In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df_tracks = pd.read_csv('C:/Users/Dell/Desktop/PROJECT/Spotify Python/tracks.csv')
```

In [3]:

```
#null values
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 = True).head(10)
```

In [6]:

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

Out[6]:

	count	mean	std	min	25%	
popularity	586672.0	27.570053	18.370642	0.0	13.0000	27.00(
duration_ms	586672.0	230051.167286	126526.087418	3344.0	175093.0000	214893.00(
explicit	586672.0	0.044086	0.205286	0.0	0.0000	0.00(
danceability	586672.0	0.563594	0.166103	0.0	0.4530	0.57
energy	586672.0	0.542036	0.251923	0.0	0.3430	0.54
key	586672.0	5.221603	3.519423	0.0	2.0000	5.00(
loudness	586672.0	-10.206067	5.089328	-60.0	-12.8910	-9.24
mode	586672.0	0.658797	0.474114	0.0	0.0000	1.00(
speechiness	586672.0	0.104864	0.179893	0.0	0.0340	0.04
acousticness	586672.0	0.449863	0.348837	0.0	0.0969	0.42
instrumentalness	586672.0	0.113451	0.266868	0.0	0.0000	0.00(
liveness	586672.0	0.213935	0.184326	0.0	0.0983	0.13
valence	586672.0	0.552292	0.257671	0.0	0.3460	0.56
tempo	586672.0	118.464857	29.764108	0.0	95.6000	117.38
time_signature	586672.0	3.873382	0.473162	0.0	4.0000	4.00(

In [7]:

```
most_popular = df_tracks.query('popularity>90',inplace = False).sort_values('popular
most_popular[0:10]
```

Out[7]:

	id	name	popularity	duration_ms	explicit	artists
93802	4iJyoBOLtHqaGxP12qzhQI	Peaches (feat. Daniel Caesar & Giveon)	100	198082	1	['Justin Bieber', 'Daniel Caesar', 'Giveon']
93803	7lPN2DXiMsVn7XUKtOW1CS	drivers license	99	242014	1	['Olivia Rodrigo']
93804	3Ofmpyhv5UAQ70mENzB277	Astronaut In The Ocean	98	132780	0	['Masked Wolf']
92810	5QO79kh1waicV47BqGRL3g	Save Your Tears	97	215627	1	['The Weeknd']
92811	6tDDoYlxWvMLTdKpjFkc1B	telepatía	97	160191	0	['Kali Uchis']
92813	0VjIjW4GlUZAMYd2vXMi3b	Blinding Lights	96	200040	0	['The Weeknd']
93805	7MAibcTli4lisCtbHKrGMh	Leave The Door Open	96	242096	0	['Bruno Mars', 'Anderson .Paak', 'Silk Sonic']
92814	6f3Slt0GbA2bPZlz0aIFXN	The Business	95	164000	0	['Tiësto']
91866	60ynsPSSKe6O3sfwRnIBRf	Streets	94	226987	1	['Doja Cat']
92816	3FAJ6O0NOHQV8Mc5Ri6ENp	Heartbreak Anniversary	94	198371	0	['Giveon']

In [8]:

```
df_tracks.set_index("release_date", inplace=True)
df_tracks.index=pd.to_datetime(df_tracks.index)
```

In [9]:

```
df_tracks[["artists"]].iloc[18]
```

Out[9]:

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

In [10]:

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

In [11]:

```
df_tracks.duration.head()
```

Out[11]:

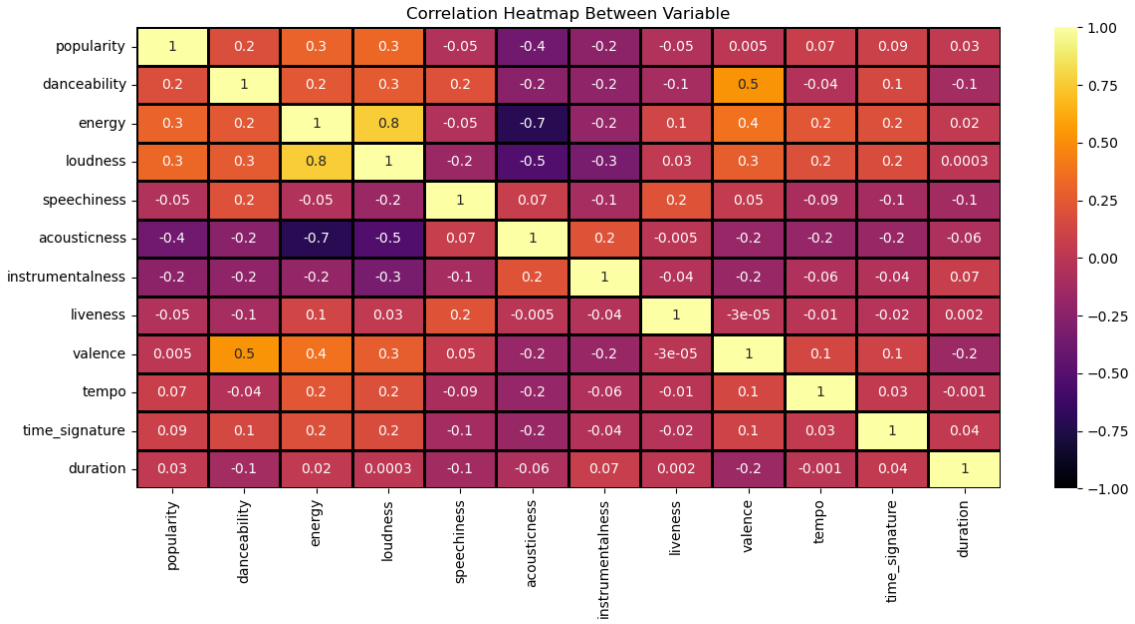
```
release_date
1922-02-22    127
1922-06-01     98
1922-03-21    182
1922-03-21    177
1922-01-01    163
Name: duration, dtype: int64
```

In [12]:

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

Out[12]:

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



In [13]:

```
sample_df=df_tracks.sample(int(0.004*len(df_tracks)))
```

In [14]:

```
print(len(sample_df))
```

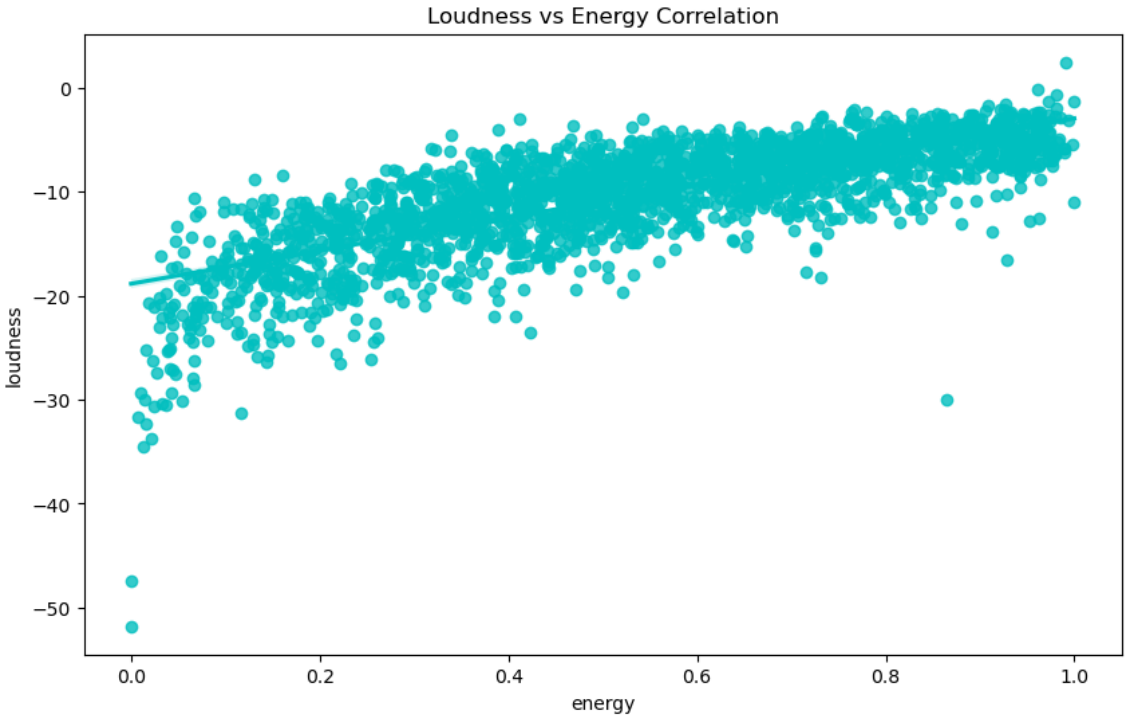
```
2346
```

In [15]:

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

Out[15]:

```
[Text(0.5, 1.0, 'Loudness vs Energy Correlation')]
```

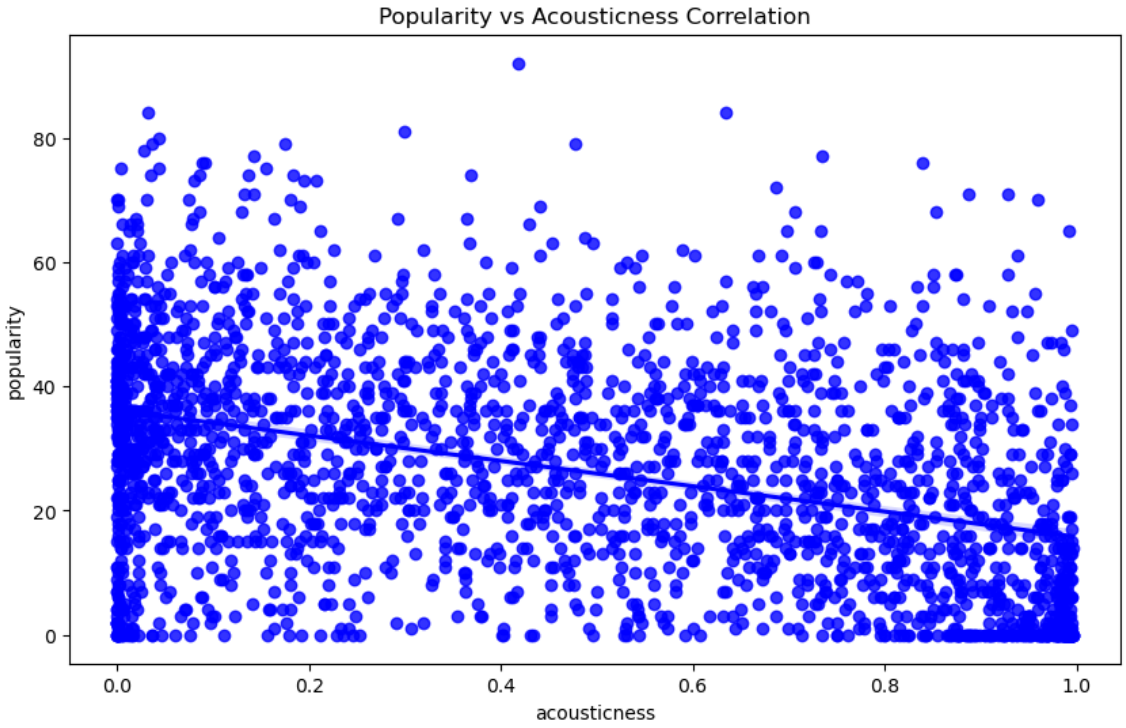


In [16]:

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

Out[16]:

```
[Text(0.5, 1.0, 'Popularity vs Acousticness Correlation')]
```



In [17]:

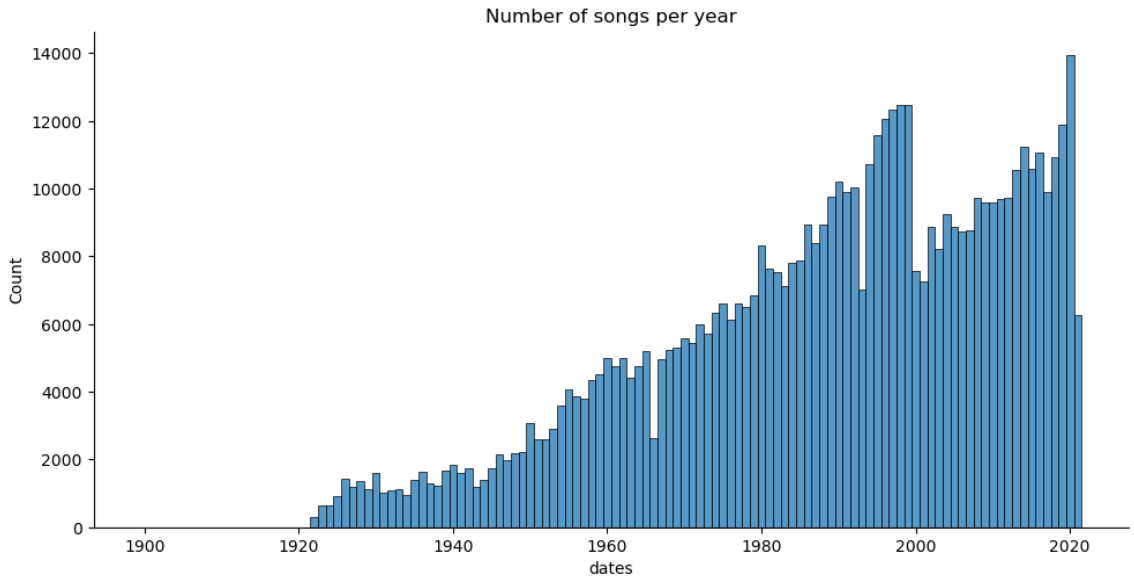
```
df_tracks['dates']=df_tracks.index.get_level_values('release_date')
df_tracks.dates=pd.to_datetime(df_tracks.dates)
years=df_tracks.dates.dt.year
```

In [18]:

```
sns.displot(years,discrete=True,aspect=2,height=5,kind="hist").set(title="Number of s
```

Out[18]:

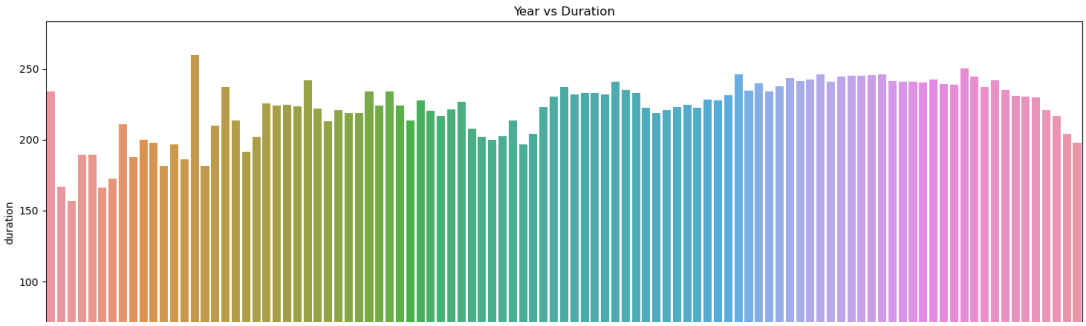
```
<seaborn.axisgrid.FacetGrid at 0x26f3a5d3520>
```



In [19]:

```
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
plt.xticks(rotation=90)
```

```
  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')])
```



In [20]:

```
df_genre=pd.read_csv("C:/Users/Dell/Desktop/PROJECT/Spotify Python/SpotifyFeatures.c
```

In [21]:

```
df_genre.head()
```

Out[21]:

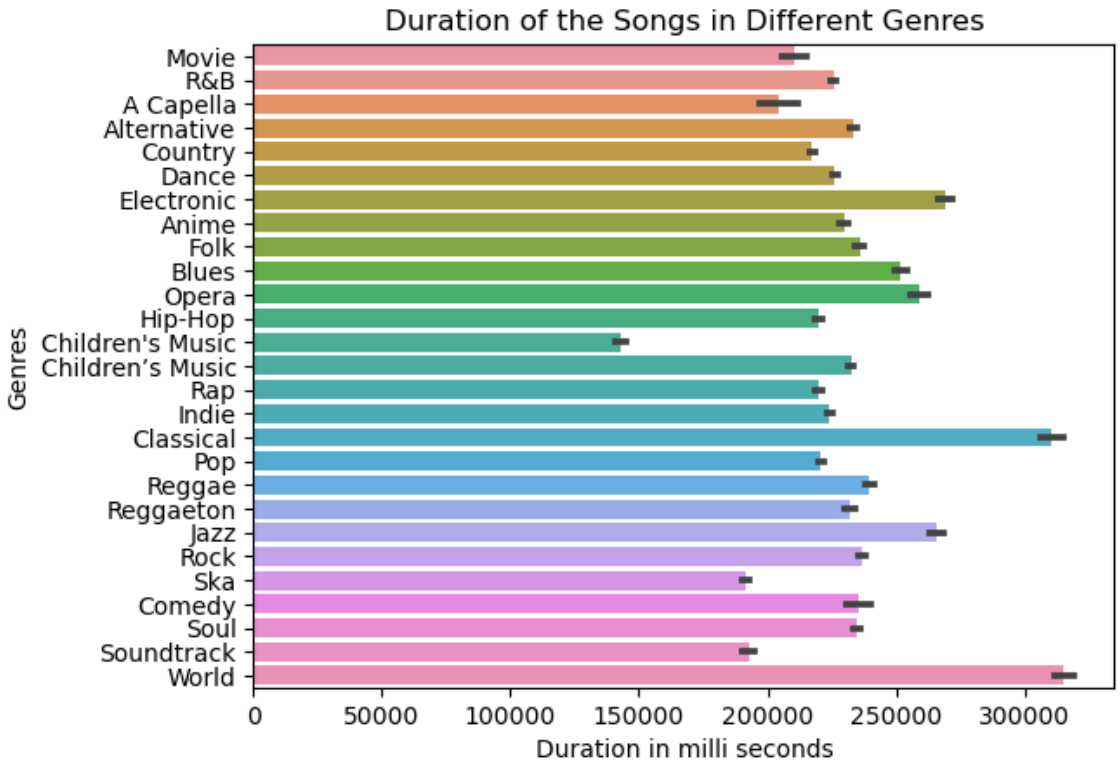
	genre	artist_name	track_name	track_id	popularity	acousticness	dar
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.611	
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	1	0.246	
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.952	
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tIvf	0	0.703	
4	Movie	Fabien Nataf	Ouverture	0IusIXpMROHdEPvSl1fTQK	4	0.950	

In [22]:

```
plt.title("Duration of the 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[22]:

```
Text(0, 0.5, 'Genres')
```



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 Po
```

Out[23]:

```
[Text(0.5, 1.0, 'Top 5 Genres by Popularity')]
```