

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: df_tracks = pd.read_csv("tracks.csv")
df_tracks.head()
```

```
Out[2]:
```

	id	name	popularity	duration_ms	explicit	artists	id_artists	re
0	35iwgR4jXetl318WEWsa1Q	Carve	6	126903	0	['Uli']	['45tlt06Xol0lio4LBEVpls']	
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']	['14jtPCOoNZwqk5wd9DxrY']	
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']	['5LiOoJbxVSAMkBS2fUm3X2']	
3	08FmqUhxytLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']	['5LiOoJbxVSAMkBS2fUm3X2']	
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0	['Dick Haymes']	['3BiJGZsyX9sJchTqcSA7Su']	

Finding Null values

```
In [3]: pd.isnull(df_tracks).sum()
```

```
Out[3]: 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
```

```
In [4]: df_tracks.info();
```

```
<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
```

```
In [5]: sorted_df = df_tracks.sort_values('popularity',ascending=False).head(10)
sorted_df
```

Out[5]:

	id	name	popularity	duration_ms	explicit	artists	id_artist
546130	181rTRhCcggZPwP2TUcVqm	Newspaper Reports On Abner, 20 February 1935	0	896575	0	['Norris Goff', 'Chester Lauck', 'Carlton Bric...	['3WCwCPDMpGzrt0Qz6quumy '7vk8UqABg0Sga78GI3.
546222	0yOCz3V5KMm8l1T8EFc60i	恋は水の上で	0	188440	0	['Hibari Misora']	['1m5pMY5blqJwdxJ7vqQtuN
546221	0y48Hhwe52099UqYjegRCO	私の誕生日	0	173467	0	['Hibari Misora']	['1m5pMY5blqJwdxJ7vqQtuN
546220	0xCmgtf9ka07hkZg3D6PaV	エル・チヨクロ (EL CHOCLO)	0	205280	0	['Hibari Misora']	['1m5pMY5blqJwdxJ7vqQtuN
546219	0tBXS3VuCPX7KWUFH2nros	恋は不思議なもの	0	185733	0	['Hibari Misora']	['1m5pMY5blqJwdxJ7vqQtuN
546218	0qrKnQtYDVJhKFAXTHYVS9	ゆうべはどうしたの (WHATSA MALLA U)	0	183427	0	['Hibari Misora']	['1m5pMY5blqJwdxJ7vqQtuN
546217	0nqsDxOeKSWEzp3AUQAAqS	Screen Director's Playhouse, Music For Million...	0	1767071	0	['Wilms Herbert', 'June Allyson', 'Joseph Kear...	['2rbm8QWvmnVwxFo84EVM1h '4yW5adMgylfHFzaL9i.
546216	0kGEdsxVLYjCdfxM9tbezD	ブルーマンボ	0	162147	0	['Hibari Misora']	['1m5pMY5blqJwdxJ7vqQtuN
546215	0bc3PUZurUUXrY7yqoOxjq	Screen Director's Playhouse, Trade Winds direc...	0	1776652	0	['Wally Maher', 'Tay Garnett', 'Lurene Tuttle'...	['7hkhJTTI3VnUGVWUt8SJXT '3kYeelpRCgJz4fQYDv.
546214	0Wwm0ruSJYMIiWG0nyAl1F	Screen Director's Playhouse, It's A Wonderful ...	0	1767576	0	['Joseph Granby', 'Jimmy Stewart', 'Irene Tedr...	['6GK59BC4LJzqR0OpHAX2S3 '58BzBaExnrnx898sby.

In [6]: df_tracks.describe().transpose()

Out[6]:

	count	mean	std	min	25%	50%	75%	max
popularity	586672.0	27.570053	18.370642	0.0	13.0000	27.000000	41.00000	100.000
duration_ms	586672.0	230051.167286	126526.087418	3344.0	175093.0000	214893.000000	263867.00000	5621218.000
explicit	586672.0	0.044086	0.205286	0.0	0.0000	0.000000	0.00000	1.000
danceability	586672.0	0.563594	0.166103	0.0	0.4530	0.577000	0.68600	0.991
energy	586672.0	0.542036	0.251923	0.0	0.3430	0.549000	0.74800	1.000
key	586672.0	5.221603	3.519423	0.0	2.0000	5.000000	8.00000	11.000
loudness	586672.0	-10.206067	5.089328	-60.0	-12.8910	-9.243000	-6.48200	5.376
mode	586672.0	0.658797	0.474114	0.0	0.0000	1.000000	1.00000	1.000
speechiness	586672.0	0.104864	0.179893	0.0	0.0340	0.044300	0.07630	0.971
acousticness	586672.0	0.449863	0.348837	0.0	0.0969	0.422000	0.78500	0.996
instrumentalness	586672.0	0.113451	0.266868	0.0	0.0000	0.000024	0.00955	1.000
liveness	586672.0	0.213935	0.184326	0.0	0.0983	0.139000	0.27800	1.000
valence	586672.0	0.552292	0.257671	0.0	0.3460	0.564000	0.76900	1.000
tempo	586672.0	118.464857	29.764108	0.0	95.6000	117.384000	136.32100	246.381
time_signature	586672.0	3.873382	0.473162	0.0	4.0000	4.000000	4.00000	5.000

```
In [7]: most_popular=df_tracks.query('popularity>90',inplace=False).sort_values('popularity',ascending=False)
most_popular[:10]
```

Out[7]:

		id	name	popularity	duration_ms	explicit	artists	id_artists
93802	4iJyoBOLtHqaGxP12qzhQl		Peaches (feat. Daniel Caesar & Giveon)	100	198082	1	['Justin Bieber', 'Daniel Caesar', 'Giveon']	['1uNFoZAHBGtllmzznpCI3s', '20wkVLutqVOYrc0kxF...']
93803	7IPN2DXiMsVn7XUKtOW1CS		drivers license	99	242014	1	['Olivia Rodrigo']	['1McMsnEElThX1knmY4oliG']
93804	3Ofmpyvhv5UAQ70mENzB277		Astronaut In The Ocean	98	132780	0	['Masked Wolf']	['1uU7g3DNSbsu0QjSEqZtEd']
92810	5QO79kh1waicV47BqGRL3g		Save Your Tears	97	215627	1	['The Weeknd']	['1Xyo4u8uXC1ZmMpatF05PJ']
92811	6tDDoYlxWvMLTdKpjFkc1B		telepatía	97	160191	0	['Kali Uchis']	['1U1e13k54VvEUzo3ybLPIM']
92813	0VjjW4GIUZAMYd2vXMi3b		Blinding Lights	96	200040	0	['The Weeknd']	['1Xyo4u8uXC1ZmMpatF05PJ']
93805	7MAibcTli4lisCtbHKrGMh		Leave The Door Open	96	242096	0	['Bruno Mars', 'Anderson .Paak', 'Silk Sonic']	['0du5cEVh5yTK9QJze8zA0C', '3jK9MiCrA42ILAdMGU...']
92814	6f3Slt0GbA2bPZlZ0aIFXN		The Business	95	164000	0	['Tiësto']	['2o5jDhtHVPPhrJdv3cEQ99Z']
91866	60ynsPSSKe6O3sfwRnIBrf		Streets	94	226987	1	['Doja Cat']	['5cj0lJcoR7YOSnhnX0Po5']
92816	3FAJ6O0NOHQV8Mc5Ri6ENp		Heartbreak Anniversary	94	198371	0	['Giveon']	['4fxd5Ee7UefO4CUXgwJ7IP']

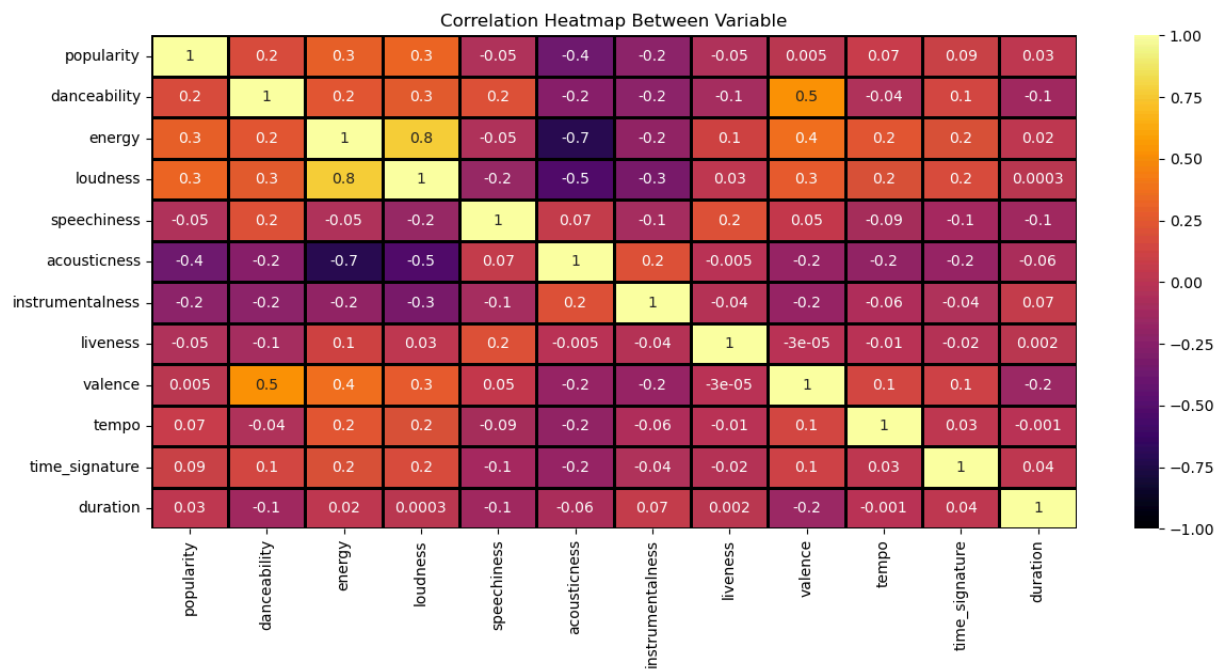
```
In [8]: df_tracks.set_index("release_date",inplace=True)
df_tracks.index=pd.to_datetime(df_tracks.index,format='mixed',dayfirst=True)
df_tracks.head()
```

Out[8]:

		id	name	popularity	duration_ms	explicit	artists	id
release_date								
1922-02-22	35iwgR4jXetl318WEWsa1Q		Carve	6	126903	0	['Uli']	['45tlt06Xol0lio4L']
1922-06-01	021ht4sdgPcrDgSk7JTbKY		Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']	['14jtPCOoNZwquk5wc']
1922-03-21	07A5yehtSnoedViJAZkNnc		Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']	['5LiOoJbxVSAMkBS2fl']
1922-03-21	08FmqUhxyLTn6pAh6bk45		El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']	['5LiOoJbxVSAMkBS2fl']
1922-01-01	08y9GfoqCWfOGsKdwojr5e		Lady of the Evening	0	163080	0	['Dick Haymes']	['3BiJGZsyX9sJchTqc']

```
In [9]: df_tracks['duration']=df_tracks['duration_ms'].apply(lambda x:round(x/1000))
df_tracks.drop('duration_ms',inplace=True,axis=1)
```

```
In [10]: corr_df=df_tracks.drop(['key','mode','explicit'],axis=1).corr(method='pearson',numeric_only=True)
plt.figure(figsize=(14,6))
heatmap=sns.heatmap(corr_df,annot=True,fmt='.1g',vmin=-1,vmax=1,center=0,cmap='inferno',linewidths=1,linecolor='black')
heatmap.set_title('Correlation Heatmap Between Variable')
heatmap.set_xticklabels(heatmap.get_xticklabels(),rotation=90);
```



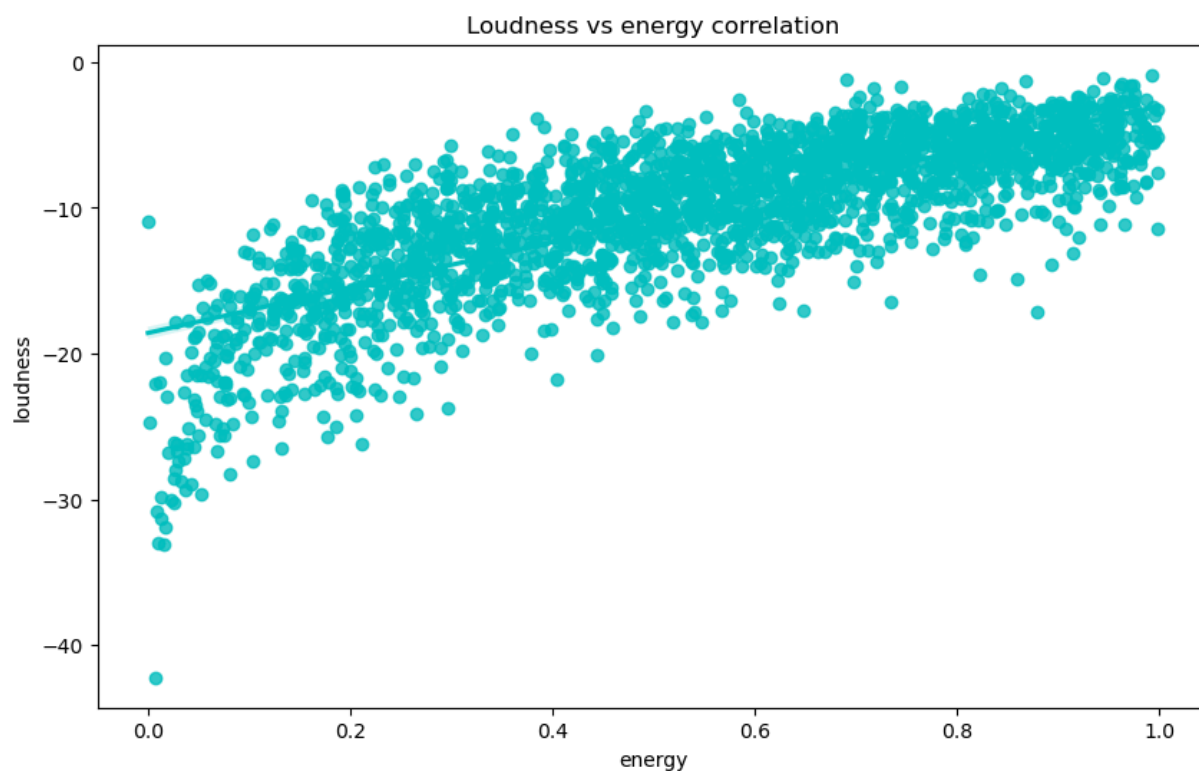
Sampling data

```
In [12]: sample_df =df_tracks.sample(int(0.004*len(df_tracks)))
```

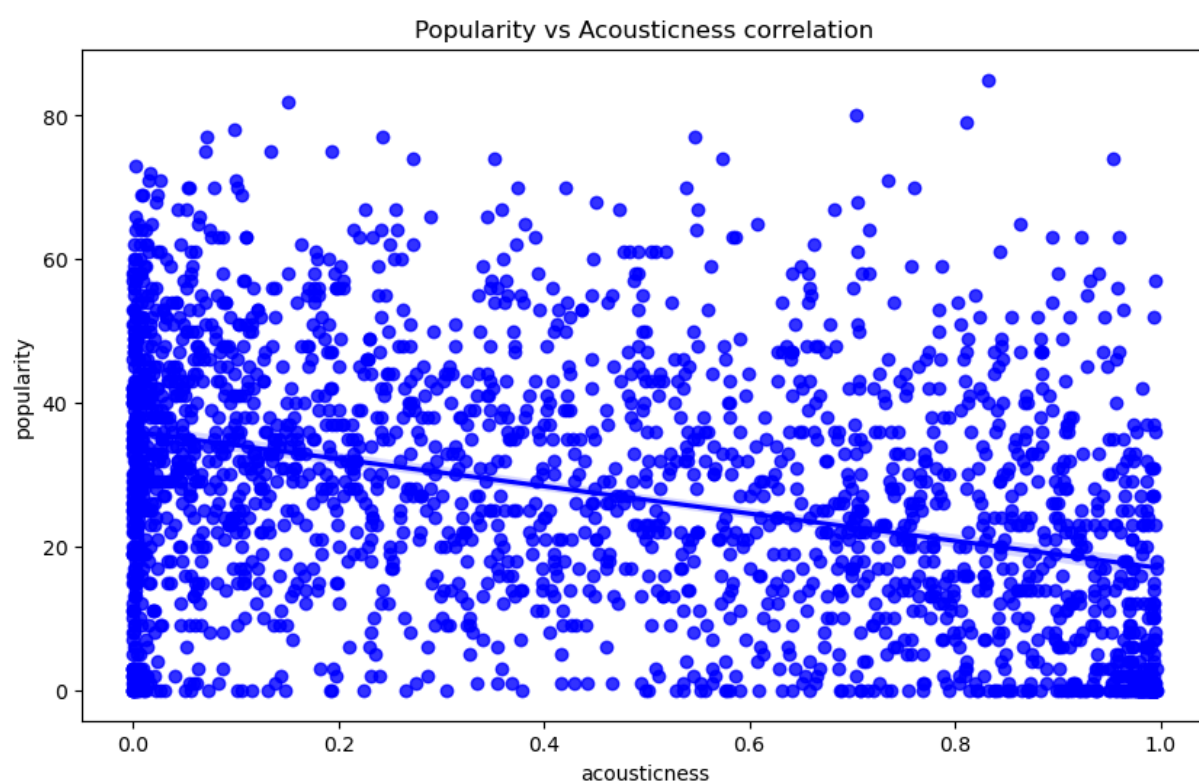
```
In [13]: print(len(sample_df))
```

2346

```
In [14]: plt.figure(figsize=(10,6))
sns.regplot(data=sample_df,y='loudness',x='energy',color='c').set(title="Loudness vs energy correlation");
```

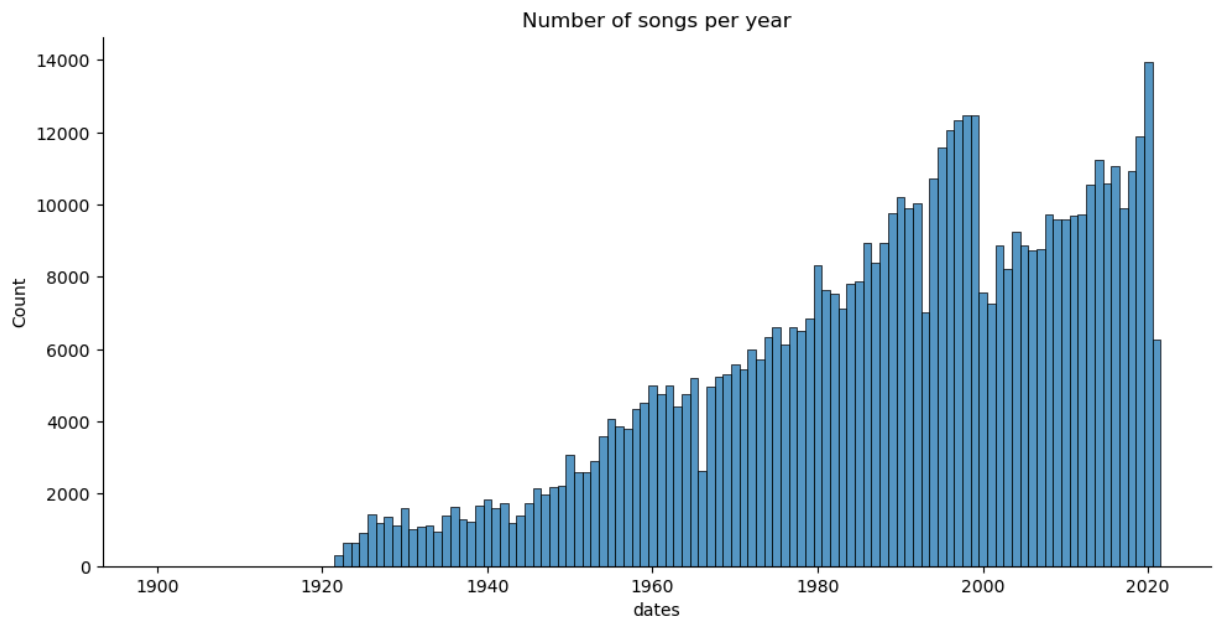


```
In [15]: plt.figure(figsize=(10,6))
sns.regplot(data=sample_df,y='popularity',x='acousticness',color='b').set(title="Popularity vs Acousticness correlation");
```

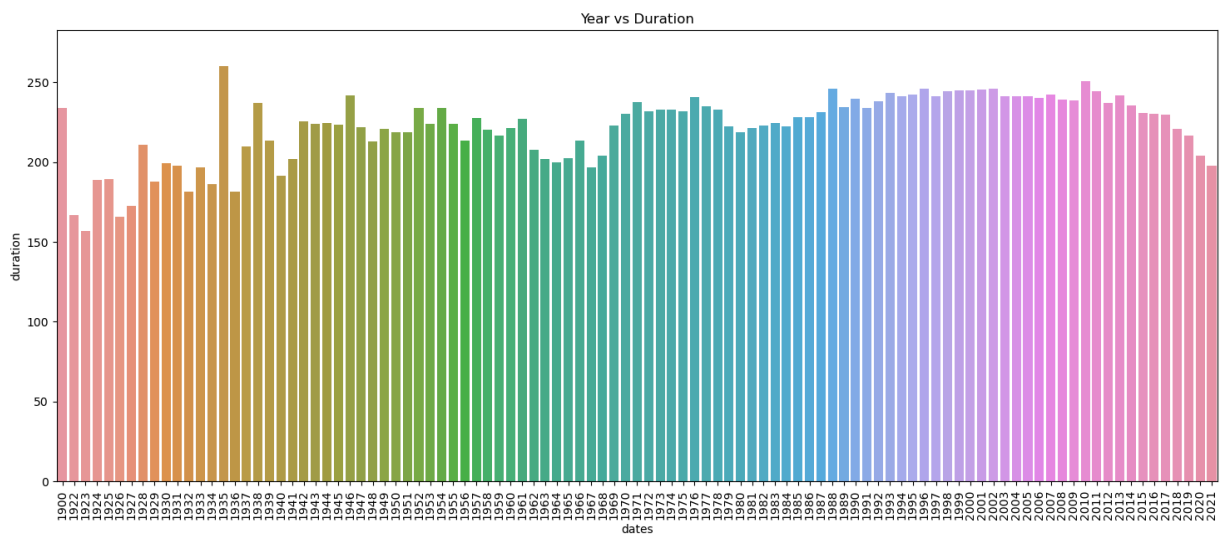


```
In [16]: df_tracks['dates']=df_tracks.index.get_level_values('release_date')
df_tracks.dates=pd.to_datetime(df_tracks.dates)
years=df_tracks.dates.dt.year
```

```
In [17]: sns.displot(years,discrete=True,aspect=2,height=5,kind='hist').set(title='Number of songs per year');
```

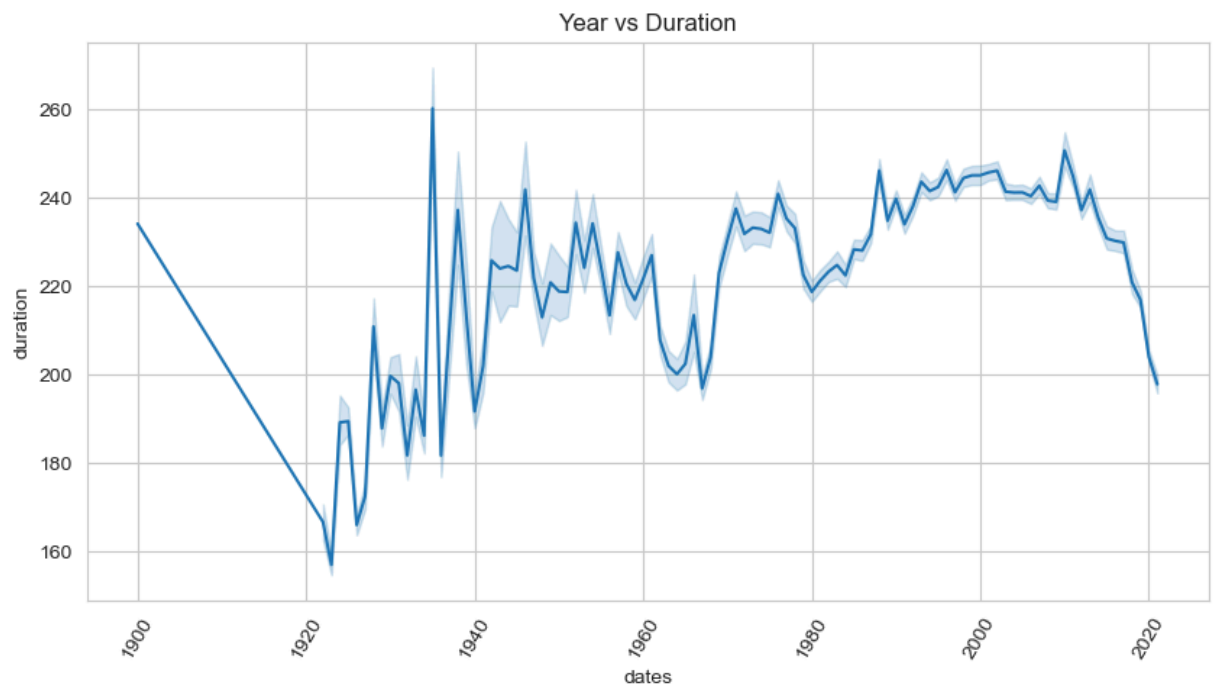


```
In [18]: total_dr=df_tracks.duration
fig_dims=(18,7)
fig,ax=plt.subplots(figsize=fig_dims)
fig=sns.barplot(x=years,y=total_dr,ax=ax,errwidth=False).set(title='Year vs Duration')
plt.xticks(rotation=90);
```



Analysing genres


```
In [19]: 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);
```



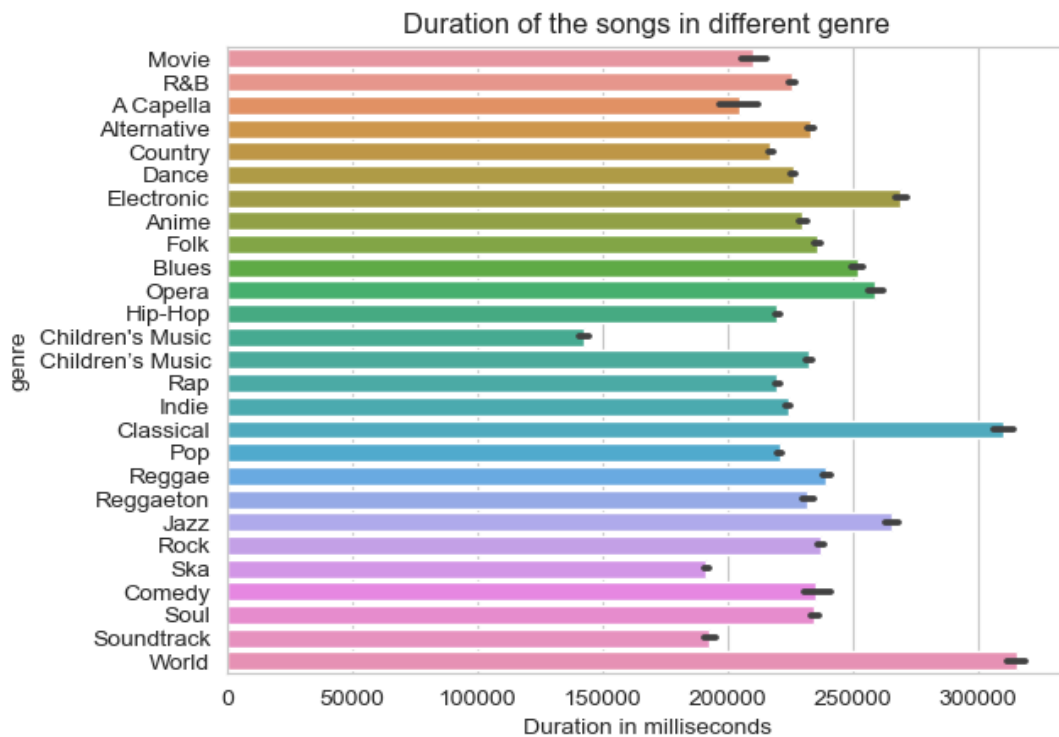
```
In [20]: df_genre=pd.read_csv("SpotifyFeatures.csv")
```

```
In [21]: df_genre.head()
```

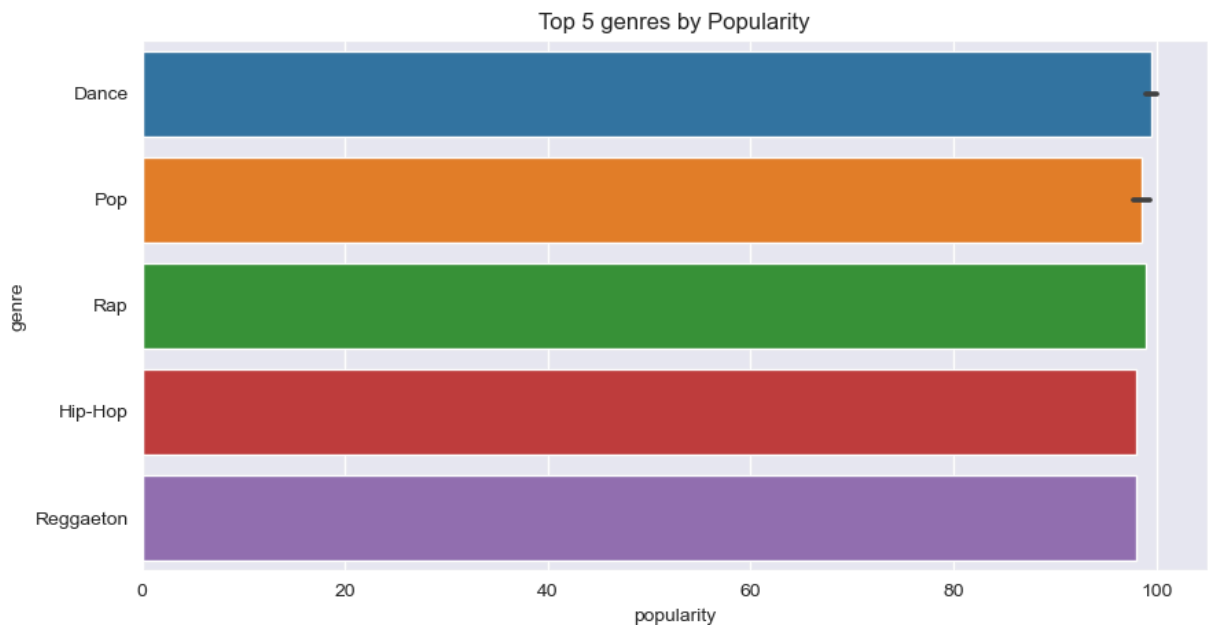
Out[21]:

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	duration_ms	energy
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.611	0.389	99373	0.91
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	1	0.246	0.590	137373	0.75
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.952	0.663	170267	0.15
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tlvf	0	0.703	0.240	152427	0.32
4	Movie	Fabien Nataf	Ouverture	0luslXpMROHdEPvSI1fTQK	4	0.950	0.331	82625	0.22

```
In [22]: plt.title("Duration of the songs in different genre")
sns.color_palette("rocket",as_cmap=True)
sns.barplot(y='genre',x='duration_ms',data=df_genre)
plt.xlabel("Duration in milliseconds")
plt.ylabel('genre');
```



```
In [23]: sns.set_style(style='darkgrid')
plt.figure(figsize=(10,5))
famous=df_genre.sort_values('popularity',ascending=False).head(10)
sns.barplot(y='genre',x='popularity',data=famous).set(title='Top 5 genres by Popularity');
```



In []:

In []:

