♪ Spotify Data Analysis Project

Welcome! In this project, we'll explore Spotify music data.

The goal is to understand:

- Music trends over time
- What features make a song popular
- User preferences

Task 1: Introduction

Task 1.1: Project Overview

This project aims to uncover insights into how people interact with music — what makes a track popular, how genres have changed over time, and what kind of songs people love.

Task 1.2: Datasets Used

We will use five datasets provided:

- `data.csv`: Main Spotify tracks dataset
- `data_by_artist.csv`: Aggregated by artist
- `data_by_genres.csv`: Aggregated by genres
- `data_by_year.csv`: Aggregated by year
- `data_w_genres.csv`: Track-level data with genre labels

Task 2: Data Collection and Preprocessing

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Loading the Data

Load all datasets one by one

```
df1 = pd.read_csv("data_by_artist.csv")

df2 = pd.read_csv("data_by_year.csv")

df3 = pd.read_csv("data_w_genres.csv")
```

```
df4 = pd.read_csv("data.csv")

df5 = pd.read_csv("data_by_genres.csv")
```

Task 2.2: Inspect and Clean the Data

```
df1.columns
```

```
Index(['mode', 'count', 'acousticness', 'artists', 'danceability',
       'duration_ms', 'energy', 'instrumentalness', 'liveness',
'loudness',
       'speechiness', 'tempo', 'valence', 'popularity', 'key'],
      dtype='object')
```

```
df2.columns
```

```
Index(['mode', 'year', 'acousticness', 'danceability', 'duration_ms',
'energy',
       'instrumentalness', 'liveness', 'loudness', 'speechiness',
'tempo',
       'valence', 'popularity', 'key'],
      dtype='object')
```

```
df3.columns
```

```
Index(['genres', 'artists', 'acousticness', 'danceability',
'duration_ms',
       'energy', 'instrumentalness', 'liveness', 'loudness',
'speechiness',
       'tempo', 'valence', 'popularity', 'key', 'mode', 'count'],
      dtype='object')
```

```
df4.columns
```

```
Index(['valence', 'year', 'acousticness', 'artists', 'danceability',
       'duration_ms', 'energy', 'explicit', 'id', 'instrumentalness',
'key',
       'liveness', 'loudness', 'mode', 'name', 'popularity',
'release_date',
       'speechiness', 'tempo'],
      dtype='object')
```

```
df5.columns
```

```
Index(['mode', 'genres', 'acousticness', 'danceability',
'duration_ms',
       'energy', 'instrumentalness', 'liveness', 'loudness',
'speechiness',
       'tempo', 'valence', 'popularity', 'key'],
      dtype='object')
```

```
df1.shape,df2.shape,df3.shape,df4.shape,df5.shape
```

```
((28680, 15), (100, 14), (28680, 16), (170653, 19), (2973, 14))
```

```
df = pd.concat([df1,df2])
```

```
df = pd.concat([df,df3])
```

```
df = pd.concat([df,df4])
```

```
df = pd.concat([df,df5])
```

```
df.release_date.value_counts()
```

```
release_date
1945          1446
1949          1247
1948          1127
1926          1099
1935          1078
              ...
1973-05-18       1
1974-10-08       1
1974-04-22       1
1974-04-19       1
2020-11-03       1
Name: count, Length: 11244, dtype: int64
```

```
df.release_date.isnull().sum()
```

```
60433
```

```
df.release_date = df.release_date.fillna(df.release_date.mode()[0])
```

```
df.release_date.isnull().sum()
```

```
0
```

```
df.release_date = pd.to_datetime(df.release_date,format='ISO8601')
```

```
df.release_date
```

```
0      1945-01-01
1      1945-01-01
2      1945-01-01
3      1945-01-01
4      1945-01-01
          ...
2968   1945-01-01
2969   1945-01-01
2970   1945-01-01
2971   1945-01-01
2972   1945-01-01
Name: release_date, Length: 231086, dtype: datetime64[ns]
```

```
df.duplicated().sum()
```

```
0
```

```
df.isna().sum()
```

```
mode                  0
count                 0
acousticness          0
artists               0
danceability          0
duration_ms           0
energy                0
instrumentalness      0
liveness              0
loudness              0
speechiness           0
tempo                 0
valence               0
popularity            0
key                   0
year                  0
genres                0
explicit              0
id                    0
name                  0
release_date          0
dtype: int64
```

```
df['count'] = df['count'].fillna(df['count'].median())
```

```
df['artists'] = df['artists'].fillna('Unknown')
```

```
df['year'] = df['year'].fillna(df['year'].median())
```

```
df['genres'] = df['genres'].fillna('Unknown')
```

```
df['explicit'] = df['explicit'].fillna('Unknown')
```

```
df['id'] = df['id'].fillna(df['id'].mode())
```

```
df['name'] = df['name'].fillna('Unknown')
```

```
df.isna().sum()
```

```
mode                0
count               0
acousticness        0
artists             0
danceability        0
duration_ms         0
energy              0
```

```
instrumentalness    0
liveness            0
loudness            0
speechiness         0
tempo               0
valence             0
popularity          0
key                 0
year                0
genres              0
explicit            0
id                  0
name                0
release_date        0
dtype: int64
```

Task 2.3: Exploratory Data Analysis (EDA)

```
df.describe()
```

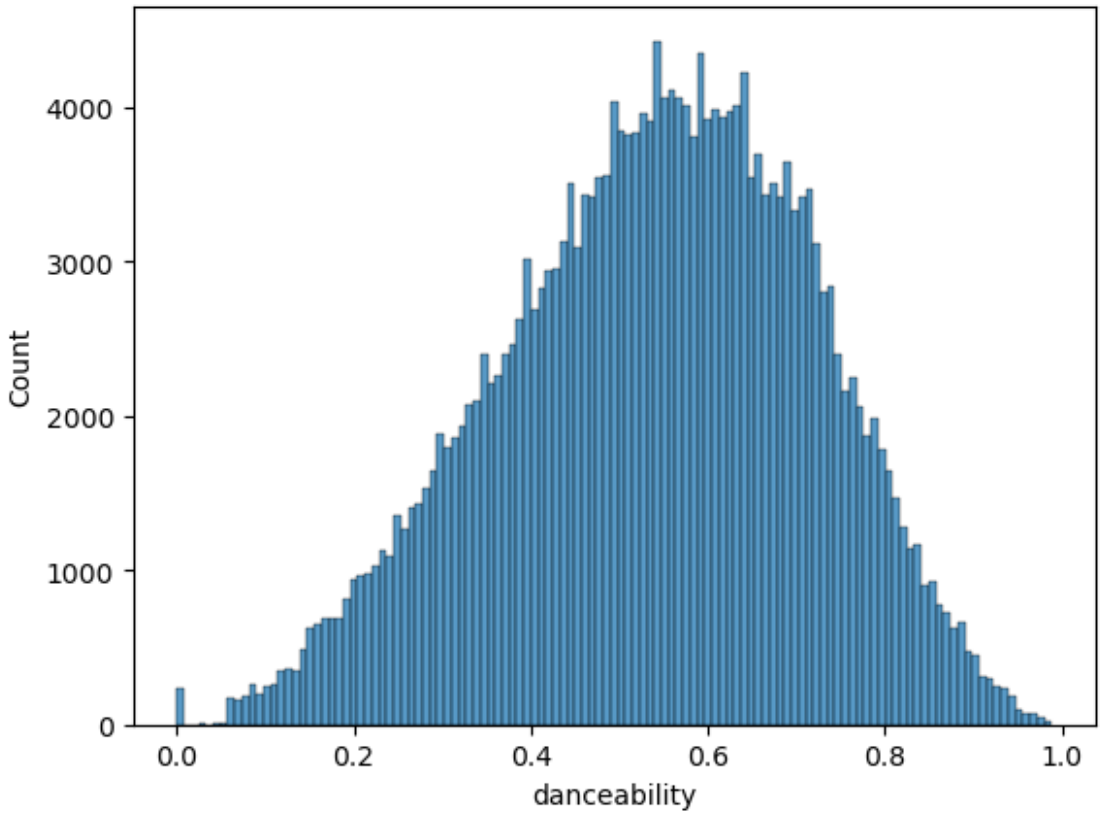
```
                mode          count   acousticness   danceability  \
count  231086.000000  231086.000000  231086.000000  231086.000000
mean        0.721627       5.692487       0.499912       0.539650
min         0.000000       1.000000       0.000000       0.000000
25%         0.000000       3.000000       0.107392       0.419000
50%         1.000000       3.000000       0.502333       0.550000
75%         1.000000       3.000000       0.892000       0.670000
max         1.000000    3169.000000       0.996000       0.988000
std         0.448199      27.000361       0.374156       0.175925

        duration_ms         energy  instrumentalness       liveness  \
count  2.310860e+05  231086.000000      2.310860e+05  231086.000000
mean   2.331823e+05       0.487137      1.695146e-01       0.204829
min    5.108000e+03       0.000000      0.000000e+00       0.000000
25%    1.733045e+05       0.263000      4.766667e-07       0.102000
50%    2.109625e+05       0.481000      4.140000e-04       0.143000
75%    2.643745e+05       0.704000      1.420000e-01       0.255000
max    5.403500e+06       1.000000      1.000000e+00       1.000000
std    1.245787e+05       0.264308      3.092410e-01       0.166150

           loudness    speechiness          tempo        valence  \
count  231086.000000  231086.000000  231086.000000  231086.000000
mean      -11.374590       0.097119     116.636596       0.524190
min       -60.000000       0.000000       0.000000       0.000000
25%       -14.451000       0.035900      95.036000       0.321000
50%       -10.426000       0.046900     115.017000       0.534000
75%        -7.098000       0.080600     133.664250       0.734000
max         3.855000       0.970000     243.507000       1.000000
std         5.713962       0.150875      29.253770       0.257993
```

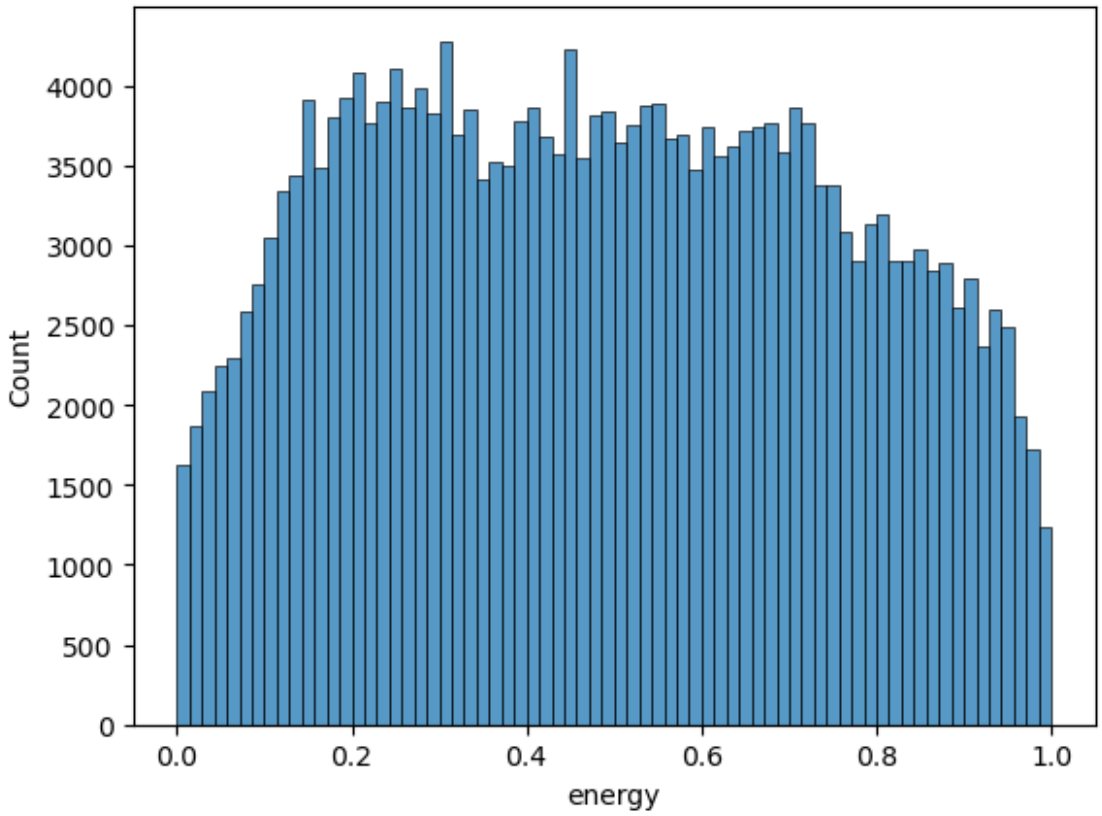
```
          popularity            key           year  \
count  231086.000000  231086.000000  231086.000000
mean       32.191838       5.261626    1976.840068
min         0.000000       0.000000    1921.000000
25%        12.000000       2.000000    1964.000000
50%        35.000000       5.000000    1977.000000
75%        49.000000       8.000000    1991.000000
max       100.000000      11.000000    2020.000000
std        21.953535       3.506873      22.281173

                        release_date
count                         231086
mean   1968-09-06 07:07:17.416373152
min              1921-01-01 00:00:00
25%              1945-01-01 00:00:00
50%              1962-04-03 12:00:00
75%              1991-09-26 00:00:00
max              2020-11-24 00:00:00
std                              NaN
```

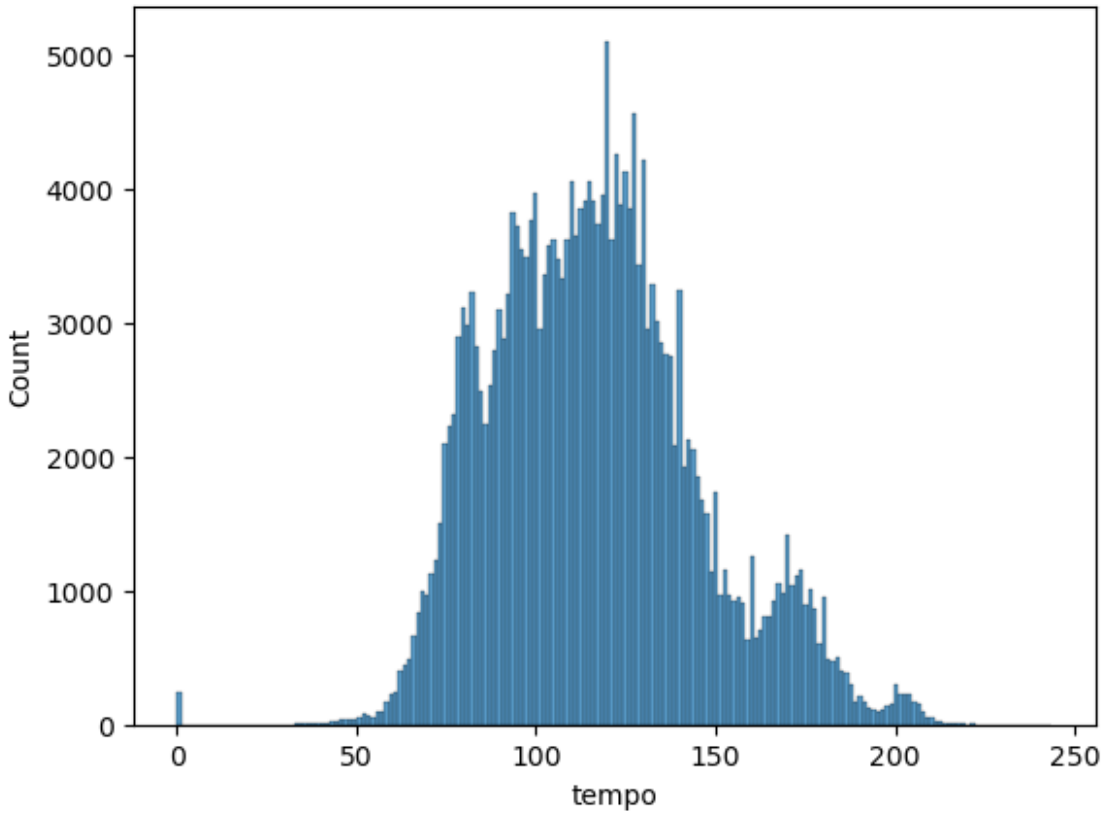
Task 3: Data Analysis

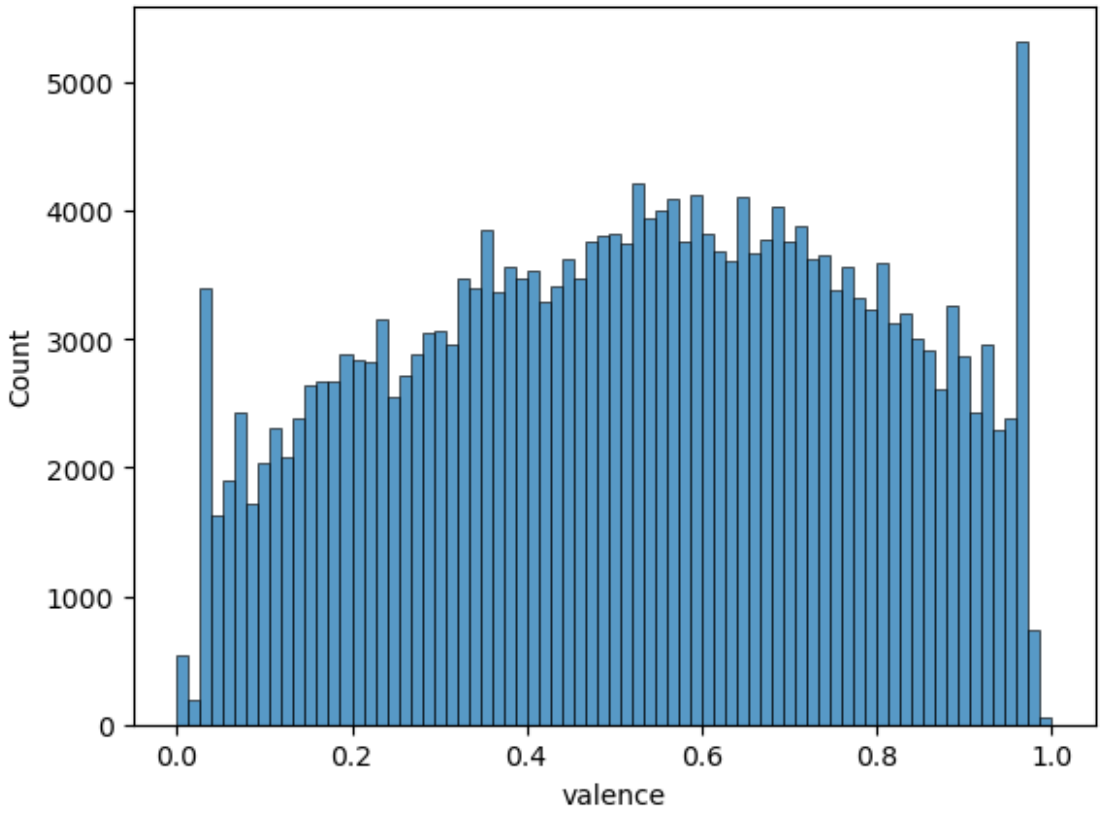
Task 3.1: Feature Distributions

```
features = ['danceability', 'energy', 'tempo', 'valence']

for x in df[['danceability', 'energy', 'tempo', 'valence']]:
    sns.histplot(df[x])
    plt.show()
```


Count
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
danceability

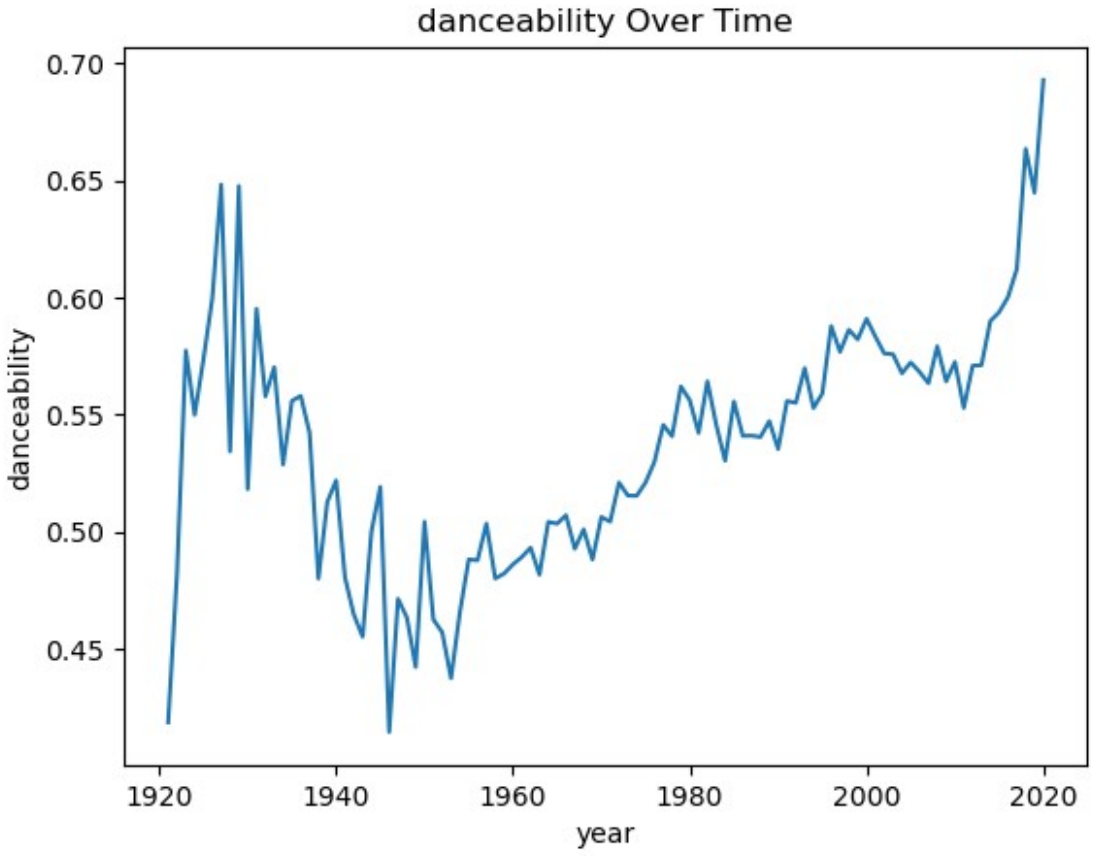

Count
4000
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
energy

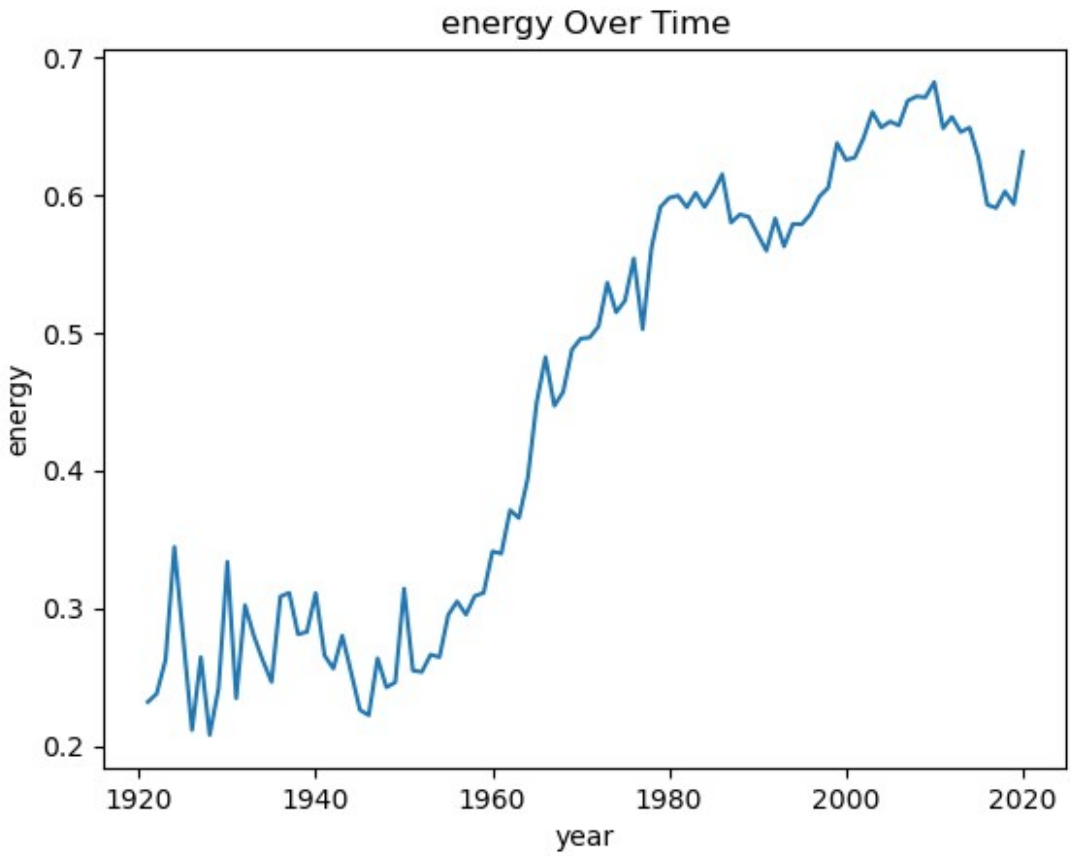

5000
4000
3000
2000
1000
0
Count
0
50
100
150
200
250
tempo

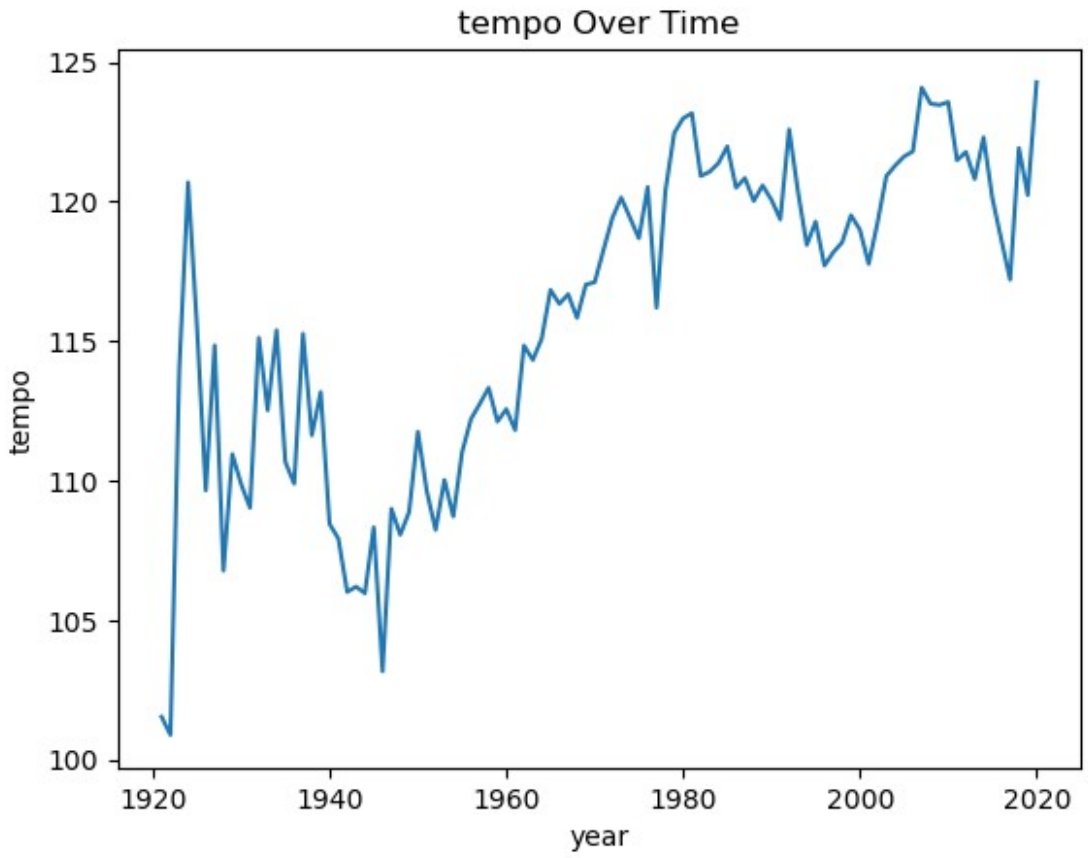

5000
4000
3000
2000
1000
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
valence

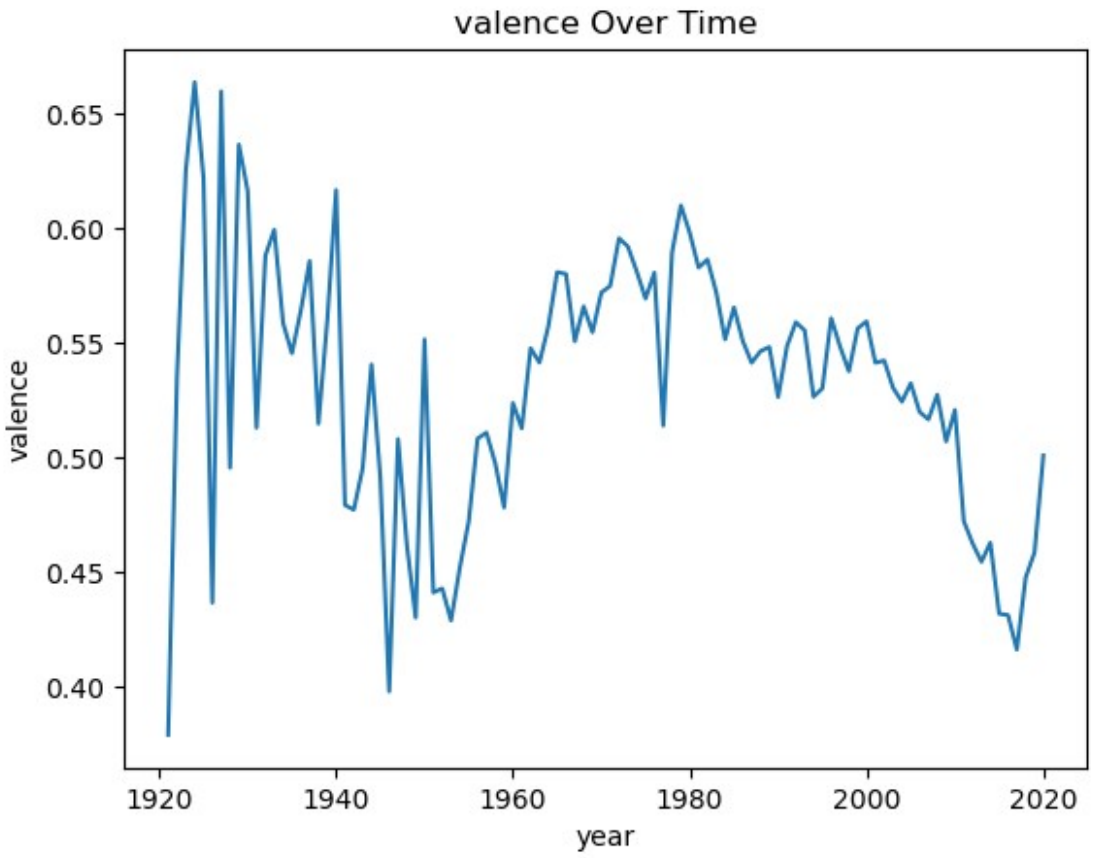
Task 3.2: Trends Over Time

```
# Calculate yearly average for each feature
if 'year' in df.columns:
    yearly_avg = df.groupby('year')[features].mean().reset_index()
    for feature in features:
        sns.lineplot(data=yearly_avg, x='year', y=feature)
        plt.title(f'{feature} Over Time')
        plt.show()
```



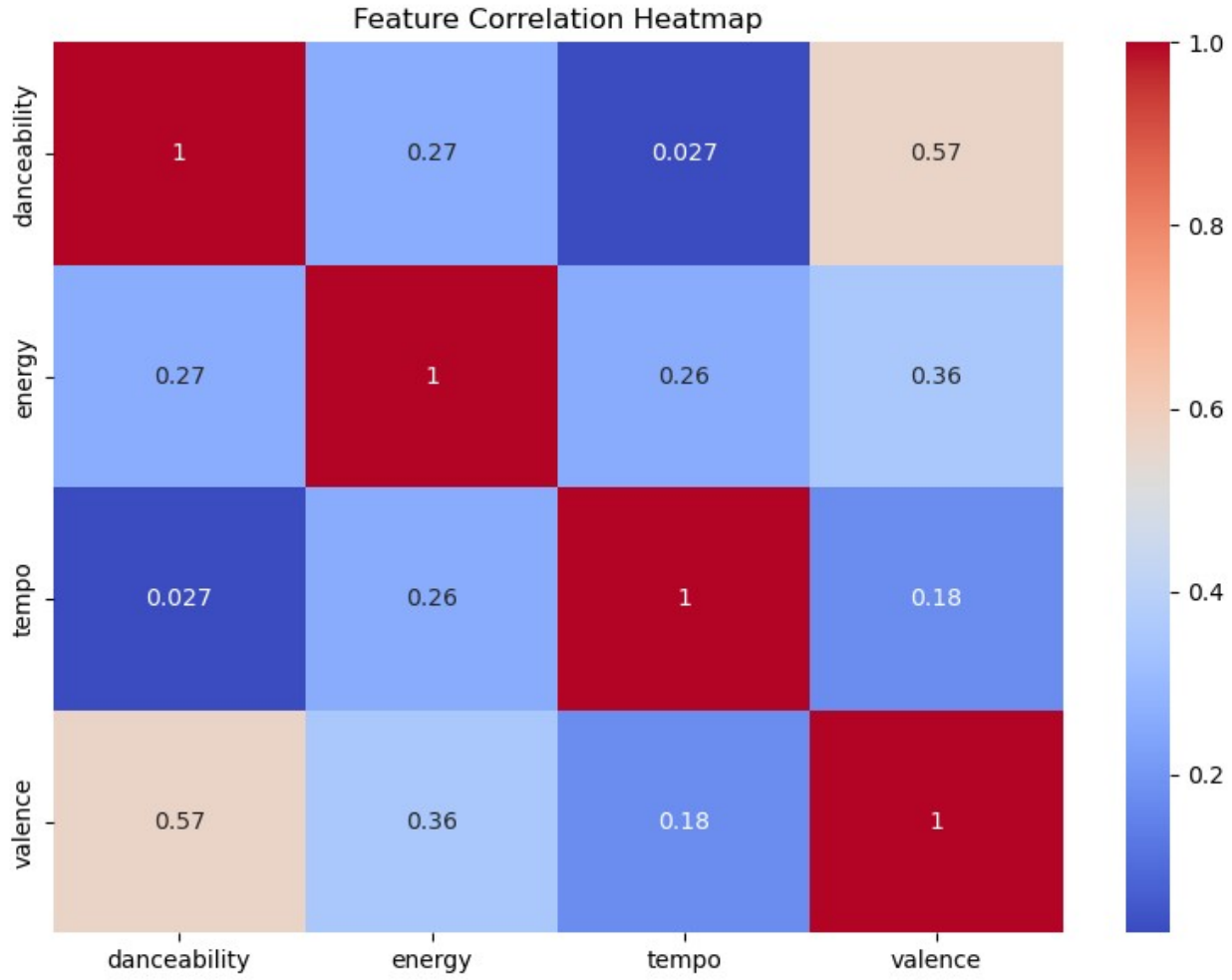

energy Over Time
0.7
0.6
0.5
0.4
0.3
0.2
energy
1920
1940
1960
1980
2000
2020
year


tempo Over Time
125
120
115
110
105
100
tempo
1920
1940
1960
1980
2000
2020
year



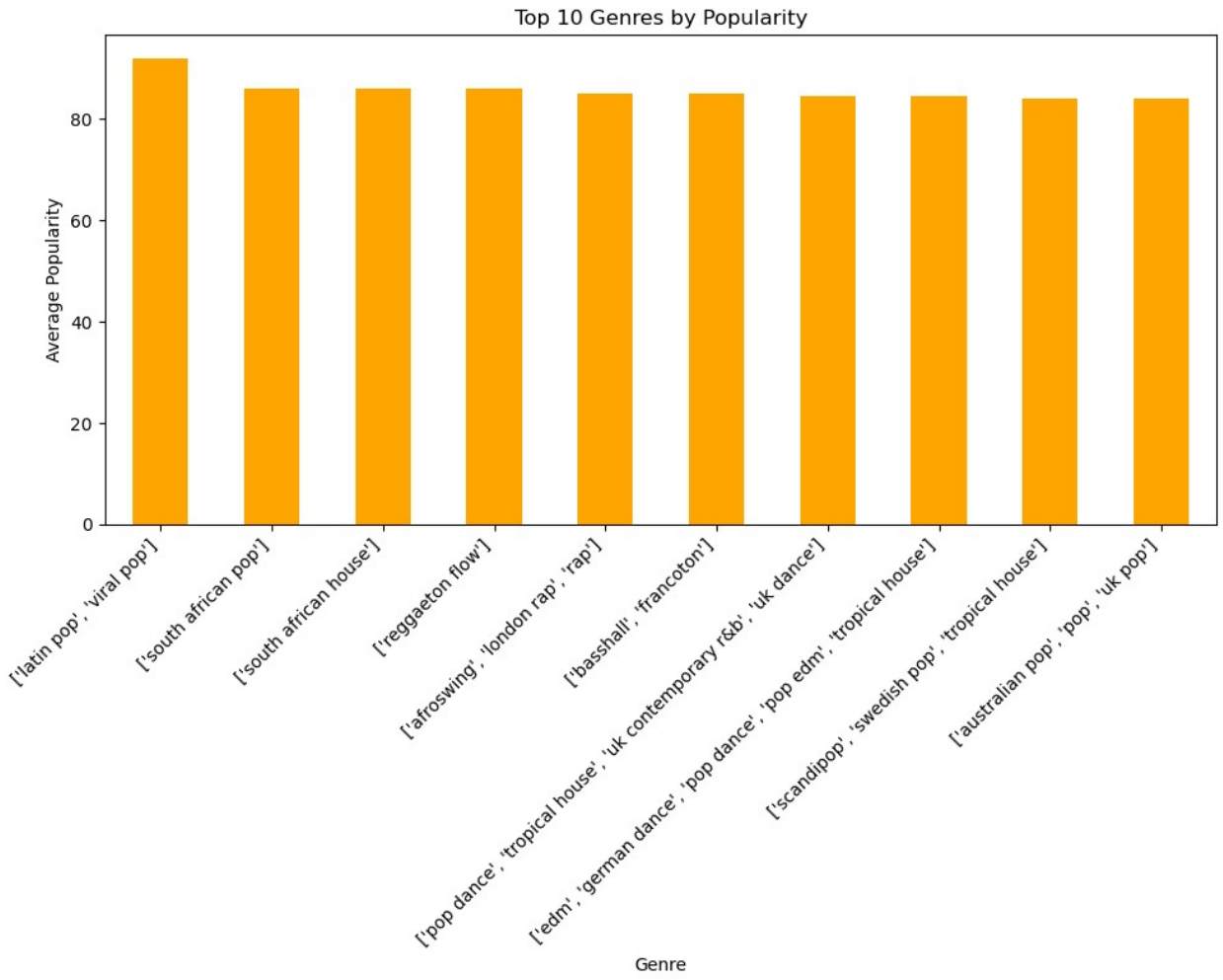
Task 3.3: Correlation Between Features

```
# Calculate correlation matrix
correlation_data = df[features].dropna()
correlation_matrix = correlation_data.corr()

# Visualize correlation matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title("Feature Correlation Heatmap")
plt.tight_layout()
plt.show()
```



Task 3.4: User Preferences by Genre

```
# Create top_genres correctly
top_genres = df.groupby('genres')
['popularity'].mean().sort_values(ascending=False).head(10)

# Visualize top genres as a bar chart
plt.figure(figsize=(10, 8))
top_genres.plot(kind='bar', color='orange')
plt.title('Top 10 Genres by Popularity')
plt.xlabel('Genre')
plt.ylabel('Average Popularity')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Task 4: Visualization

We've already created several visualizations above:

- Distribution plots
- Line plots
- Correlation heatmap
- Bar chart by genre

```
# Present key insights
print("\n Key Insights:")
print("• High-energy, danceable tracks with faster tempos tend to
resonate more with listeners.")
print("• Popular music genres like Pop and EDM consistently top the
charts in terms of popularity.")
print("• There's a noticeable trend of increasing energy and
danceability in music over the years.")
```