoNN | CARNIVAL | ¥ , *KanyeWest, TyDolla* ign, Rich The Kid, P... | 4 | 0 | -2 | NaN | 2024-03-1 |
| 4 | 3rUGC1vUnkDG9CZFHMur1t | greedy | Tate McRae | 5 | 0 | 1 | NaN | 2024-03-1 |



```python
# tail() will display the last 5 observations of the dataset
df.tail()
```

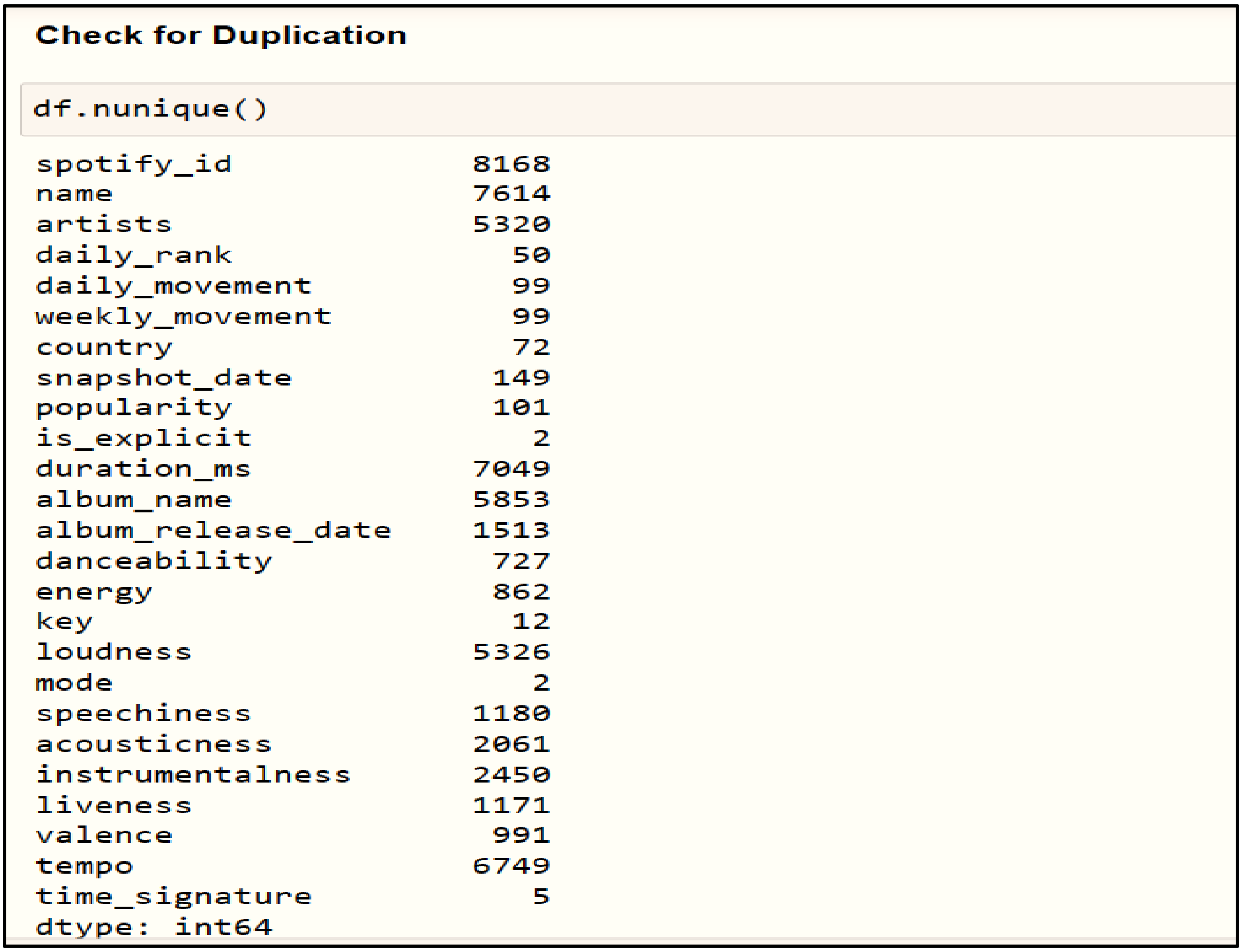
| | spotify_id | name | artists | daily_rank | daily_movement | weekly_movement | country | snapshot_date |
|---|---|---|---|---|---|---|---|---|
| **543386** | 0AYt6NMyyLd0rLuvr0UkMH | Slime You Out (feat. SZA) | Drake, SZA | 46 | 4 | 0 | AE | 2023-10-18 |
| **543387** | 2Gk6fi0dqt91NKvlzGsmm7 | SAY MY GRACE (feat. Travis Scott) | Offset, Travis Scott | 47 | 3 | 0 | AE | 2023-10-18 |
| **543388** | 26b3oVLrRUaaybJulow9kz | People | Libianca | 48 | 2 | 0 | AE | 2023-10-18 |
| **543389** | 5ydjxBSUIDn26MFzU3asP4 | Rainy Days | V | 49 | 1 | 0 | AE | 2023-10-18 |
| **543390** | 59NraMJsLaMCVtwXTSia8i | Prada | cassö, RAYE, D-Block Europe | 50 | 0 | 0 | AE | 2023-10-18 |

5 rows × 25 columns



```python
df.info()
```

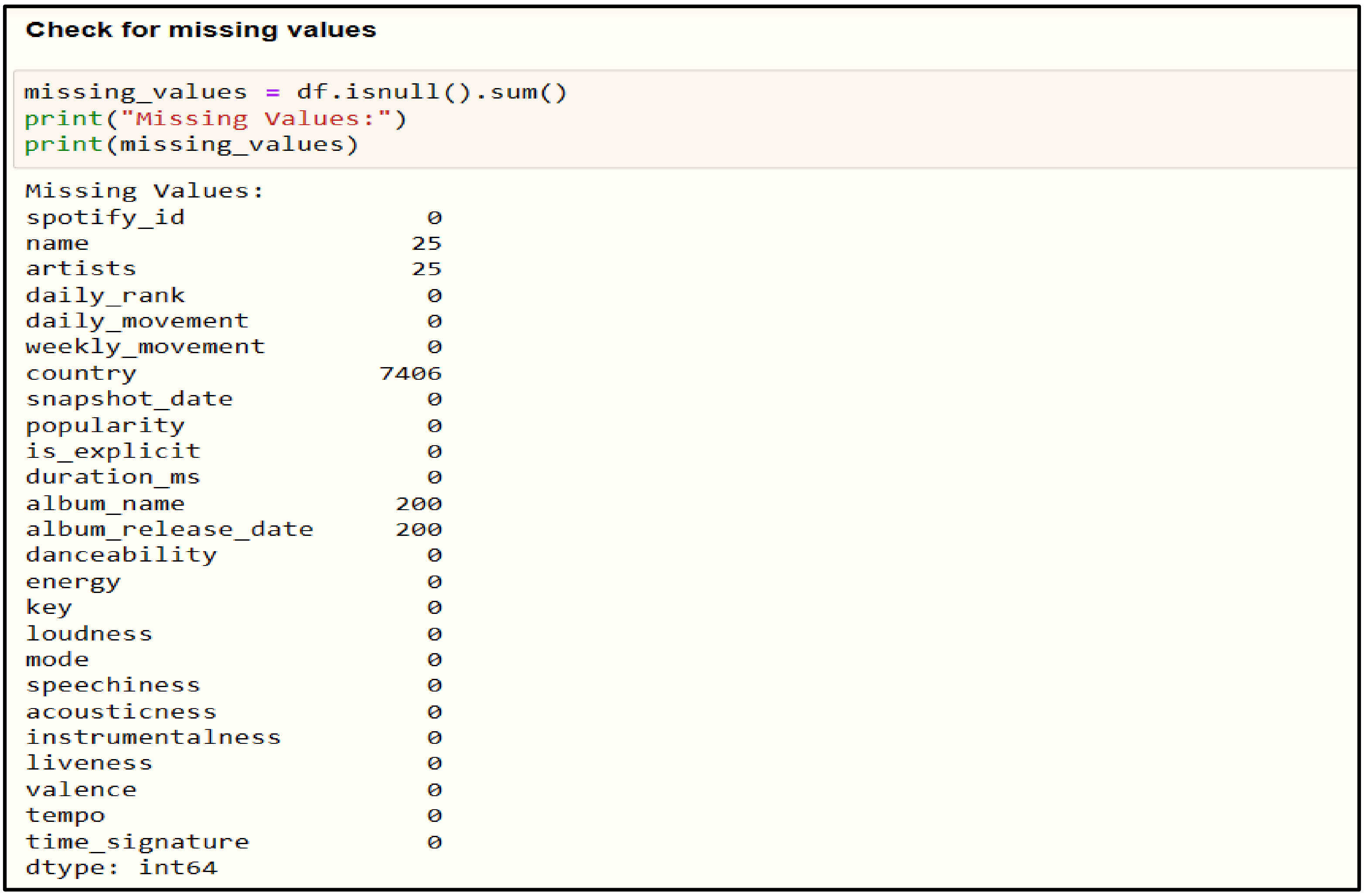
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 543391 entries, 0 to 543390
Data columns (total 25 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   spotify_id          543391 non-null  object 
 1   name                543366 non-null  object 
 2   artists             543366 non-null  object 
 3   daily_rank          543391 non-null  int64  
 4   daily_movement      543391 non-null  int64  
 5   weekly_movement     543391 non-null  int64  
 6   country             535985 non-null  object 
 7   snapshot_date       543391 non-null  object 
 8   popularity          543391 non-null  int64  
 9   is_explicit         543391 non-null  bool   
 10  duration_ms         543391 non-null  int64  
 11  album_name          543191 non-null  object 
 12  album_release_date  543191 non-null  object 
 13  danceability        543391 non-null  float64
 14  energy              543391 non-null  float64
 15  key                 543391 non-null  int64  
 16  loudness            543391 non-null  float64
 17  mode                543391 non-null  int64  
 18  speechiness         543391 non-null  float64
 19  acousticness        543391 non-null  float64
 20  instrumentalness    543391 non-null  float64
 21  liveness            543391 non-null  float64
 22  valence             543391 non-null  float64
 23  tempo               543391 non-null  float64
 24  time_signature      543391 non-null  int64  
dtypes: bool(1), float64(9), int64(8), object(7)
memory usage: 100.0+ MB
```



**Check for Duplication**

```
df.nunique()
```

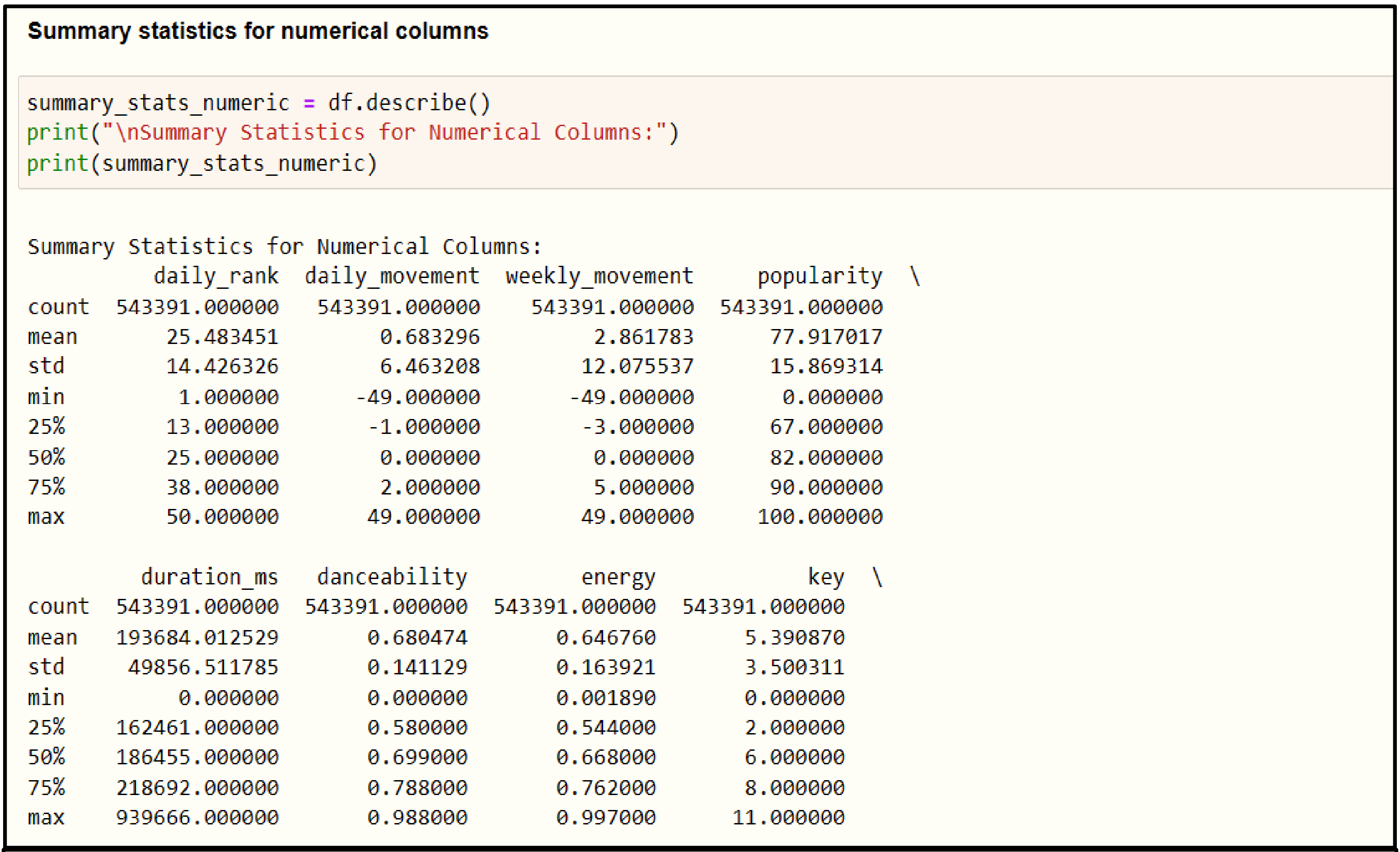
```
spotify_id            8168
name                  7614
artists               5320
daily_rank              50
daily_movement          99
weekly_movement         99
country                 72
snapshot_date          149
popularity             101
is_explicit              2
duration_ms           7049
album_name            5853
album_release_date    1513
danceability           727
energy                 862
key                     12
loudness              5326
mode                     2
speechiness           1180
acousticness          2061
instrumentalness      2450
liveness              1171
valence                991
tempo                 6749
time_signature           5
dtype: int64
```



**Check for missing values**

```
missing_values = df.isnull().sum()
print("Missing Values:")
print(missing_values)
```

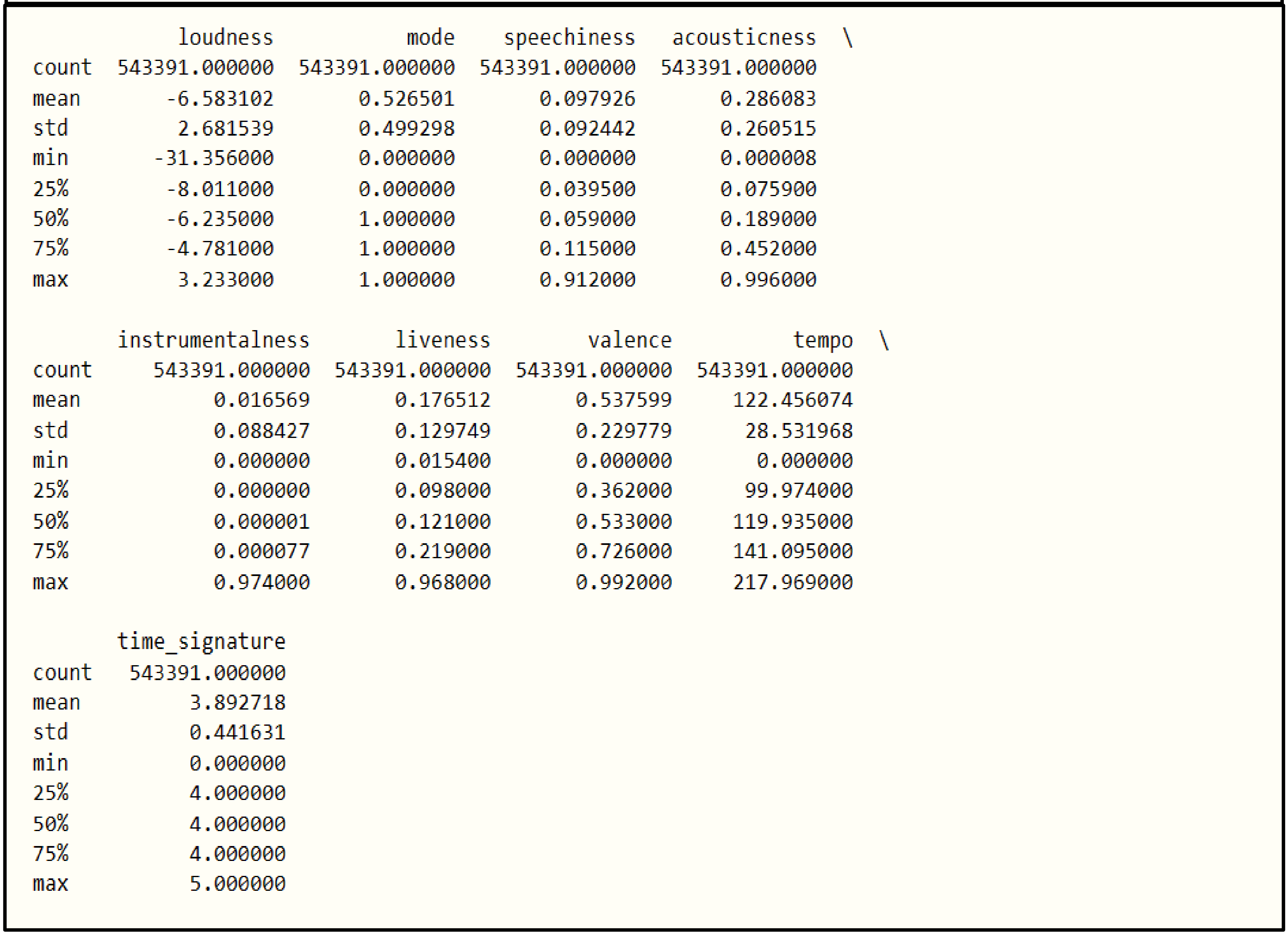
```
Missing Values:
spotify_id               0
name                    25
artists                 25
daily_rank               0
daily_movement           0
weekly_movement          0
country               7406
snapshot_date            0
popularity               0
is_explicit              0
duration_ms              0
album_name             200
album_release_date     200
danceability             0
energy                   0
key                      0
loudness                 0
mode                     0
speechiness              0
acousticness             0
instrumentalness         0
liveness                 0
valence                  0
tempo                    0
time_signature           0
dtype: int64
```



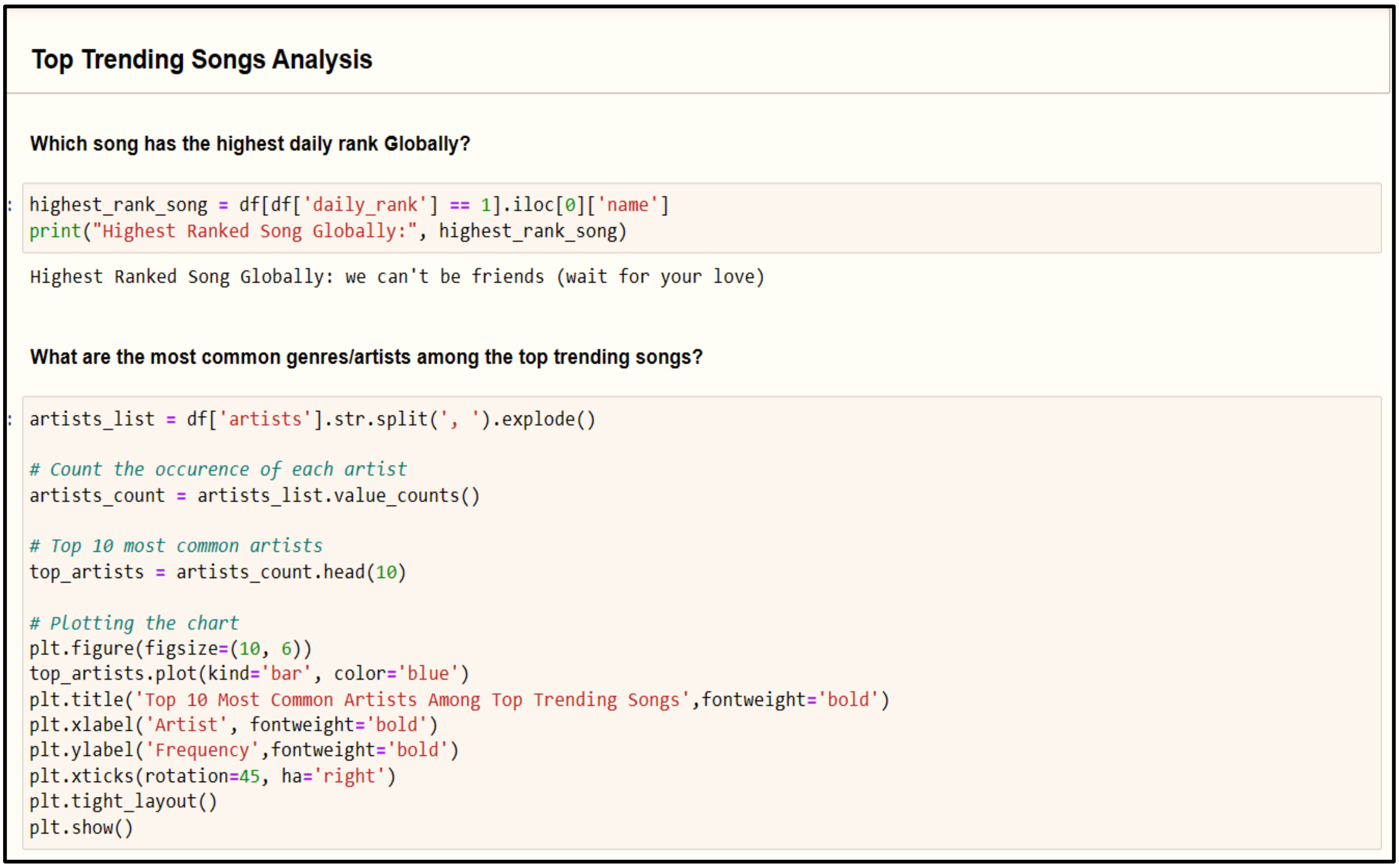
**Summary statistics for numerical columns**

```python
summary_stats_numeric = df.describe()
print("\nSummary Statistics for Numerical Columns:")
print(summary_stats_numeric)
```

```
Summary Statistics for Numerical Columns:
          daily_rank  daily_movement  weekly_movement     popularity  \
count  543391.000000   543391.000000    543391.000000  543391.000000   
mean       25.483451        0.683296         2.861783      77.917017   
std        14.426326        6.463208        12.075537      15.869314   
min         1.000000      -49.000000       -49.000000       0.000000   
25%        13.000000       -1.000000        -3.000000      67.000000   
50%        25.000000        0.000000         0.000000      82.000000   
75%        38.000000        2.000000         5.000000      90.000000   
max        50.000000       49.000000        49.000000     100.000000   

         duration_ms   danceability         energy            key  \
count  543391.000000  543391.000000  543391.000000  543391.000000   
mean   193684.012529       0.680474       0.646760       5.390870   
std     49856.511785       0.141129       0.163921       3.500311   
min         0.000000       0.000000       0.001890       0.000000   
25%    162461.000000       0.580000       0.544000       2.000000   
50%    186455.000000       0.699000       0.668000       6.000000   
75%    218692.000000       0.788000       0.762000       8.000000   
max    939666.000000       0.988000       0.997000      11.000000   
```



```
            loudness           mode    speechiness   acousticness  \
count  543391.000000  543391.000000  543391.000000  543391.000000   
mean       -6.583102       0.526501       0.097926       0.286083   
std         2.681539       0.499298       0.092442       0.260515   
min       -31.356000       0.000000       0.000000       0.000008   
25%        -8.011000       0.000000       0.039500       0.075900   
50%        -6.235000       1.000000       0.059000       0.189000   
75%        -4.781000       1.000000       0.115000       0.452000   
max         3.233000       1.000000       0.912000       0.996000   

       instrumentalness       liveness        valence          tempo  \
count     543391.000000  543391.000000  543391.000000  543391.000000   
mean           0.016569       0.176512       0.537599     122.456074   
std            0.088427       0.129749       0.229779      28.531968   
min            0.000000       0.015400       0.000000       0.000000   
25%            0.000000       0.098000       0.362000      99.974000   
50%            0.000001       0.121000       0.533000     119.935000   
75%            0.000077       0.219000       0.726000     141.095000   
max            0.974000       0.968000       0.992000     217.969000   

       time_signature  
count   543391.000000  
mean         3.892718  
std          0.441631  
min          0.000000  
25%          4.000000  
50%          4.000000  
75%          4.000000  
max          5.000000  
```



## Top Trending Songs Analysis

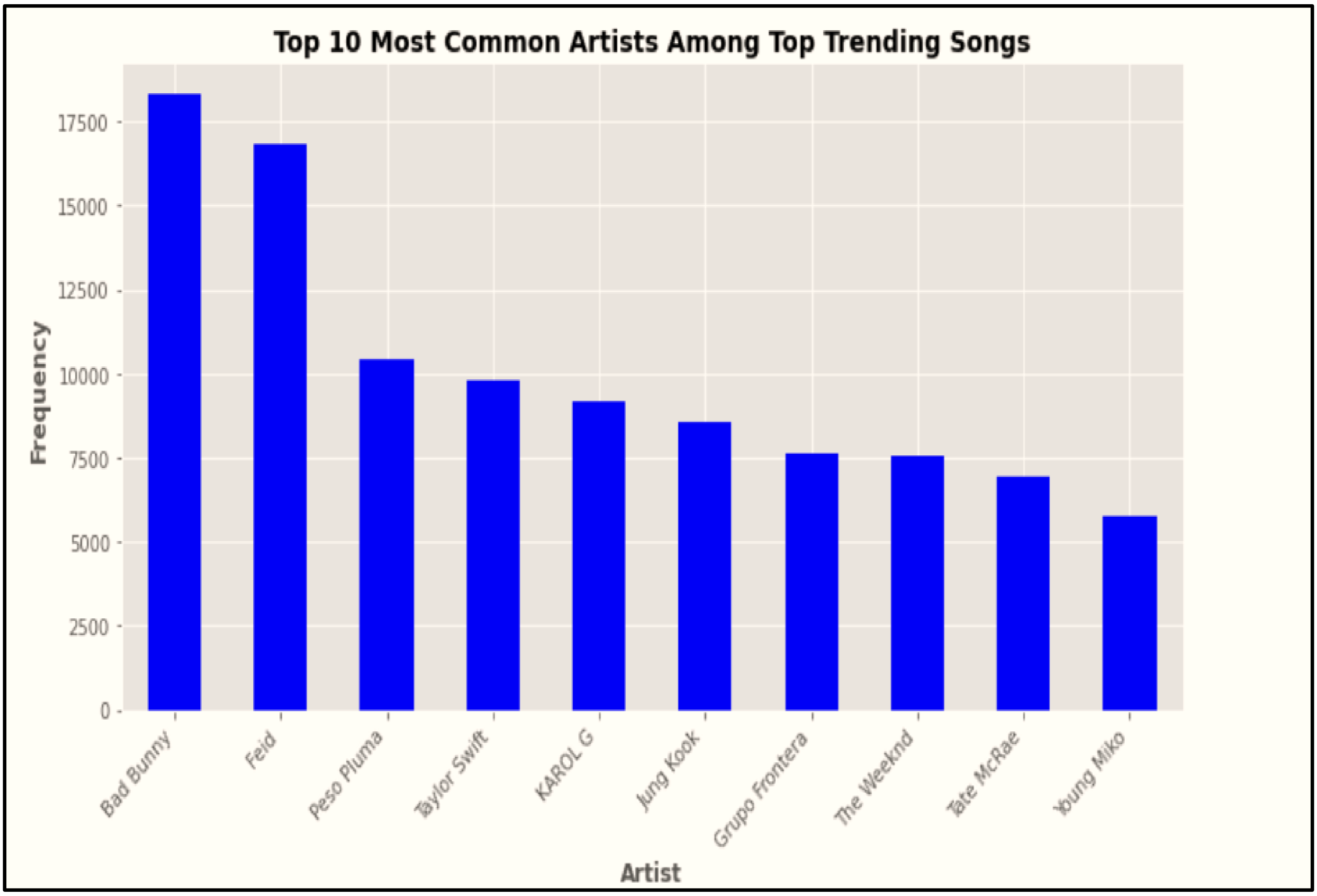
**Which song has the highest daily rank Globally?**

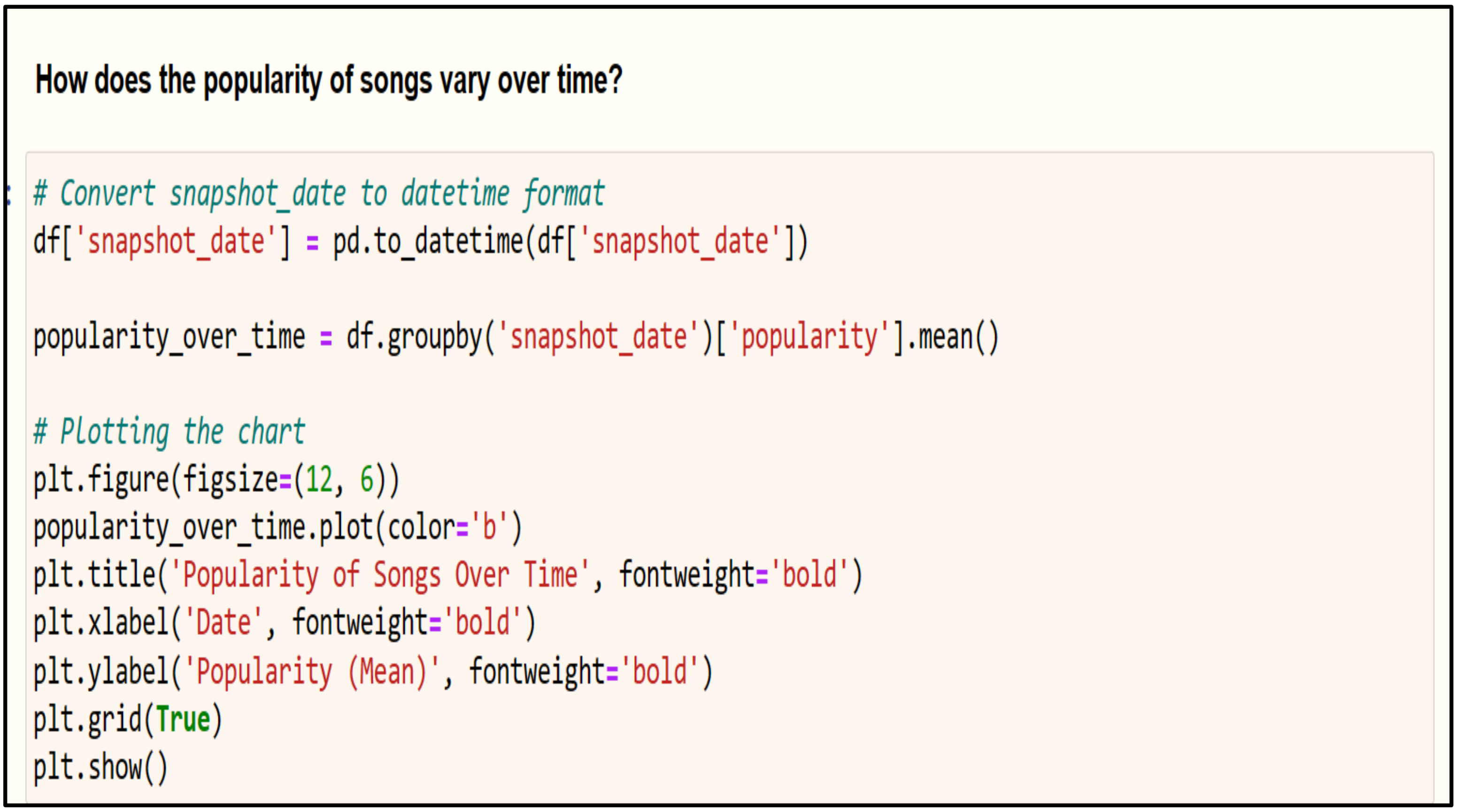
```
highest_rank_song = df[df['daily_rank'] == 1].iloc[0]['name']
print("Highest Ranked Song Globally:", highest_rank_song)
```

```
Highest Ranked Song Globally: we can't be friends (wait for your love)
```

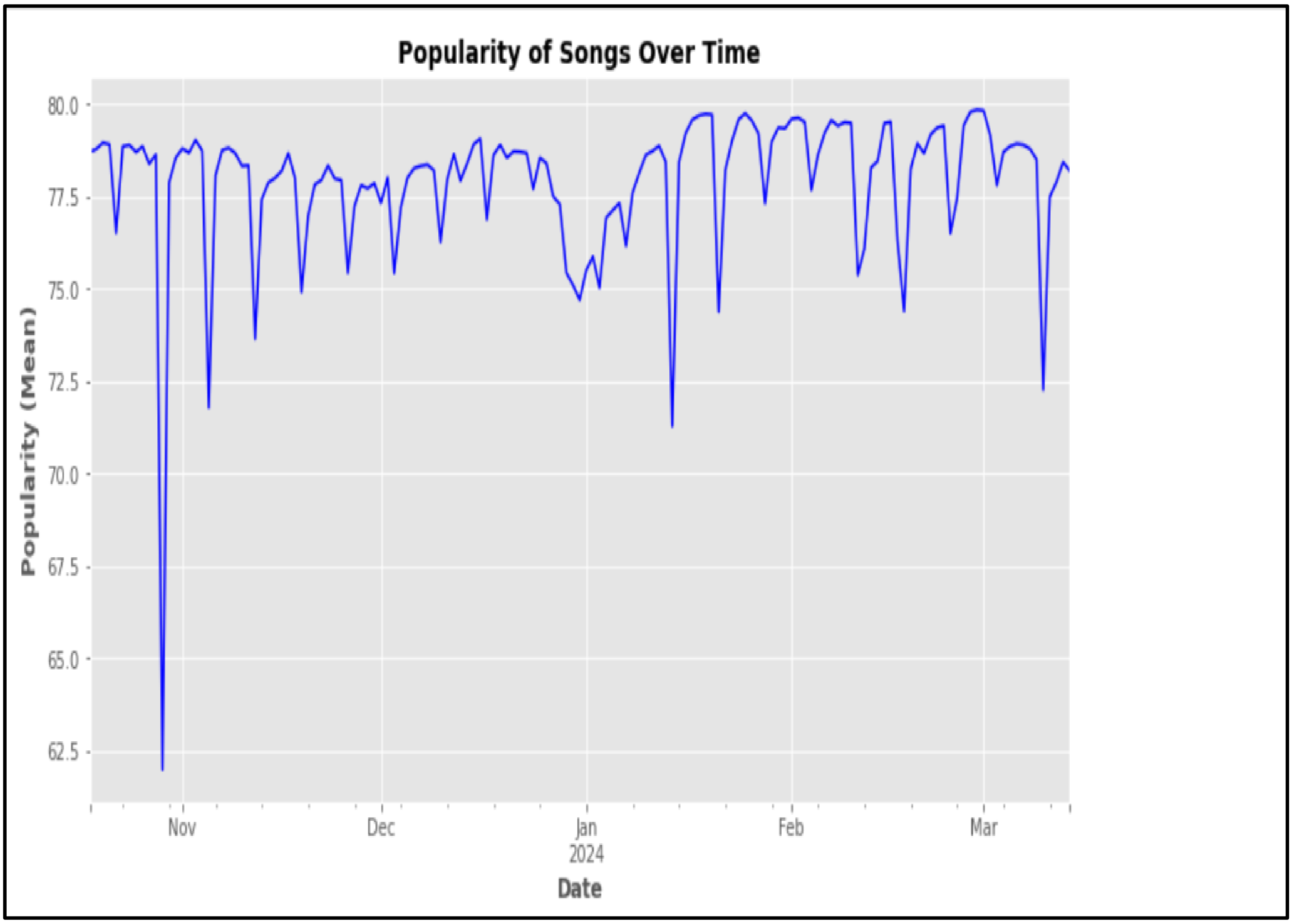
**What are the most common genres/artists among the top trending songs?**

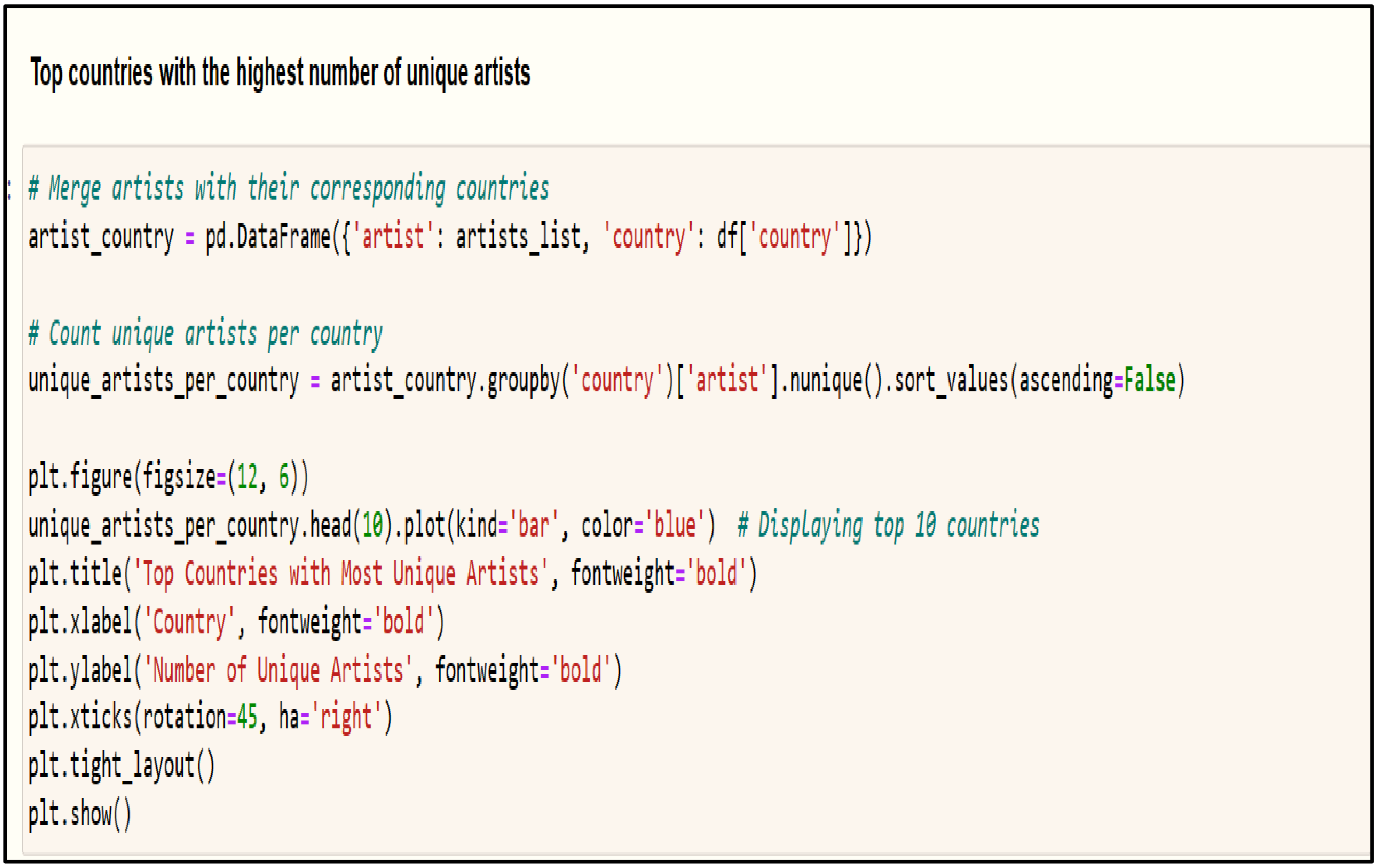
```
artists_list = df['artists'].str.split(', ').explode()

# Count the occurence of each artist
artists_count = artists_list.value_counts()

# Top 10 most common artists
top_artists = artists_count.head(10)

# Plotting the chart
plt.figure(figsize=(10, 6))
top_artists.plot(kind='bar', color='blue')
plt.title('Top 10 Most Common Artists Among Top Trending Songs',fontweight='bold')
plt.xlabel('Artist', fontweight='bold')
plt.ylabel('Frequency',fontweight='bold')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



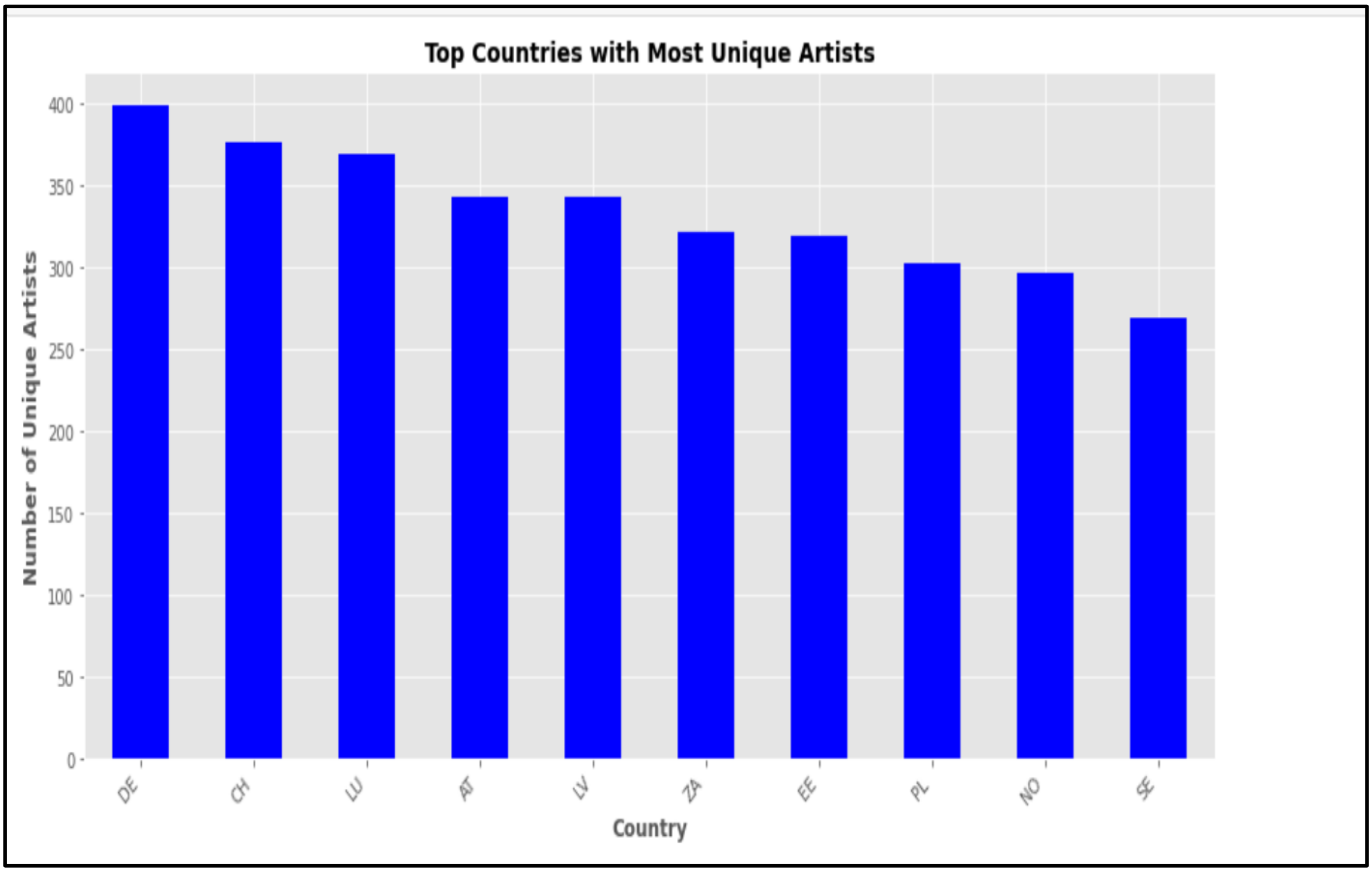

**How does the popularity of songs vary over time?**

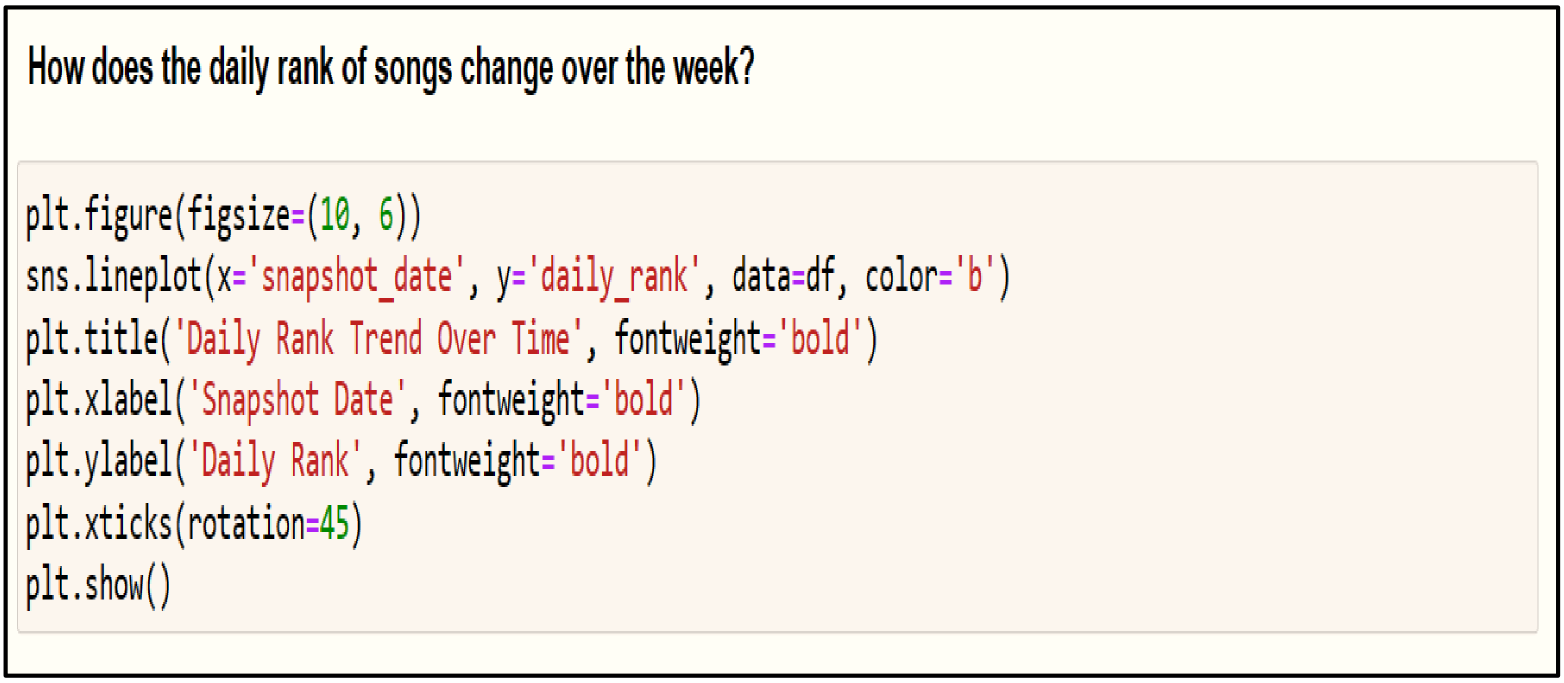
```
# Convert snapshot_date to datetime format
df['snapshot_date'] = pd.to_datetime(df['snapshot_date'])

popularity_over_time = df.groupby('snapshot_date')['popularity'].mean()

# Plotting the chart
plt.figure(figsize=(12, 6))
popularity_over_time.plot(color='b')
plt.title('Popularity of Songs Over Time', fontweight='bold')
plt.xlabel('Date', fontweight='bold')
plt.ylabel('Popularity (Mean)', fontweight='bold')
plt.grid(True)
plt.show()
```




Top countries with the highest number of unique artists

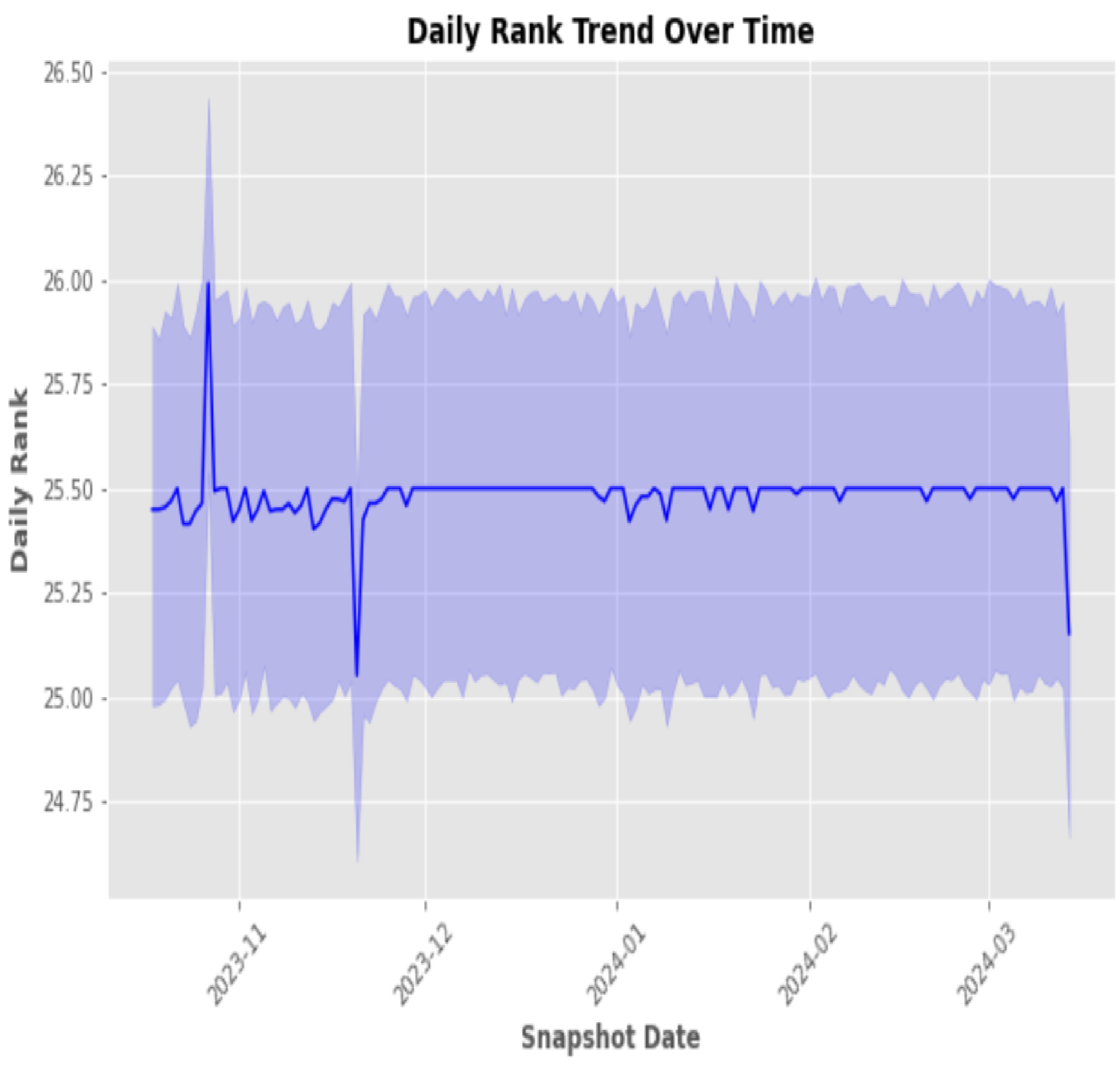
```
# Merge artists with their corresponding countries
artist_country = pd.DataFrame({'artist': artists_list, 'country': df['country']})

# Count unique artists per country
unique_artists_per_country = artist_country.groupby('country')['artist'].nunique().sort_values(ascending=False)

plt.figure(figsize=(12, 6))
unique_artists_per_country.head(10).plot(kind='bar', color='blue')  # Displaying top 10 countries
plt.title('Top Countries with Most Unique Artists', fontweight='bold')
plt.xlabel('Country', fontweight='bold')
plt.ylabel('Number of Unique Artists', fontweight='bold')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

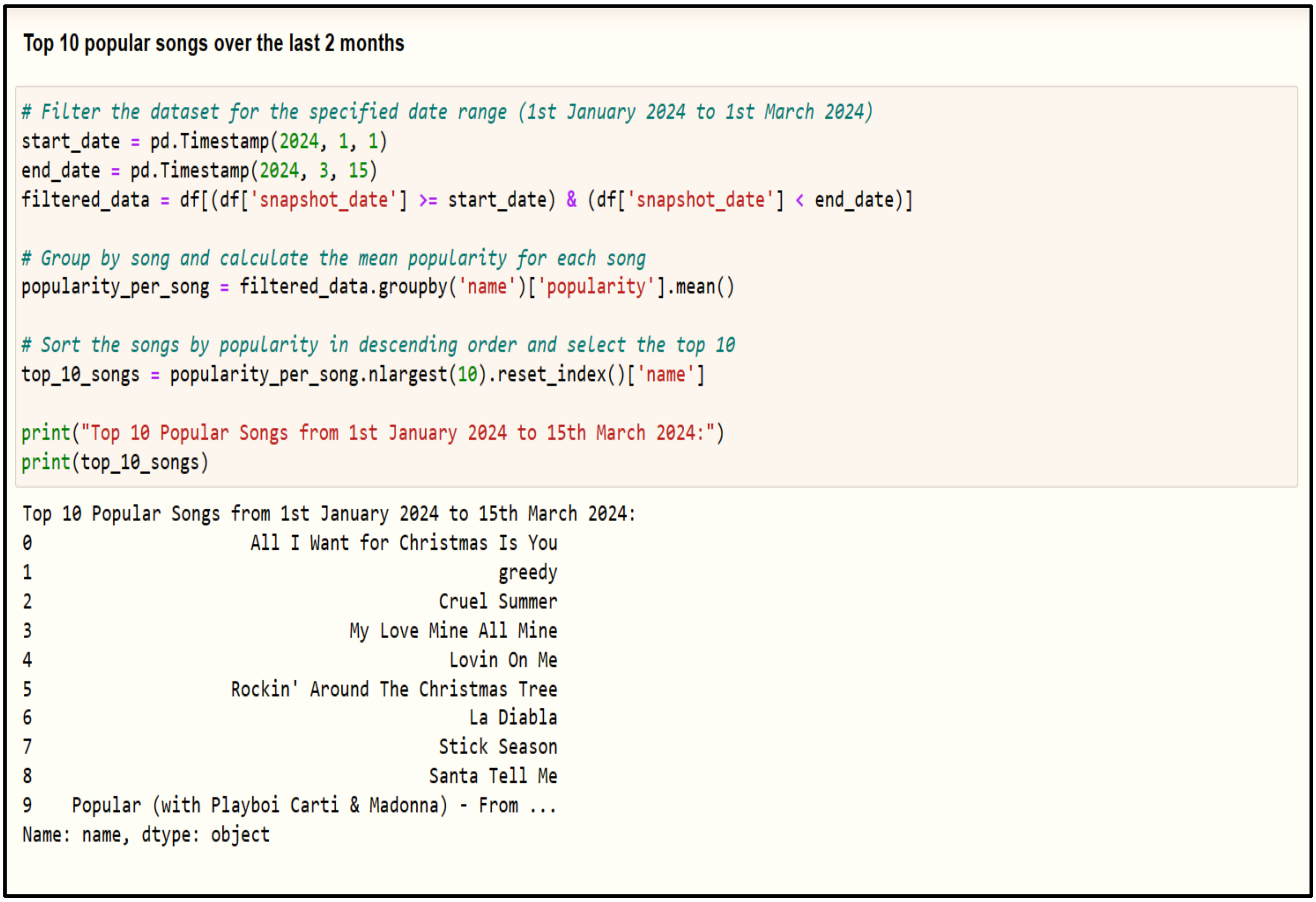




How does the daily rank of songs change over the week?

```
plt.figure(figsize=(10, 6))
sns.lineplot(x='snapshot_date', y='daily_rank', data=df, color='b')
plt.title('Daily Rank Trend Over Time', fontweight='bold')
plt.xlabel('Snapshot Date', fontweight='bold')
plt.ylabel('Daily Rank', fontweight='bold')
plt.xticks(rotation=45)
plt.show()
```



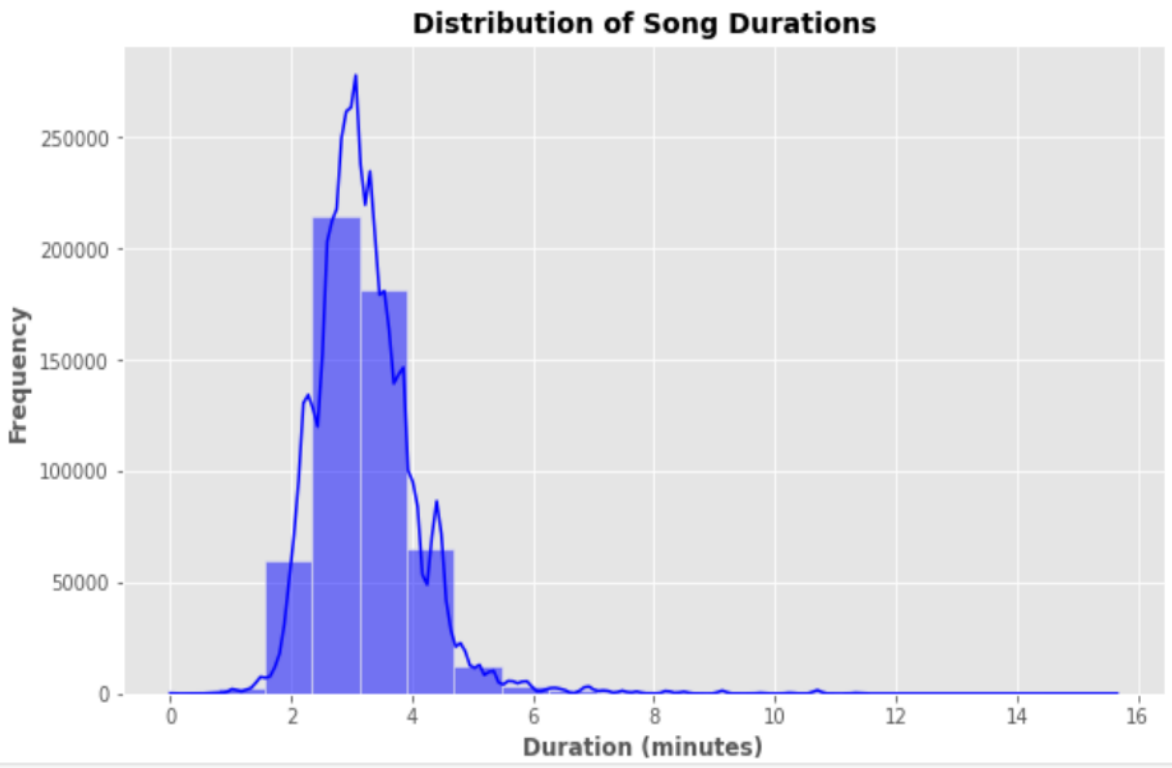


Top 10 popular songs over the last 2 months

```
# Filter the dataset for the specified date range (1st January 2024 to 1st March 2024)
start_date = pd.Timestamp(2024, 1, 1)
end_date = pd.Timestamp(2024, 3, 15)
filtered_data = df[(df['snapshot_date'] >= start_date) & (df['snapshot_date'] < end_date)]

# Group by song and calculate the mean popularity for each song
popularity_per_song = filtered_data.groupby('name')['popularity'].mean()

# Sort the songs by popularity in descending order and select the top 10
top_10_songs = popularity_per_song.nlargest(10).reset_index()['name']

print("Top 10 Popular Songs from 1st January 2024 to 15th March 2024:")
print(top_10_songs)
```

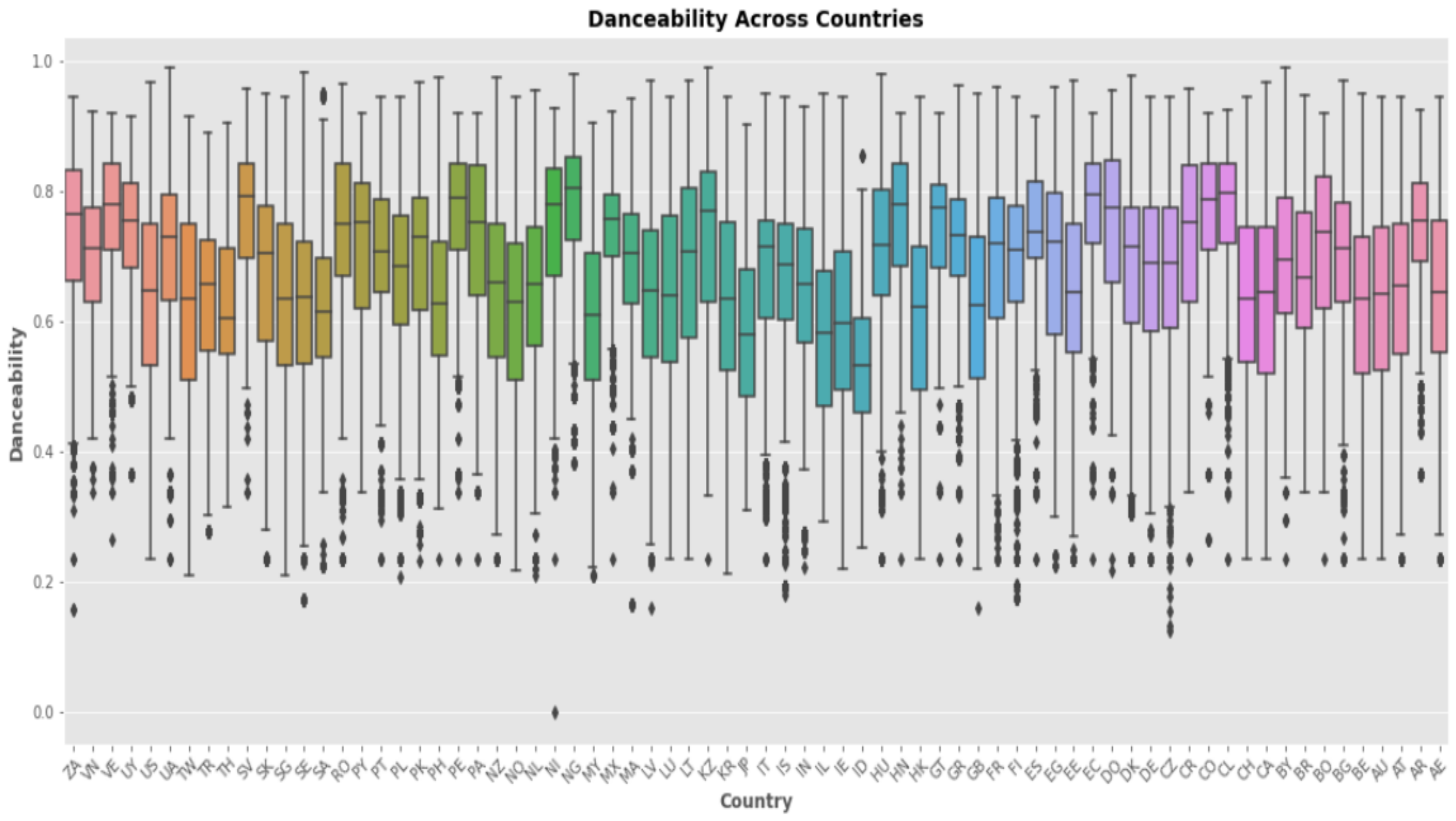
```
Top 10 Popular Songs from 1st January 2024 to 15th March 2024:
0                       All I Want for Christmas Is You
1                                                greedy
2                                          Cruel Summer
3                                      My Love Mine All Mine
4                                            Lovin On Me
5                        Rockin' Around The Christmas Tree
6                                              La Diabla
7                                           Stick Season
8                                           Santa Tell Me
9    Popular (with Playboi Carti & Madonna) - From ...
Name: name, dtype: object
```

```
plt.figure(figsize=(10, 6))
sns.histplot(df['duration_ms'] / 60000, bins=20, kde=True, color='b')
plt.xlabel('Duration (minutes)', fontweight='bold')
plt.ylabel('Frequency', fontweight='bold')
plt.title('Distribution of Song Durations', fontweight='bold')
plt.show()
```



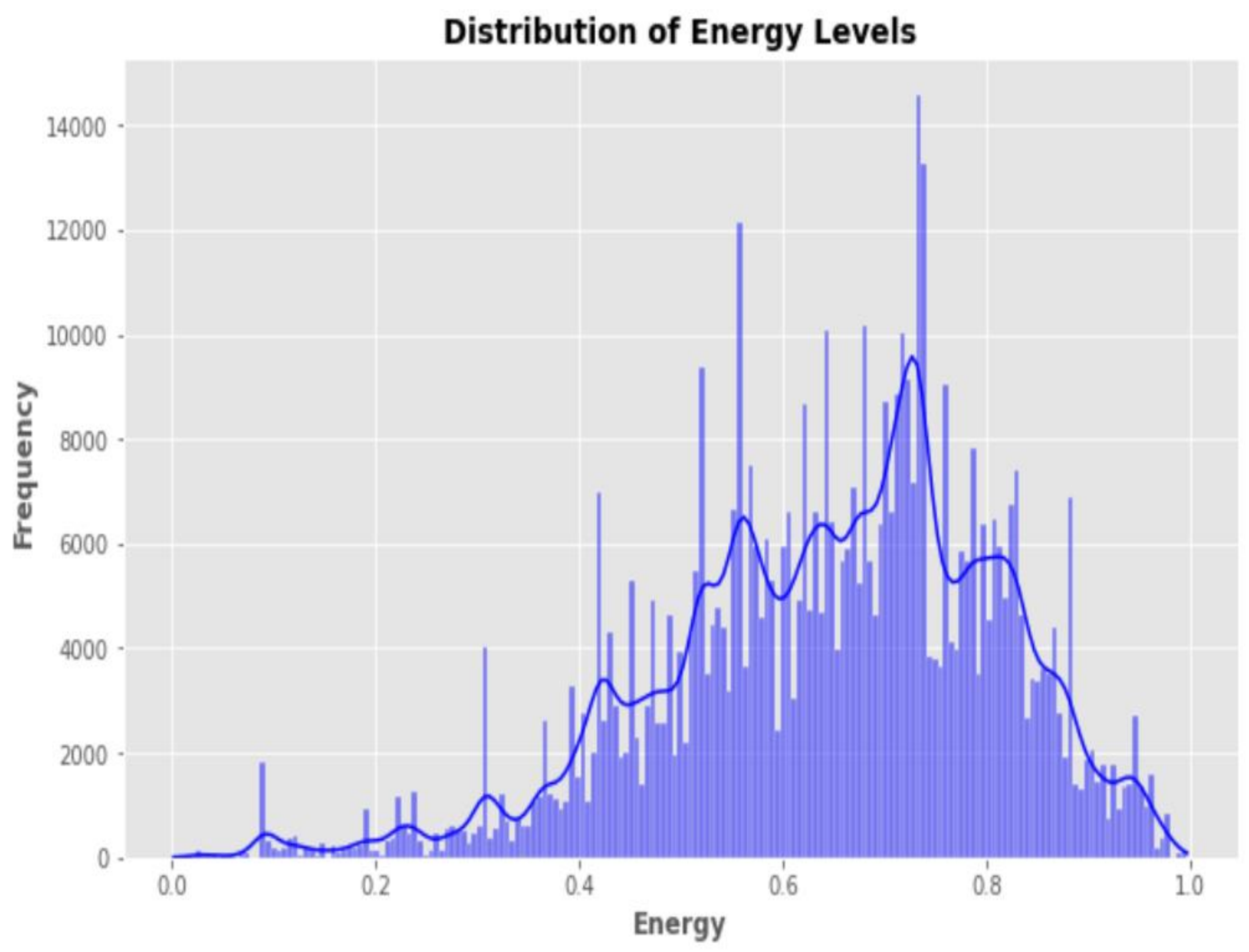
**Box plot showing danceability across different countries**

```
plt.figure(figsize=(18, 8))
sns.boxplot(x='country', y='danceability', data=df)
plt.xlabel('Country', fontweight='bold')
plt.ylabel('Danceability', fontweight='bold')
plt.title('Danceability Across Countries', fontweight='bold')
plt.xticks(rotation=45)
plt.show()
```



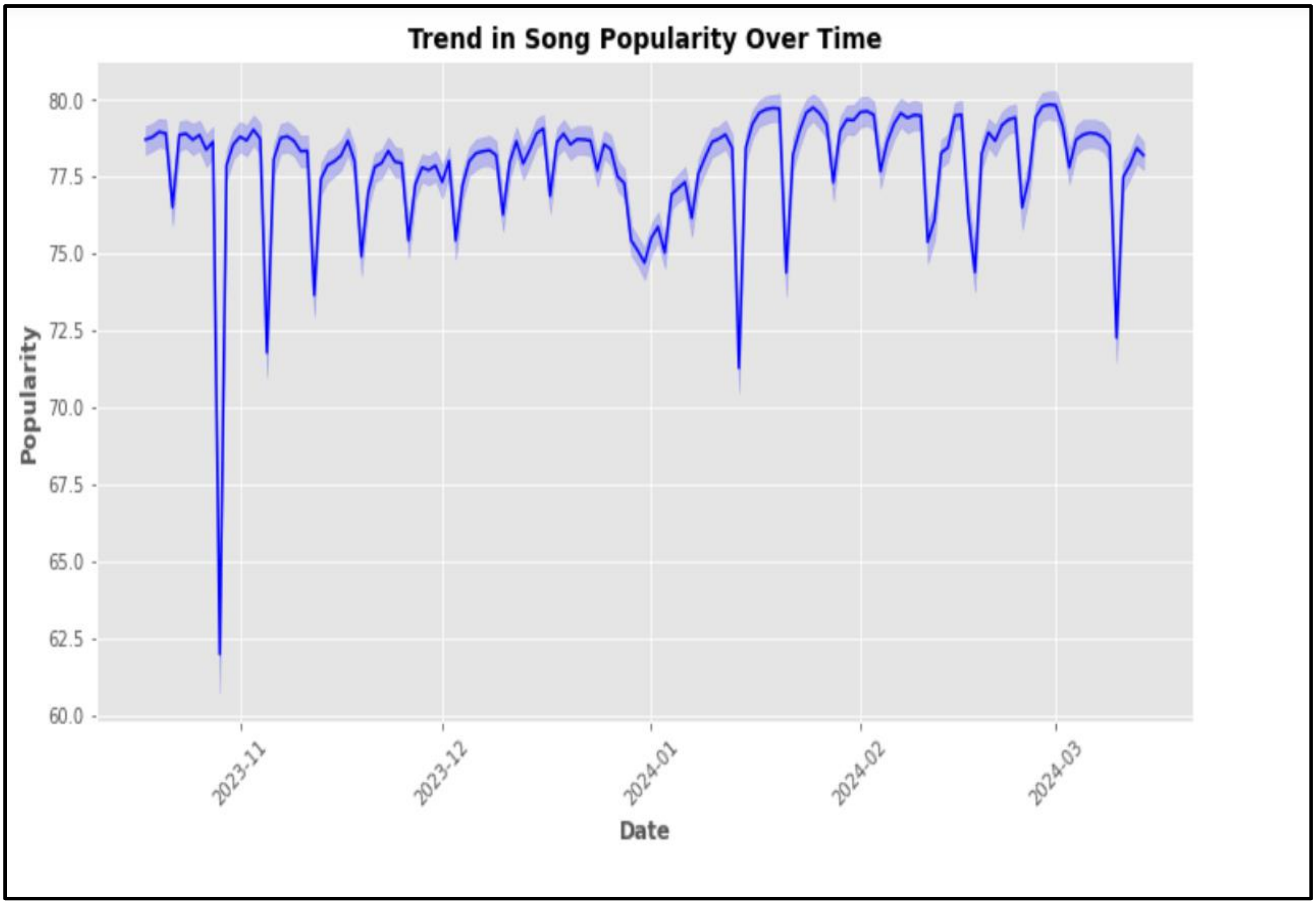
## Distribution of energy levels in the dataset

```
plt.figure(figsize=(10, 6))
sns.histplot(df['energy'], kde=True, color='b')
plt.xlabel('Energy', fontweight='bold')
plt.ylabel('Frequency', fontweight='bold')
plt.title('Distribution of Energy Levels', fontweight='bold')
plt.show()
```



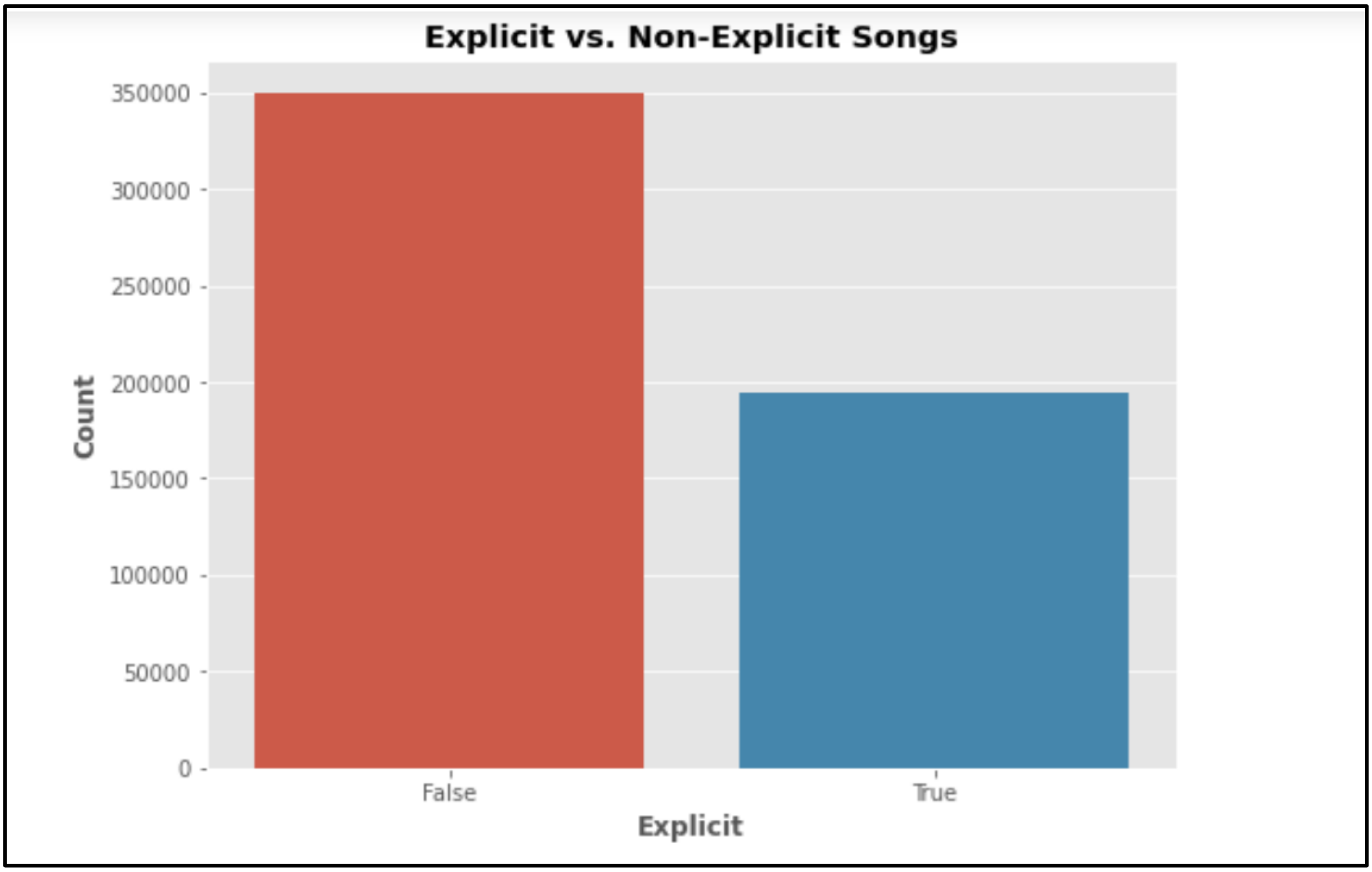
## Trend in song popularity over time

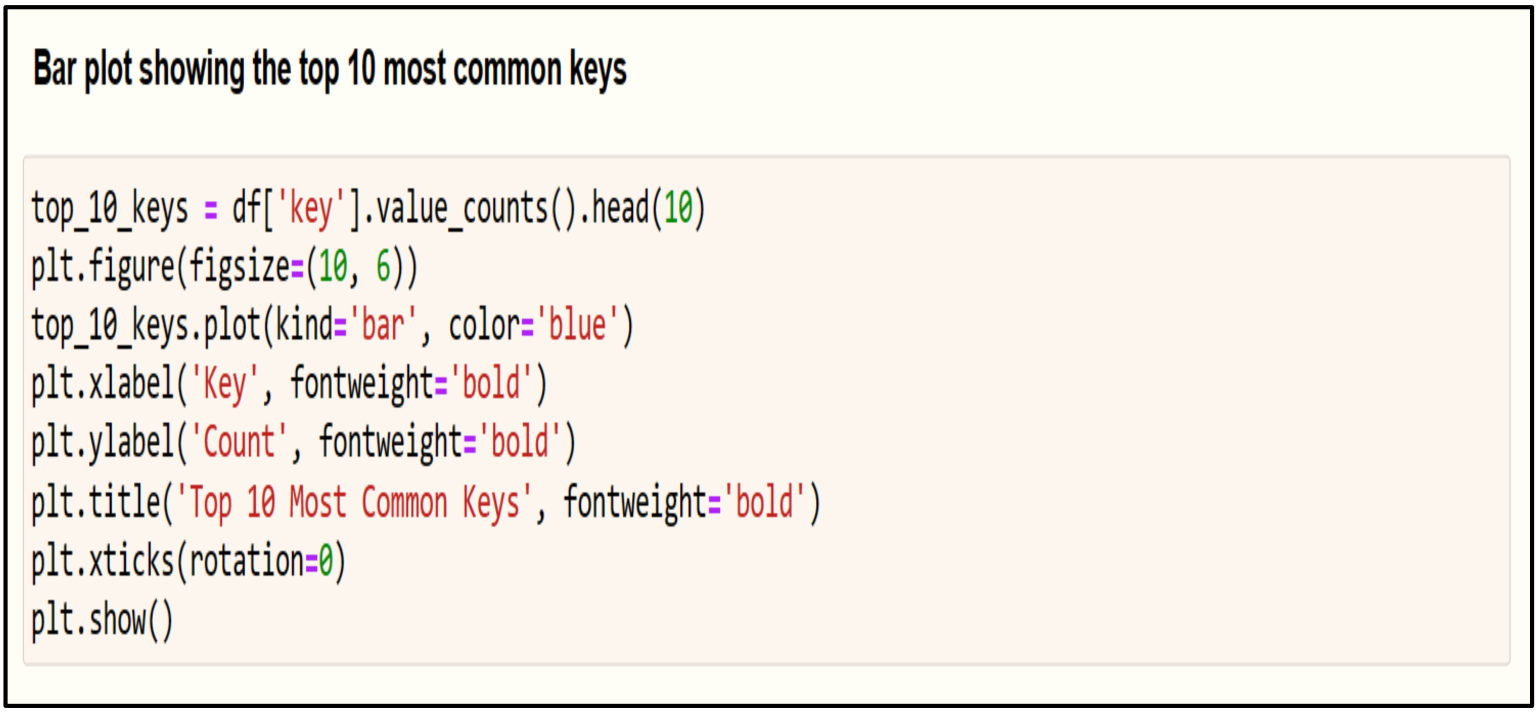
```
plt.figure(figsize=(12, 6))
sns.lineplot(x='snapshot_date', y='popularity', data=df, color='b')
plt.xlabel('Date', fontweight='bold')
plt.ylabel('Popularity',fontweight='bold')
plt.title('Trend in Song Popularity Over Time', fontweight='bold')
plt.xticks(rotation=45)
plt.show()
```



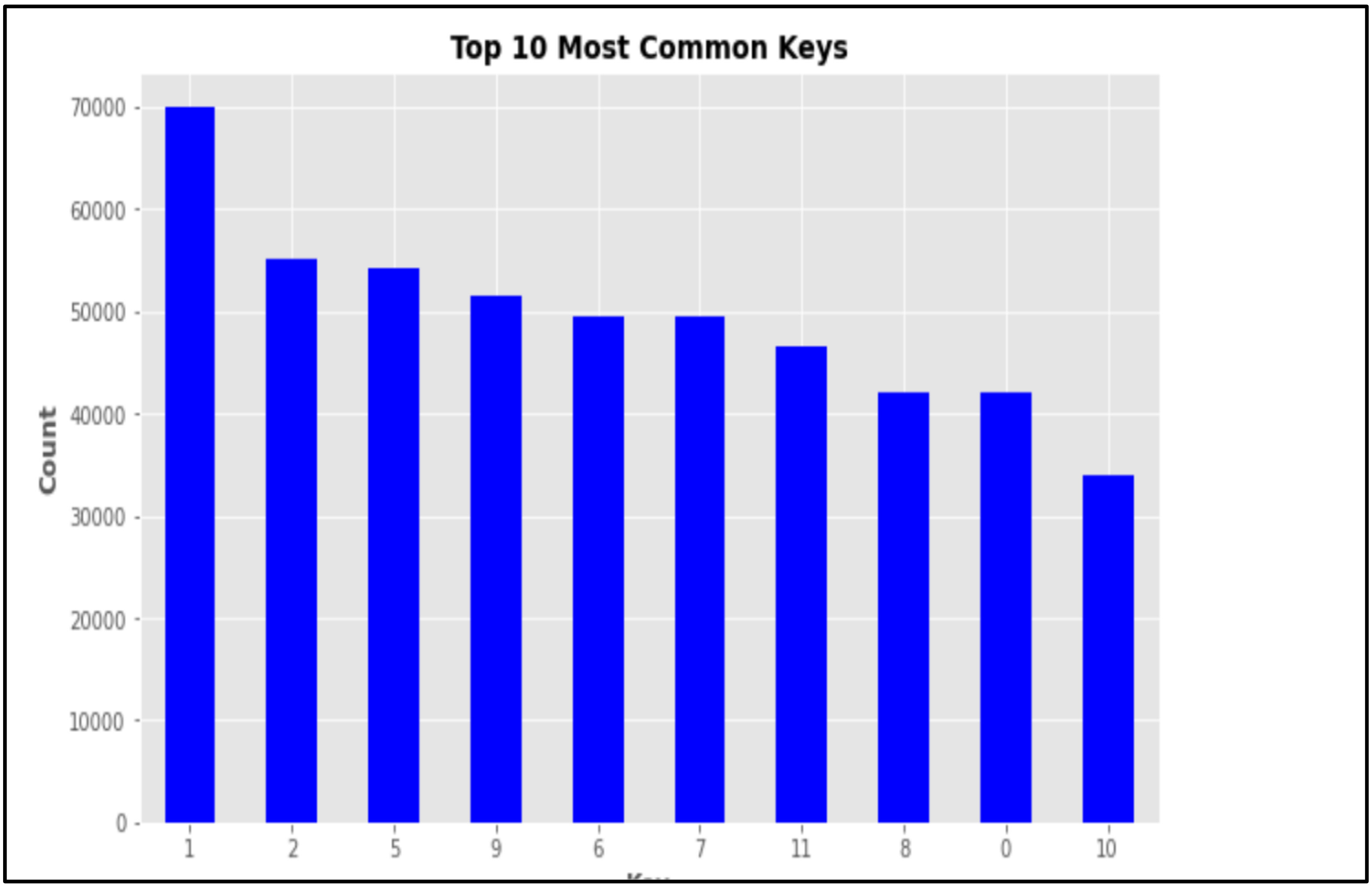
## Count plot showing explicit vs. non-explicit songs

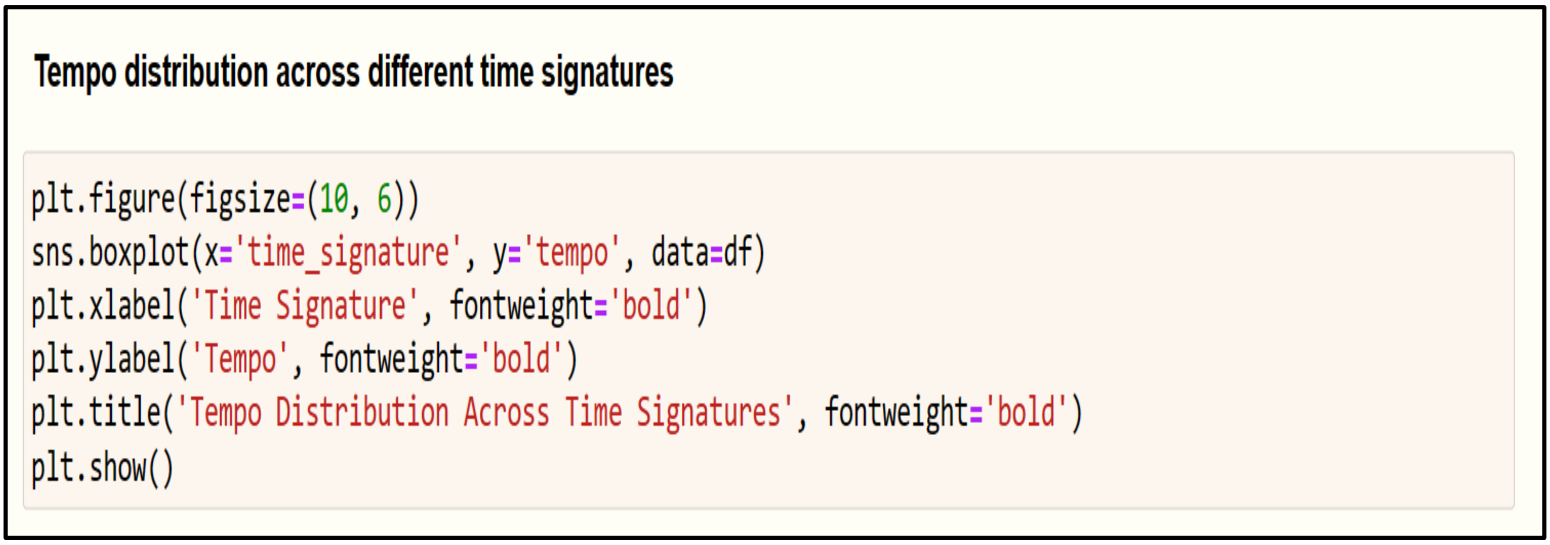
```
plt.figure(figsize=(8, 6))
sns.countplot(x='is_explicit', data=df)
plt.xlabel('Explicit', fontweight='bold')
plt.ylabel('Count', fontweight='bold')
plt.title('Explicit vs. Non-Explicit Songs', fontweight='bold')
plt.show()
```




Bar plot showing the top 10 most common keys

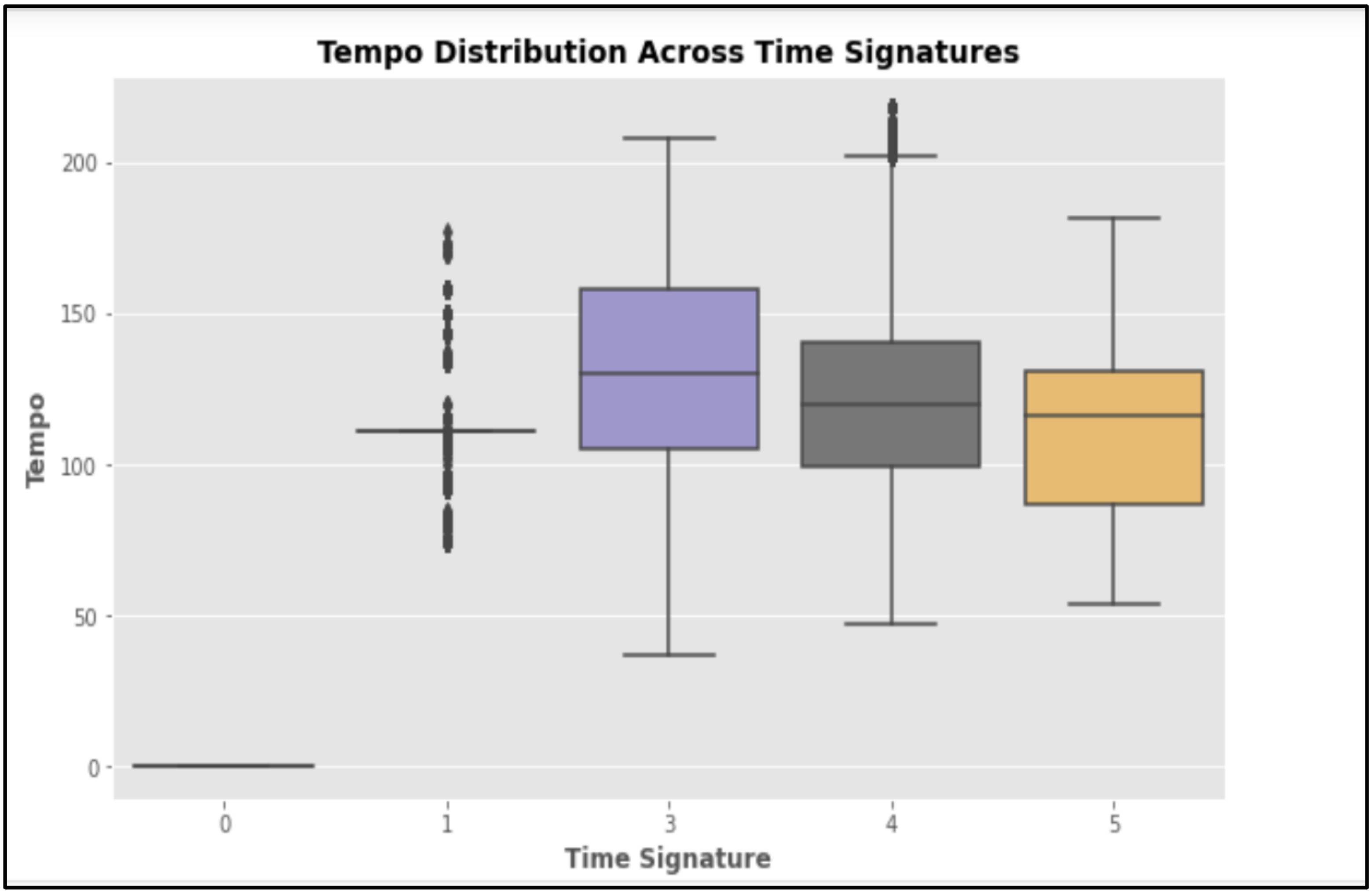
```
top_10_keys = df['key'].value_counts().head(10)
plt.figure(figsize=(10, 6))
top_10_keys.plot(kind='bar', color='blue')
plt.xlabel('Key', fontweight='bold')
plt.ylabel('Count', fontweight='bold')
plt.title('Top 10 Most Common Keys', fontweight='bold')
plt.xticks(rotation=0)
plt.show()
```





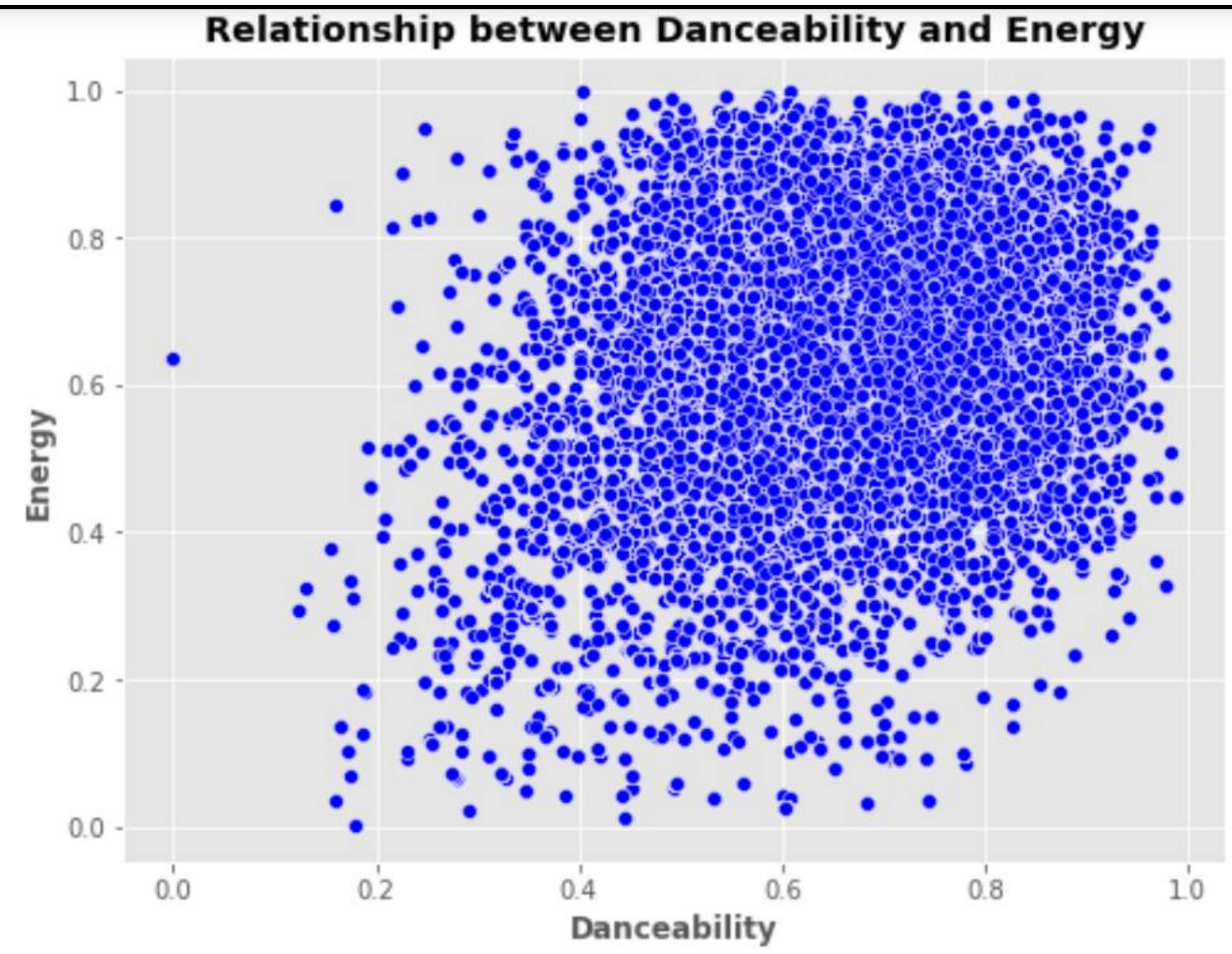
**Tempo distribution across different time signatures**

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='time_signature', y='tempo', data=df)
plt.xlabel('Time Signature', fontweight='bold')
plt.ylabel('Tempo', fontweight='bold')
plt.title('Tempo Distribution Across Time Signatures', fontweight='bold')
plt.show()
```



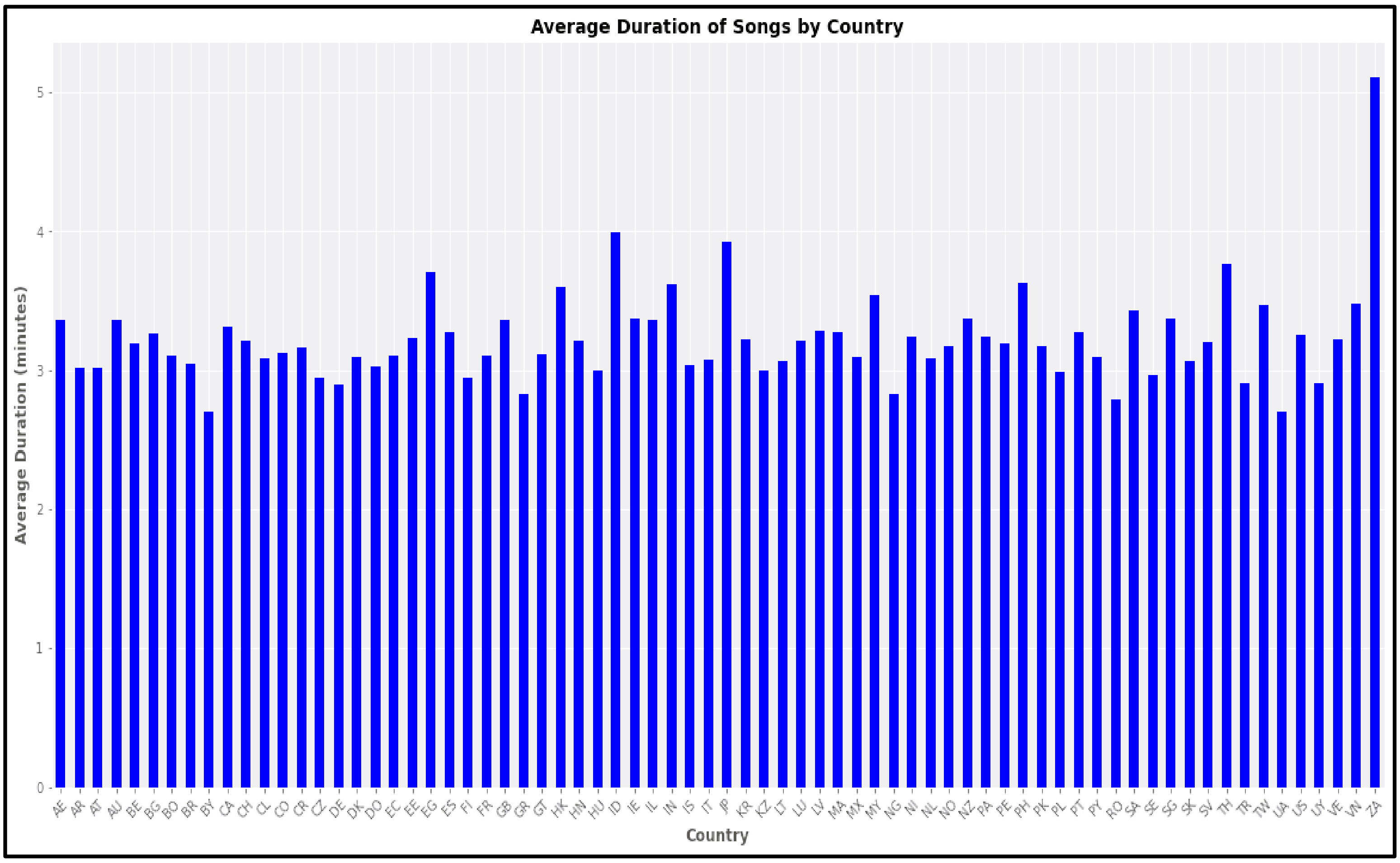
Relationship between danceability and energy

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='danceability', y='energy', data=df, color='b')
plt.xlabel('Danceability', fontweight='bold')
plt.ylabel('Energy', fontweight='bold')
plt.title('Relationship between Danceability and Energy', fontweight='bold')
plt.show()
```



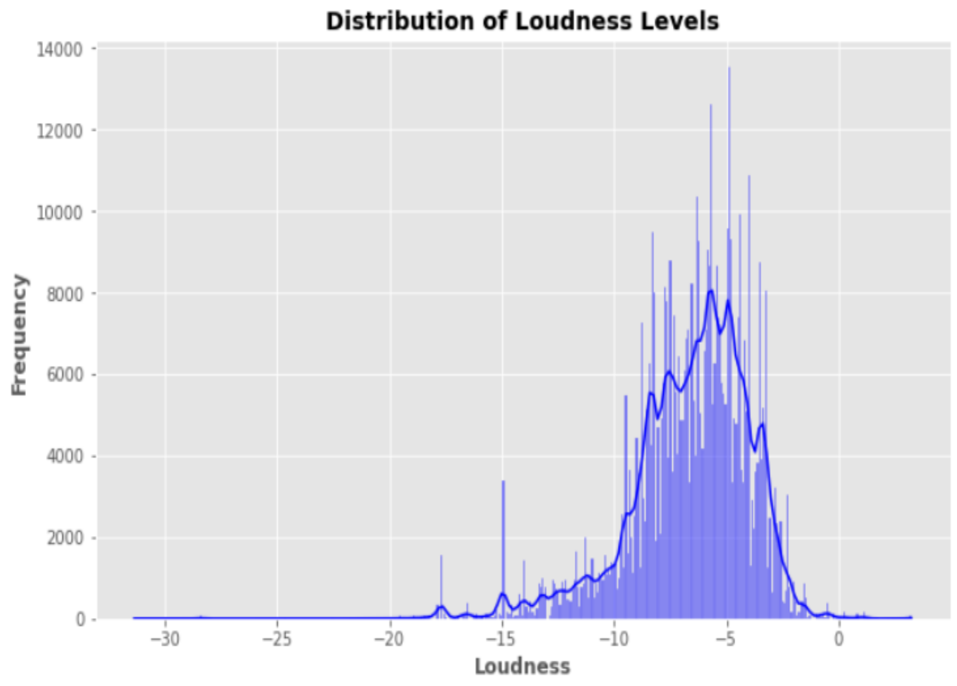
## What is the average duration of songs for each country?

```
avg_duration_by_country = df.groupby('country')['duration_ms'].mean() / 60000
plt.figure(figsize=(20, 10))
avg_duration_by_country.plot(kind='bar', color='blue')
plt.xlabel('Country', fontweight='bold')
plt.ylabel('Average Duration (minutes)', fontweight='bold')
plt.title('Average Duration of Songs by Country', fontweight='bold')
plt.xticks(rotation=45)
plt.show()
```



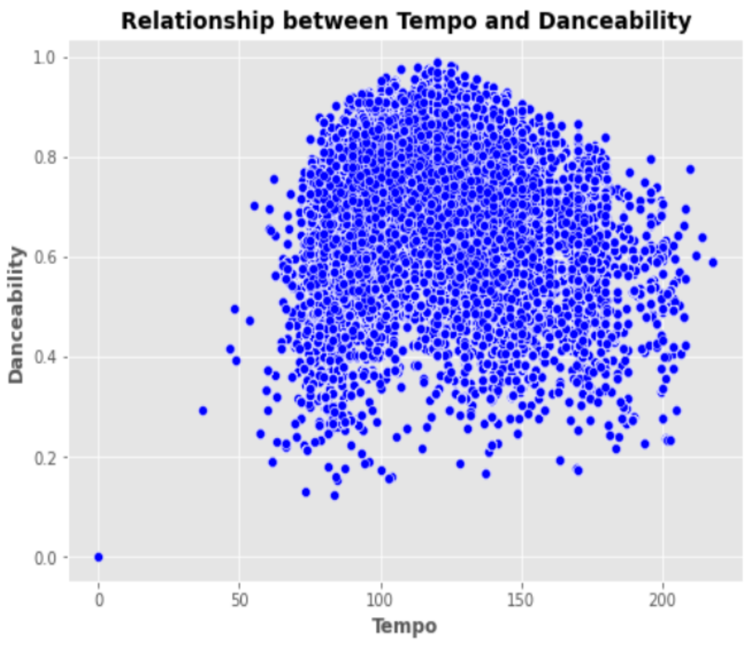
**What is the distribution of loudness levels in the dataset?**

```
plt.figure(figsize=(10, 6))
sns.histplot(df['loudness'], kde=True, color='b')
plt.xlabel('Loudness', fontweight='bold')
plt.ylabel('Frequency', fontweight='bold')
plt.title('Distribution of Loudness Levels', fontweight='bold')
plt.show()
```

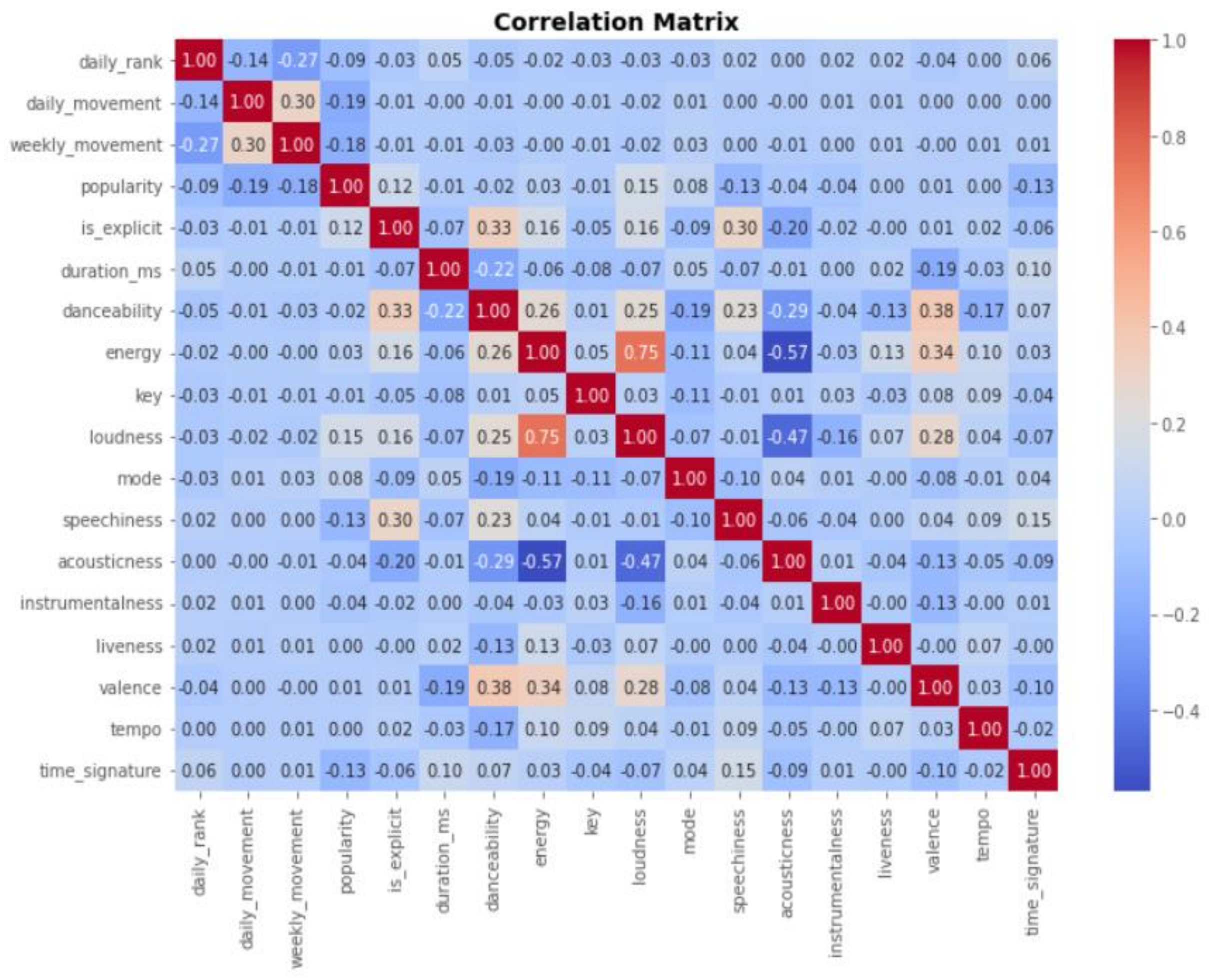


**Do songs with higher tempo tend to have higher danceability?**

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='tempo', y='danceability', data=df, color='b')
plt.xlabel('Tempo', fontweight='bold')
plt.ylabel('Danceability', fontweight='bold')
plt.title('Relationship between Tempo and Danceability', fontweight='bold')
plt.show()
```



```
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix', fontweight='bold')
plt.show()
```



| | daily_rank | daily_movement | weekly_movement | popularity | is_explicit | duration_ms | danceability | energy | key | loudness | mode | speechiness | acousticness | instrumentalness | liveness | valence | tempo | time_signature |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| daily_rank | 1.00 | -0.14 | -0.27 | -0.09 | -0.03 | 0.05 | -0.05 | -0.02 | -0.03 | -0.03 | -0.03 | 0.02 | 0.00 | 0.02 | 0.02 | -0.04 | 0.00 | 0.06 |
| daily_movement | -0.14 | 1.00 | 0.30 | -0.19 | -0.01 | -0.00 | -0.01 | -0.00 | -0.01 | -0.02 | 0.01 | 0.00 | -0.00 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 |
| weekly_movement | -0.27 | 0.30 | 1.00 | -0.18 | -0.01 | -0.01 | -0.03 | -0.00 | -0.01 | -0.02 | 0.03 | 0.00 | -0.01 | 0.00 | 0.01 | -0.00 | 0.01 | 0.01 |
| popularity | -0.09 | -0.19 | -0.18 | 1.00 | 0.12 | -0.01 | -0.02 | 0.03 | -0.01 | 0.15 | 0.08 | -0.13 | -0.04 | -0.04 | 0.00 | 0.01 | 0.00 | -0.13 |
| is_explicit | -0.03 | -0.01 | -0.01 | 0.12 | 1.00 | -0.07 | 0.33 | 0.16 | -0.05 | 0.16 | -0.09 | 0.30 | -0.20 | -0.02 | -0.00 | 0.01 | 0.02 | -0.06 |
| duration_ms | 0.05 | -0.00 | -0.01 | -0.01 | -0.07 | 1.00 | -0.22 | -0.06 | -0.08 | -0.07 | 0.05 | -0.07 | -0.01 | 0.00 | 0.02 | -0.19 | -0.03 | 0.10 |
| danceability | -0.05 | -0.01 | -0.03 | -0.02 | 0.33 | -0.22 | 1.00 | 0.26 | 0.01 | 0.25 | -0.19 | 0.23 | -0.29 | -0.04 | -0.13 | 0.38 | -0.17 | 0.07 |
| energy | -0.02 | -0.00 | -0.00 | 0.03 | 0.16 | -0.06 | 0.26 | 1.00 | 0.05 | 0.75 | -0.11 | 0.04 | -0.57 | -0.03 | 0.13 | 0.34 | 0.10 | 0.03 |
| key | -0.03 | -0.01 | -0.01 | -0.01 | -0.05 | -0.08 | 0.01 | 0.05 | 1.00 | 0.03 | -0.11 | -0.01 | 0.01 | 0.03 | -0.03 | 0.08 | 0.09 | -0.04 |
| loudness | -0.03 | -0.02 | -0.02 | 0.15 | 0.16 | -0.07 | 0.25 | 0.75 | 0.03 | 1.00 | -0.07 | -0.01 | -0.47 | -0.16 | 0.07 | 0.28 | 0.04 | -0.07 |
| mode | -0.03 | 0.01 | 0.03 | 0.08 | -0.09 | 0.05 | -0.19 | -0.11 | -0.11 | -0.07 | 1.00 | -0.10 | 0.04 | 0.01 | -0.00 | -0.08 | -0.01 | 0.04 |
| speechiness | 0.02 | 0.00 | 0.00 | -0.13 | 0.30 | -0.07 | 0.23 | 0.04 | -0.01 | -0.01 | -0.10 | 1.00 | -0.06 | -0.04 | 0.00 | 0.04 | 0.09 | 0.15 |
| acousticness | 0.00 | -0.00 | -0.01 | -0.04 | -0.20 | -0.01 | -0.29 | -0.57 | 0.01 | -0.47 | 0.04 | -0.06 | 1.00 | 0.01 | -0.04 | -0.13 | -0.05 | -0.09 |
| instrumentalness | 0.02 | 0.01 | 0.00 | -0.04 | -0.02 | 0.00 | -0.04 | -0.03 | 0.03 | -0.16 | 0.01 | -0.04 | 0.01 | 1.00 | -0.00 | -0.13 | -0.00 | 0.01 |
| liveness | 0.02 | 0.01 | 0.01 | 0.00 | -0.00 | 0.02 | -0.13 | 0.13 | -0.03 | 0.07 | -0.00 | 0.00 | -0.04 | -0.00 | 1.00 | -0.00 | 0.07 | -0.00 |
| valence | -0.04 | 0.00 | -0.00 | 0.01 | 0.01 | -0.19 | 0.38 | 0.34 | 0.08 | 0.28 | -0.08 | 0.04 | -0.13 | -0.13 | -0.00 | 1.00 | 0.03 | -0.10 |
| tempo | 0.00 | 0.00 | 0.01 | 0.00 | 0.02 | -0.03 | -0.17 | 0.10 | 0.09 | 0.04 | -0.01 | 0.09 | -0.05 | -0.00 | 0.07 | 0.03 | 1.00 | -0.02 |
| time_signature | 0.06 | 0.00 | 0.01 | -0.13 | -0.06 | 0.10 | 0.07 | 0.03 | -0.04 | -0.07 | 0.04 | 0.15 | -0.09 | 0.01 | -0.00 | -0.10 | -0.02 | 1.00 |

# Thank You ☺