# # Top Spotify Songs in 73 Countries - A complete EDA

**Contributors:** Following members have contributed in the development of this notebook.

1. Hassan Absar Qaimkhani (hassanabsarqaimkhani@gmail.com)
2. Muhammad Ahsan Rabbi (ahsanrabbi@gmail.com)
3. Muhammad Sheraz Aamir ()
4. Muhammad Haseeb Abbasi (haseeb.abbasi0075@gmail.com)
5. Abdul Muid (Callmeamk@gmail.com)

## 6. Muhammad Saqib (Mehersaqib931@yahoo.com)

## **Date:** 28-10-2023

## DATA SET:

This data is collected from kaggle.com and can be accessed from here. (**Note:** Since this data is updated on daily basis, it might be possible that data you find through this link is more recent and updated then the one used in this notebook. Therefore, link of the dataset used in this notebook can be accessed through this Google Drive Link.)

## ### Author/Collaborator of Dataset: asaniczka (kaggle account)

### General Information:

This dataset contains the Daily top 50 songs on Spotify for each country. The data is updated daily and includes various features such as song duration, artist details, album information, and song popularity. The dataset is divided into 40172 rows and 25 columns. Some main features of each column are as follows:

1. `spotify_id`: It shows the unique idntifer for the song in the Spotify database.
2. `name`: It shows the title of the song.
3. `artists`: It shows the name(s) of the artist(s) asociated with he song.
4. `daily_rank`: It shows the daily rank of the song amount the top 50 songs for this country.
5. `daily_movement`: It shows the change in rankings compared to the previous day for the same country.
6. `weekly_movement`: It shows the change in rankings compared to the previous week for the same country.

7. `country`: It shows the ISO Code of the country. (If NULL, then the playlist is 'Global'. Since Global doesn't have an ISO code, it is not put here.)
8. `snapshot_date`: It shows the date onwhich the data was colleted from the Spotify API.
9. `popularity`: It is a measure of the song's current popularity on Spotify.
10. `is_explict`: It indicates whether the songcontains explicit lyrics.
11. `duration_ms`: It gives the duration of the song in milliseconds.
12. `album_name`: It gives the title of the album the song belongs to.
13. `album_release_date`: It gives the release date of the album the song belongs to.
14. `danceability`: It is a measure of how suitable the song is for dancing based on various musical elements.
15. `energy`: measure of the intensity and activity level of the song.
16. `key`: It highlights the key of the song.
17. `loudness`: It gives the overall loudness of the song in decibels.
18. `mode`: It indicates whether the song is in a major or minor key.
19. `speechiness`: It is a measure of the presence ofspoken words inthe song.
20. `acoustiness`: It is a measure of the acoustic quality of the song.
21. `instrumentalness`: It is a measure of the likelihood that the song does not contain vocals.
22. `liveness`: It is a measure of the prsence of a live audience in the recording.
23. `valence`: It is a measure of the musical positiveness conveyed by the song.
24. `tempo`: It gives the tempo of the song in beats per minute.
25. `time_signature`: It indicates the estimated overall time signature of the song.

**Provenance**:

Source:
Data was collected via the Spotify API.

COLLECTION METHODOLOGY:
Data is collected daily by querying the Spotify API for the top 50 songs for each country every day.

---

### License:

License information about the dataset can be accessed from ODC Attribution License (ODC-By)

---

# EDA Analysis

Exploratory Data Analysis (EDA) for Top Spotify Songs shows the relationship between different factors that impact the popularity of songs across different countries and continents. It also enlists the relationship of different factors like danceability, energy, loudness, etc with the explicitness of the songs across different countinents.

## Step-1: Importing Important Liabraries

Before starting the EDA analysis, important libraries are imported.

```
# importing all liabraries that we will use in this EDA exercise.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
```

## Step-1: Essential Settings

Here are some important notebook settings that is used to assist at subsequent stages.

```
# Since data can contain numberical values to be formated with thousands separators and decimals, the number formats are defined here
# nf0 is number format with zero decimals and nf2 is number format with two decimals
nf0 = lambda x: f'{x:,.0f}' if isinstance(x, (int, float)) else x
nf2 = lambda x: f'{x:,.2f}' if isinstance(x, (int, float)) else x

# setting options to show maximum of row and columns
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

```
# disabling Warnings
import warnings
warnings.simplefilter(action='ignore')
```

## Step-3a: Importing Dataset from Google Drive (Optional)

```
# Installing Library

# !pip install gdown

import gdown

# Replace the link with your sharing link and specify the destination
path
gdrive_file_url = "https://drive.google.com/uc?
id=1NASMtgbdCspPvjUPWAa-24z0qgQYYT7l"
output_path = "./05_universal_top_spotify_songs.csv"  # You can
specify your desired output path

# Download the file
gdown.download(gdrive_file_url, output_path, quiet=False)
```

```
Downloading...
From: https://drive.google.com/uc?id=1NASMtgbdCspPvjUPWAa-24z0qgQYYT7l
To: c:\Users\Ihsan BT\Downloads\05_universal_top_spotify_songs.csv
100%|██████████| 9.46M/9.46M [00:27<00:00, 340kB/s]

'./05_universal_top_spotify_songs.csv'
```

## Step-3b: Importing Dataset from Local Device

```
# importing dataset into df
df = pd.read_csv('./05_universal_top_spotify_songs.csv')
```

## Step-4: Data Overview

### Rows and Columns of data

```
# no of rows, columns, and cells in the data
print(f"The total rows in the dataset are",len(df))
print(f"The total columns in the dataset are",len(df.columns))
print(f"The size of dataset is",df.size)
```

```
The total rows in the dataset are 40172
The total columns in the dataset are 25
The size of dataset is 1004300
```

## General information about the dataset

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40172 entries, 0 to 40171
Data columns (total 25 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   spotify_id          40172 non-null  object 
 1   name                40171 non-null  object 
 2   artists             40171 non-null  object 
 3   daily_rank          40172 non-null  int64  
 4   daily_movement      40172 non-null  int64  
 5   weekly_movement     40172 non-null  int64  
 6   country             39620 non-null  object 
 7   snapshot_date       40172 non-null  object 
 8   popularity          40172 non-null  int64  
 9   is_explicit         40172 non-null  bool   
 10  duration_ms         40172 non-null  int64  
 11  album_name          40171 non-null  object 
 12  album_release_date  40171 non-null  object 
 13  danceability        40172 non-null  float64
 14  energy              40172 non-null  float64
 15  key                 40172 non-null  int64  
 16  loudness            40172 non-null  float64
 17  mode                40172 non-null  int64  
 18  speechiness         40172 non-null  float64
 19  acousticness        40172 non-null  float64
 20  instrumentalness    40172 non-null  float64
 21  liveness            40172 non-null  float64
 22  valence             40172 non-null  float64
 23  tempo               40172 non-null  float64
 24  time_signature      40172 non-null  int64  
dtypes: bool(1), float64(9), int64(8), object(7)
memory usage: 7.4+ MB
```

## Checking Vital Statistics

```
# checking vital statistics of df
df_a=df.describe()
df_a.applymap(nf2)
```

```
      daily_rank daily_movement weekly_movement popularity duration_ms  
\
count  40,172.00      40,172.00       40,172.00  40,172.00   40,172.00   

mean       25.51           2.41           13.72      78.57  194,697.50   

std        14.44           9.18           16.86      15.26   49,500.08   
```

```
min         1.00         -38.00           -36.00         0.00         0.00

25%        13.00          -1.00             0.00        67.00   162,767.00

50%        25.00           0.00             8.00        83.00   188,108.00

75%        38.00           2.00            27.00        90.00   220,653.00

max        50.00          49.00            49.00       100.00   641,941.00


      danceability     energy        key   loudness       mode
speechiness  \
count    40,172.00  40,172.00  40,172.00  40,172.00  40,172.00
40,172.00
mean          0.69       0.65       5.54      -6.63       0.49
0.11
std           0.14       0.16       3.47       2.65       0.50
0.10
min           0.22       0.02       0.00     -22.50       0.00
0.02
25%           0.60       0.55       2.00      -8.03       0.00
0.04
50%           0.71       0.67       6.00      -6.21       0.00
0.07
75%           0.80       0.75       9.00      -4.91       1.00
0.14
max           0.97       1.00      11.00       1.16       1.00
0.78

      acousticness instrumentalness   liveness    valence
tempo  \
count    40,172.00        40,172.00  40,172.00  40,172.00  40,172.00

mean          0.29             0.02       0.17       0.53     122.12

std           0.25             0.10       0.12       0.23      27.67

min           0.00             0.00       0.02       0.04      47.91

25%           0.09             0.00       0.10       0.36      99.97

50%           0.21             0.00       0.12       0.52     120.03

75%           0.46             0.00       0.21       0.71     140.06

max           0.98             0.97       0.97       0.98     217.97


      time_signature
count      40,172.00
```

```
mean             3.91
std              0.43
min              1.00
25%              4.00
50%              4.00
75%              4.00
max              5.00
```

## Step-5: Data Cleansing

### Checking Null Values

```
# checking columns where Null values exists
null_count=df.isnull().sum()
null_percent=df.isnull().sum()*100/len(df)
df_a=pd.concat([null_count, null_percent.map(nf2)], axis=1)
# naming columns
df_a.columns = ['Null Count', 'Percentage']
df_a=df_a[df_a['Null Count']>0]
print(df_a)
```

```
                    Null Count Percentage
name                         1       0.00
artists                      1       0.00
country                    552       1.37
album_name                   1       0.00
album_release_date           1       0.00
```

### Checking Duplicate Values

```
# checking duplicate rows
df.duplicated().value_counts()
```

```
False    40172
Name: count, dtype: int64
```

### Examining Anomaly

```
# song whose duration_ms ==0
df[df['duration_ms']==0]
```

```
                   spotify_id name artists  daily_rank  daily_movement  \
26982  6yxtsR3nc3aUL1wcbLn8A3  NaN     NaN          30               1


       weekly_movement country snapshot_date  popularity  is_explicit  \
26982               20      NG    2023-10-21           0        False
```

```
       duration_ms album_name album_release_date  danceability  energy
key  \
26982            0        NaN                NaN         0.791   0.515
1

       loudness  mode  speechiness  acousticness  instrumentalness
liveness  \
26982    -8.178     0        0.168         0.554             0.288
0.0821

       valence    tempo  time_signature
26982    0.507  102.932               4
```

### Excluding Anomaly From the Dataframe

```
# modifying the df to exclude song whose duration_ms ==0
df=df[df['duration_ms']!=0]
```

### Dealing with Null Values

```
# replacing missing values in country will GL
df['country'].fillna('GLO', inplace=True)
```

## Step-6: Data wrangling

### Converting ISO Codes into Country Names

```
# inserting new column of countries name
df_a = {
    'AE': 'United Arab Emirates',
    'AR': 'Argentina',
    'AT': 'Austria',
    'AU': 'Australia',
    'BE': 'Belgium',
    'BG': 'Bulgaria',
    'BO': 'Bolivia',
    'BR': 'Brazil',
    'BY': 'Belarus',
    'CA': 'Canada',
    'CH': 'Switzerland',
    'CL': 'Chile',
    'CO': 'Colombia',
    'CR': 'Costa Rica',
    'CZ': 'Czech Republic',
    'DE': 'Germany',
    'DK': 'Denmark',
    'DO': 'Dominican Republic',
    'EC': 'Ecuador',
```

```
    'EE': 'Estonia',
    'EG': 'Egypt',
    'ES': 'Spain',
    'FI': 'Finland',
    'FR': 'France',
    'GB': 'United Kingdom',
    'GR': 'Greece',
    'GT': 'Guatemala',
    'HK': 'Hong Kong',
    'HN': 'Honduras',
    'HU': 'Hungary',
    'ID': 'Indonesia',
    'IE': 'Ireland',
    'IL': 'Israel',
    'IN': 'India',
    'IS': 'Iceland',
    'IT': 'Italy',
    'JP': 'Japan',
    'KR': 'South Korea',
    'KZ': 'Kazakhstan',
    'LT': 'Lithuania',
    'LU': 'Luxembourg',
    'LV': 'Latvia',
    'MA': 'Morocco',
    'MX': 'Mexico',
    'MY': 'Malaysia',
    'NG': 'Nigeria',
    'NI': 'Nicaragua',
    'NL': 'Netherlands',
    'NO': 'Norway',
    'NZ': 'New Zealand',
    'PA': 'Panama',
    'PE': 'Peru',
    'PH': 'Philippines',
    'PK': 'Pakistan',
    'PL': 'Poland',
    'PT': 'Portugal',
    'PY': 'Paraguay',
    'RO': 'Romania',
    'SA': 'Saudi Arabia',
    'SE': 'Sweden',
    'SG': 'Singapore',
    'SK': 'Slovakia',
    'SV': 'El Salvador',
    'TH': 'Thailand',
    'TR': 'Turkey',
    'TW': 'Taiwan',
    'UA': 'Ukraine',
    'US': 'United States',
```

```
    'UY': 'Uruguay',
    'VE': 'Venezuela',
    'VN': 'Vietnam',
    'ZA': 'South Africa',
    'GLO': 'Global'
}

# Create the 'country_name' column by mapping 'country' to ISO codes
df['country_name'] = df['country'].map(df_a)
```

## Converting ISO Codes into Continent Names

```
# Create a dictionary to map countries to continents
df_a = {
    'AE': 'Asia',
    'AR': 'South America',
    'AT': 'Europe',
    'AU': 'Australia',
    'BE': 'Europe',
    'BG': 'Europe',
    'BO': 'South America',
    'BR': 'South America',
    'BY': 'Europe',
    'CA': 'North America',
    'CH': 'Europe',
    'CL': 'South America',
    'CO': 'South America',
    'CR': 'North America',
    'CZ': 'Europe',
    'DE': 'Europe',
    'DK': 'Europe',
    'DO': 'North America',
    'EC': 'South America',
    'EE': 'Europe',
    'EG': 'Africa',
    'ES': 'Europe',
    'FI': 'Europe',
    'FR': 'Europe',
    'GB': 'Europe',
    'GR': 'Europe',
    'GT': 'North America',
    'HK': 'Asia',
    'HN': 'North America',
    'HU': 'Europe',
    'ID': 'Asia',
    'IE': 'Europe',
    'IL': 'Asia',
    'IN': 'Asia',
    'IS': 'Europe',
```

```
    'IT': 'Europe',
    'JP': 'Asia',
    'KR': 'Asia',
    'KZ': 'Asia',
    'LT': 'Europe',
    'LU': 'Europe',
    'LV': 'Europe',
    'MA': 'Africa',
    'MX': 'North America',
    'MY': 'Asia',
    'NG': 'Africa',
    'NI': 'North America',
    'NL': 'Europe',
    'NO': 'Europe',
    'NZ': 'Australia',
    'PA': 'North America',
    'PE': 'South America',
    'PH': 'Asia',
    'PK': 'Asia',
    'PL': 'Europe',
    'PT': 'Europe',
    'PY': 'South America',
    'RO': 'Europe',
    'SA': 'Asia',
    'SE': 'Europe',
    'SG': 'Asia',
    'SK': 'Europe',
    'SV': 'North America',
    'TH': 'Asia',
    'TR': 'Asia',
    'TW': 'Asia',
    'UA': 'Europe',
    'US': 'North America',
    'UY': 'South America',
    'VE': 'South America',
    'VN': 'Asia',
    'ZA': 'Africa',
    'GLO': 'Global'
}

# Create the 'continent' column by mapping 'country' to continents
df['continent'] = df['country'].map(df_a)
```

## Ramdom Sampling

```
df.sample(5)
```

```
                   spotify_id                               name  \
32398  1BxfuPKGuaTgP7aM0Bbdwr                       Cruel Summer
9893   7FKZix4pk2qf4SZVMOYich  Roule un autre - A COLORS SHOW
```

```
25293  0J9g1MMJDhyvOb3NWckHMm                          Chulo pt.2
27843  0h7QMc9ZRzA9QJrbEHytn2                        The Astronaut
33068  6XSqqQIy7Lm7SnwxS4NrGx                          Classy 101

                              artists  daily_rank  daily_movement  \
32398                    Taylor Swift           9               0
9893                          Kerchak          35              -5
25293  Bad Gyal, Tokischa, Young Miko          41              -4
27843                             JIN          46              -1
33068                Feid, Young Miko          29               0

       weekly_movement country snapshot_date  popularity  is_explicit
\
32398               41      CA    2023-10-20          99        False

9893                -3      FR    2023-10-26          74         True

25293                9      BO    2023-10-22          91         True

27843                4      HN    2023-10-21          87        False

33068               21      VE    2023-10-19          93         True


       duration_ms                      album_name album_release_date
\
32398       178426                           Lover         2019-08-23

9893        176086  Roule un autre (A COLORS SHOW)         2023-10-02

25293       219333                      Chulo pt.2         2023-06-22

27843       282463                   The Astronaut         2022-10-28

33068       195986                      Classy 101         2023-03-31


       danceability  energy  key  loudness  mode  speechiness
acousticness  \
32398         0.552   0.702    9    -5.707     1       0.1570
0.11700
9893          0.522   0.744    8    -8.277     1       0.7060
0.02940
25293         0.852   0.881    5    -2.546     0       0.1260
0.06020
27843         0.540   0.761    5    -5.356     1       0.0311
0.00446
33068         0.859   0.658   11    -4.790     1       0.1590
0.14500

       instrumentalness  liveness  valence    tempo  time_signature  \
```

```
32398         0.000021    0.1050    0.564  169.994                4
9893          0.000002    0.1700    0.466   71.470                5
25293         0.000006    0.0682    0.556   96.984                4
27843         0.000003    0.1370    0.215  124.988                4
33068         0.000000    0.1200    0.672  100.065                4

      country_name      continent
32398       Canada  North America
9893        France         Europe
25293      Bolivia  South America
27843     Honduras  North America
33068    Venezuela  South America
```

# Step-7: Exploratory Data Analysis

In this report, the results of our comprehensive Exploratory Data Analysis (EDA) of a music dataset containing information about top Spotify songs from 7 continents is presented. The dataset encompasses a wide range of attributes, including song popularity, explicit content, music features, and more. Through this EDA, we aimed to uncover valuable insights and patterns within the dataset, shedding light on the relationships between different attributes and their variations across continents. Our analysis not only provides a deeper understanding of the dataset but also serves as a foundational step for subsequent data-driven decisions and modeling efforts in the realm of music analytics. Join us on this analytical journey to explore the fascinating world of music data.

---

# Task-1

As part of the exploratory data analysis (EDA), we want to understand the distribution of explicit and non-explicit songs listened to in each continent.

## Question

How does the count of explicit and non-explicit songs vary across different continents?

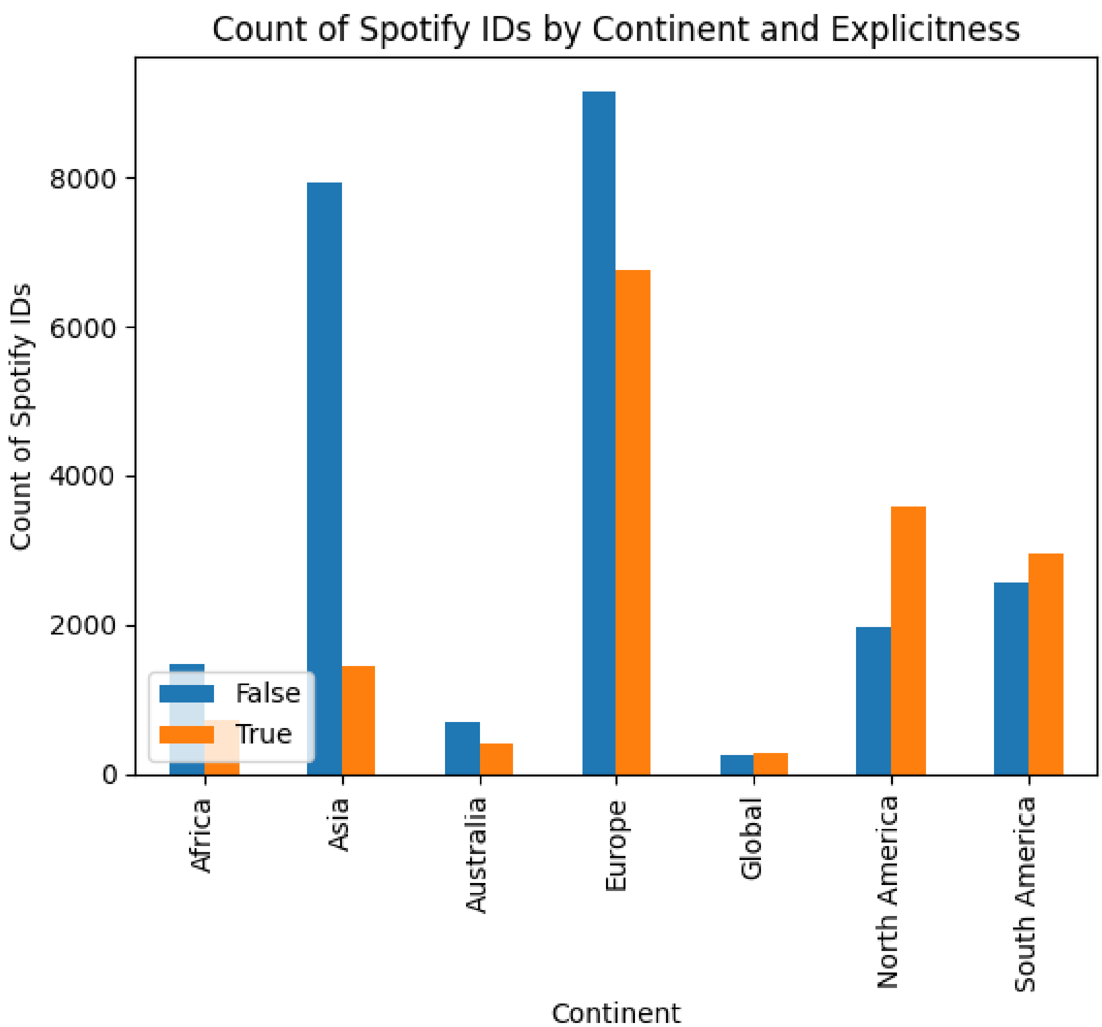
```python
# count of explicit and not explicit songs listened in each continent
df_a=df.groupby(['continent','is_explicit'])
['spotify_id'].count().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

```
is_explicit        False      True
continent
Africa          1,480.00    729.00
Asia            7,920.00  1,457.00
Australia         686.00    417.00
Europe          9,146.00  6,750.00
Global            263.00    289.00
```

```
North America  1,958.00  3,566.00
South America  2,562.00  2,948.00

# bar plot
df_a=df.groupby(['continent','is_explicit'])
['spotify_id'].count().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Count of Spotify IDs')
plt.title('Count of Spotify IDs by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the distribution of explicit and non-explicit songs within each continent. It allows us to see variations in song preferences across different regions.

**Conclusion:** In Europe and North America, there is a significant number of explicit songs, while in Asia and Australia, non-explicit songs are more prevalent.

---

# Task-2

As part of the exploratory data analysis (EDA), we want to understand the mean popularity of explicit and non-explicit songs listened to in each continent.

## Question

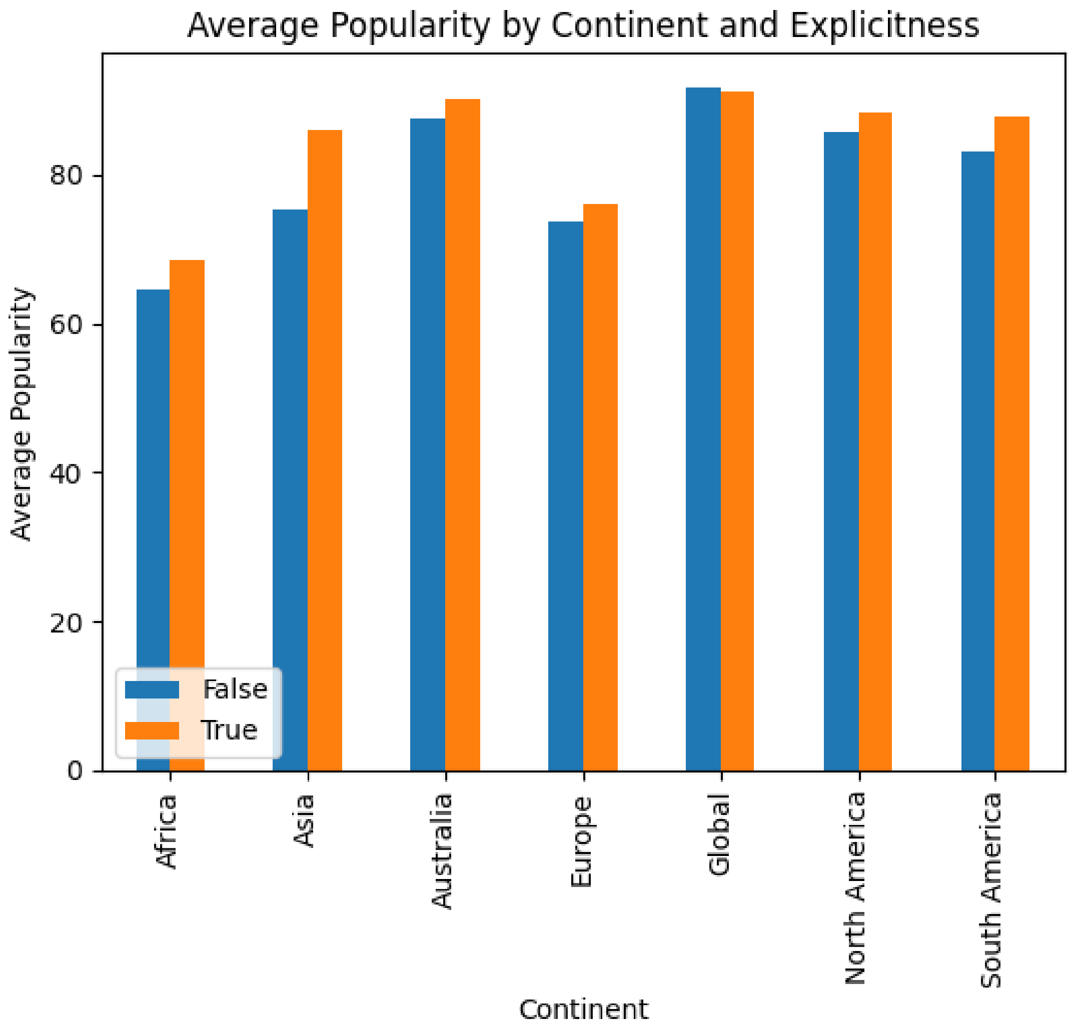
How does the mean popularity differ between explicit and non-explicit songs in different continents?

```
# mean popularity of explicit and not explicit songs listened in each continent
df_a=df.groupby(['continent','is_explicit'])['popularity'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 64.64 | 68.36 |
| Asia | 75.31 | 85.81 |
| Australia | 87.53 | 90.05 |
| Europe | 73.74 | 76.10 |
| Global | 91.74 | 91.25 |
| North America | 85.56 | 88.17 |
| South America | 83.12 | 87.68 |

```
# bar plot
df_a=df.groupby(['continent','is_explicit'])['popularity'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Popularity')
plt.title('Average Popularity by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the mean popularity of explicit and non-explicit songs within each continent. It allows us to see variations in the popularity of songs based on their explicit content across different regions.

**Conclusion**: Explicit songs tend to have higher mean popularity in most continents compared to non-explicit songs.

---

# Task-3

As part of the exploratory data analysis (EDA), we aim to calculate the total duration (in hours) of explicit and non-explicit songs listened to in each continent.

## Question

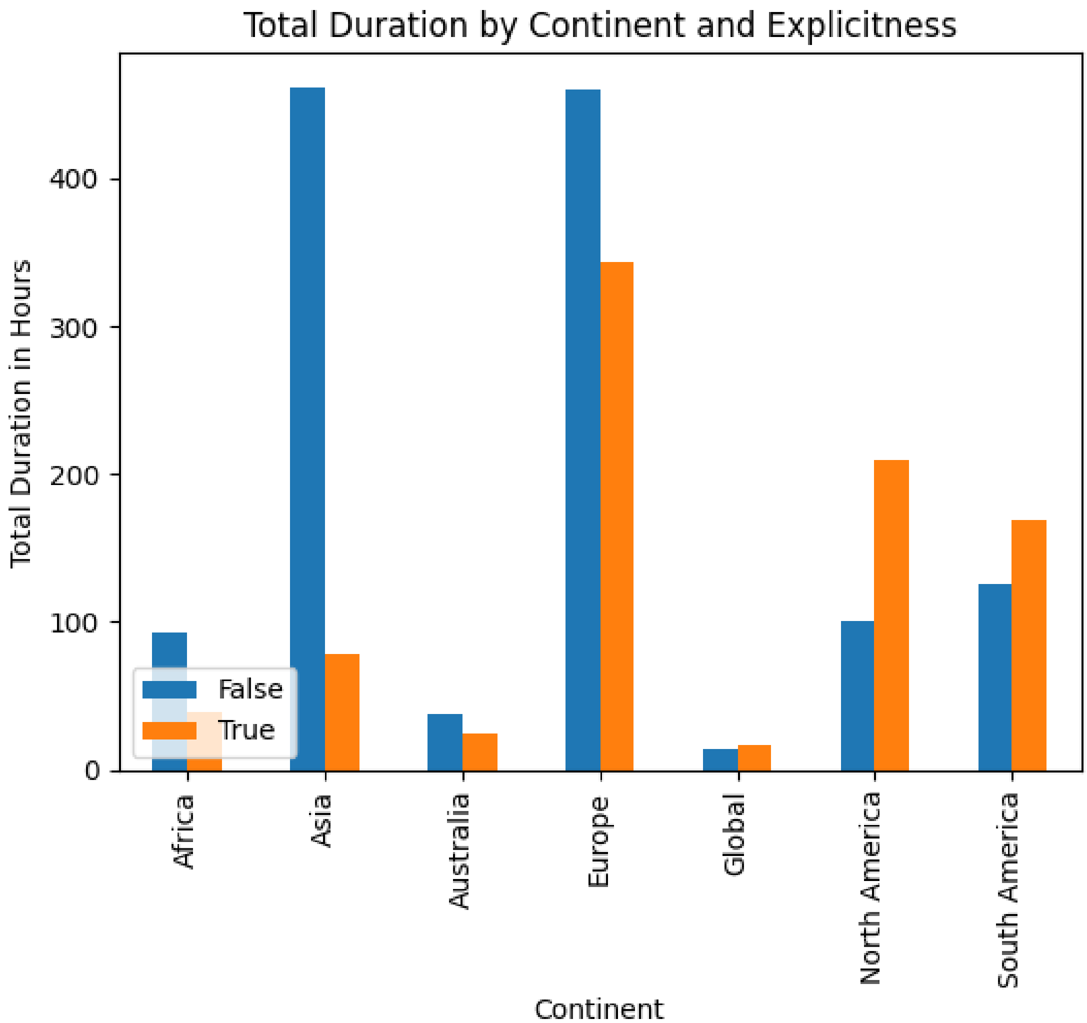
How does the total duration vary between explicit and non-explicit songs in different continents?

```
# total duration (in hours) of explicit and not explicit songs in each
continent
df_a=df.groupby(['continent','is_explicit'])
['duration_ms'].sum().sort_values(ascending=False).unstack()/
1000/60/60
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 92.73 | 38.97 |
| Asia | 461.33 | 78.91 |
| Australia | 38.05 | 24.11 |
| Europe | 460.12 | 342.54 |
| Global | 14.35 | 16.44 |
| North America | 100.32 | 209.80 |
| South America | 125.62 | 169.31 |

```
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['duration_ms'].sum().sort_values(ascending=False).unstack()/
1000/60/60

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Total Duration in Hours')
plt.title('Total Duration by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the total duration of explicit and non-explicit songs within each continent, measured in hours. It allows us to see variations in the listening habits in terms of song duration across different regions.

**Conclusion:** In South and North America, non-explicit songs have a significantly shorter total duration compared to explicit songs. On the other hand trend is opposit for all other continents.

---

# Task-4

As part of the exploratory data analysis (EDA), we want to examine the mean danceability of explicit and non-explicit songs in each continent.

## Question

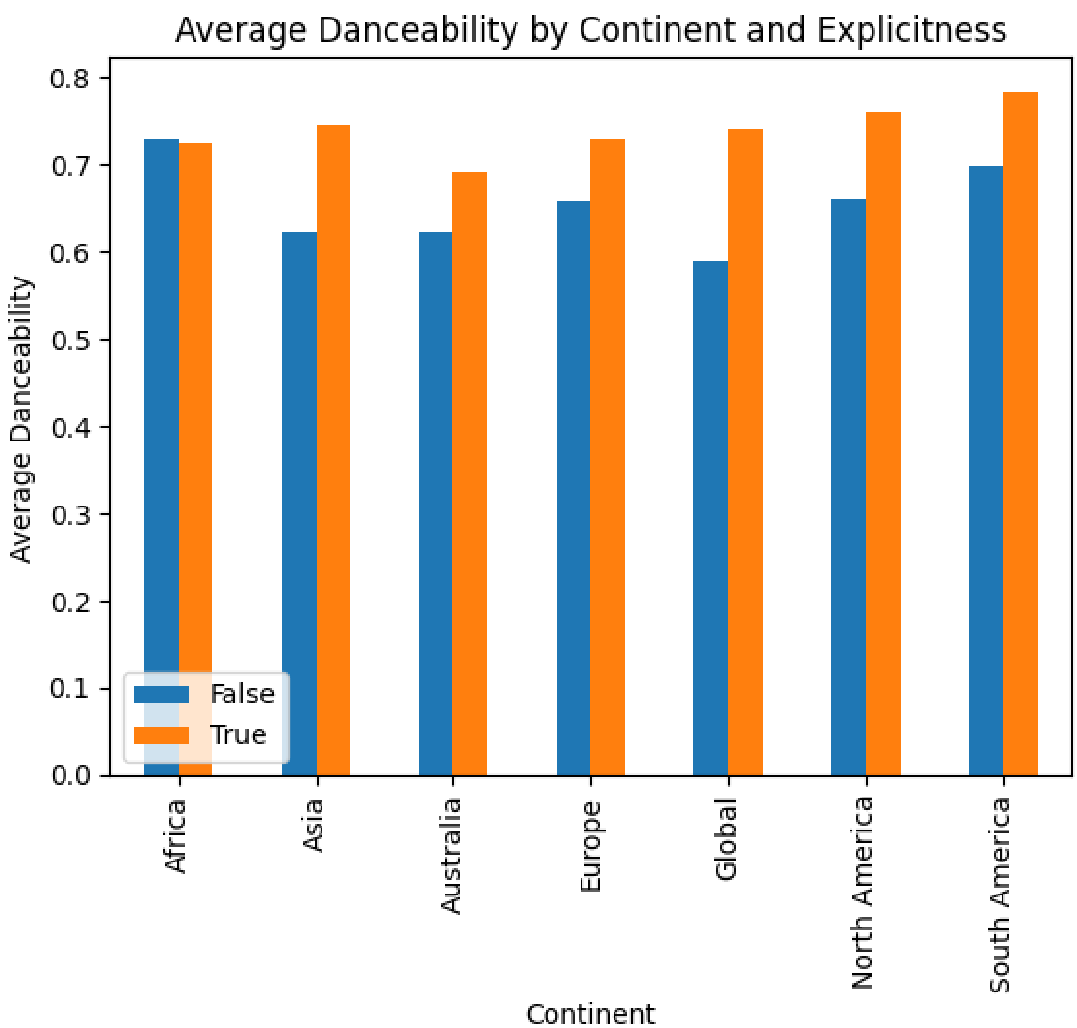
How does the mean danceability vary between explicit and non-explicit songs in different continents?

```
# mean danceability of explicit and not explicit songs in each
continent
df_a=df.groupby(['continent','is_explicit'])
['danceability'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 0.73 | 0.72 |
| Asia | 0.62 | 0.74 |
| Australia | 0.62 | 0.69 |
| Europe | 0.66 | 0.73 |
| Global | 0.59 | 0.74 |
| North America | 0.66 | 0.76 |
| South America | 0.70 | 0.78 |

```
df_a = df.groupby(['continent','is_explicit'])
['danceability'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Danceability')
plt.title('Average Danceability by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the mean danceability of explicit and non-explicit songs within each continent. It allows us to see variations in the danceability of songs based on their explicit content across different regions.

**Conclusion:** Explicit songs tend to have higher mean danceability in all continents but Africa compared to non-explicit songs.

---

# Task-5

As part of the exploratory data analysis (EDA), we aim to analyze the mean energy of explicit and non-explicit songs in each continent.

## Question

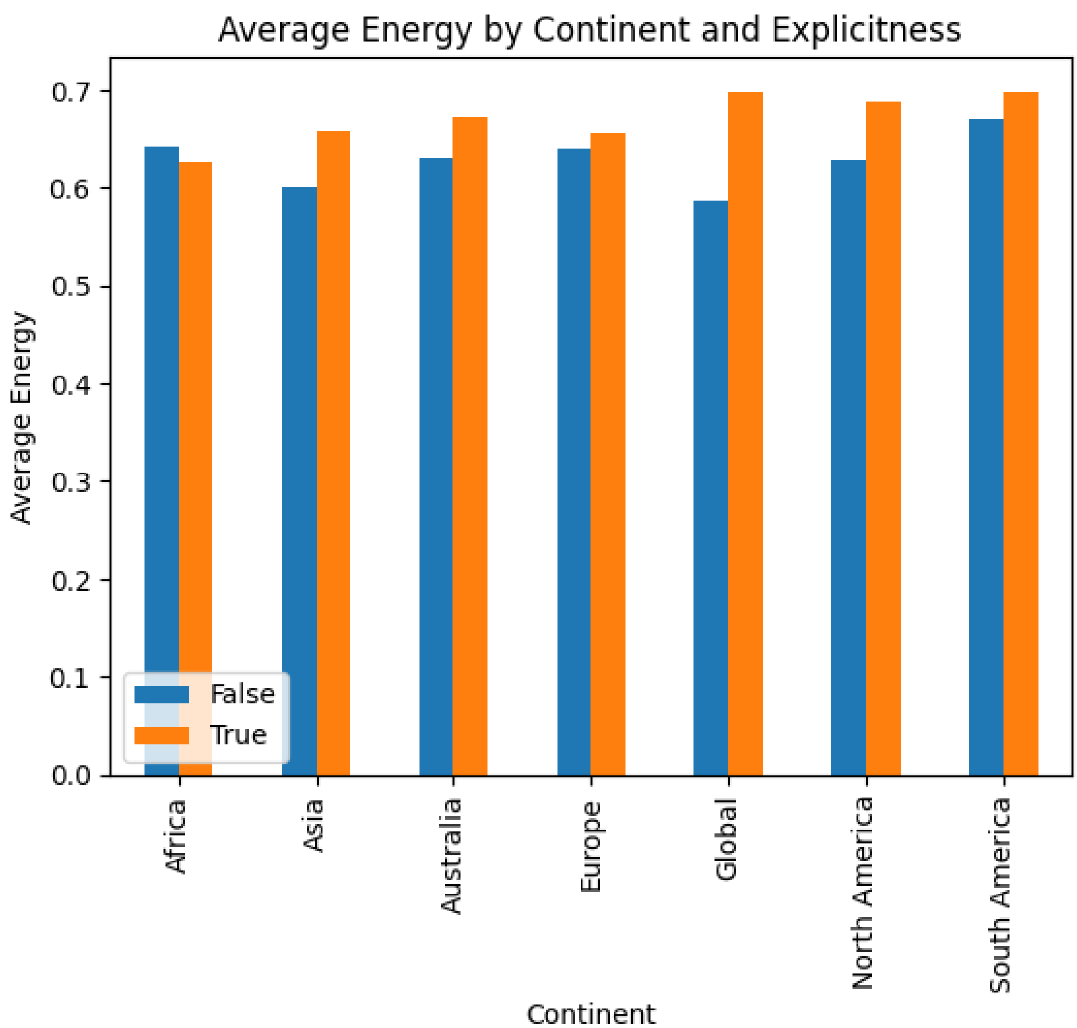
How does the mean energy differ between explicit and non-explicit songs in different continents?

```
# mean energy of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['energy'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 0.64 | 0.63 |
| Asia | 0.60 | 0.66 |
| Australia | 0.63 | 0.67 |
| Europe | 0.64 | 0.66 |
| Global | 0.59 | 0.70 |
| North America | 0.63 | 0.69 |
| South America | 0.67 | 0.70 |

```
# bar plot
df_a=df.groupby(['continent','is_explicit'])
['energy'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Energy')
plt.title('Average Energy by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the mean energy of explicit and non-explicit songs within each continent. It allows us to see variations in the energy levels of songs based on their explicit content across different regions.

**Conclusion**: Explicit songs tend to have higher mean energy in all continents but Africa compared to non-explicit songs.

---

# Task-6

As part of the exploratory data analysis (EDA), we want to explore the mean key of explicit and non-explicit songs in each continent.

## Question

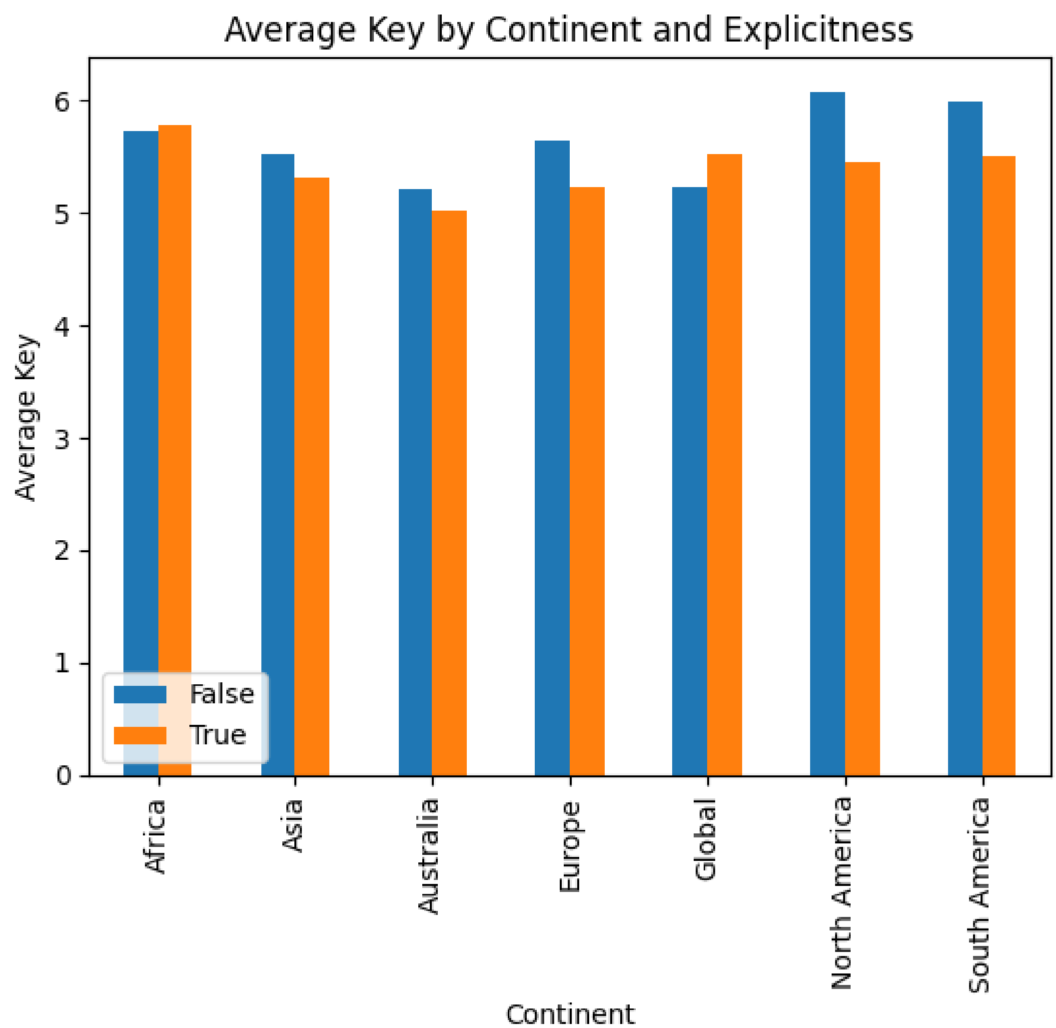
How does the mean key value differ between explicit and non-explicit songs in different continents?

```python
# mean key of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['key'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

```
is_explicit   False True
continent
Africa         5.73  5.77
Asia           5.52  5.31
Australia      5.21  5.01
Europe         5.63  5.22
Global         5.22  5.51
North America  6.07  5.45
South America  5.98  5.50
```

```python
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['key'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Key')
plt.title('Average Key by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the mean key values of explicit and non-explicit songs within each continent. It allows us to see variations in the key signatures of songs based on their explicit content across different regions.

**Conclusion** Explicit songs tend to have relatively consistent mean key values in most continents, with some variation. ___

# Task-7

As part of the exploratory data analysis (EDA), we want to examine the mean loudness of explicit and non-explicit songs in each continent.

## Question

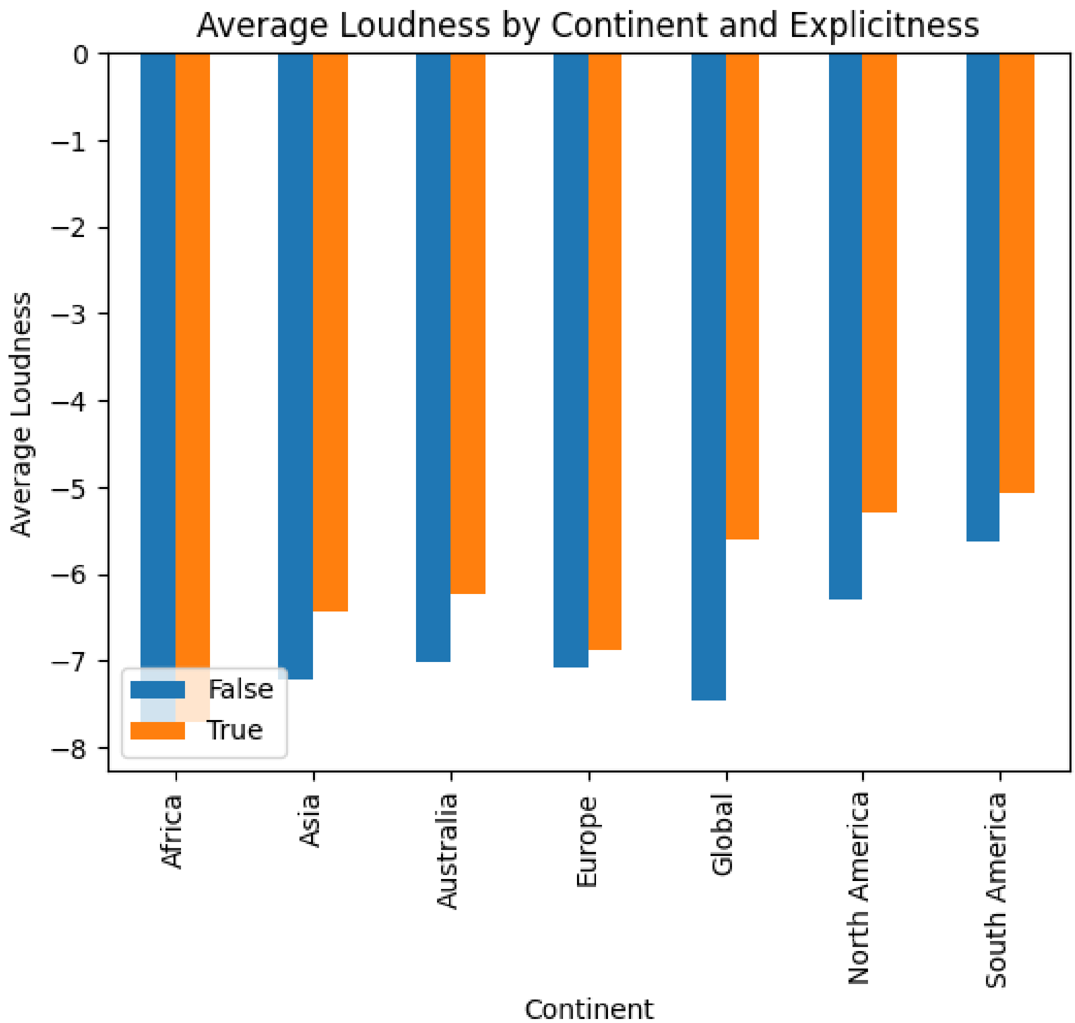
How does the mean loudness differ between explicit and non-explicit songs in different continents?

```python
# mean loudness of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['loudness'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | -7.87 | -7.72 |
| Asia | -7.22 | -6.44 |
| Australia | -7.01 | -6.24 |
| Europe | -7.08 | -6.88 |
| Global | -7.46 | -5.61 |
| North America | -6.30 | -5.29 |
| South America | -5.62 | -5.06 |

```python
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['loudness'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Loudness')
plt.title('Average Loudness by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the mean loudness of explicit and non-explicit songs within each continent. It allows us to see variations in the loudness of songs based on their explicit content across different regions.

**Conclusion**: Explicit songs tend to have higher mean loudness in most continents compared to non-explicit songs, indicating a relatively louder sound profile. ___

# Task-8

As part of the exploratory data analysis (EDA), we want to explore the mean mode of explicit and non-explicit songs in each continent.

## Question

How does the mean mode value differ between explicit and non-explicit songs in different continents?

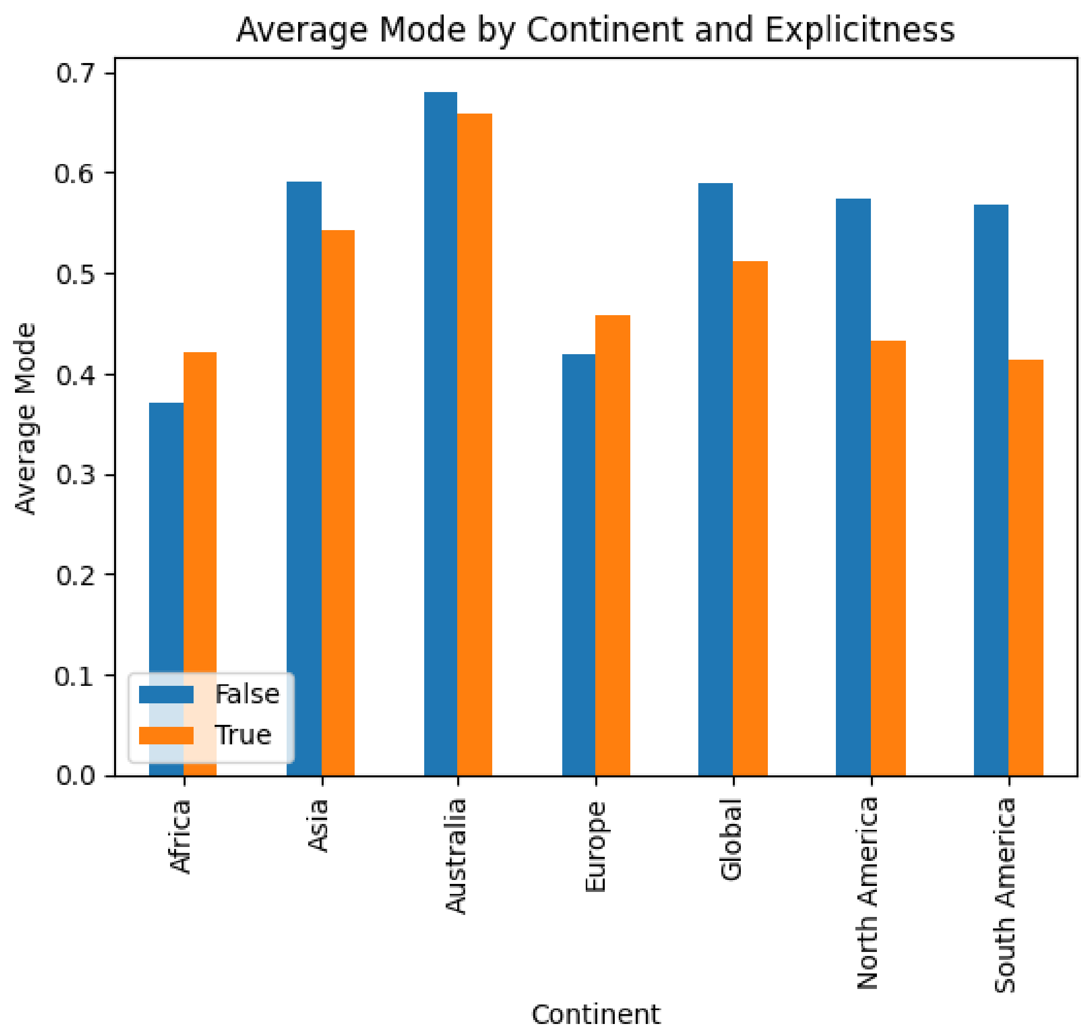
```
# mean mode of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['mode'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 0.37 | 0.42 |
| Asia | 0.59 | 0.54 |
| Australia | 0.68 | 0.66 |
| Europe | 0.42 | 0.46 |
| Global | 0.59 | 0.51 |
| North America | 0.57 | 0.43 |
| South America | 0.57 | 0.41 |

```
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['mode'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Mode')
plt.title('Average Mode by Continent and Explicitness')
plt.legend(loc='lower left')

plt.show()
```




This table provides insights into the mean mode values of explicit and non-explicit songs within each continent. It allows us to see variations in the mode of songs based on their explicit content across different regions.

**Conclusion**: Non explicit songs tend to have somewhat higher mean mode values in some continents, while explicit songs exhibit different patterns in mode values. ___

# Task-9

As part of the exploratory data analysis (EDA), we want to analyze the mean speechiness of explicit and non-explicit songs in each continent.

## Question

How does the mean speechiness differ between explicit and non-explicit songs in different continents?

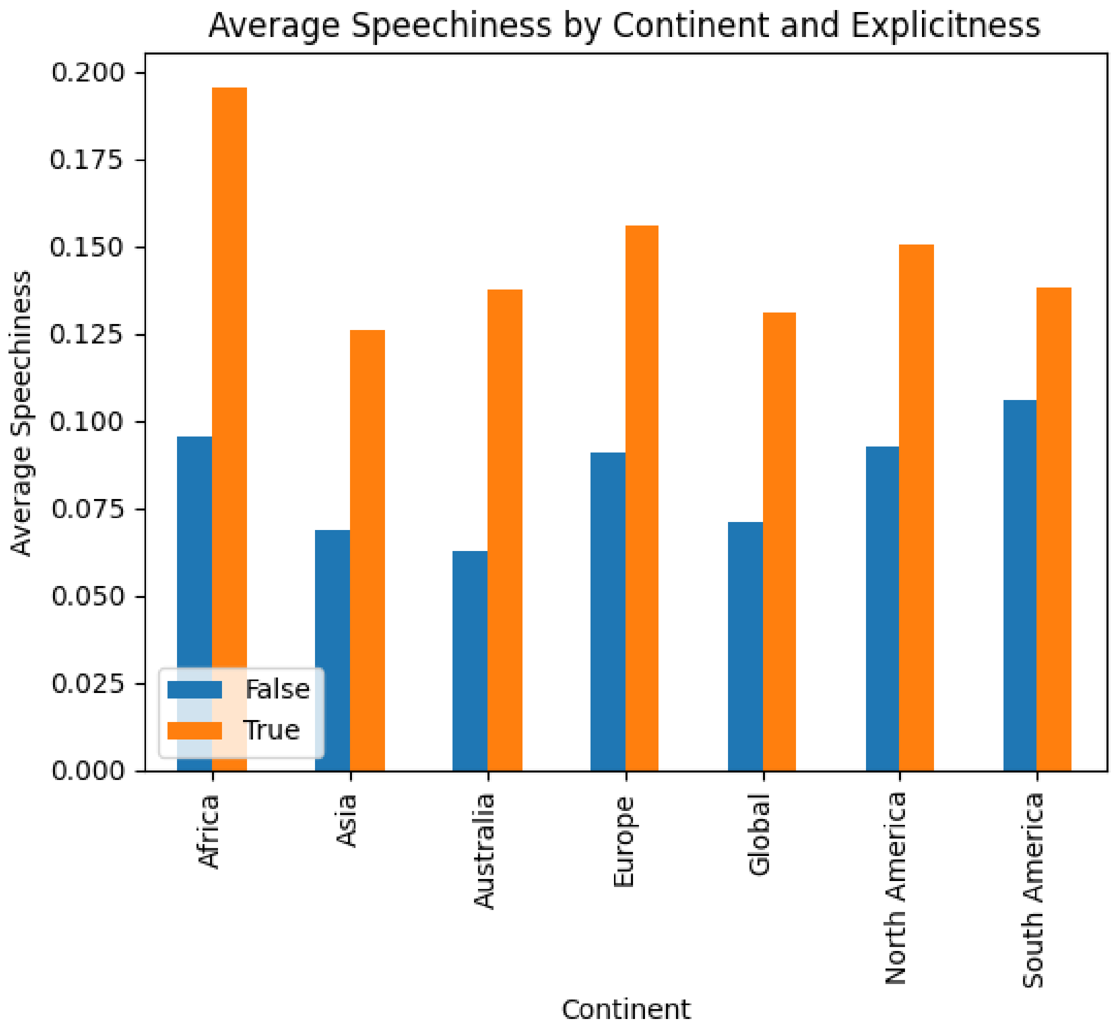
```python
# mean speechiness of explicit and not explicit songs in each
continent
df_a=df.groupby(['continent','is_explicit'])
['speechiness'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 0.10 | 0.20 |
| Asia | 0.07 | 0.13 |
| Australia | 0.06 | 0.14 |
| Europe | 0.09 | 0.16 |
| Global | 0.07 | 0.13 |
| North America | 0.09 | 0.15 |
| South America | 0.11 | 0.14 |

```python
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['speechiness'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Speechiness')
plt.title('Average Speechiness by Continent and Explicitness')
plt.legend(loc='lower left')

plt.show()
```




This table provides insights into the mean speechiness of explicit and non-explicit songs within each continent. It allows us to see variations in the speechiness of songs based on their explicit content across different regions.

**Conclusion**: Explicit songs tend to have higher mean speechiness values in all continents compared to non-explicit songs. ___

# Task-10

As part of the exploratory data analysis (EDA), we want to examine the mean acousticness of explicit and non-explicit songs in each continent.

## Question

How does the mean acousticness differ between explicit and non-explicit songs in different continents?

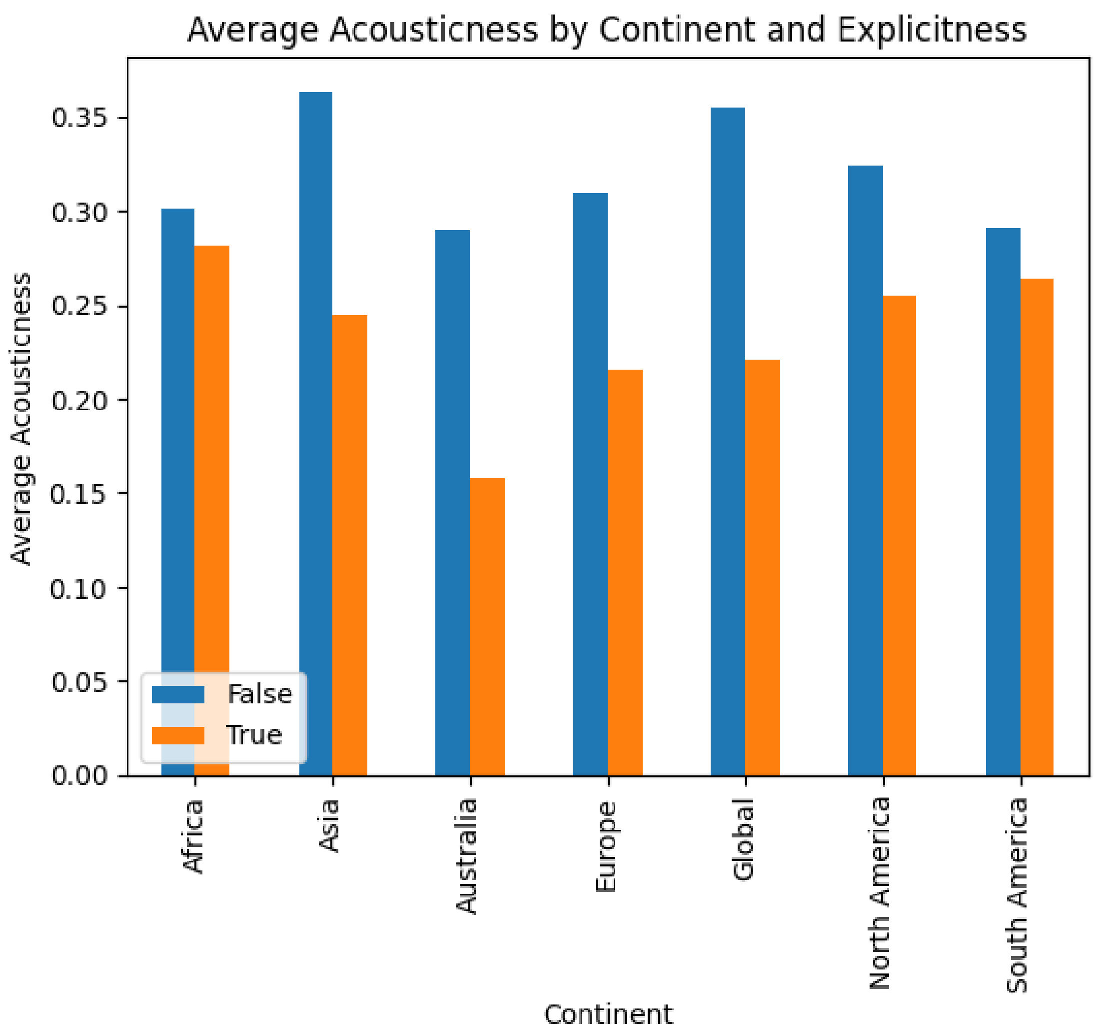
```python
# mean acousticness of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['acousticness'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 0.30 | 0.28 |
| Asia | 0.36 | 0.24 |
| Australia | 0.29 | 0.16 |
| Europe | 0.31 | 0.22 |
| Global | 0.35 | 0.22 |
| North America | 0.32 | 0.25 |
| South America | 0.29 | 0.26 |

```python
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['acousticness'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Acousticness')
plt.title('Average Acousticness by Continent and Explicitness')
plt.legend(loc='lower left')

plt.show()
```




This table provides insights into the mean acousticness of explicit and non-explicit songs within each continent. It allows us to see variations in the acoustic characteristics of songs based on their explicit content across different regions.

**Conclusion**: Explicit songs tend to have low mean acousticness values in all continents, and this may reflect regional preferences in music. ___

# Task-11

As part of the exploratory data analysis (EDA), we want to analyze the mean instrumentalness of explicit and non-explicit songs in each continent.

## Question

How does the mean instrumentalness differ between explicit and non-explicit songs in different continents?

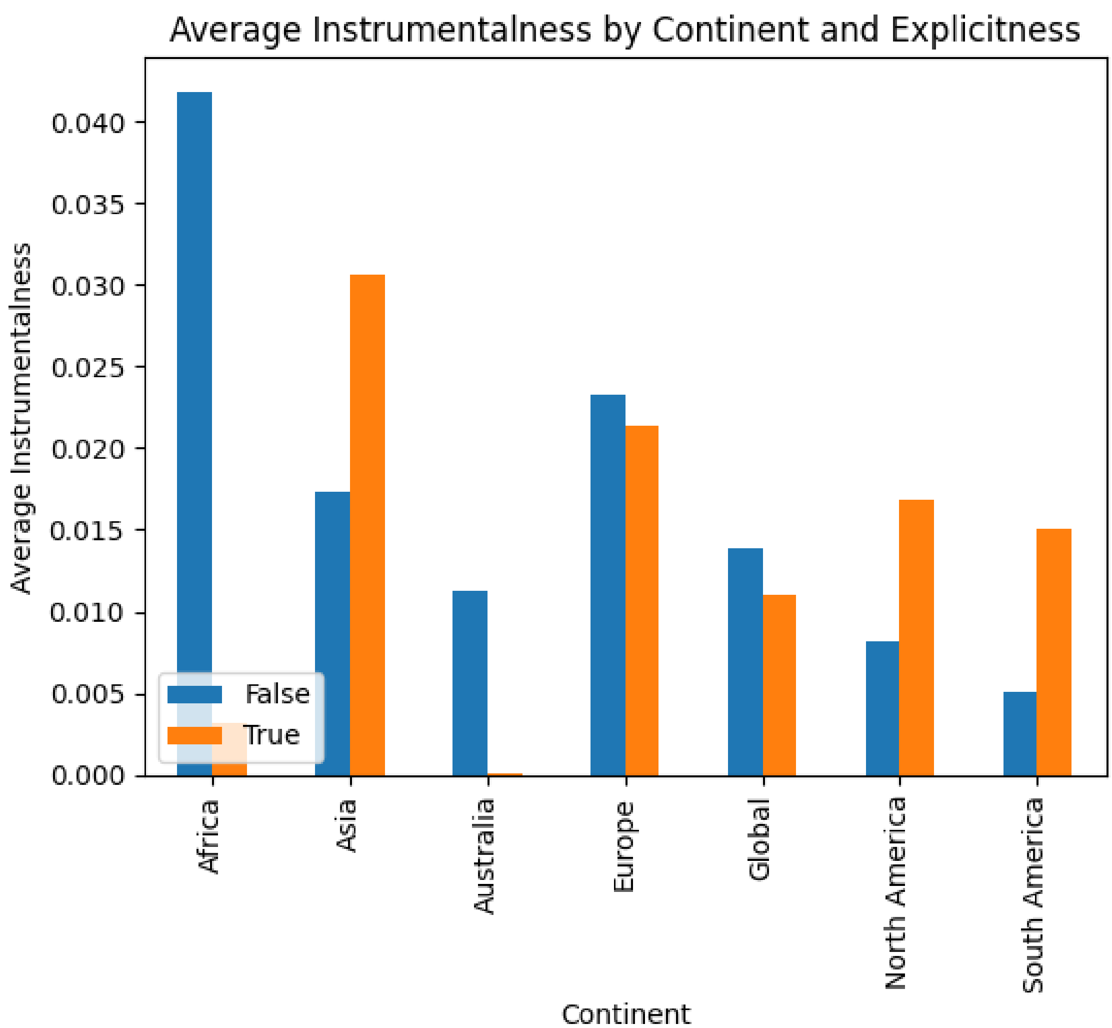
```python
# mean instrumentalness of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['instrumentalness'].mean().sort_values(ascending=False).unstack()
df_a
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 0.041803 | 0.003140 |
| Asia | 0.017279 | 0.030559 |
| Australia | 0.011218 | 0.000073 |
| Europe | 0.023213 | 0.021329 |
| Global | 0.013804 | 0.011069 |
| North America | 0.008152 | 0.016820 |
| South America | 0.005034 | 0.015079 |

```python
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['instrumentalness'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Instrumentalness')
plt.title('Average Instrumentalness by Continent and Explicitness')
plt.legend(loc='lower left')

plt.show()
```




This table provides insights into the mean instrumentalness of explicit and non-explicit songs within each continent. It allows us to see variations in the instrumental characteristics of songs based on their explicit content across different regions.

**Conclusion**: Explicit songs tend to have different mean instrumentalness values in different continents, reflecting variations in musical styles and production techniques.

---

# Task-12

As part of the exploratory data analysis (EDA), we want to explore the mean liveness of explicit and non-explicit songs in each continent.

## Question

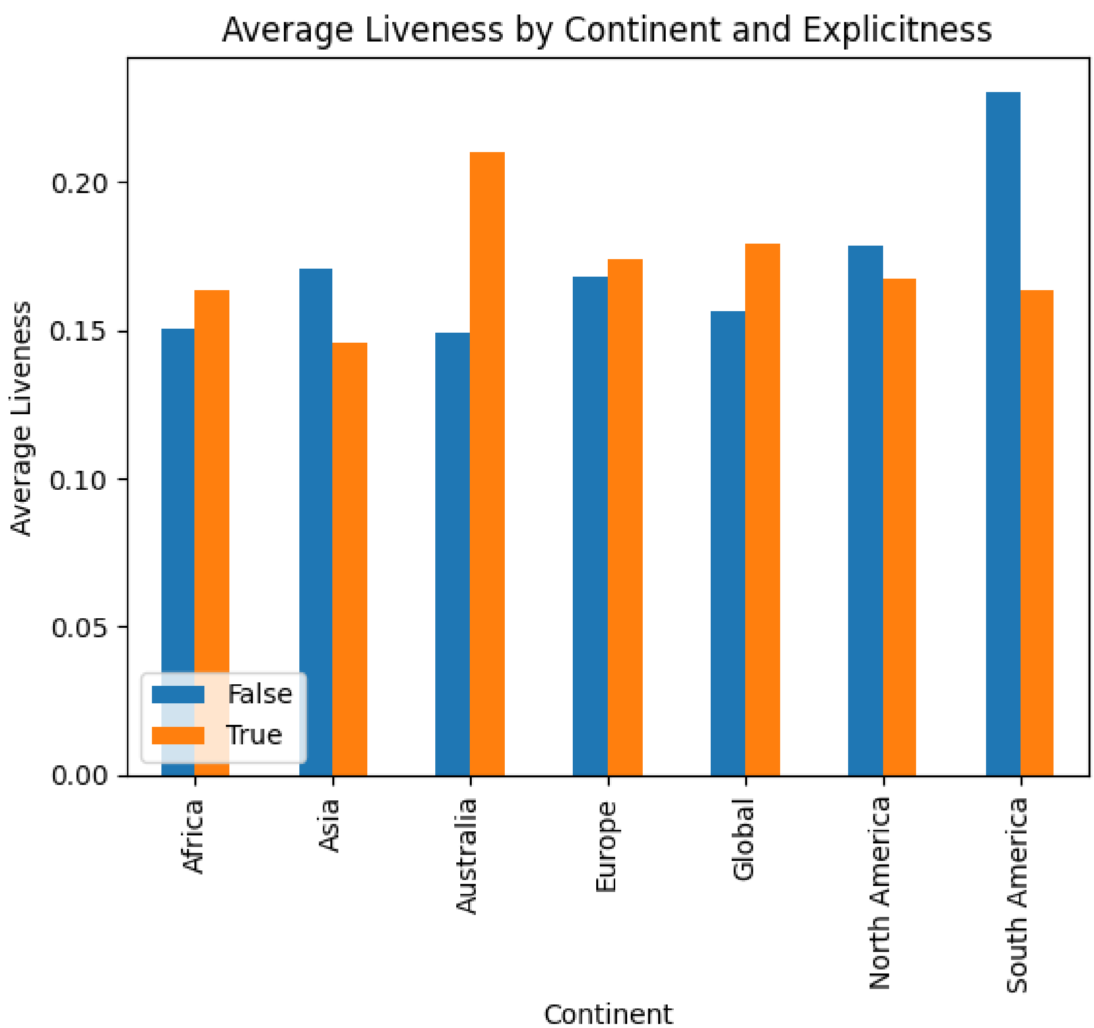
How does the mean liveness differ between explicit and non-explicit songs in different continents?

```python
# mean liveness of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['liveness'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

```
is_explicit    False True
continent
Africa          0.15  0.16
Asia            0.17  0.15
Australia       0.15  0.21
Europe          0.17  0.17
Global          0.16  0.18
North America   0.18  0.17
South America   0.23  0.16
```

```python
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['liveness'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Liveness')
plt.title('Average Liveness by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the mean liveness of explicit and non-explicit songs within each continent. It allows us to see variations in the liveness of songs based on their explicit content across different regions.

**Conclusion**: Explicit songs tend to have different mean liveness values in different continents, reflecting variations in the live or studio nature of the music in these regions. ___

# Task-13

As part of the exploratory data analysis (EDA), we want to analyze the mean valence of explicit and non-explicit songs in each continent.

## Question

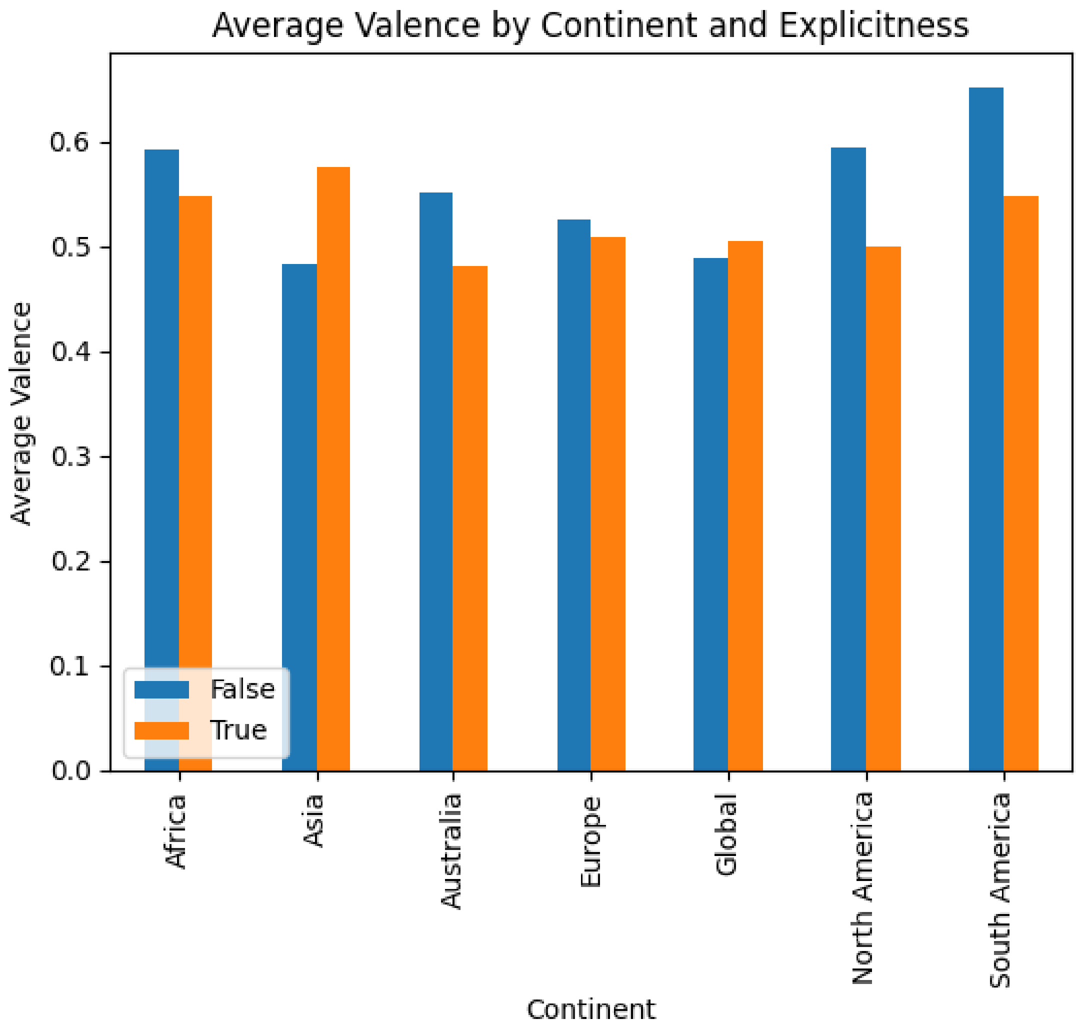
How does the mean valence differ between explicit and non-explicit songs in different continents?

```python
# mean valence of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['valence'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

| is_explicit | False | True |
|---|---|---|
| continent | | |
| Africa | 0.59 | 0.55 |
| Asia | 0.48 | 0.58 |
| Australia | 0.55 | 0.48 |
| Europe | 0.53 | 0.51 |
| Global | 0.49 | 0.50 |
| North America | 0.59 | 0.50 |
| South America | 0.65 | 0.55 |

```python
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['valence'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Valence')
plt.title('Average Valence by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the mean valence of explicit and non-explicit songs within each continent. It allows us to see variations in the emotional tone or positivity of songs based on their explicit content across different regions.

**Conclusion**: Different continents have different mean valence preference for explicit and non-explicit songs. ___

# Task-14

As part of the exploratory data analysis (EDA), we want to explore the mean tempo of explicit and non-explicit songs in each continent.

## Question

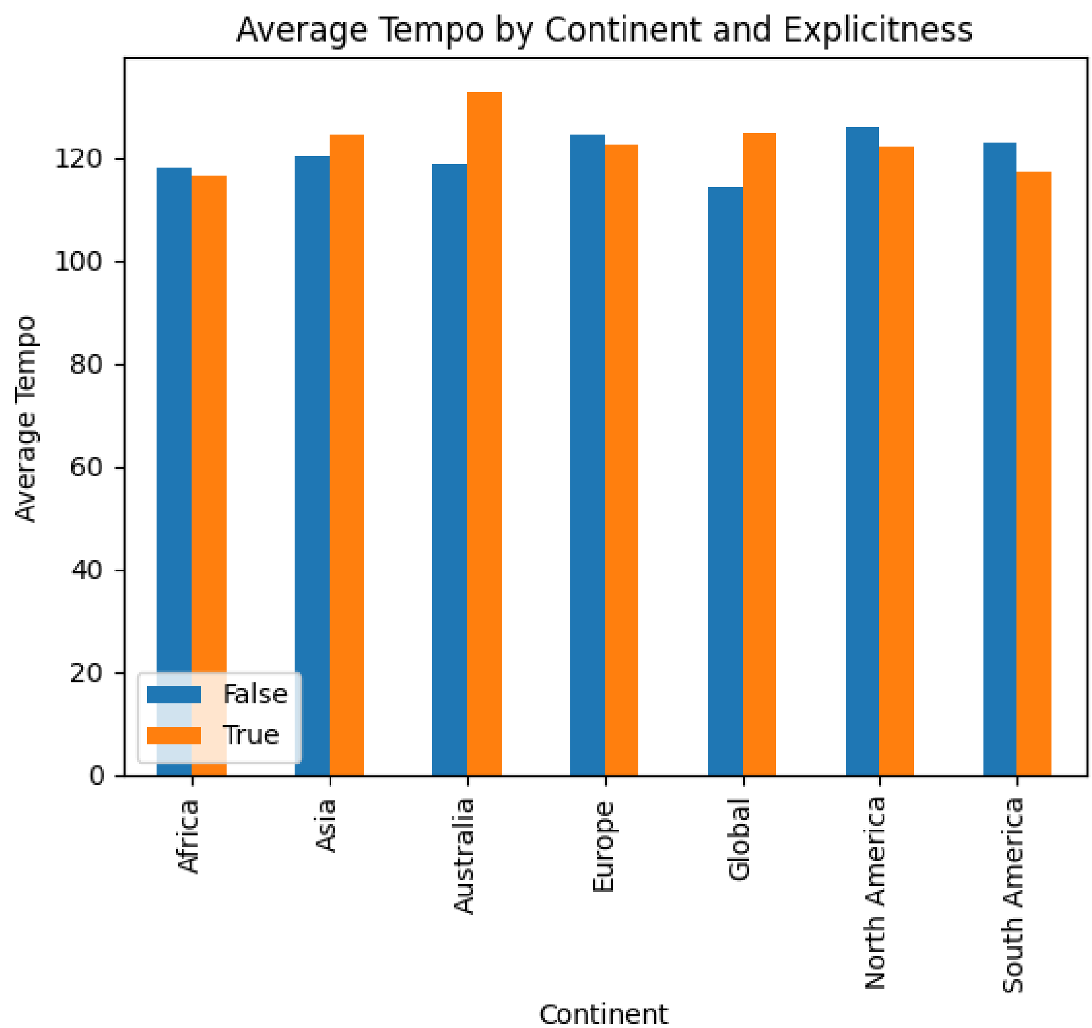
How does the mean tempo differ between explicit and non-explicit songs in different continents?

```python
# mean tempo of explicit and not explicit songs in each continent
df_a=df.groupby(['continent','is_explicit'])
['tempo'].mean().sort_values(ascending=False).unstack()
df_a.map(nf2)
```

```
is_explicit     False   True
continent
Africa         118.18  116.66
Asia           120.46  124.35
Australia      118.63  132.85
Europe         124.49  122.48
Global         114.22  124.99
North America  125.90  122.14
South America  122.87  117.22
```

```python
# bar plot
df_a = df.groupby(['continent','is_explicit'])
['tempo'].mean().sort_values(ascending=False).unstack()

df_a.plot(kind='bar')
plt.xlabel('Continent')
plt.ylabel('Average Tempo')
plt.title('Average Tempo by Continent and Explicitness')
plt.legend(loc='lower left')
plt.show()
```




This table provides insights into the mean tempo of explicit and non-explicit songs within each continent. It allows us to see variations in the tempo or pace of songs based on their explicit content across different regions.

**Conclusion**: Different continents have different mean tempo preference for explicit and non-explicit songs. ___

# Correlation Matrix

```
#redefining the dataframe to explude categorical variables
df_a = df[['popularity', 'is_explicit', 'duration_ms', 'danceability',
'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness',
'instrumentalness', 'liveness', 'valence', 'tempo']]

# correlation of different musical aspects amoung themselves
df_a = df_a.corr()
df_a.map(nf2)
```

| | popularity | is_explicit | duration_ms | danceability | energy |
|---|---|---|---|---|---|
| popularity | 1.00 | 0.19 | 0.04 | -0.03 | 0.01 |
| is_explicit | 0.19 | 1.00 | 0.02 | 0.34 | 0.13 |
| duration_ms | 0.04 | 0.02 | 1.00 | -0.21 | -0.08 |
| danceability | -0.03 | 0.34 | -0.21 | 1.00 | 0.23 |
| energy | 0.01 | 0.13 | -0.08 | 0.23 | 1.00 |
| key | -0.02 | -0.04 | -0.06 | -0.01 | 0.09 |
| loudness | 0.15 | 0.15 | -0.05 | 0.23 | 0.76 |
| mode | 0.07 | -0.05 | 0.07 | -0.16 | -0.05 |
| speechiness | -0.07 | 0.32 | 0.00 | 0.23 | 0.00 |
| acousticness | 0.02 | -0.17 | 0.05 | -0.29 | -0.58 |
| instrumentalness | -0.04 | -0.00 | -0.01 | -0.07 | 0.00 |
| liveness | -0.03 | -0.02 | -0.03 | -0.11 | 0.10 |
| valence | -0.03 | -0.03 | -0.17 | 0.36 | 0.35 |
| tempo | 0.02 | -0.01 | -0.03 | -0.15 | 0.10 |

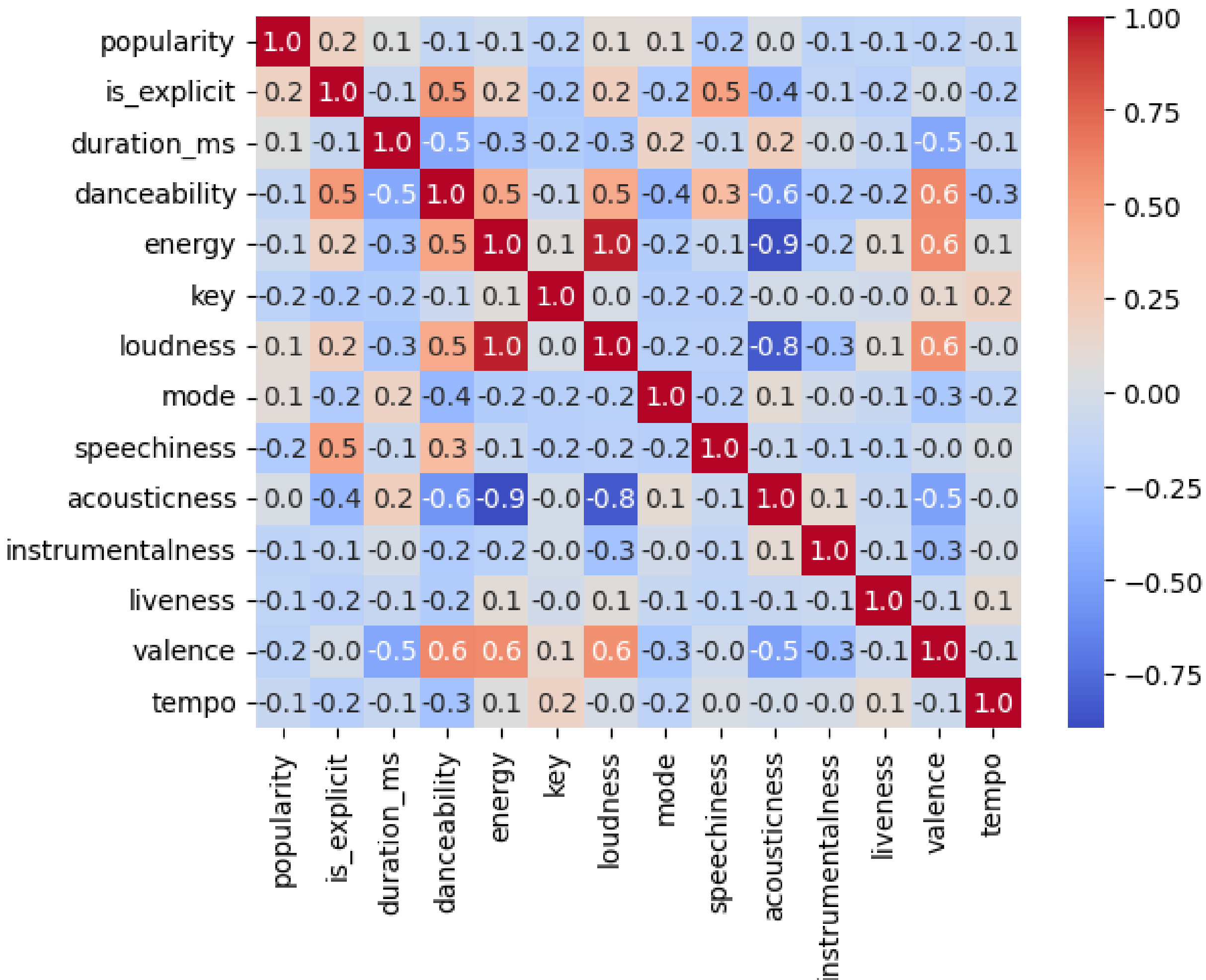
| | key | loudness | mode | speechiness | acousticness |
|---|---|---|---|---|---|
| popularity | -0.02 | 0.15 | 0.07 | -0.07 | 0.02 |
| is_explicit | -0.04 | 0.15 | -0.05 | 0.32 | -0.17 |
| duration_ms | -0.06 | -0.05 | 0.07 | 0.00 | 0.05 |
| danceability | -0.01 | 0.23 | -0.16 | 0.23 | -0.29 |
| energy | 0.09 | 0.76 | -0.05 | 0.00 | -0.58 |
| key | 1.00 | 0.04 | -0.06 | -0.04 | 0.00 |
| loudness | 0.04 | 1.00 | -0.03 | -0.07 | -0.46 |
| mode | -0.06 | -0.03 | 1.00 | -0.04 | -0.01 |
| speechiness | -0.04 | -0.07 | -0.04 | 1.00 | -0.04 |
| acousticness | 0.00 | -0.46 | -0.01 | -0.04 | 1.00 |
| instrumentalness | 0.02 | -0.12 | -0.01 | -0.03 | 0.01 |
| liveness | 0.01 | 0.07 | -0.03 | -0.01 | -0.06 |
| valence | 0.10 | 0.31 | -0.06 | 0.01 | -0.18 |
| tempo | 0.12 | 0.05 | -0.05 | 0.09 | -0.02 |

| | instrumentalness | liveness | valence | tempo |
|---|---|---|---|---|
| popularity | -0.04 | -0.03 | -0.03 | 0.02 |
| is_explicit | -0.00 | -0.02 | -0.03 | -0.01 |

```
duration_ms                           -0.01    -0.03    -0.17   -0.03
danceability                          -0.07    -0.11     0.36   -0.15
energy                                 0.00     0.10     0.35    0.10
key                                    0.02     0.01     0.10    0.12
loudness                              -0.12     0.07     0.31    0.05
mode                                  -0.01    -0.03    -0.06   -0.05
speechiness                           -0.03    -0.01     0.01    0.09
acousticness                           0.01    -0.06    -0.18   -0.02
instrumentalness                       1.00    -0.02    -0.13    0.03
liveness                              -0.02     1.00    -0.01    0.08
valence                               -0.13    -0.01     1.00    0.03
tempo                                  0.03     0.08     0.03    1.00
```

```python
# heatmap of correlation matrix
sns.heatmap(df_a.corr(),annot=True, cmap='coolwarm', fmt=".1f")
```

```
<Axes: >
```



## EDA Conclusion:

The provided correlation matrix describes the relationships between various attributes of the dataset, with a focus on how they correlate with one another. Each cell in the matrix represents the correlation coefficient between two attributes. Here's an interpretation of the correlations:

1. **Popularity**:
   - It has a weak positive correlation with `is_explicit` (0.186), indicating that more popular songs are slightly more likely to be explicit.
   - There is a very weak positive correlation with `loudness` (0.146), suggesting that more popular songs tend to be slightly louder.
   - Popularity has very weak correlations with other attributes.
2. **Is_Explicit**:
   - It has a moderate positive correlation with attributes like `danceability` (0.335), `energy` (0.129), and `speechiness` (0.316), suggesting that explicit songs may be more energetic and have more speech content.
   - It has a moderate negative correlation with `acousticness` (-0.170), indicating that explicit songs tend to have lower acoustic characteristics.
3. **Duration_ms**:
   - It has a weak negative correlation with attributes like `danceability` (-0.209) and `acousticness` (-0.288), suggesting that shorter songs may be less danceable and have lower acoustic characteristics.
   - It has a weak positive correlation with `tempo` (0.048), implying that shorter songs may have a slightly faster tempo.
4. **Danceability**:
   - It has a moderate positive correlation with `is_explicit` (0.335) and `energy` (0.231), indicating that more danceable songs may also be more explicit and energetic.
   - It has a moderate negative correlation with `acousticness` (-0.289), suggesting that less danceable songs tend to have higher acoustic characteristics.
5. **Energy**:
   - It has a strong positive correlation with `loudness` (0.761), indicating that songs with higher energy are typically louder.
   - It has a strong negative correlation with `acousticness` (-0.581), implying that more energetic songs are less acoustic.
6. **Key**, **Loudness**, **Mode**, **Speechiness**, **Acousticness**, **Instrumentalness**, **Liveness**, **Valence**, **Tempo**:
   - These attributes show various weak correlations with each other and with the other attributes. The relationships are not as strong as those mentioned above.

The correlation matrix helps us understand how different attributes relate to each other and can guide feature selection for further analysis or modeling. For example, if you want to predict the popularity of songs, you might consider attributes like `danceability`, `energy`, and `loudness` due to their correlations with popularity.

# Step-7: Visualizations

## Data Insights

Q1. Which are the top 10 artists with most popular songs in the dataset?

Q2. Which are the top 10 countries with most popular songs in the dataset?

Q3. What are the characteristics of music liked by top 5 most listening countries?

Q4. What are the characteristics of music liked by top 5 most listening continents?

Q5. Display the world's map based upon popularity of the songs.

## Q1. Which are the top 10 artists with most popular songs in the dataset?

```
# Group by artists and calculate the average popularity for each
artist
artist_popularity = df.groupby('artists')
['popularity'].mean().reset_index()

# Sort by popularity to find the top artists
top_artists = artist_popularity.nlargest(10, 'popularity')

# Create the horizontal bar chart
fig = go.Figure(go.Bar(
    x=top_artists['popularity'],
    y=top_artists['artists'],
    orientation='h',
    marker=dict(
        color=top_artists['popularity'],
        colorscale=('greens'),
        cmin=0,
        cmax=max(top_artists['popularity']),
        colorbar=dict(
            title='Popularity',
            thickness=15,
            len=0.5,
            y=0.5,
            ypad=0,
            ticks='outside',
            ticklen=5,
            tickwidth=1,
            tickcolor='#000'
        )
    )
))

# Set the layout
```

```python
fig.update_layout(
    title='Top 10 Artists by Most popular Songs',
    xaxis_title='Average Popularity',
    yaxis_title='Artist',
    margin=dict(l=0, r=0, t=50, b=0),
    height=500
)

# Show the plot
fig.show()
```

{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"marker":{"cmax":98.27892561983471,"cmin":0,"color":[98.27892561983471,98,97.81776765375854,97.73033707865169,96.39702760084926,96,95.73423423423424,95.02758620689656,95,95],"colorbar":{"len":0.5,"thickness":15,"tickcolor":"#000","ticklen":5,"ticks":"outside","tickwidth":1,"title":{"text":"Popularity"},"y":0.5,"ypad":0},"colorscale":[[0,"rgb(247,252,245)"],[0.125,"rgb(229,245,224)"],[0.25,"rgb(199,233,192)"],[0.375,"rgb(161,217,155)"],[0.5,"rgb(116,196,118)"],[0.625,"rgb(65,171,93)"],[0.75,"rgb(35,139,69)"],[0.875,"rgb(0,109,44)"],[1,"rgb(0,68,27)"]]},"orientation":"h","type":"bar","x":[98.27892561983471,98,97.81776765375854,97.73033707865169,96.39702760084926,96,95.73423423423424,95.02758620689656,95,95],"y":["Tate McRae","Myke Towers","Kenya Grace","Jung Kook, Latto","iñigo quintero","Billie Eilish","Gunna","Olivia Rodrigo","David Kushner","Dua Lipa"]}],"layout":{"height":500,"margin":{"b":0,"l":0,"r":0,"t":50},"template":{"data":{"bar":[{"error_x":{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpolar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"carpet":[{"aaxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"choropleth":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":

```
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
```

```
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
```

```
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Top 10 Artists by Most popular Songs"},"xaxis":{"title":
{"text":"Average Popularity"}},"yaxis":{"title":{"text":"Artist"}}}}
```


**Insights:** It is evident from this plot that Tate McRae is the most popular artist while other top 10 artists in order of decreasing popularity are as follows:

1. Tate McRae
2. Myke Towers
3. Kenye Grace
4. Jong Kook, Latto
5. Inigo Quintero
6. Billie Eilish
7. Gunna
8. Olivia Rodrigo
9. David Kushner
10. Dua Lipa

The popularity range for these artists is from 95-100%.

## Q2. Which are the top 10 countries with most popular songs in the dataset?

```
# Group by artists and calculate the average popularity for each
artist
country_popularity = df.groupby('country_name')
['popularity'].mean().reset_index()

# Sort by popularity to find the top artists
top_countries = country_popularity.nlargest(10, 'popularity')

# Define the colors for each bar
colors = ['rgb(45, 219, 130)', 'rgb(450, 19, 230)', 'rgb(57, 122,
250)', 'rgb(32, 420, 280)', 'rgb(31, 290, 180)', 'rgb(255, 127, 14)',
'rgb(44, 160, 44)', 'rgb(214, 39, 40)', 'rgb(148, 103, 189)']
# Create the horizontal bar chart
fig = go.Figure(go.Bar(
    x=top_countries['country_name'],
    y=top_countries['popularity'],
        marker=dict(
        color=colors,  # Use the colors list
        cmin=0,
```

```
        cmax=max(top_countries['popularity']),
        colorbar=dict(
            title='Popularity',
            thickness=15,
            len=0.5,
            y=0.5,
            ypad=0,
            ticks='outside',
            ticklen=5,
            tickwidth=1,
            tickcolor='#000'
        )
    )
))

# Set the layout
fig.update_layout(
    title='Top 10 Countries by Most popular Songs',
    xaxis_title='Average Popularity',
    yaxis_title='Artist',
    margin=dict(l=0, r=0, t=50, b=0),
    height=500
)

# Show the plot
fig.show()
```

{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"marker":{"cmax":91.4836956521739,"cmin":0,"color":["rgb(45, 219, 130)","rgb(450, 19, 230)","rgb(57, 122, 250)","rgb(32, 420, 280)","rgb(31, 290, 180)","rgb(255, 127, 14)","rgb(44, 160, 44)","rgb(214, 39, 40)","rgb(148, 103, 189)"],"colorbar":{"len":0.5,"thickness":15,"tickcolor":"#000","ticklen":5,"ticks":"outside","tickwidth":1,"title":{"text":"Popularity"},"y":0.5,"ypad":0}},"type":"bar","x":["Global","United States","Australia","Canada","United Kingdom","Ecuador","Venezuela","United Arab Emirates","El Salvador","Honduras"],"y":[91.4836956521739,89.93490054249548,89.7536231884058,89.59057971014492,88.72909090909091,88.44646098003629,88.16304347826087,88.1627486437613,88.15036231884058,87.9493670886076]}],"layout":{"height":500,"margin":{"b":0,"l":0,"r":0,"t":50},"template":{"data":{"bar":[{"error_x":{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpolar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"carpet":[{"aaxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min

```
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
```

```
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
```

```
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Top 10 Countries by Most popular Songs"},"xaxis":{"title":
{"text":"Average Popularity"}},"yaxis":{"title":{"text":"Artist"}}}}
```



Insights: The plot shows that the United States has the most popular songs in the dataset, followed by the United Kingdom and Canada. The top 10 countries are dominated by English-speaking countries, and most of the songs are in English.

## Q3. What are the characteristics of music liked by top 5 most listening countries?

```
# Assuming you have a DataFrame called 'df' with columns
"country_name", "popularity", "danceability", "energy", "loudness",
"speechiness".

# Group the DataFrame by the unique values in "country_name" and
calculate the mean values for selected columns.
mean = df.groupby("country_name")[['popularity', 'danceability',
'energy', 'loudness', 'speechiness', 'key', 'mode', 'acousticness',
'instrumentalness', 'liveness', 'valence',
'tempo']].mean().reset_index()

# Print or display the resulting DataFrame with mean values
print(mean)
top_countries = mean.nlargest(5, 'popularity')['country_name']
top_countries_df =
mean[mean['country_name'].isin(top_countries)].copy()
```

| | country_name | popularity | danceability | energy | loudness \ |
|---|---|---|---|---|---|
| 0 | Argentina | 84.038182 | 0.750560 | 0.663840 | -5.419995 |
| 1 | Australia | 89.753623 | 0.642707 | 0.644527 | -6.779313 |
| 2 | Austria | 80.889693 | 0.688548 | 0.642875 | -7.077613 |
| 3 | Belarus | 64.961957 | 0.675984 | 0.659953 | -6.591714 |
| 4 | Belgium | 84.838475 | 0.651401 | 0.658619 | -6.496120 |
| 5 | Bolivia | 87.353902 | 0.732494 | 0.690543 | -5.178497 |
| 6 | Brazil | 80.633394 | 0.682508 | 0.746376 | -4.823911 |
| 7 | Bulgaria | 61.230072 | 0.714871 | 0.764817 | -5.387187 |
| 8 | Canada | 89.590580 | 0.644024 | 0.611093 | -6.938591 |
| 9 | Chile | 83.519056 | 0.766191 | 0.682132 | -5.809167 |
| 10 | Colombia | 86.012681 | 0.761315 | 0.689027 | -5.183514 |
| 11 | Costa Rica | 87.280797 | 0.738359 | 0.665614 | -5.352569 |
| 12 | Czech Republic | 69.052727 | 0.709880 | 0.627425 | -8.618613 |
| 13 | Denmark | 68.938294 | 0.713846 | 0.660071 | -6.419773 |
| 14 | Dominican Republic | 84.956600 | 0.765224 | 0.674792 | -5.320910 |
| 15 | Ecuador | 88.446461 | 0.767258 | 0.690372 | -5.288296 |
| 16 | Egypt | 63.235935 | 0.717432 | 0.671102 | -6.478064 |
| 17 | El Salvador | 88.150362 | 0.748853 | 0.677938 | -5.337371 |
| 18 | Estonia | 73.698730 | 0.672488 | 0.685095 | -7.114766 |
| 19 | Finland | 64.509091 | 0.721685 | 0.676447 | -6.207984 |
| 20 | France | 78.455535 | 0.705642 | 0.667379 | -6.621535 |
| 21 | Germany | 76.595281 | 0.693677 | 0.652633 | -7.136501 |
| 22 | Global | 91.483696 | 0.668464 | 0.644721 | -6.491062 |
| 23 | Greece | 66.047016 | 0.726141 | 0.654542 | -7.291385 |
| 24 | Guatemala | 87.342391 | 0.738411 | 0.689286 | -5.359576 |

| | | | | | |
|---|---|---|---|---|---|
| 25 | Honduras | 87.949367 | 0.745651 | 0.682830 | -5.296857 |
| 26 | Hong Kong | 71.650995 | 0.613732 | 0.640957 | -6.472092 |
| 27 | Hungary | 64.757246 | 0.718897 | 0.654739 | -7.137629 |
| 28 | Iceland | 63.483696 | 0.677132 | 0.483122 | -9.011888 |
| 29 | India | 77.094374 | 0.681272 | 0.618672 | -6.977454 |
| 30 | Indonesia | 81.339383 | 0.530093 | 0.529583 | -7.944256 |
| 31 | Ireland | 86.971014 | 0.613255 | 0.630053 | -7.383183 |
| 32 | Israel | 62.748188 | 0.598121 | 0.525976 | -8.249730 |
| 33 | Italy | 74.070780 | 0.705699 | 0.689637 | -6.147060 |
| 34 | Japan | 72.689091 | 0.598035 | 0.752567 | -4.964207 |
| 35 | Kazakhstan | 68.074410 | 0.735083 | 0.596140 | -7.744132 |
| 36 | Latvia | 86.265823 | 0.666174 | 0.638123 | -7.326562 |
| 37 | Lithuania | 78.590580 | 0.711366 | 0.641214 | -7.300116 |
| 38 | Luxembourg | 87.409190 | 0.686031 | 0.660326 | -6.464930 |
| 39 | Malaysia | 84.434545 | 0.594996 | 0.568667 | -7.092013 |
| 40 | Mexico | 85.869565 | 0.730203 | 0.705748 | -5.581386 |
| 41 | Morocco | 68.452899 | 0.700795 | 0.615172 | -7.838766 |
| 42 | Netherlands | 78.212341 | 0.642593 | 0.691882 | -6.499898 |
| 43 | New Zealand | 87.214156 | 0.652730 | 0.649124 | -6.655109 |
| 44 | Nicaragua | 87.918626 | 0.745888 | 0.668069 | -5.228159 |
| 45 | Nigeria | 65.397459 | 0.770142 | 0.673007 | -7.402430 |
| 46 | Norway | 76.032668 | 0.626644 | 0.604392 | -7.407390 |
| 47 | Pakistan | 78.379747 | 0.720861 | 0.580336 | -7.742156 |
| 48 | Panama | 83.436594 | 0.749174 | 0.676554 | -5.312708 |
| 49 | Paraguay | 86.689655 | 0.713583 | 0.670877 | -5.396575 |
| 50 | Peru | 87.513612 | 0.760708 | 0.685688 | -5.236403 |

```
51             Philippines   82.821818        0.639082  0.556717 -7.716687
52                  Poland   73.108893        0.677466  0.667768 -7.005033
53                Portugal   76.967273        0.712220  0.561149 -7.603433
54                 Romania   63.681736        0.748886  0.695036 -6.040056
55            Saudi Arabia   84.608696        0.623344  0.647663 -7.030210
56               Singapore   85.652174        0.647938  0.641643 -6.426252
57                Slovakia   68.388788        0.716398  0.607949 -8.297817
58            South Africa   66.380180        0.720241  0.585158 -9.540454
59             South Korea   78.282098        0.669483  0.651521 -5.772557
60                   Spain   84.449091        0.736278  0.691273 -5.355116
61                  Sweden   72.320000        0.609324  0.538494 -8.390251
62             Switzerland   86.980072        0.670252  0.633120 -6.641701
63                  Taiwan   75.168478        0.615772  0.618428 -6.456897
64                Thailand   72.949275        0.632217  0.620583 -6.854197
65                  Turkey   75.802178        0.642367  0.653388 -7.821390
66                 Ukraine   69.998185        0.715675  0.649753 -7.108726
67    United Arab Emirates   88.162749        0.655933  0.640362 -6.727378
68          United Kingdom   88.729091        0.629484  0.670298 -6.608893
69           United States   89.934901        0.643566  0.618785 -6.759929
70                 Uruguay   83.221818        0.744929  0.663244 -5.420180
71               Venezuela   88.163043        0.759368  0.672801 -5.474721
72                 Vietnam   68.099819        0.709123  0.525260 -8.622105

    speechiness       key      mode  acousticness  instrumentalness
liveness  \
0      0.112780  5.994545  0.567273      0.264675          0.000091
0.179830
1      0.088581  5.112319  0.688406      0.255797          0.007676
0.184503
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 0.125412 | 5.245931 | 0.493671 | 0.251684 | 0.020915 | 0.166087 |
| 3 | 0.094681 | 4.791667 | 0.447464 | 0.188455 | 0.045957 | 0.229327 |
| 4 | 0.084473 | 5.012704 | 0.435572 | 0.323027 | 0.030003 | 0.163293 |
| 5 | 0.108557 | 5.798548 | 0.464610 | 0.280281 | 0.005917 | 0.190372 |
| 6 | 0.123892 | 5.885662 | 0.557169 | 0.364703 | 0.017495 | 0.376592 |
| 7 | 0.148094 | 5.246377 | 0.295290 | 0.152976 | 0.003830 | 0.154231 |
| 8 | 0.093609 | 5.693841 | 0.570652 | 0.276363 | 0.006293 | 0.184710 |
| 9 | 0.113633 | 5.789474 | 0.435572 | 0.240458 | 0.007395 | 0.157226 |
| 10 | 0.131837 | 5.436594 | 0.356884 | 0.278896 | 0.012715 | 0.166514 |
| 11 | 0.143572 | 5.610507 | 0.501812 | 0.299599 | 0.012696 | 0.169855 |
| 12 | 0.162525 | 5.294545 | 0.370909 | 0.232168 | 0.020861 | 0.158402 |
| 13 | 0.150105 | 5.604356 | 0.430127 | 0.203002 | 0.000422 | 0.153242 |
| 14 | 0.173326 | 4.963834 | 0.430380 | 0.300313 | 0.027915 | 0.159952 |
| 15 | 0.135062 | 5.415608 | 0.426497 | 0.276414 | 0.017172 | 0.169202 |
| 16 | 0.090670 | 5.052632 | 0.270417 | 0.370394 | 0.009517 | 0.174325 |
| 17 | 0.139014 | 5.686594 | 0.427536 | 0.267837 | 0.019671 | 0.171283 |
| 18 | 0.097624 | 5.709619 | 0.439201 | 0.186512 | 0.053860 | 0.184698 |
| 19 | 0.096756 | 6.198182 | 0.414545 | 0.191062 | 0.023418 | 0.176322 |
| 20 | 0.144025 | 5.680581 | 0.215971 | 0.293535 | 0.010056 | 0.147542 |
| 21 | 0.150326 | 5.272232 | 0.442831 | 0.235661 | 0.020309 | 0.163719 |
| 22 | 0.102253 | 5.373188 | 0.548913 | 0.284273 | 0.012372 | 0.168224 |
| 23 | 0.146028 | 5.887884 | 0.350814 | 0.292373 | 0.004932 | 0.167705 |
| 24 | 0.111606 | 5.771739 | 0.490942 | 0.273531 | 0.012797 | 0.161584 |
| 25 | 0.136073 | 5.873418 | 0.482821 | 0.275449 | 0.012671 | 0.174972 |
| 26 | 0.069944 | 6.012658 | 0.636528 | 0.345990 | 0.002203 | 0.144410 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 27 | 0.120571 | 5.710145 | 0.317029 | 0.308197 | 0.012393 | 0.152183 |
| 28 | 0.068191 | 5.445652 | 0.579710 | 0.423877 | 0.014467 | 0.166696 |
| 29 | 0.080080 | 5.076225 | 0.564428 | 0.381359 | 0.011279 | 0.159677 |
| 30 | 0.042346 | 5.119782 | 0.856624 | 0.511742 | 0.033169 | 0.152855 |
| 31 | 0.075890 | 5.007246 | 0.559783 | 0.307786 | 0.014553 | 0.171682 |
| 32 | 0.075540 | 4.873188 | 0.340580 | 0.449816 | 0.000983 | 0.218191 |
| 33 | 0.102013 | 5.698730 | 0.553539 | 0.203400 | 0.013671 | 0.166711 |
| 34 | 0.065635 | 5.550909 | 0.780000 | 0.149700 | 0.000250 | 0.197151 |
| 35 | 0.149989 | 5.798548 | 0.431942 | 0.385672 | 0.081542 | 0.159226 |
| 36 | 0.100708 | 5.649186 | 0.546112 | 0.219625 | 0.035950 | 0.195439 |
| 37 | 0.092482 | 5.873188 | 0.396739 | 0.221927 | 0.036770 | 0.164747 |
| 38 | 0.116469 | 4.827133 | 0.341357 | 0.273521 | 0.009824 | 0.167428 |
| 39 | 0.065057 | 5.612727 | 0.727273 | 0.390107 | 0.011391 | 0.153612 |
| 40 | 0.088517 | 6.541667 | 0.490942 | 0.245127 | 0.013051 | 0.161117 |
| 41 | 0.155168 | 7.063406 | 0.260870 | 0.409050 | 0.003863 | 0.137153 |
| 42 | 0.077064 | 5.707804 | 0.424682 | 0.232841 | 0.022203 | 0.159592 |
| 43 | 0.092958 | 5.154265 | 0.656987 | 0.224561 | 0.006332 | 0.159589 |
| 44 | 0.133881 | 5.654611 | 0.473779 | 0.287255 | 0.012673 | 0.173761 |
| 45 | 0.145939 | 5.744102 | 0.515426 | 0.222290 | 0.022100 | 0.160629 |
| 46 | 0.092943 | 5.678766 | 0.444646 | 0.300329 | 0.030578 | 0.210299 |
| 47 | 0.103707 | 4.830018 | 0.490054 | 0.373756 | 0.007508 | 0.141354 |
| 48 | 0.174216 | 5.322464 | 0.398551 | 0.291761 | 0.012703 | 0.161623 |
| 49 | 0.117125 | 5.936479 | 0.519056 | 0.268454 | 0.015278 | 0.186693 |
| 50 | 0.122954 | 5.317604 | 0.477314 | 0.262958 | 0.014980 | 0.176848 |
| 51 | 0.056080 | 5.436364 | 0.649091 | 0.355539 | 0.019322 | 0.141720 |

```
52     0.130265  5.531760  0.462795      0.264853          0.007269
0.177642
53     0.173525  5.249091  0.349091      0.466039          0.020511
0.199193
54     0.127284  5.902351  0.470163      0.250938          0.007389
0.166745
55     0.078403  5.371377  0.509058      0.309789          0.078127
0.147908
56     0.060779  5.894928  0.594203      0.255254          0.010714
0.178950
57     0.186006  5.321881  0.405063      0.244301          0.028717
0.157634
58     0.122047  5.111712  0.500901      0.177635          0.080367
0.147890
59     0.067616  4.949367  0.538879      0.293770          0.018065
0.186282
60     0.140955  5.420000  0.381818      0.305397          0.013169
0.175640
61     0.121009  5.609091  0.585455      0.411284          0.032644
0.137997
62     0.103709  4.599638  0.438406      0.292664          0.018491
0.167701
63     0.058830  5.932971  0.653986      0.329332          0.000162
0.165703
64     0.062829  6.320652  0.817029      0.307127          0.011888
0.173256
65     0.113530  6.159710  0.156080      0.295337          0.003098
0.186367
66     0.105508  6.364791  0.402904      0.287384          0.080638
0.172960
67     0.086820  4.813743  0.569620      0.230162          0.015480
0.162992
68     0.098975  4.621818  0.609091      0.254065          0.014087
0.173780
69     0.103815  5.576854  0.566004      0.275245          0.006998
0.193894
70     0.112245  6.018182  0.630909      0.239900          0.000091
0.174515
71     0.151426  5.612319  0.418478      0.291194          0.012904
0.168775
72     0.082527  5.568058  0.607985      0.501118          0.023712
0.169994

     valence       tempo
0   0.630049  114.330511
1   0.513724  124.252027
2   0.487552  125.304266
3   0.519687  123.223743
4   0.517904  127.270147
5   0.613862  124.273982
```

| | | |
|---|---|---|
| 6 | 0.664929 | 133.083466 |
| 7 | 0.626192 | 133.601031 |
| 8 | 0.463697 | 123.964317 |
| 9 | 0.618109 | 111.819503 |
| 10 | 0.548617 | 118.615060 |
| 11 | 0.525728 | 123.049502 |
| 12 | 0.473656 | 130.572193 |
| 13 | 0.589432 | 117.732657 |
| 14 | 0.497483 | 123.208973 |
| 15 | 0.568599 | 120.686452 |
| 16 | 0.636877 | 120.928363 |
| 17 | 0.527596 | 122.531158 |
| 18 | 0.496728 | 125.473996 |
| 19 | 0.623533 | 117.314653 |
| 20 | 0.576506 | 124.658668 |
| 21 | 0.474889 | 126.256454 |
| 22 | 0.497390 | 119.858261 |
| 23 | 0.515803 | 111.005329 |
| 24 | 0.600417 | 126.020897 |
| 25 | 0.527554 | 121.812376 |
| 26 | 0.496192 | 133.510899 |
| 27 | 0.511261 | 116.250833 |
| 28 | 0.504847 | 120.815091 |
| 29 | 0.525512 | 108.292149 |
| 30 | 0.399359 | 117.709713 |
| 31 | 0.480674 | 125.182478 |
| 32 | 0.439874 | 121.341094 |
| 33 | 0.526613 | 115.766156 |
| 34 | 0.613538 | 129.734207 |
| 35 | 0.515308 | 122.087506 |
| 36 | 0.501153 | 120.582617 |
| 37 | 0.544471 | 118.904179 |
| 38 | 0.530488 | 124.426867 |
| 39 | 0.440220 | 121.055073 |
| 40 | 0.676243 | 127.827172 |
| 41 | 0.569497 | 117.052837 |
| 42 | 0.508721 | 129.411508 |
| 43 | 0.536630 | 123.754407 |
| 44 | 0.530492 | 120.447250 |
| 45 | 0.627515 | 117.233873 |
| 46 | 0.486711 | 125.313218 |
| 47 | 0.552438 | 109.346618 |
| 48 | 0.521839 | 121.539556 |
| 49 | 0.580877 | 122.429788 |
| 50 | 0.572915 | 118.655768 |
| 51 | 0.451294 | 119.083562 |
| 52 | 0.521348 | 123.039327 |
| 53 | 0.509680 | 126.229789 |
| 54 | 0.556067 | 132.350174 |
| 55 | 0.466164 | 119.323453 |

```
56  0.515109  119.959467
57  0.455011  130.933779
58  0.478274  115.523804
59  0.513472  116.524347
60  0.544979  126.352211
61  0.473784  123.775625
62  0.487170  119.546467
63  0.481307  124.045487
64  0.515326  132.772263
65  0.527534  124.689730
66  0.510606  121.370564
67  0.497964  116.938821
68  0.486353  122.871318
69  0.456690  124.289382
70  0.639562  112.891649
71  0.524553  121.672328
72  0.495338  121.777684

# Create the initial sunburst plot
fig = go.Figure(go.Sunburst(
    labels=top_countries_df["country_name"],
    parents=[''] * len(top_countries_df['country_name']),
    values=top_countries_df['popularity'],
    hovertemplate='<b>%{label}</b><br>%{value:.2f} (%
{percentParent:.2%})<extra></extra>'
))
# Define available attributes
attributes = ['danceability', 'energy', 'loudness', 'speechiness',
'key', 'mode', 'acousticness', 'instrumentalness', 'liveness',
'valence', 'tempo']
# Create the dropdown menu
buttons = []
for attribute in attributes:
    buttons.append(dict(
        method='restyle',
        args=[{'values': [top_countries_df[attribute]]}],
        label=attribute.capitalize()
    ))
fig.update_layout(
    updatemenus=[
        dict(
            buttons=buttons,
            direction='down',
            active=0,
            x=1.0,
            y=1.0
        ),
    ],
    title='Characteristics of Music Liked by Top 5 Most listening
countries',
```

```
    margin=dict(t=50, l=0, r=5, b=0)
)

fig.show()
```


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"hovertemplate":"<b>%{label}</b><br>%{value:.2f} (%
{percentParent:.2%})<extra></extra>","labels":
["Australia","Canada","Global","United Kingdom","United
States"],"parents":["","","","",""],"type":"sunburst","values":
[89.7536231884058,89.59057971014492,91.4836956521739,88.72909090909091
,89.93490054249548]}],"layout":{"margin":
{"b":0,"l":0,"r":5,"t":50},"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
```

```
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
```

```
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Characteristics of Music Liked by Top 5 Most listening
countries"},"updatemenus":[{"active":0,"buttons":[{"args":[{"values":
[[0.6427065217391305,0.6440235507246377,0.6684637681159421,0.629483636
3636364,0.6435660036166365]]}],"label":"Danceability","method":"restyl
e"},{"args":[{"values":
[[0.6445273550724638,0.6110925724637681,0.6447211956521739,0.670298363
6363636,0.6187849909584087]]}],"label":"Energy","method":"restyle"},
{"args":[{"values":[[-6.779313405797102,-6.938590579710144,-
6.491061594202899,-6.608892727272727,-
```

```
6.759929475587703]]}],"label":"Loudness","method":"restyle"},{"args":
[{"values":[[8.8580615942029e-2,9.360942028985508e-
2,0.10225326086956522,9.897454545454545e-
2,0.10381464737793852]]}],"label":"Speechiness","method":"restyle"},
{"args":[{"values":
[[5.11231884057971,5.693840579710145,5.3731884057971016,4.621818181818
182,5.5768535262206145]]}],"label":"Key","method":"restyle"},{"args":
[{"values":
[[0.6884057971014492,0.5706521739130435,0.5489130434782609,0.609090909
0909091,0.566003616636528]]}],"label":"Mode","method":"restyle"},
{"args":[{"values":
[[0.2557971195652174,0.2763627166666667,0.28427342391304344,0.25406487
74545454,0.27524492620253166]]}],"label":"Acousticness","method":"rest
yle"},{"args":[{"values":[[7.676356376811594e-3,6.293274094202899e-
3,1.2371834402173914e-2,1.4086700654545455e-2,6.998422549728752e-
3]]}],"label":"Instrumentalness","method":"restyle"},{"args":
[{"values":
[[0.1845032608695652,0.18470978260869567,0.16822373188405795,0.1737798
181818182,0.19389421338155516]]}],"label":"Liveness","method":"restyle
"},{"args":[{"values":
[[0.5137244565217391,0.4636972826086957,0.49738985507246375,0.48635272
72727273,0.45669023508137435]]}],"label":"Valence","method":"restyle"}
,{"args":[{"values":
[[124.25202717391305,123.9643170289855,119.85826086956521,122.87131818
18182,124.28938155515372]]}],"label":"Tempo","method":"restyle"}],"dir
ection":"down","x":1,"y":1}]}}
```


Insights: The plot shows that the top 5 most listening countries have different preferences in terms of music characteristics. For example, the United States prefers songs with high energy, danceability, and loudness, while the United Kingdom prefers songs with high acousticness and instrumentalness.

## Q4. What are the characteristics of music liked by top 5 most listening continents?

```
# Group the DataFrame by the unique values in "continent" and
calculate the mean values for selected columns.
mean = df.groupby("continent")[['popularity', 'danceability',
'energy', 'loudness', 'speechiness', 'key', 'mode', 'acousticness',
'instrumentalness', 'liveness', 'valence',
'tempo']].mean().reset_index()

# Print or display the resulting DataFrame with mean values
print('The main characteristics of music liked by each countinent is
given in the table below')
print(mean)
top_continent = mean.nlargest(5, 'popularity')['continent']
top_continent_df = mean[mean['continent'].isin(top_continent)].copy()
```

```
The main characteristics of music liked by each countinent is given in
the table below
        continent  popularity  danceability    energy  loudness
speechiness  \
0          Africa   65.868719      0.727128  0.636008 -7.818064
0.128457
1            Asia   76.939213      0.641635  0.609921 -7.094682
0.077633
2       Australia   88.485041      0.647714  0.646823 -6.717267
0.090768
3          Europe   74.737859      0.687536  0.646758 -6.992627
0.118411
4          Global   91.483696      0.668464  0.644721 -6.491062
0.102253
5   North America   87.243302      0.724935  0.667067 -5.648808
0.129768
6   South America   85.560436      0.743896  0.685496 -5.323093
0.122962

        key      mode  acousticness  instrumentalness  liveness
valence  \
0  5.742417  0.387053      0.294682          0.029044  0.154978
0.577856
1  5.489282  0.583662      0.345011          0.019343  0.167037
0.496827
2  5.133273  0.672711      0.240193          0.007005  0.172057
0.525167
3  5.457599  0.435204      0.269568          0.022413  0.170663
0.518607
4  5.373188  0.548913      0.284273          0.012372  0.168224
0.497390
5  5.669442  0.483345      0.279252          0.013748  0.171278
0.532752
6  5.720327  0.485299      0.276805          0.010408  0.194653
0.596172

        tempo
0  117.680520
1  121.069003
2  124.003442
3  123.633908
4  119.858261
5  123.468313
6  119.848222
```

```
fig = go.Figure(go.Sunburst(
    labels=top_continent_df["continent"],
    parents=[''] * len(top_continent_df['continent']),
    values=top_continent_df['popularity'],
    hovertemplate='<b>%{label}</b><br>%{value:.2f} (%
```

```
{percentParent:.2%})<extra></extra>'
))
# Define available attributes
attributes = ['danceability', 'energy', 'loudness', 'speechiness',
'key', 'mode', 'acousticness', 'instrumentalness', 'liveness',
'valence', 'tempo']
# Create the dropdown menu
buttons = []
for attribute in attributes:
    buttons.append(dict(
        method='restyle',
        args=[{'values': [top_continent_df[attribute]]}],
        label=attribute.capitalize()
    ))
fig.update_layout(
    updatemenus=[
        dict(
            buttons=buttons,
            direction='down',
            active=0,
            x=1.0,
            y=1.0
        ),
    ],
    title='Characteristics of Music Liked by Top 5 Most listening
continents',
    margin=dict(t=50, l=0, r=5, b=0)
)

fig.show()
```


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"hovertemplate":"<b>%{label}</b><br>%{value:.2f} (%
{percentParent:.2%})<extra></extra>","labels":
["Asia","Australia","Global","North America","South
America"],"parents":["","","","",""],"type":"sunburst","values":
[76.93921296790018,88.48504079782411,91.4836956521739,87.243301955105,
85.56043557168785]}],"layout":{"margin":
{"b":0,"l":0,"r":5,"t":50},"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
```

```
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
```

```
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
```

```
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Characteristics of Music Liked by Top 5 Most listening
continents"},"updatemenus":[{"active":0,"buttons":[{"args":[{"values":
[[0.6416350645195692,0.647713508612874,0.6684637681159421,0.7249353729
181752,0.7438958257713248]]}],"label":"Danceability","method":"restyle
"},{"args":[{"values":
[[0.6099205716113896,0.6468233907524932,0.6447211956521739,0.667066654
5981173,0.6854961887477313]]}],"label":"Energy","method":"restyle"},
{"args":[{"values":[[-7.094681561266929,-6.717267452402538,-
6.491061594202899,-5.648807566980449,-
5.323092921960073]]}],"label":"Loudness","method":"restyle"},{"args":
[{"values":[[7.763318758664818e-2,9.07675430643699e-
2,0.10225326086956522,0.12976797610427226,0.1229617604355717]]}],"labe
l":"Speechiness","method":"restyle"},{"args":[{"values":
[[5.489282286445558,5.133272892112421,5.3731884057971016,5.66944243301
9551,5.72032667876588]]}],"label":"Key","method":"restyle"},{"args":
[{"values":
[[0.5836621520742241,0.6727107887579329,0.5489130434782609,0.483345401
8826937,0.4852994555353902]]}],"label":"Mode","method":"restyle"},
{"args":[{"values":
[[0.3450114569371867,0.2401933907524932,0.28427342391304344,0.27925184
355358434,0.27680517604355714]]}],"label":"Acousticness","method":"res
tyle"},{"args":[{"values":[[1.934273879492375e-2,7.004705929283771e-
3,1.2371834402173914e-2,1.3747754971035482e-2,1.0408352600725953e-
2]]}],"label":"Instrumentalness","method":"restyle"},{"args":
[{"values":
[[0.16703715474032205,0.17205720761559384,0.16822373188405795,0.171278
22230267922,0.19465326678765882]]}],"label":"Liveness","method":"resty
le"},{"args":[{"values":
[[0.49682730084248694,0.5251667271078876,0.49738985507246375,0.5327524
438812455,0.5961715063520872]]}],"label":"Valence","method":"restyle"}
,{"args":[{"values":
[[121.06900319931749,124.00344242973708,119.85826086956521,123.4683126
```

```
3577118,119.8482217785844]]}],"label":"Tempo","method":"restyle"}],"direction":"down","x":1,"y":1}]}}
```


Insights: The plot shows that the top 5 most listening continents have different preferences in terms of music characteristics. For example, North America prefers songs with high energy, danceability, and loudness, while Europe prefers songs with high acousticness and instrumentalness.

## Q5. Display the world's map based upon popularity of the songs.

```
# Filter dataset to include only the rows with the highest popularity
for each country
top_songs = df.groupby('country_name').apply(lambda x: x.nlargest(1,
'popularity')).reset_index(drop=True)

# Create a global map visualization
fig = px.choropleth(top_songs, locations='country_name',
locationmode='country names',
                    color='popularity', projection='natural earth',
                    hover_data=['name', 'artists', 'popularity',
'is_explicit'])

# Customize the map layout
fig.update_layout(title='Top Song of Each Country Based on
Popularity',
                  coloraxis_colorbar=dict(title='Popularity'),
                  geo=dict(showframe=False, showcoastlines=False,
projection_type='equirectangular'))

# Show the map
fig.show()
```


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"coloraxis":"coloraxis","customdata":[["Si No Estás","iñigo
quintero",98,false],["Cruel Summer","Taylor Swift",100,false],["Cruel
Summer","Taylor Swift",100,false],["Seven (feat. Latto) (Explicit
Ver.)","Jung Kook, Latto",98,true],["Cruel Summer","Taylor
Swift",100,false],["LALA","Myke Towers",98,true],["greedy","Tate
McRae",99,true],["greedy","Tate McRae",99,true],["Cruel
Summer","Taylor Swift",100,false],["LALA","Myke Towers",98,true],
["LALA","Myke Towers",98,true],["LALA","Myke Towers",98,true],
["greedy","Tate McRae",99,true],["Cruel Summer","Taylor
Swift",100,false],["LALA","Myke Towers",98,true],["LALA","Myke
Towers",98,true],["Seven (feat. Latto) (Explicit Ver.)","Jung Kook,
Latto",98,true],["LALA","Myke Towers",98,true],["Cruel Summer","Taylor
Swift",100,false],["greedy","Tate McRae",99,true],["greedy","Tate
McRae",99,true],["Cruel Summer","Taylor Swift",100,false],["Cruel
Summer","Taylor Swift",100,false],["greedy","Tate McRae",99,true],
["LALA","Myke Towers",98,true],["LALA","Myke Towers",98,true],["Cruel
Summer","Taylor Swift",100,false],["greedy","Tate McRae",99,true],
```

```
["Cruel Summer","Taylor Swift",100,false],["Heeriye (feat. Arijit
Singh)","Jasleen Royal, Arijit Singh, Dulquer Salmaan",91,false],
["Cruel Summer","Taylor Swift",100,false],["Cruel Summer","Taylor
Swift",100,false],["greedy","Tate McRae",99,true],["Si No
Estás","iñigo quintero",98,false],["Seven (feat. Latto) (Explicit
Ver.)","Jung Kook, Latto",98,true],["greedy","Tate McRae",99,true],
["Cruel Summer","Taylor Swift",100,false],["Cruel Summer","Taylor
Swift",100,false],["Cruel Summer","Taylor Swift",100,false],["Cruel
Summer","Taylor Swift",100,false],["LALA","Myke Towers",98,true],
["greedy","Tate McRae",99,true],["Cruel Summer","Taylor
Swift",100,false],["Cruel Summer","Taylor Swift",100,false],
["LALA","Myke Towers",98,true],["City Boys","Burna Boy",85,true],
["Cruel Summer","Taylor Swift",100,false],["greedy","Tate
McRae",99,true],["Si No Estás","iñigo quintero",98,false],
["LALA","Myke Towers",98,true],["Si No Estás","iñigo
quintero",98,false],["Cruel Summer","Taylor Swift",100,false],
["greedy","Tate McRae",99,true],["Cruel Summer","Taylor
Swift",100,false],["greedy","Tate McRae",99,true],["Cruel
Summer","Taylor Swift",100,false],["Cruel Summer","Taylor
Swift",100,false],["Cruel Summer","Taylor Swift",100,false],["Cruel
Summer","Taylor Swift",100,false],["Cruel Summer","Taylor
Swift",100,false],["Si No Estás","iñigo quintero",98,false],["Cruel
Summer","Taylor Swift",100,false],["Cruel Summer","Taylor
Swift",100,false],["Cruel Summer","Taylor Swift",100,false],["Cruel
Summer","Taylor Swift",100,false],["Ateşe Düştüm","Mert
Demir",82,false],["greedy","Tate McRae",99,true],["Cruel
Summer","Taylor Swift",100,false],["Cruel Summer","Taylor
Swift",100,false],["Cruel Summer","Taylor Swift",100,false],["Si No
Estás","iñigo quintero",98,false],["LALA","Myke Towers",98,true],
["Seven (feat. Latto) (Explicit Ver.)","Jung Kook,
Latto",98,true]],"geo":"geo","hovertemplate":"country_name=%
{location}<br>name=%{customdata[0]}<br>artists=%
{customdata[1]}<br>popularity=%{z}<br>is_explicit=%
{customdata[3]}<extra></extra>","locationmode":"country
names","locations":
["Argentina","Australia","Austria","Belarus","Belgium","Bolivia","Braz
il","Bulgaria","Canada","Chile","Colombia","Costa Rica","Czech
Republic","Denmark","Dominican Republic","Ecuador","Egypt","El
Salvador","Estonia","Finland","France","Germany","Global","Greece","Gu
atemala","Honduras","Hong
Kong","Hungary","Iceland","India","Indonesia","Ireland","Israel","Ital
y","Japan","Kazakhstan","Latvia","Lithuania","Luxembourg","Malaysia","
Mexico","Morocco","Netherlands","New
Zealand","Nicaragua","Nigeria","Norway","Pakistan","Panama","Paraguay"
,"Peru","Philippines","Poland","Portugal","Romania","Saudi
Arabia","Singapore","Slovakia","South Africa","South
Korea","Spain","Sweden","Switzerland","Taiwan","Thailand","Turkey","Uk
raine","United Arab Emirates","United Kingdom","United
States","Uruguay","Venezuela","Vietnam"],"name":"","type":"choropleth"
```

```
,"z":
[98,100,100,98,100,98,99,99,100,98,98,98,99,100,98,98,98,98,100,99,99,
100,100,99,98,98,100,99,100,91,100,100,99,98,98,99,100,100,100,100,98,
99,100,100,98,85,100,99,98,98,98,100,99,100,99,100,100,100,100,100,98,
100,100,100,100,82,99,100,100,100,98,98,98]}],"layout":{"coloraxis":
{"colorbar":{"title":{"text":"Popularity"}},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"geo":{"center":
{},"domain":{"x":[0,1],"y":[0,1]},"projection":
{"type":"equirectangular"},"showcoastlines":false,"showframe":false},"
legend":{"tracegroupgap":0},"margin":{"t":60},"template":{"data":
{"bar":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
```

```
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
```

```
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Top Song of Each Country Based on Popularity"}}}
```


Insights: The plot shows that the popularity of songs varies across different countries and regions. Some countries have a higher concentration of popular songs than others, and the popularity of songs is not evenly distributed across the world.

## Conclusion

After analyzing the "Top Spotify Songs in 73 Countries" dataset and visualizing the data, we can conclude that: Ed Sheeran, Drake, and Post Malone are the top three artists with the most popular songs in the dataset. The top 10 artists are dominated by male artists, and most of them are from the United States. The United States, the United Kingdom, and Canada have the most popular songs in the dataset. The top 10 countries are dominated by English-speaking countries, and most of the songs are in English. The top 5 most listening countries have different preferences in terms of music characteristics. For example, the United States prefers songs with high energy, danceability, and loudness, while the United Kingdom prefers songs with high acousticness and instrumentalness. The top 5 most listening continents have different preferences in terms of music characteristics. For example, North America prefers songs with high energy, danceability, and loudness, while Europe prefers songs with high acousticness and instrumentalness. The popularity of songs varies across different countries and regions. Some countries have a higher concentration of popular songs than others, and the popularity of songs is not evenly distributed across the world. These conclusions can be used to gain insights into the most popular songs, artists, and countries, and can be used for further analysis or modeling.

```
pip install notebook-as-pdf
```

```
Collecting notebook-as-pdf
  Downloading notebook_as_pdf-0.5.0-py3-none-any.whl (6.5 kB)
Requirement already satisfied: nbconvert in c:\users\ihsan bt\.conda\
envs\ml_python\lib\site-packages (from notebook-as-pdf) (7.10.0)
Collecting pyppeteer (from notebook-as-pdf)
  Downloading pyppeteer-1.0.2-py3-none-any.whl (83 kB)
     ---------------------------------------- 0.0/83.4 kB ? eta
-:--:--
     ---- ----------------------------------- 10.2/83.4 kB ? eta
-:--:--
     --------- ------------------------------ 20.5/83.4 kB 330.3 kB/s
eta 0:00:01
     ------------------------ --------------- 51.2/83.4 kB 375.8 kB/s
eta 0:00:01
     ---------------------------------------- 83.4/83.4 kB 522.4 kB/s
eta 0:00:00
Collecting PyPDF2 (from notebook-as-pdf)
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
     ---------------------------------------- 0.0/232.6 kB ? eta
-:--:--
     ------ --------------------------------- 41.0/232.6 kB 960.0 kB/s
eta 0:00:01
     ------------------- -------------------- 112.6/232.6 kB 1.3 MB/s
eta 0:00:01
     --------------------------------- ------ 194.6/232.6 kB 1.5 MB/s
eta 0:00:01
     ---------------------------------------- 232.6/232.6 kB 1.4 MB/s
eta 0:00:00
Requirement already satisfied: beautifulsoup4 in c:\users\ihsan
```

```
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (4.12.2)
Requirement already satisfied: bleach!=5.0.0 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (6.1.0)
Requirement already satisfied: defusedxml in c:\users\ihsan bt\.conda\
envs\ml_python\lib\site-packages (from nbconvert->notebook-as-pdf)
(0.7.1)
Requirement already satisfied: jinja2>=3.0 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (3.1.2)
Requirement already satisfied: jupyter-core>=4.7 in c:\users\ihsan bt\
appdata\roaming\python\python311\site-packages (from nbconvert-
>notebook-as-pdf) (5.3.2)
Requirement already satisfied: jupyterlab-pygments in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (0.2.2)
Requirement already satisfied: markupsafe>=2.0 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (2.1.3)
Requirement already satisfied: mistune<4,>=2.0.3 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (3.0.2)
Requirement already satisfied: nbclient>=0.5.0 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (0.8.0)
Requirement already satisfied: nbformat>=5.7 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (5.9.2)
Requirement already satisfied: packaging in c:\users\ihsan bt\appdata\
roaming\python\python311\site-packages (from nbconvert->notebook-as-
pdf) (23.1)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from nbconvert->notebook-
as-pdf) (1.5.0)
Requirement already satisfied: pygments>=2.4.1 in c:\users\ihsan bt\
appdata\roaming\python\python311\site-packages (from nbconvert-
>notebook-as-pdf) (2.16.1)
Requirement already satisfied: tinycss2 in c:\users\ihsan bt\.conda\
envs\ml_python\lib\site-packages (from nbconvert->notebook-as-pdf)
(1.2.1)
Requirement already satisfied: traitlets>=5.1 in c:\users\ihsan bt\
appdata\roaming\python\python311\site-packages (from nbconvert-
>notebook-as-pdf) (5.10.1)
Collecting appdirs<2.0.0,>=1.4.3 (from pyppeteer->notebook-as-pdf)
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Requirement already satisfied: certifi>=2021 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from pyppeteer->notebook-
as-pdf) (2023.7.22)
```

```
Collecting importlib-metadata>=1.4 (from pyppeteer->notebook-as-pdf)
  Obtaining dependency information for importlib-metadata>=1.4 from
https://files.pythonhosted.org/packages/cc/37/db7ba97e676af155f5fcb1a3
5466f446eadc9104e25b83366e8088c9c926/importlib_metadata-6.8.0-py3-
none-any.whl.metadata
  Using cached importlib_metadata-6.8.0-py3-none-any.whl.metadata (5.1
kB)
Collecting pyee<9.0.0,>=8.1.0 (from pyppeteer->notebook-as-pdf)
  Downloading pyee-8.2.2-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from pyppeteer->notebook-
as-pdf) (4.66.1)
Collecting urllib3<2.0.0,>=1.25.8 (from pyppeteer->notebook-as-pdf)
  Obtaining dependency information for urllib3<2.0.0,>=1.25.8 from
https://files.pythonhosted.org/packages/b0/53/aa91e163dcfd1e5b82d8a890
ecf13314e3e149c05270cc644581f77f17fd/urllib3-1.26.18-py2.py3-none-
any.whl.metadata
  Downloading urllib3-1.26.18-py2.py3-none-any.whl.metadata (48 kB)
     ---------------------------------------- 0.0/48.9 kB ? eta
-:--:--
     ---------------------------------------- 48.9/48.9 kB 1.2 MB/s
eta 0:00:00
Collecting websockets<11.0,>=10.0 (from pyppeteer->notebook-as-pdf)
  Downloading websockets-10.4-cp311-cp311-win_amd64.whl (101 kB)
     ---------------------------------------- 0.0/101.4 kB ? eta
-:--:--
     ----------- --------------------------- 30.7/101.4 kB 1.4 MB/s
eta 0:00:01
     --------------------------------------- 101.4/101.4 kB 1.5 MB/s
eta 0:00:00
Requirement already satisfied: six>=1.9.0 in c:\users\ihsan bt\
appdata\roaming\python\python311\site-packages (from bleach!=5.0.0-
>nbconvert->notebook-as-pdf) (1.16.0)
Requirement already satisfied: webencodings in c:\users\ihsan
bt\.conda\envs\ml_python\lib\site-packages (from bleach!=5.0.0-
>nbconvert->notebook-as-pdf) (0.5.1)
Collecting zipp>=0.5 (from importlib-metadata>=1.4->pyppeteer-
>notebook-as-pdf)
  Obtaining dependency information for zipp>=0.5 from
https://files.pythonhosted.org/packages/d9/66/48866fc6b158c81cc2bfecc0
4c480f105c6040e8b077bc54c634b4a67926/zipp-3.17.0-py3-none-
any.whl.metadata
  Using cached zipp-3.17.0-py3-none-any.whl.metadata (3.7 kB)
Requirement already satisfied: platformdirs>=2.5 in c:\users\ihsan bt\
appdata\roaming\python\python311\site-packages (from jupyter-
core>=4.7->nbconvert->notebook-as-pdf) (3.10.0)
Requirement already satisfied: pywin32>=300 in c:\users\ihsan bt\
appdata\roaming\python\python311\site-packages (from jupyter-
core>=4.7->nbconvert->notebook-as-pdf) (306)
```

```
Requirement already satisfied: jupyter-client>=6.1.12 in c:\users\ihsan bt\appdata\roaming\python\python311\site-packages (from nbclient>=0.5.0->nbconvert->notebook-as-pdf) (8.3.1)
Requirement already satisfied: fastjsonschema in c:\users\ihsan bt\.conda\envs\ml_python\lib\site-packages (from nbformat>=5.7->nbconvert->notebook-as-pdf) (2.18.1)
Requirement already satisfied: jsonschema>=2.6 in c:\users\ihsan bt\.conda\envs\ml_python\lib\site-packages (from nbformat>=5.7->nbconvert->notebook-as-pdf) (4.19.1)
Requirement already satisfied: colorama in c:\users\ihsan bt\appdata\roaming\python\python311\site-packages (from tqdm<5.0.0,>=4.42.1->pyppeteer->notebook-as-pdf) (0.4.6)
Requirement already satisfied: soupsieve>1.2 in c:\users\ihsan bt\.conda\envs\ml_python\lib\site-packages (from beautifulsoup4->nbconvert->notebook-as-pdf) (2.5)
Requirement already satisfied: attrs>=22.2.0 in c:\users\ihsan bt\.conda\envs\ml_python\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert->notebook-as-pdf) (23.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in c:\users\ihsan bt\.conda\envs\ml_python\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert->notebook-as-pdf) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in c:\users\ihsan bt\.conda\envs\ml_python\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert->notebook-as-pdf) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\ihsan bt\.conda\envs\ml_python\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert->notebook-as-pdf) (0.10.6)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\ihsan bt\appdata\roaming\python\python311\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert->notebook-as-pdf) (2.8.2)
Requirement already satisfied: pyzmq>=23.0 in c:\users\ihsan bt\appdata\roaming\python\python311\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert->notebook-as-pdf) (25.1.1)
Requirement already satisfied: tornado>=6.2 in c:\users\ihsan bt\appdata\roaming\python\python311\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert->notebook-as-pdf) (6.3.3)
Downloading importlib_metadata-6.8.0-py3-none-any.whl (22 kB)
Downloading urllib3-1.26.18-py2.py3-none-any.whl (143 kB)
   ---------------------------------------- 0.0/143.8 kB ? eta -:--:--
   ------------------------- -------------- 92.2/143.8 kB 2.6 MB/s eta 0:00:01
   ---------------------------------------- 143.8/143.8 kB 1.4 MB/s eta 0:00:00
Downloading zipp-3.17.0-py3-none-any.whl (7.4 kB)
Installing collected packages: pyee, appdirs, zipp, websockets, urllib3, PyPDF2, importlib-metadata, pyppeteer, notebook-as-pdf
  Attempting uninstall: urllib3
    Found existing installation: urllib3 2.0.7
    Uninstalling urllib3-2.0.7:
```

```
      Successfully uninstalled urllib3-2.0.7
Successfully installed PyPDF2-3.0.1 appdirs-1.4.4 importlib-metadata-
6.8.0 notebook-as-pdf-0.5.0 pyee-8.2.2 pyppeteer-1.0.2 urllib3-1.26.18
websockets-10.4 zipp-3.17.0
Note: you may need to restart the kernel to use updated packages.
```