Importing Libraries

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

Loading Dataset

```
df = pd.read_csv("/content/Spotify.csv")
df. head()
```

	Unnamed: 0	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo
0	0	0.0102	0.833	204600	0.434	0.021900	2	0.1650	-8.795	1	0.4310	150.062
1	1	0.1990	0.743	326933	0.359	0.006110	1	0.1370	-10.401	1	0.0794	160.083
2	2	0.0344	0.838	185707	0.412	0.000234	2	0.1590	-7.148	1	0.2890	75.044
3	3	0.6040	0.494	199413	0.338	0.510000	5	0.0922	-15.236	1	0.0261	86.468
4	4	0.1800	0.678	392893	0.561	0.512000	5	0.4390	-11.648	0	0.0694	174.004

Data Cleaning

```
df.drop("Unnamed: 0", axis = 1, inplace = True)
df.head()
```

	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signa
0	0.0102	0.833	204600	0.434	0.021900	2	0.1650	-8.795	1	0.4310	150.062	
1	0.1990	0.743	326933	0.359	0.006110	1	0.1370	-10.401	1	0.0794	160.083	
2	0.0344	0.838	185707	0.412	0.000234	2	0.1590	-7.148	1	0.2890	75.044	
3	0.6040	0.494	199413	0.338	0.510000	5	0.0922	-15.236	1	0.0261	86.468	
4	0.1800	0.678	392893	0.561	0.512000	5	0.4390	-11.648	0	0.0694	174.004	

Next steps: View recommended plots

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

```
acousticness        0
danceability        0
duration_ms         0
energy              0
instrumentalness    0
key                 0
liveness            0
loudness            0
mode                0
speechiness         0
tempo               0
time_signature      0
valence             0
target              0
song_title          0
artist              0
dtype: int64
```

```
df.shape
```

```
(2017, 16)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2017 entries, 0 to 2016
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   acousticness      2017 non-null   float64
 1   danceability      2017 non-null   float64
 2   duration_ms       2017 non-null   int64  
 3   energy            2017 non-null   float64
 4   instrumentalness  2017 non-null   float64
 5   key               2017 non-null   int64  
 6   liveness          2017 non-null   float64
 7   loudness          2017 non-null   float64
 8   mode              2017 non-null   int64  
 9   speechiness       2017 non-null   float64
 10  tempo             2017 non-null   float64
 11  time_signature    2017 non-null   float64
 12  valence           2017 non-null   float64
 13  target            2017 non-null   int64  
 14  song_title        2017 non-null   object 
 15  artist            2017 non-null   object 
dtypes: float64(10), int64(4), object(2)
memory usage: 252.2+ KB
```

```
df.columns
```

```
Index(['acousticness', 'danceability', 'duration_ms', 'energy',
       'instrumentalness', 'key', 'liveness', 'loudness', 'mode',
       'speechiness', 'tempo', 'time_signature', 'valence', 'target',
       'song_title', 'artist'],
      dtype='object')
```

```
df.describe()
```

	acousticness	danceability	duration_ms	energy	instrumentalness	
count	2017.000000	2017.000000	2.017000e+03	2017.000000	2017.000000	2017.0
mean	0.187590	0.618422	2.463062e+05	0.681577	0.133286	5.3
std	0.259989	0.161029	8.198181e+04	0.210273	0.273162	3.6
min	0.000003	0.122000	1.604200e+04	0.014800	0.000000	0.0
25%	0.009630	0.514000	2.000150e+05	0.563000	0.000000	2.0
50%	0.063300	0.631000	2.292610e+05	0.715000	0.000076	6.0
75%	0.265000	0.738000	2.703330e+05	0.846000	0.054000	9.0
max	0.995000	0.984000	1.004627e+06	0.998000	0.976000	11.0

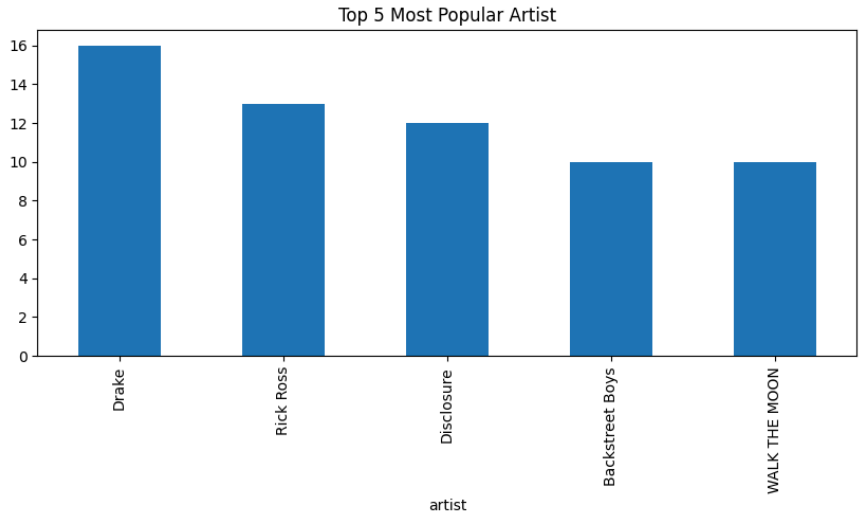
Data Analysis

Top 5 most popular artist

```
top_five_artist = df.groupby("artist").count().sort_values(by ="song_title", ascending = False)["song_title"][:5]
top_five_artist
```

```
artist
Drake              16
Rick Ross          13
Disclosure         12
Backstreet Boys    10
WALK THE MOON      10
Name: song_title, dtype: int64
```

```
top_five_artist.plot(kind='bar', figsize=(10, 4))
plt.title("Top 5 Most Popular Artist")
plt.show()
```



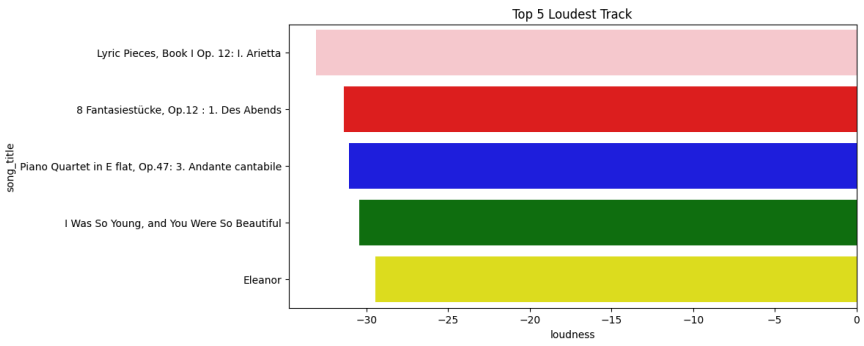
Top 5 loudest track

```
top_five_tracks = df[["song_title","loudness"]].sort_values(by ="loudness", ascending = True)[:5]
top_five_tracks
```

	song_title	loudness
1594	Lyric Pieces, Book I Op. 12: I. Arietta	-33.097
1596	8 Fantasiestücke, Op.12 : 1. Des Abends	-31.367
1598	Piano Quartet in E flat, Op.47: 3. Andante can...	-31.082
1531	I Was So Young, and You Were So Beautiful	-30.447
1549	Eleanor	-29.460

Next steps: View recommended plots

```
plt.figure(figsize = (10,5))
colors = ["pink","red","blue","green","yellow"]
sns.barplot(x="loudness", y="song_title", data=top_five_tracks, hue="song_title", palette=colors, dodge=False, legend=False)
plt.title("Top 5 Loudest Track")
plt.show()
```

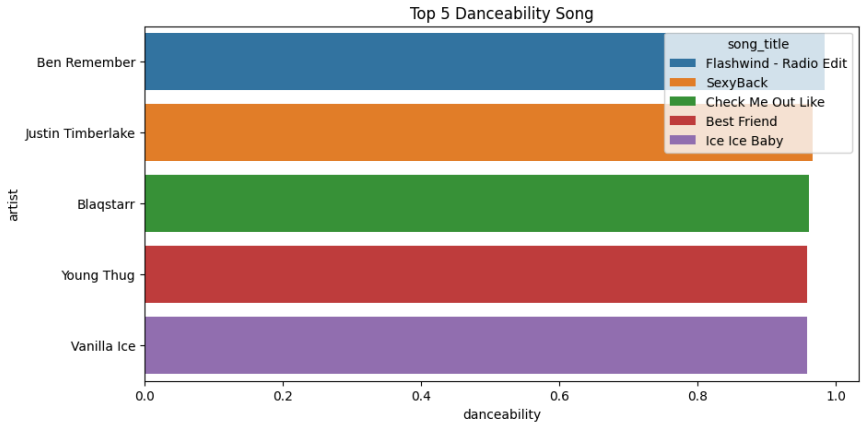


```
top_danceability_song = df[["artist","danceability","song_title"]].sort_values(by = "danceability", ascending = False)[:5]
top_danceability_song
```

	artist	danceability	song_title
1433	Ben Remember	0.984	Flashwind - Radio Edit
1901	Justin Timberlake	0.967	SexyBack
604	Blaqstarr	0.962	Check Me Out Like
32	Young Thug	0.959	Best Friend
1957	Vanilla Ice	0.959	Ice Ice Baby

Next steps: View recommended plots

```
plt.figure(figsize = (10,5))
sns.barplot(x ="danceability", y="artist", data=top_danceability_song, hue = "song_title")
plt.title("Top 5 Danceability Song")
plt.show()
```



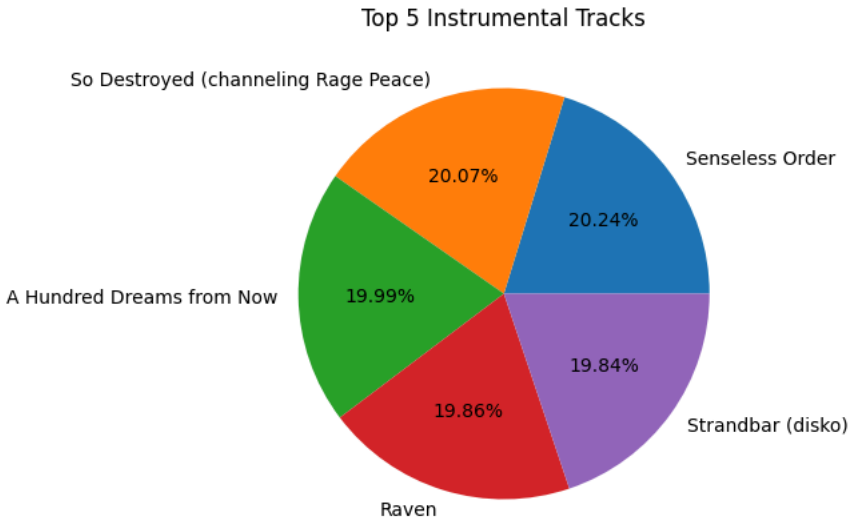
Top 5 instrumental tracks

```
top_ten_instrumental_tracks = df[["instrumentalness", "song_title", "artist"]].sort_values(by = "instrumentalness", ascending = False)[
top_ten_instrumental_tracks
```

	instrumentalness	song_title	artist
1313	0.976	Senseless Order	Signs of the Swarm
271	0.968	So Destroyed (channeling Rage Peace)	Prince Rama
1575	0.964	A Hundred Dreams from Now	Ray Bryant
1619	0.958	Raven	John Dahlbäck
725	0.957	Strandbar (disko)	Todd Terje

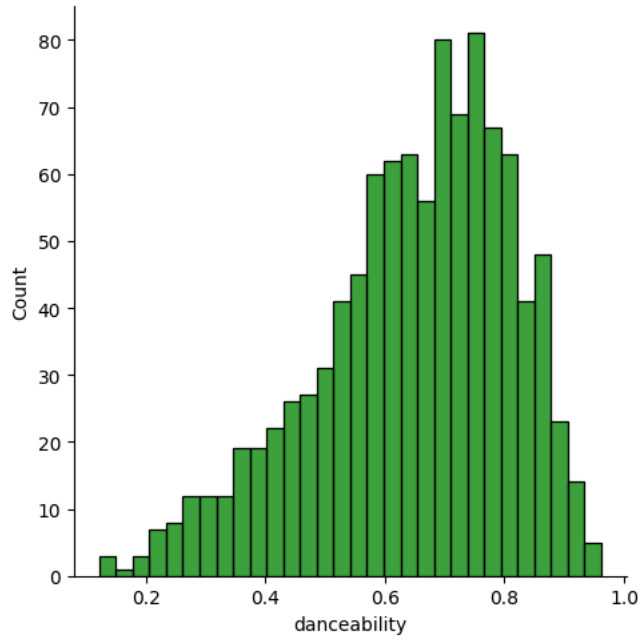
Next steps: View recommended plots

```
plt.figure(figsize = (10,5))
plt.pie(x = "instrumentalness", data = top_ten_instrumental_tracks, labels ="song_title", autopct = "%1.2f%%")
plt.title("Top 5 Instrumental Tracks")
plt.show()
```

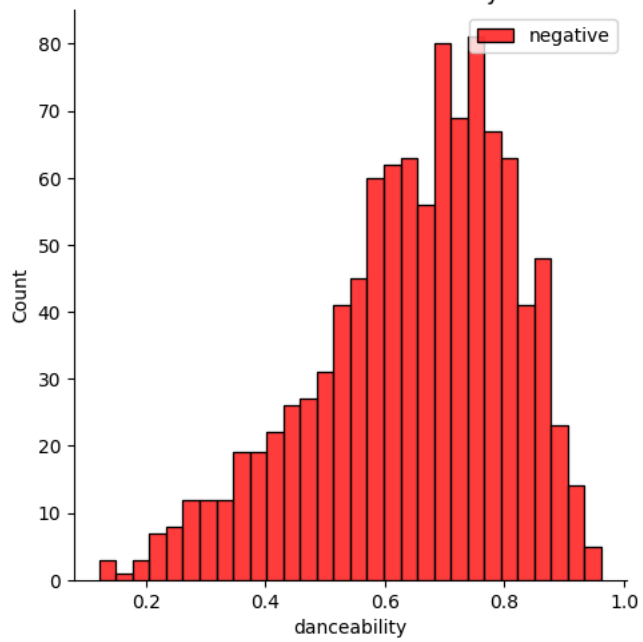



```
interest_feature_cols = ["danceability", "energy", "loudness", "speechiness", "acousticness", "instrumentalness", "liveness", "valence",

for feature_col in interest_feature_cols:
  pos_data = df[df["target"]== 1][feature_col]
  neg_data = df[df["target"]== 1][feature_col]


  plt.figure(figsize= (10,5))
  sns.displot(pos_data, bins = 30, label = "positive", color = "green")
  sns.displot(neg_data, bins = 30, label = "negative", color = "red")
  plt.title(f"Distribution of {feature_col}")

  plt.legend(loc = "upper right")
  plt.show()
```

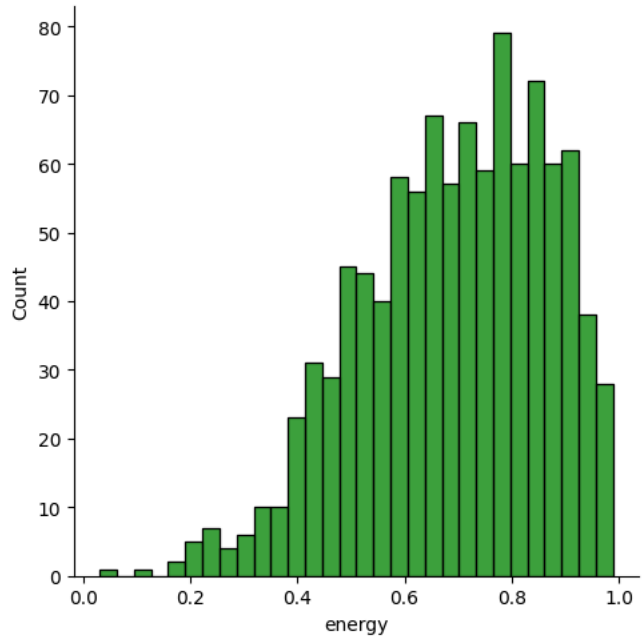
```
<Figure size 1000x500 with 0 Axes>
```



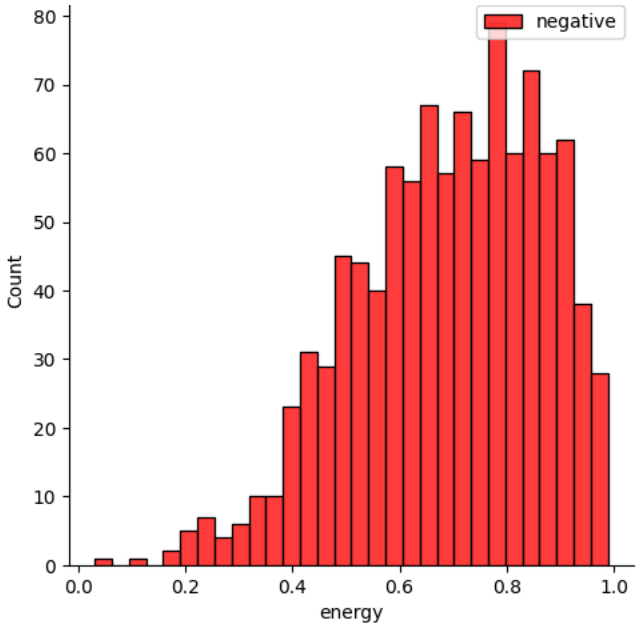
Distribution of danceability



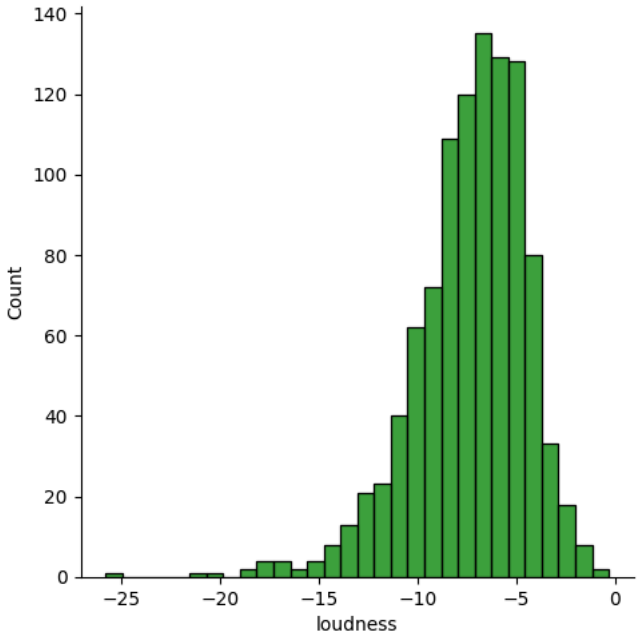

```
<Figure size 1000x500 with 0 Axes>
```

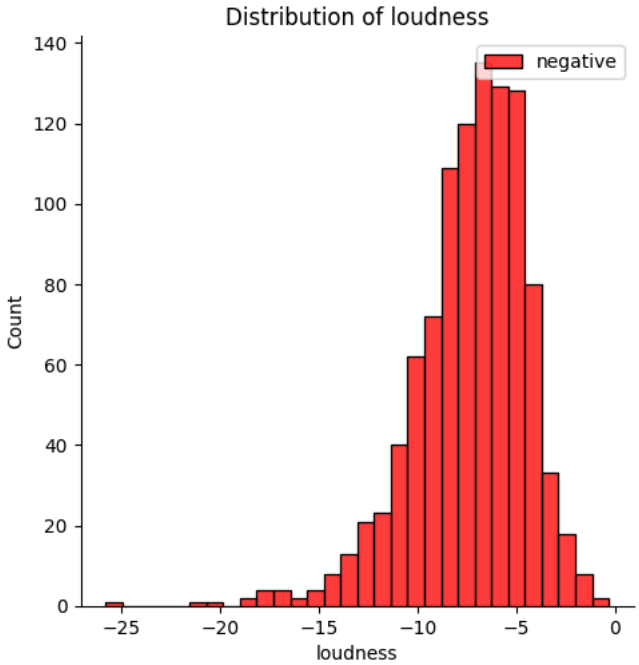


Distribution of energy

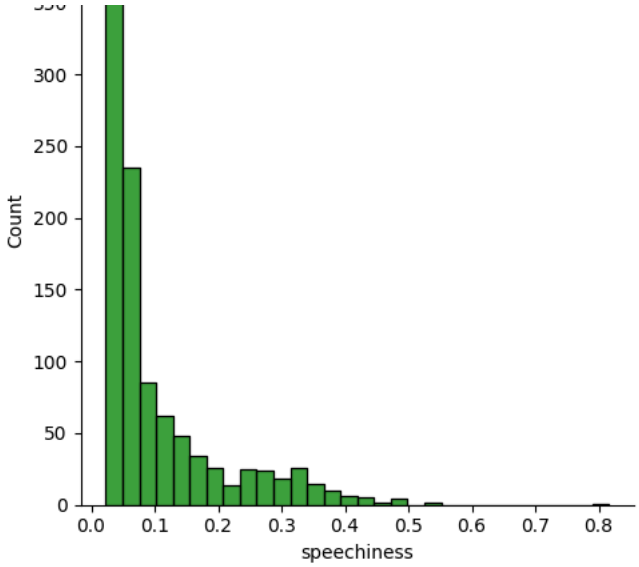



```
<Figure size 1000x500 with 0 Axes>
```

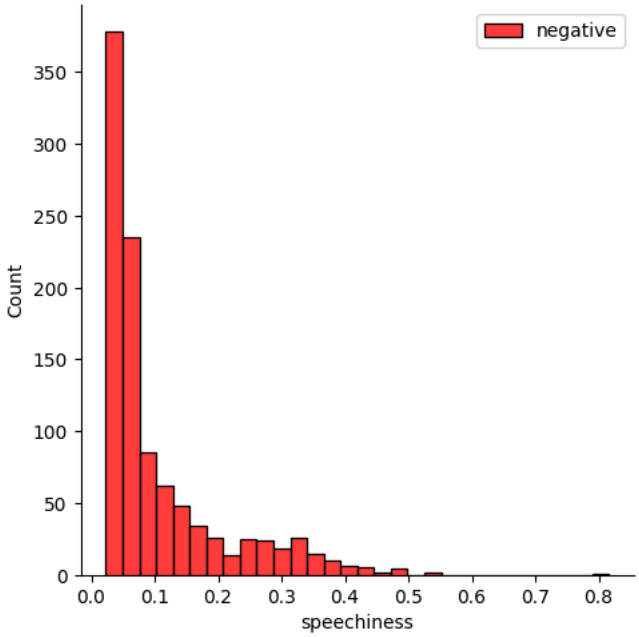




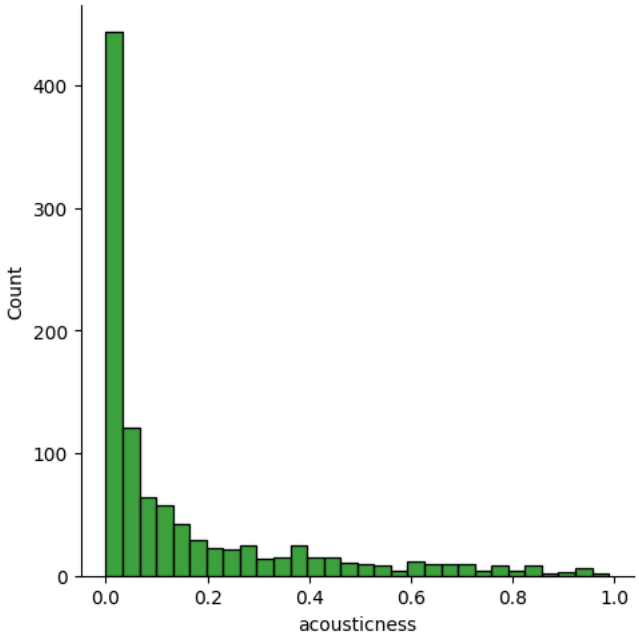

```
<Figure size 1000x500 with 0 Axes>
```



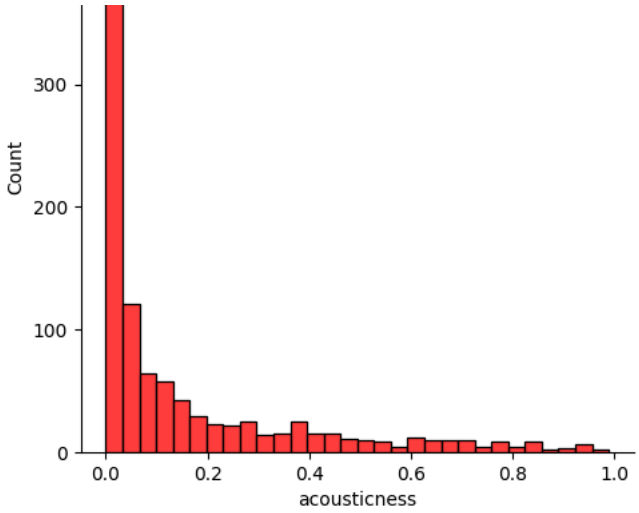
Distribution of speechiness



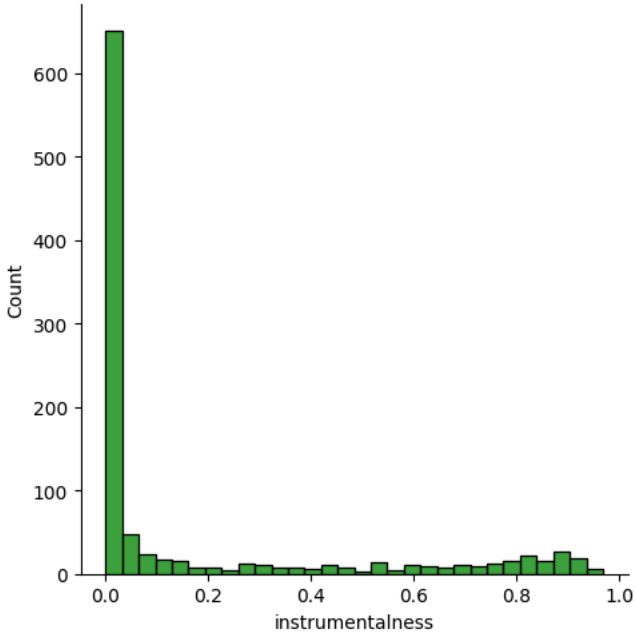

```
<Figure size 1000x500 with 0 Axes>
```

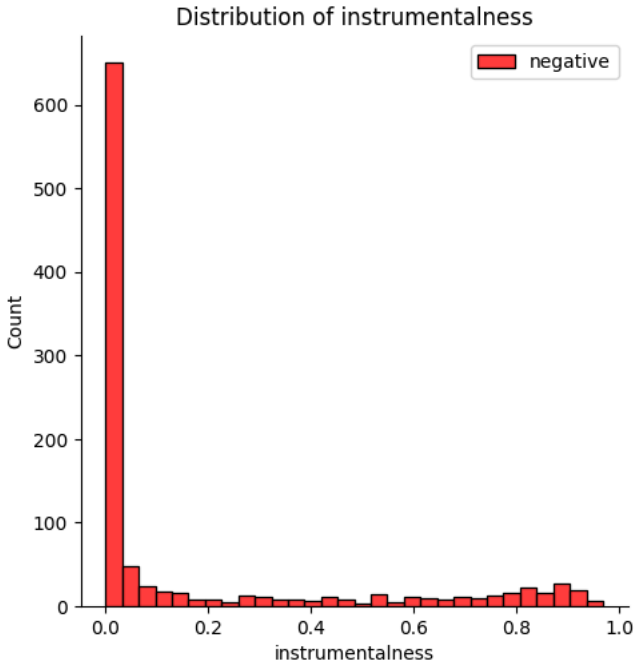


Distribution of acousticness

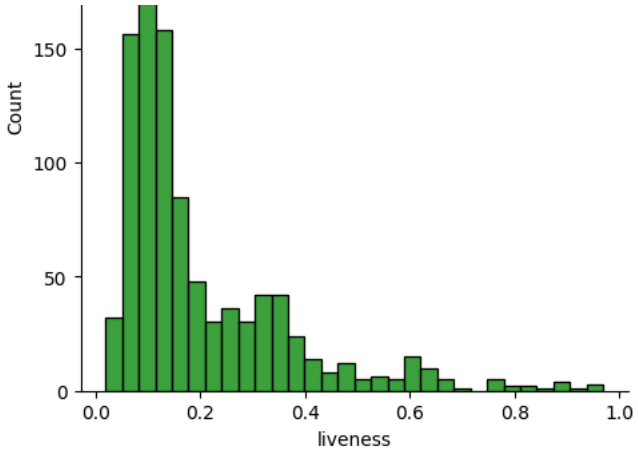


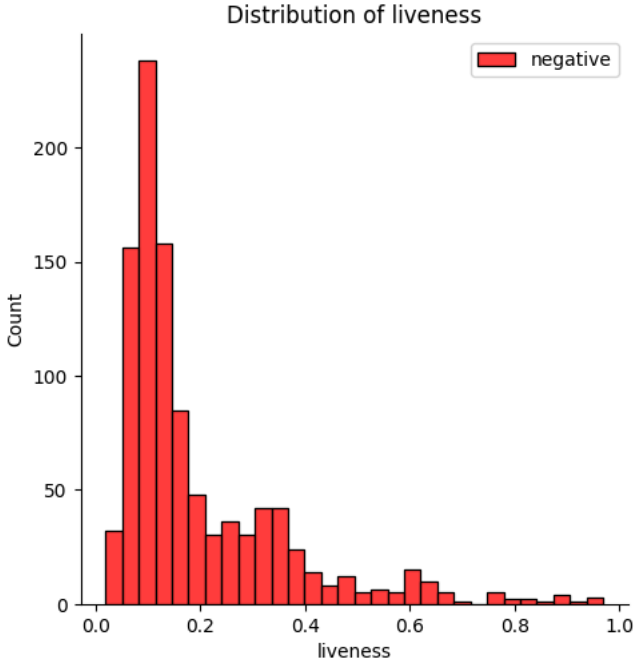

```
<Figure size 1000x500 with 0 Axes>
```



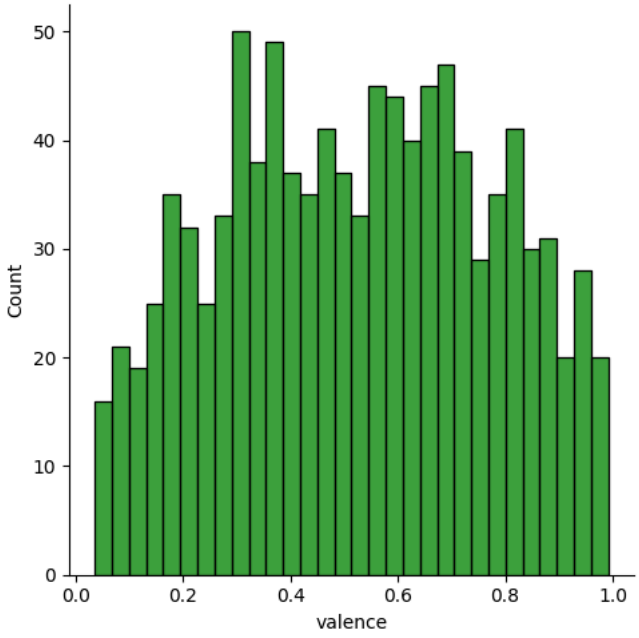


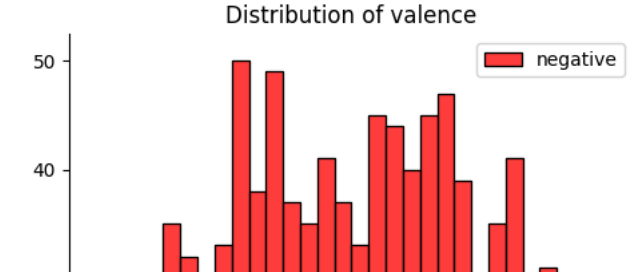

```
<Figure size 1000x500 with 0 Axes>
```

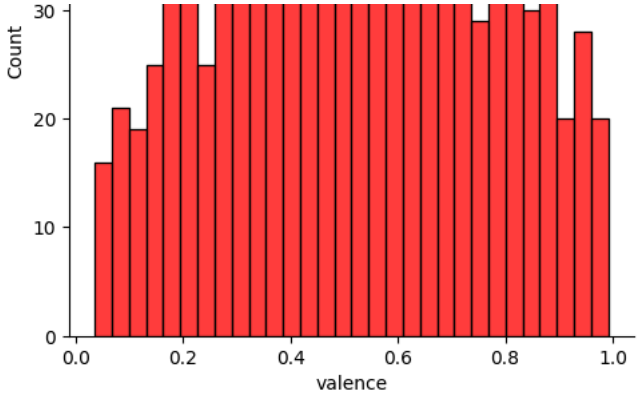




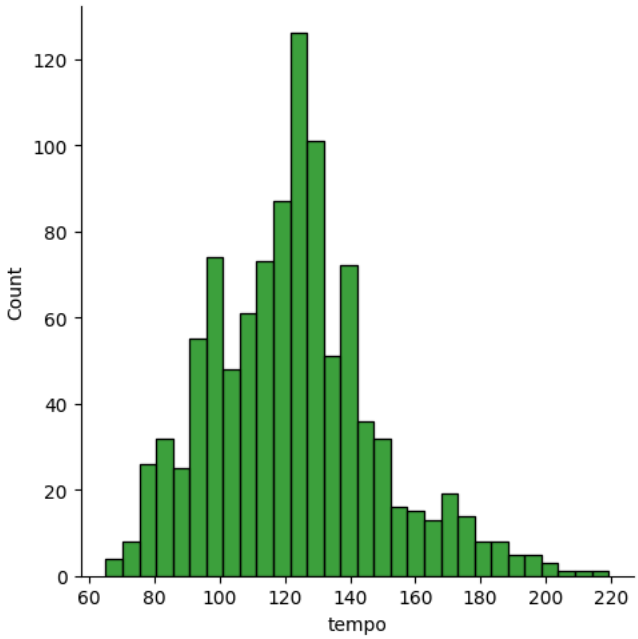

```
<Figure size 1000x500 with 0 Axes>
```

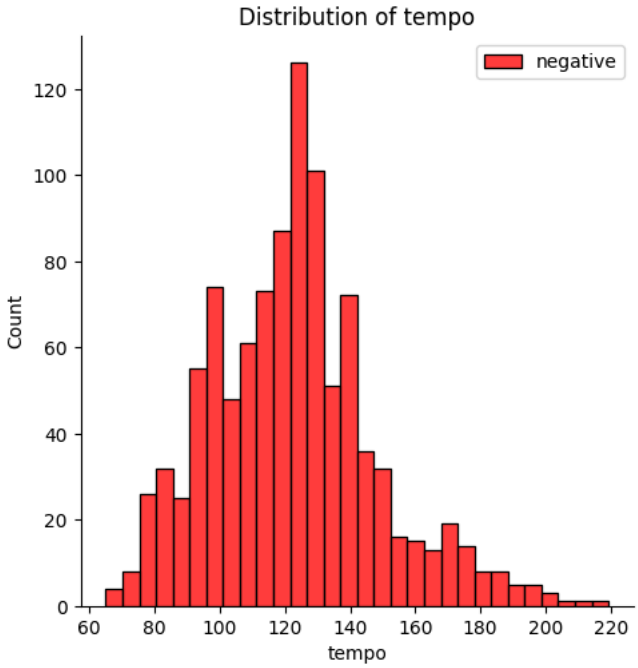




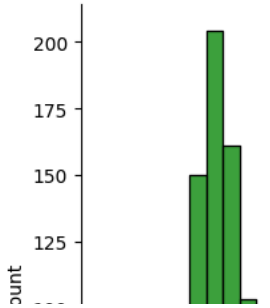


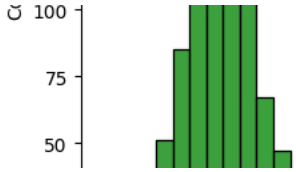

```
<Figure size 1000x500 with 0 Axes>
```






```
<Figure size 1000x500 with 0 Axes>
```




Ct
100
75
50