Spotify Data Analysis

The Spotify Data Analysis set has information about the various artists and tracks available on the Spotify application.

There are two datasets that are being used in this analysis, both of which are available on Kaggle for free. The first dataset tracks.csv contains data about all the tracks while the second dataset SpotifyFeatures.csv has additional information like genre and artist of the tracks.

Import Library

In [45]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Uploading CSV file tracks.csv

In [83]:
```
df_tracks = pd.read_csv('E:/Projects/Data Analysis/tracks.csv')
df_tracks.head()
```

Out[83]:

	id	name	popularity	duration_ms	explicit	artists
0	35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0	['Uli']
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']
3	08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0	['Dick Haymes']

.isnull()

In [84]:
```
#null values

pd.isnull(df_tracks).sum()
```

Out[84]:
```
id                  0
name               71
popularity          0
duration_ms         0
explicit            0
artists             0
id_artists          0
release_date        0
danceability        0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
acousticness        0
instrumentalness    0
liveness            0
valence             0
tempo               0
time_signature      0
dtype: int64
```

.info()

In [86]:
```
df_tracks.info()    #to check number of entries and details
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 586672 entries, 0 to 586671
Data columns (total 20 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   id                586672 non-null  object
 1   name              586601 non-null  object
 2   popularity        586672 non-null  int64
 3   duration_ms       586672 non-null  int64
 4   explicit          586672 non-null  int64
 5   artists           586672 non-null  object
 6   id_artists        586672 non-null  object
 7   release_date      586672 non-null  object
 8   danceability      586672 non-null  float64
 9   energy            586672 non-null  float64
 10  key               586672 non-null  int64
 11  loudness          586672 non-null  float64
 12  mode              586672 non-null  int64
 13  speechiness       586672 non-null  float64
 14  acousticness      586672 non-null  float64
 15  instrumentalness  586672 non-null  float64
 16  liveness          586672 non-null  float64
 17  valence           586672 non-null  float64
 18  tempo             586672 non-null  float64
 19  time_signature    586672 non-null  int64
dtypes: float64(9), int64(6), object(5)
memory usage: 89.5+ MB
```

Displaiyng the 10 least popular songs using sorting

In [87]:
```
#10 least popular songs

sorted_df = df_tracks.sort_values('popularity', ascending = True).head(10)
sorted_df
```

Out[87]:

	id	name	popularity	duration_ms	explicit	artist
546130	181rTRhCcggZPwP2TUcVqm	Newspaper Reports On Abner, 20 February 1935	0	896575	0	['Norr Gof 'Chest Lauck 'Carlto Bric
546222	0yOCz3V5KMm8l1T8EFc60i	恋は水の上で	0	188440	0	['Hiba Misora
546221	0y48Hhwe52099UqYjegRCO	私の誕生日	0	173467	0	['Hiba Misora
546220	0xCmgtf9ka07hkZg3D6PaV	エル・チョクロ (EL CHOCLO)	0	205280	0	['Hiba Misora
546219	0tBXS3VuCPX7KWUFH2nros	恋は不思議なもの	0	185733	0	['Hiba Misora
546218	0qrKnQtYDVJhKFAXTHYVS9	ゆうべはどうしたの (WHATSA MALLA U)	0	183427	0	['Hiba Misora
546217	0nqsDxOeKSwEzp3AUQAAqS	Screen Director's Playhouse, Music For Million...	0	1767071	0	['Wilm Herber 'Jur Allysor 'Josep Kear
546216	0kGEdsxVLYjCdfxM9tbezd	ブルーマンボ	0	162147	0	['Hiba Misora
546215	0bc3PUZurUUXrY7yqoOxjq	Screen Director's Playhouse, Trade Winds direc...	0	1776652	0	['Wal Mahe 'Ta Garnet 'Lurer Tuttle'
546214	0Wwm0ruSjYMIiWG0nyAI1F	Screen Director's Playhouse, It's A Wonderful ...	0	1767576	0	['Josep Granby 'Jimm Stewar 'Irer Tedr

.describe().transpose() for descriptive statistics

In [88]:
```
#descriptive statistics

df_tracks.describe().transpose()
```

Out[88]:

	count	mean	std	min	25%	
popularity	586672.0	27.570053	18.370642	0.0	13.0000	27.0
duration_ms	586672.0	230051.167286	126526.087418	3344.0	175093.0000	214893.0
explicit	586672.0	0.044086	0.205286	0.0	0.0000	0.0
danceability	586672.0	0.563594	0.166103	0.0	0.4530	0.5
energy	586672.0	0.542036	0.251923	0.0	0.3430	0.5
key	586672.0	5.221603	3.519423	0.0	2.0000	5.0
loudness	586672.0	-10.206067	5.089328	-60.0	-12.8910	-9.2
mode	586672.0	0.658797	0.474114	0.0	0.0000	1.0
speechiness	586672.0	0.104864	0.179893	0.0	0.0340	0.0
acousticness	586672.0	0.449863	0.348837	0.0	0.0969	0.4
instrumentalness	586672.0	0.113451	0.266868	0.0	0.0000	0.0
liveness	586672.0	0.213935	0.184326	0.0	0.0983	0.1
valence	586672.0	0.552292	0.257671	0.0	0.3460	0.5
tempo	586672.0	118.464857	29.764108	0.0	95.6000	117.3
time_signature	586672.0	3.873382	0.473162	0.0	4.0000	4.0

10 most popular songs with popularity > 90

In [89]:
```
#10 most popular songs with popularity > 90

#inplace = False - to not change the original dataset
most_popular = df_tracks.query('popularity>90', inplace = False).sort_values
most_popular[:10]
```

Out[89]:

	id	name	popularity	duration_ms	explicit	artis
93802	4iJyoBOLtHqaGxP12qzhQI	Peaches (feat. Daniel Caesar & Giveon)	100	198082	1	['Jus Biebe 'Dan Caesa 'Giveo
93803	7lPN2DXiMsVn7XUKtOW1CS	drivers license	99	242014	1	['Oliv Rodrig
93804	3Ofmpyhv5UAQ70mENzB277	Astronaut In The Ocean	98	132780	0	['Mask Wo
92810	5QO79kh1waicV47BqGRL3g	Save Your Tears	97	215627	1	['T Weekn
92811	6tDDoYIxWvMLTdKpjFkc1B	telepatía	97	160191	0	['K Uchi
92813	0VjIjW4GlUZAMYd2vXMi3b	Blinding Lights	96	200040	0	['T Weekn
93805	7MAibcTli4IisCtbHKrGMh	Leave The Door Open	96	242096	0	['Bru Mar 'Anders .Paa 'S Soni
92814	6f3Slt0GbA2bPZlz0aIFXN	The Business	95	164000	0	['Tiëst
91866	60ynsPSSKe6O3sfwRnIBRf	Streets	94	226987	1	['Do Ca
92816	3FAJ6O0NOHQV8Mc5Ri6ENp	Heartbreak Anniversary	94	198371	0	['Giveo

.columns

In [90]:
```
print(df_tracks.columns)
```

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

Setting the release_date column as index

In [91]:
```
df_tracks.set_index('release_date', inplace=True)
df_tracks.index=pd.to_datetime(df_tracks.index, format='mixed')
df_tracks.head()
```

Out[91]:

release_date	id	name	popularity	duration_ms	explicit
1922-02-22	35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0
1922-06-01	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0
1922-03-21	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0
1922-03-21	08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0
1922-01-01	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0

Artist at the 18th row

In [92]:
```
#artist at 18th row
df_tracks[['artists']].iloc[18]
```

Out[92]:
```
artists    ['Victor Boucher']
Name: 1922-01-01 00:00:00, dtype: object
```

Converting duration into seconds

In [93]:
```
#convert duration into seconds
df_tracks['duration']=df_tracks['duration_ms'].apply(lambda x: round(x/1000)
df_tracks.drop('duration_ms', inplace=True, axis=1)
```

In [94]:
```
df_tracks.duration.head()
```

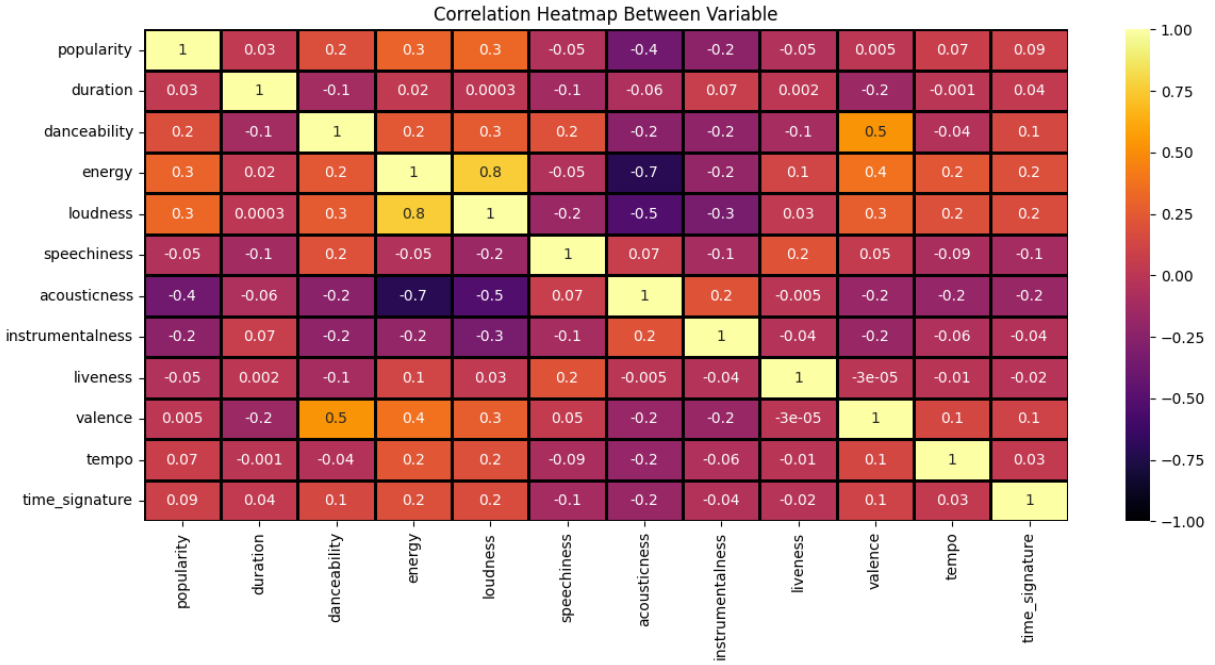
Out[94]:
```
release_date
1922-02-22    127
1922-06-01     98
1922-03-21    182
1922-03-21    177
1922-01-01    163
Name: duration, dtype: int64
```

Correlation heatmap between variables

In [95]:
```
df = df_tracks.drop(["key", "mode", "explicit"],axis=1)
cols = ['popularity', 'duration', 'danceability', 'energy', 'loudness', 'spe
df = df[cols]
corr_df = df.corr(method="pearson")
plt.figure(figsize=(14,6))
```

```
heatmap = sns.heatmap(corr_df, annot=True,fmt=".1g", vmin=-1, vmax=1, center
heatmap.set_title("Correlation Heatmap Between Variable")
heatmap.set_xticklabels(heatmap.get_xticklabels(), rotation=90)
```

Out[95]:
```
[Text(0.5, 0, 'popularity'),
 Text(1.5, 0, 'duration'),
 Text(2.5, 0, 'danceability'),
 Text(3.5, 0, 'energy'),
 Text(4.5, 0, 'loudness'),
 Text(5.5, 0, 'speechiness'),
 Text(6.5, 0, 'acousticness'),
 Text(7.5, 0, 'instrumentalness'),
 Text(8.5, 0, 'liveness'),
 Text(9.5, 0, 'valence'),
 Text(10.5, 0, 'tempo'),
 Text(11.5, 0, 'time_signature')]
```



Sampling the data

In [96]:
```
#sample the data
sample_df = df_tracks.sample(int(0.004*len(df_tracks)))
```

In [97]:
```
print(len(sample_df))
```

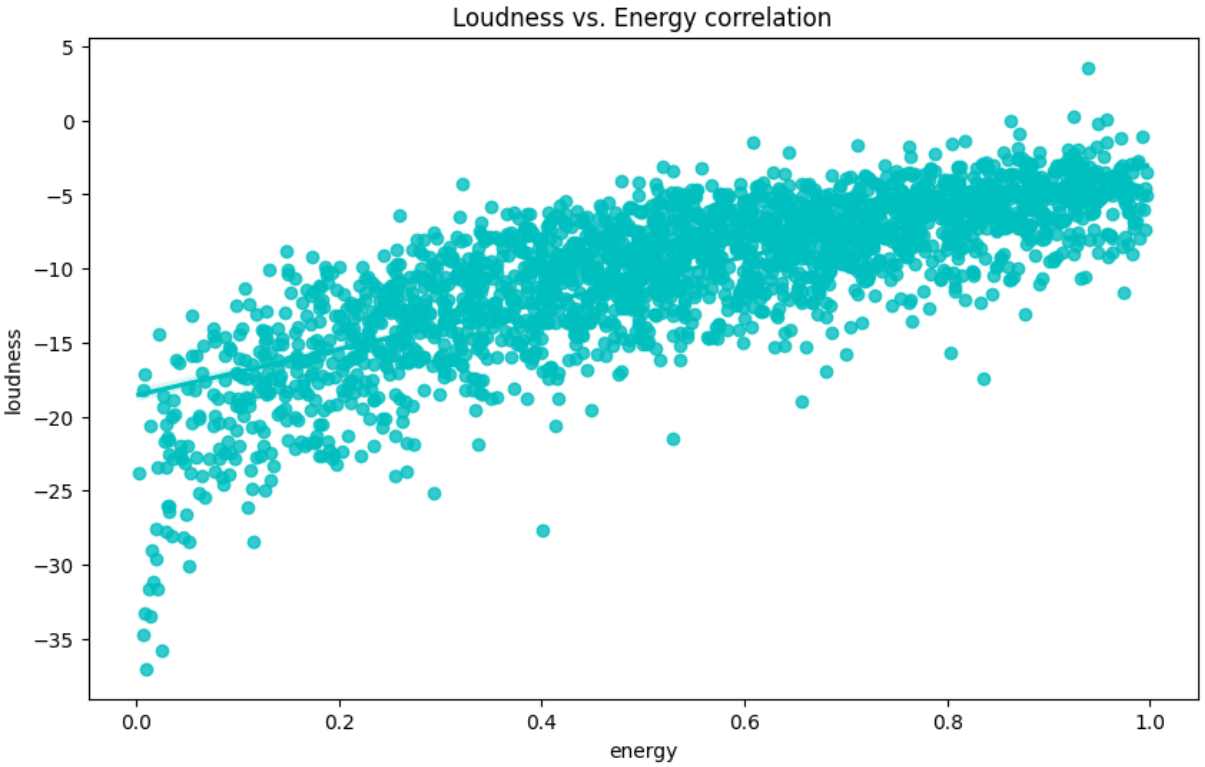
2346

Regression plot between loudness and energy

In [98]:
```
#regression plot between loudness and energy
#from correlation heatmap: loudness and energy have a positive correlation c
```

```
plt.figure(figsize=(10,6))
sns.regplot(data = sample_df, y="loudness", x="energy", color="c").set(title
```

Out[98]: [Text(0.5, 1.0, 'Loudness vs. Energy correlation')]

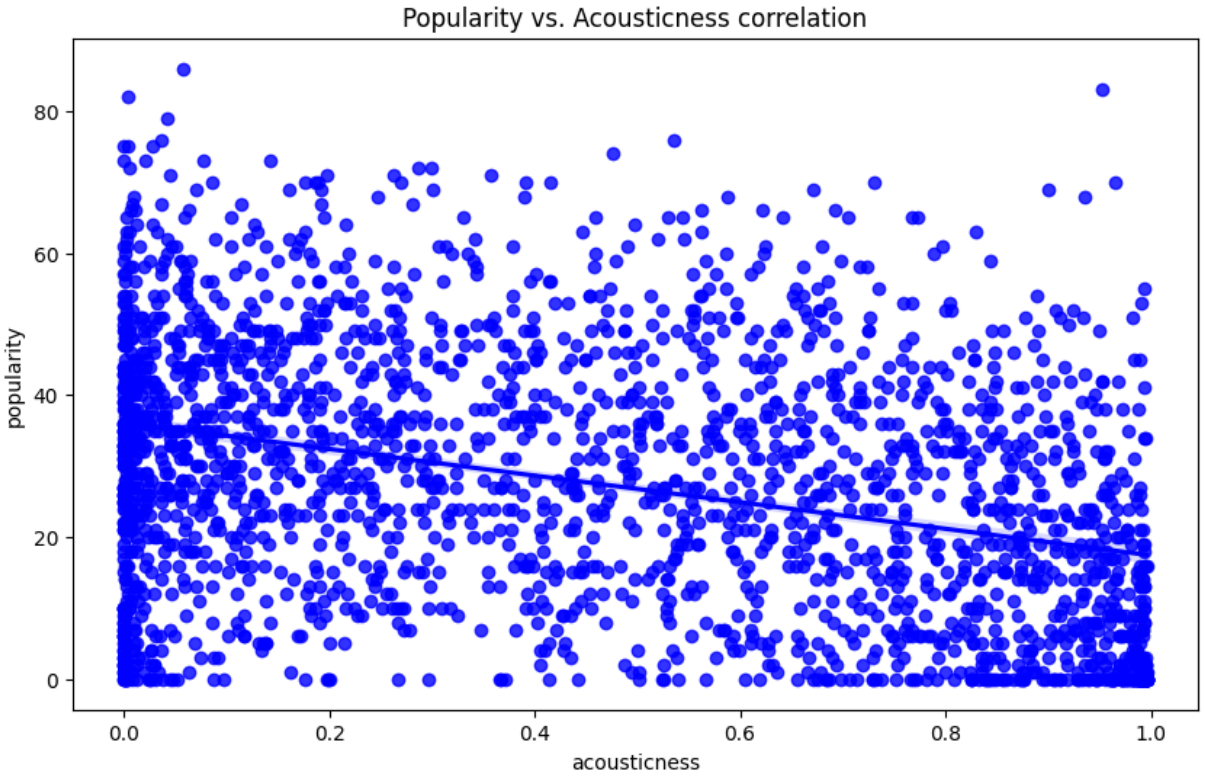


Regression plot between popularity and acousticness

In [99]:
```
#regression plot for popularity and acousticness

plt.figure(figsize=(10,6))
sns.regplot(data = sample_df, y="popularity", x="acousticness", color="b").s
```

Out[99]: [Text(0.5, 1.0, 'Popularity vs. Acousticness correlation')]



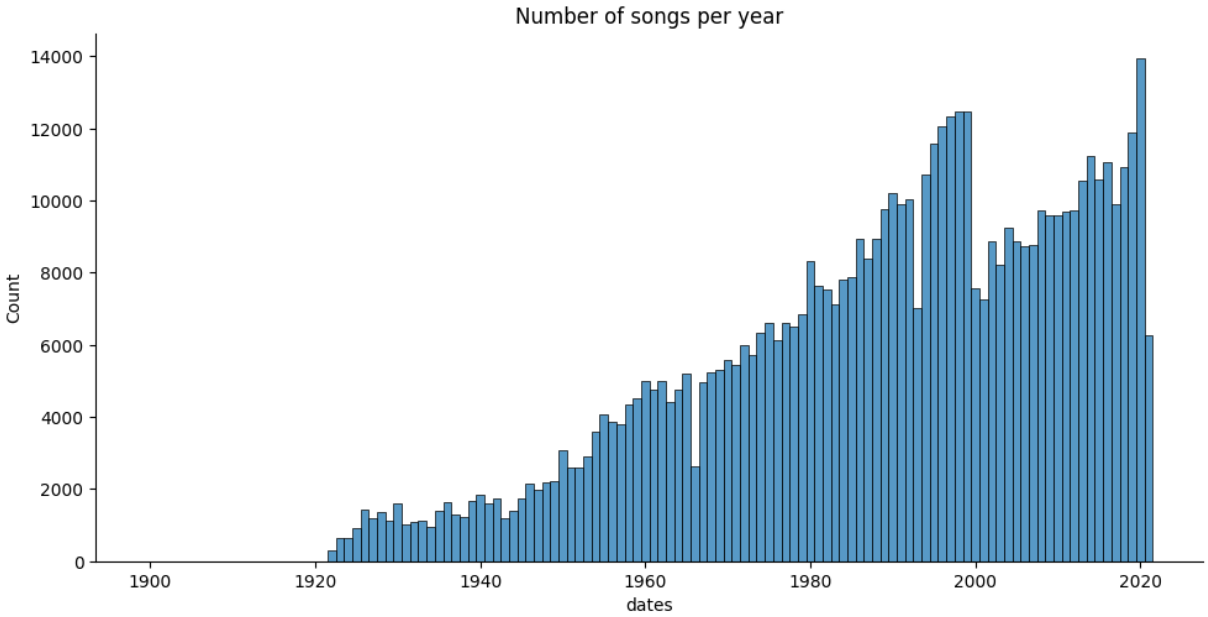
In [100…
```
df_tracks['dates']=df_tracks.index.get_level_values('release_date')
df_tracks.dates=pd.to_datetime(df_tracks.dates)
years=df_tracks.dates.dt.year
```

Distribution plot (histogram) for total songs each year since 1922

In [102…
```
#distribution plot (histogram) for total songs each year since 1922 that is
sns.displot(years, discrete=True, aspect=2, height=5, kind="hist").set(title
```

```
C:\Users\Mehtab K Sidhu\AppData\Local\Programs\Python\Python311\Lib\site-pack
ages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to t
ight
  self._figure.tight_layout(*args, **kwargs)
```

Out[102]: <seaborn.axisgrid.FacetGrid at 0x2988ed0ac90>



Bar plot for duration of songs over the years

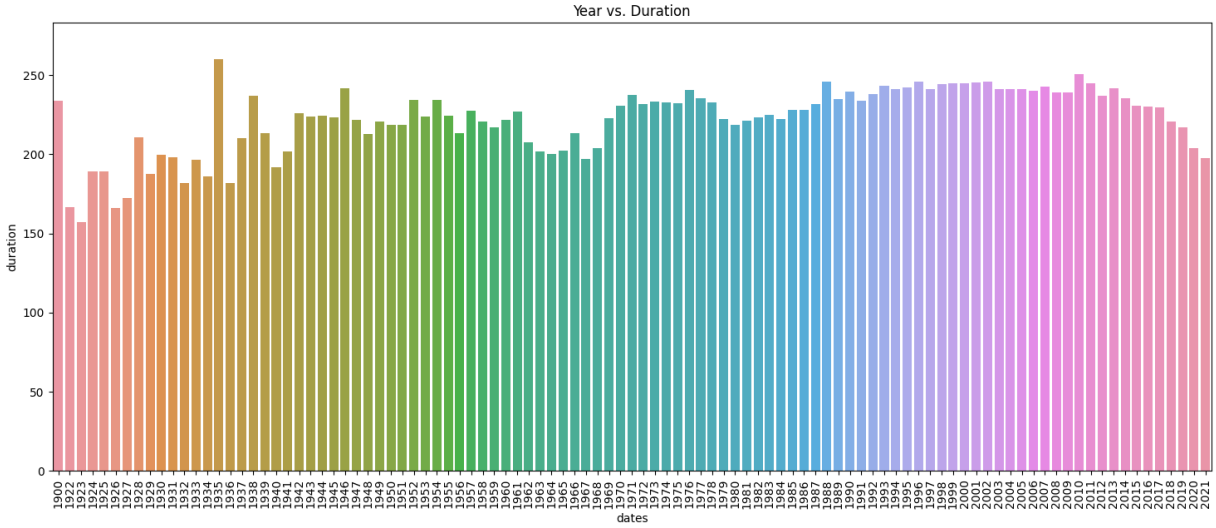
In [103…

```
#bar plot for duration of songs over the years

total_dr = df_tracks.duration
fig_dims = (18, 7)
fig, ax = plt.subplots(figsize = fig_dims)
fig = sns.barplot(x = years, y = total_dr, ax = ax, errwidth = False).set(ti
plt.xticks(rotation=90)
```

```
Out[103]: (array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
                   13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
                   26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
                   39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
                   52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
                   65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
                   78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
                   91,  92,  93,  94,  95,  96,  97,  98,  99, 100]),
           [Text(0, 0, '1900'),
            Text(1, 0, '1922'),
            Text(2, 0, '1923'),
            Text(3, 0, '1924'),
            Text(4, 0, '1925'),
            Text(5, 0, '1926'),
            Text(6, 0, '1927'),
            Text(7, 0, '1928'),
            Text(8, 0, '1929'),
            Text(9, 0, '1930'),
            Text(10, 0, '1931'),
            Text(11, 0, '1932'),
            Text(12, 0, '1933'),
            Text(13, 0, '1934'),
            Text(14, 0, '1935'),
            Text(15, 0, '1936'),
            Text(16, 0, '1937'),
            Text(17, 0, '1938'),
            Text(18, 0, '1939'),
            Text(19, 0, '1940'),
            Text(20, 0, '1941'),
            Text(21, 0, '1942'),
            Text(22, 0, '1943'),
            Text(23, 0, '1944'),
            Text(24, 0, '1945'),
            Text(25, 0, '1946'),
            Text(26, 0, '1947'),
            Text(27, 0, '1948'),
            Text(28, 0, '1949'),
            Text(29, 0, '1950'),
            Text(30, 0, '1951'),
            Text(31, 0, '1952'),
            Text(32, 0, '1953'),
            Text(33, 0, '1954'),
            Text(34, 0, '1955'),
            Text(35, 0, '1956'),
            Text(36, 0, '1957'),
            Text(37, 0, '1958'),
            Text(38, 0, '1959'),
            Text(39, 0, '1960'),
            Text(40, 0, '1961'),
            Text(41, 0, '1962'),
            Text(42, 0, '1963'),
            Text(43, 0, '1964'),
            Text(44, 0, '1965'),
            Text(45, 0, '1966'),
            Text(46, 0, '1967'),
            Text(47, 0, '1968'),
```

```
Text(48, 0, '1969'),
Text(49, 0, '1970'),
Text(50, 0, '1971'),
Text(51, 0, '1972'),
Text(52, 0, '1973'),
Text(53, 0, '1974'),
Text(54, 0, '1975'),
Text(55, 0, '1976'),
Text(56, 0, '1977'),
Text(57, 0, '1978'),
Text(58, 0, '1979'),
Text(59, 0, '1980'),
Text(60, 0, '1981'),
Text(61, 0, '1982'),
Text(62, 0, '1983'),
Text(63, 0, '1984'),
Text(64, 0, '1985'),
Text(65, 0, '1986'),
Text(66, 0, '1987'),
Text(67, 0, '1988'),
Text(68, 0, '1989'),
Text(69, 0, '1990'),
Text(70, 0, '1991'),
Text(71, 0, '1992'),
Text(72, 0, '1993'),
Text(73, 0, '1994'),
Text(74, 0, '1995'),
Text(75, 0, '1996'),
Text(76, 0, '1997'),
Text(77, 0, '1998'),
Text(78, 0, '1999'),
Text(79, 0, '2000'),
Text(80, 0, '2001'),
Text(81, 0, '2002'),
Text(82, 0, '2003'),
Text(83, 0, '2004'),
Text(84, 0, '2005'),
Text(85, 0, '2006'),
Text(86, 0, '2007'),
Text(87, 0, '2008'),
Text(88, 0, '2009'),
Text(89, 0, '2010'),
Text(90, 0, '2011'),
Text(91, 0, '2012'),
Text(92, 0, '2013'),
Text(93, 0, '2014'),
Text(94, 0, '2015'),
Text(95, 0, '2016'),
Text(96, 0, '2017'),
Text(97, 0, '2018'),
Text(98, 0, '2019'),
Text(99, 0, '2020'),
Text(100, 0, '2021')])
```

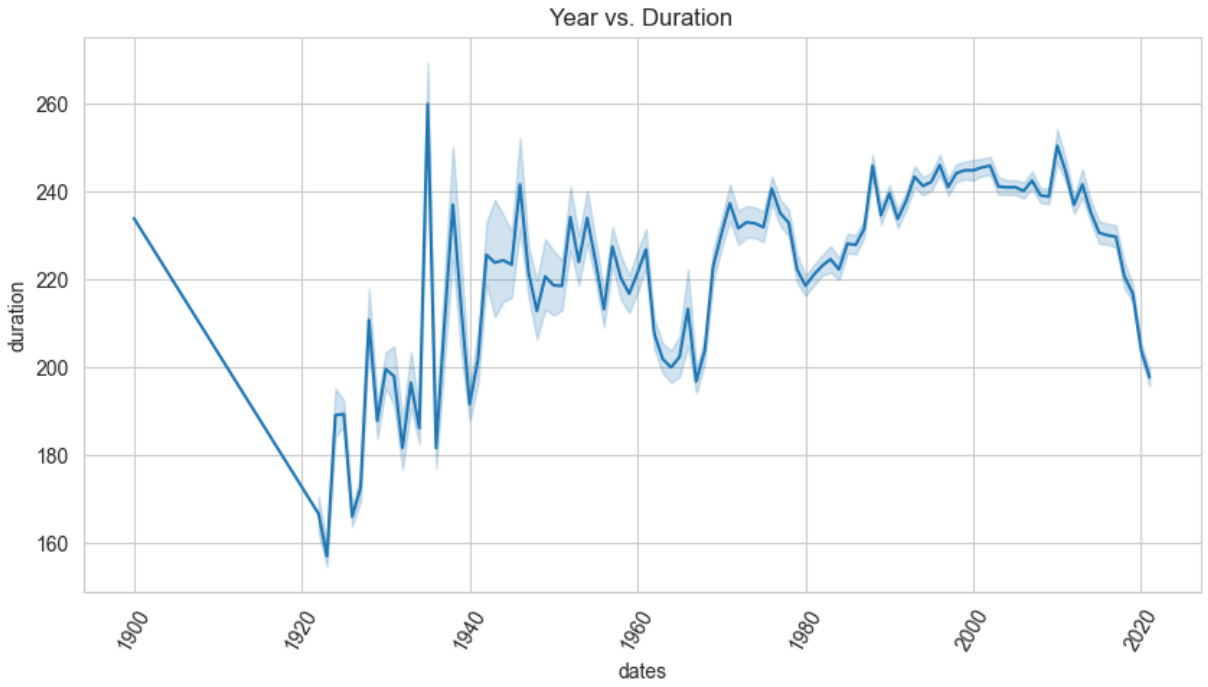


Line plot for average duration of songs over the years

In [108…
```
#line plot for average duration of songs over the years

total_dr = df_tracks.duration
sns.set_style(style="whitegrid")
fig_dims = (10, 5)
fig, ax = plt.subplots(figsize=fig_dims)
fig=sns.lineplot(x=years, y=total_dr, ax=ax).set(title="Year vs. Duration")
plt.xticks(rotation = 60)
```

Out[108]:
```
(array([1880., 1900., 1920., 1940., 1960., 1980., 2000., 2020., 2040.]),
 [Text(1880.0, 0, '1880'),
  Text(1900.0, 0, '1900'),
  Text(1920.0, 0, '1920'),
  Text(1940.0, 0, '1940'),
  Text(1960.0, 0, '1960'),
  Text(1980.0, 0, '1980'),
  Text(2000.0, 0, '2000'),
  Text(2020.0, 0, '2020'),
  Text(2040.0, 0, '2040')])
```



Uploading CSV file SpotifyFeatures.csv

In [109…
```
df_genre = pd.read_csv('E:/Projects/Data Analysis/SpotifyFeatures.csv')
```

.head()

In [110…
```
df_genre.head()
```

Out[110]:

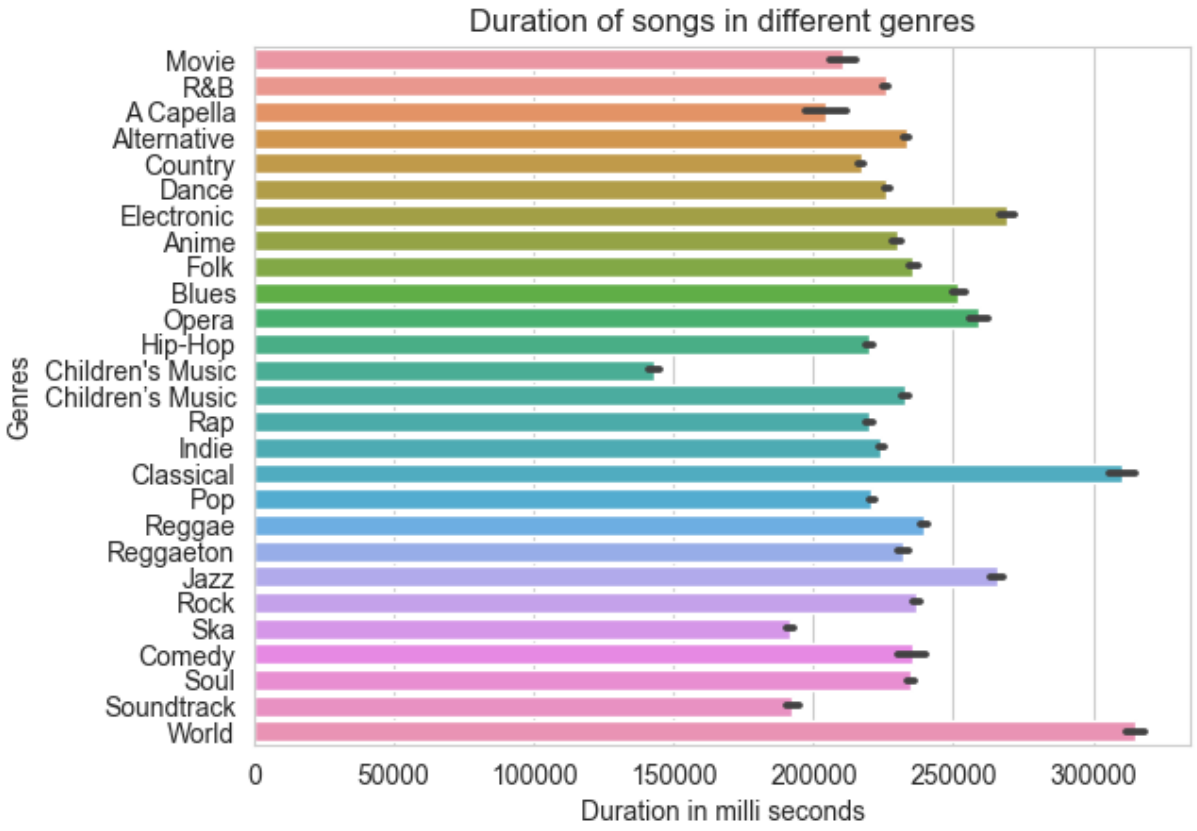
	genre	artist_name	track_name	track_id	popularity	acousticness
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.611
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	1	0.246
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.952
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tIvf	0	0.703
4	Movie	Fabien Nataf	Ouverture	0IuslXpMROHdEPvSl1fTQK	4	0.950

Bar plot for duration of songs for different genres

In [111…
```
#duration of songs for different genres

plt.title("Duration of songs in different genres")
sns.color_palette("rocket", as_cmap=True)
sns.barplot(y='genre', x='duration_ms', data=df_genre)
plt.xlabel("Duration in milli seconds")
plt.ylabel("Genres")
```

Out[111]: Text(0, 0.5, 'Genres')



Top five genres by popularity

In [112…
```
#top five genres by popularity

sns.set_style(style="darkgrid")
plt.figure(figsize=(10, 5))
famous = df_genre.sort_values("popularity", ascending=False).head(10)
#head(10) because some genres are repetitive
sns.barplot(y='genre', x='popularity', data=famous).set(title="Top 5 genres
```

Out[112]: [Text(0.5, 1.0, 'Top 5 genres by popularity')]