```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving spotify-2023.csv to spotify-2023.csv
```

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
df = pd.read_csv('spotify-2023.csv', encoding='latin-1')
df
```

	track_name	artist(s)_name	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts
0	Seven (feat. Latto) (Explicit Ver.)	Latto, Jung Kook	2	2023	7	14	553	147
1	LALA	Myke Towers	1	2023	3	23	1474	48
2	vampire	Olivia Rodrigo	1	2023	6	30	1397	113
3	Cruel Summer	Taylor Swift	1	2019	8	23	7858	100
4	WHERE SHE GOES	Bad Bunny	1	2023	5	18	3133	50
...	...	...	...	...	...	...	...	...
948	My Mind & Me	Selena Gomez	1	2022	11	3	953	0
949	Bigger Than The Whole Sky	Taylor Swift	1	2022	10	21	1180	0
950	A Veces (feat. Feid)	Feid, Paulo Londra	2	2022	11	3	573	0
951	En La De Ella	Feid, Sech, Jhayco	3	2022	10	20	1320	0
952	Alone	Burna Boy	1	2022	11	4	782	2

953 rows × 24 columns

```
df.head()
```

	track_name	artist(s)_name	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts	
0	Seven (feat. Latto) (Explicit Ver.)	Latto, Jung Kook	2	2023	7	14	553	147	1
1	LALA	Myke Towers	1	2023	3	23	1474	48	1
2	vampire	Olivia Rodrigo	1	2023	6	30	1397	113	1
3	Cruel Summer	Taylor Swift	1	2019	8	23	7858	100	8
4	WHERE SHE GOES	Bad Bunny	1	2023	5	18	3133	50	3

5 rows × 24 columns

```
df.tail()
```

	track_name	artist(s)_name	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts
948	My Mind & Me	Selena Gomez	1	2022	11	3	953	0
949	Bigger Than The Whole Sky	Taylor Swift	1	2022	10	21	1180	0
950	A Veces (feat. Feid)	Feid, Paulo Londra	2	2022	11	3	573	0
951	En La De Ella	Feid, Sech, Jhayco	3	2022	10	20	1320	0
952	Alone	Burna Boy	1	2022	11	4	782	2

5 rows × 24 columns

```
df.columns
```

```
Index(['track_name', 'artist(s)_name', 'artist_count', 'released_year',
       'released_month', 'released_day', 'in_spotify_playlists',
       'in_spotify_charts', 'streams', 'in_apple_playlists', 'in_apple_charts',
       'in_deezer_playlists', 'in_deezer_charts', 'in_shazam_charts', 'bpm',
       'key', 'mode', 'danceability_%', 'valence_%', 'energy_%',
       'acousticness_%', 'instrumentalness_%', 'liveness_%', 'speechiness_%'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 953 entries, 0 to 952
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   track_name            953 non-null    object
 1   artist(s)_name        953 non-null    object
 2   artist_count          953 non-null    int64
 3   released_year         953 non-null    int64
 4   released_month        953 non-null    int64
 5   released_day          953 non-null    int64
 6   in_spotify_playlists  953 non-null    int64
 7   in_spotify_charts     953 non-null    int64
 8   streams               953 non-null    object
 9   in_apple_playlists    953 non-null    int64
 10  in_apple_charts       953 non-null    int64
 11  in_deezer_playlists   953 non-null    object
 12  in_deezer_charts      953 non-null    int64
 13  in_shazam_charts      903 non-null    object
 14  bpm                   953 non-null    int64
 15  key                   858 non-null    object
 16  mode                  953 non-null    object
 17  danceability_%        953 non-null    int64
 18  valence_%             953 non-null    int64
 19  energy_%              953 non-null    int64
 20  acousticness_%        953 non-null    int64
 21  instrumentalness_%    953 non-null    int64
 22  liveness_%            953 non-null    int64
 23  speechiness_%         953 non-null    int64
dtypes: int64(17), object(7)
memory usage: 178.8+ KB
```

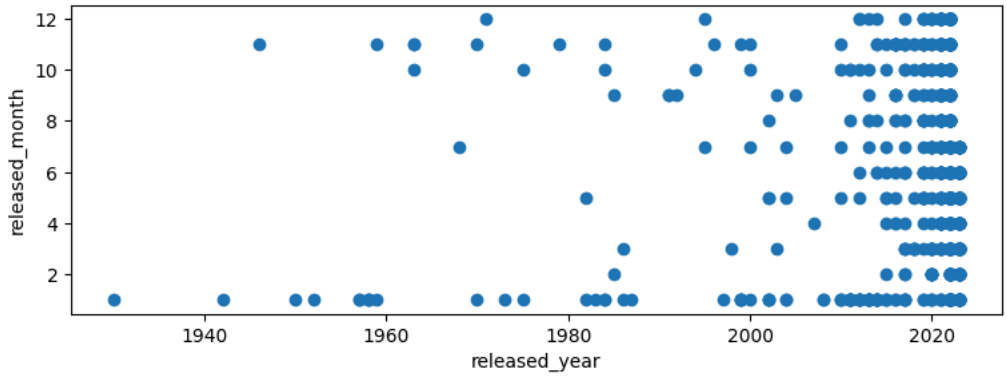
```
df.describe()
```

	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts	in_apple_playlists	in_ap
count	953.000000	953.000000	953.000000	953.000000	953.000000	953.000000	953.000000	
mean	1.556139	2018.238195	6.033578	13.930745	5200.124869	12.009444	67.812172	
std	0.893044	11.116218	3.566435	9.201949	7897.608990	19.575992	86.441493	
min	1.000000	1930.000000	1.000000	1.000000	31.000000	0.000000	0.000000	
25%	1.000000	2020.000000	3.000000	6.000000	875.000000	0.000000	13.000000	
50%	1.000000	2022.000000	6.000000	13.000000	2224.000000	3.000000	34.000000	
75%	2.000000	2022.000000	9.000000	22.000000	5542.000000	16.000000	88.000000	
max	8.000000	2023.000000	12.000000	31.000000	52898.000000	147.000000	672.000000	

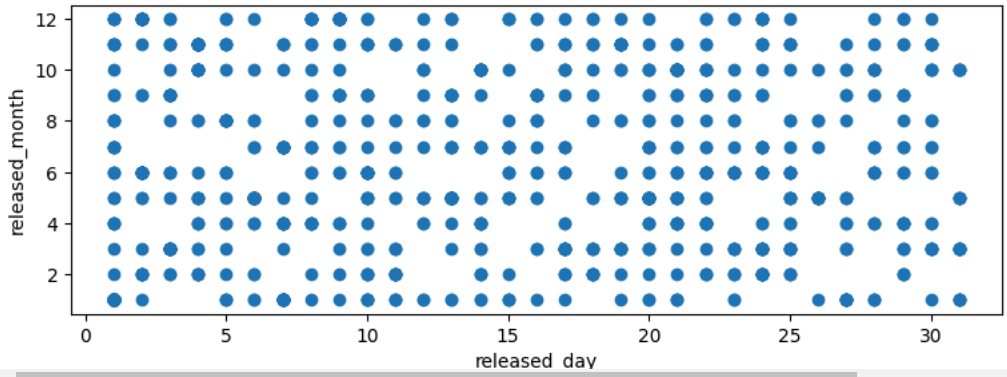
```
df.columns
```

```
Index(['track_name', 'artist(s)_name', 'artist_count', 'released_year',
       'released_month', 'released_day', 'in_spotify_playlists',
       'in_spotify_charts', 'streams', 'in_apple_playlists', 'in_apple_charts',
       'in_deezer_playlists', 'in_deezer_charts', 'in_shazam_charts', 'bpm',
       'key', 'mode', 'danceability_%', 'valence_%', 'energy_%',
       'acousticness_%', 'instrumentalness_%', 'liveness_%', 'speechiness_%'],
      dtype='object')
```

```
plt.figure(figsize=(9,3))
plt.scatter(df['released_year'], df['released_month'])
plt.xlabel('released_year')
plt.ylabel("released_month")
plt.show()
```




```
plt.figure(figsize=(9,3))
plt.scatter(df['released_day'], df['released_month'])
plt.xlabel('released_day')
plt.ylabel("released_month")
plt.show()
```




```
def bar_plot(variable) :
  var = df[variable]
  varValue = var.value_counts()
  plt.figure(figsize = (39, 33))
  plt.bar(varValue.index, varValue)
  plt.xticks(varValue.index, varValue.index.values)
  plt.ylabel("Frequency")
  plt.title(variable)
  plt.show()
  print("{}: \n {}".format(variable, varValue))
```

```
df.columns
```

```
Index(['track_name', 'artist(s)_name', 'artist_count', 'released_year',
       'released_month', 'released_day', 'in_spotify_playlists',
       'in_spotify_charts', 'streams', 'in_apple_playlists', 'in_apple_charts',
       'in_deezer_playlists', 'in_deezer_charts', 'in_shazam_charts', 'bpm',
       'key', 'mode', 'danceability_%', 'valence_%', 'energy_%',
       'acousticness_%', 'instrumentalness_%', 'liveness_%', 'speechiness_%'],
      dtype='object')
```

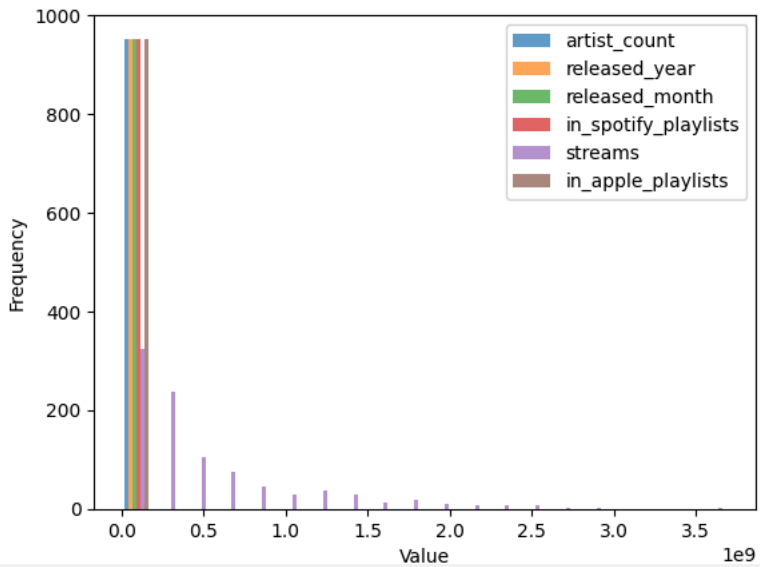
```
columns_plot = ['artist_count', 'released_year','released_month', 'released_day', 'in_spotify_playlists','in_spotify_charts', 'streams',
for i in columns_plot:
  bar_plot(i)
```

Show hidden output

```
df.columns
```

```
Index(['track_name', 'artist(s)_name', 'artist_count', 'released_year',
       'released_month', 'released_day', 'in_spotify_playlists',
       'in_spotify_charts', 'streams', 'in_apple_playlists', 'in_apple_charts',
       'in_deezer_playlists', 'in_deezer_charts', 'in_shazam_charts', 'bpm',
       'key', 'mode', 'danceability_%', 'valence_%', 'energy_%',
       'acousticness_%', 'instrumentalness_%', 'liveness_%', 'speechiness_%'],
      dtype='object')
```

```
df['streams'] = pd.to_numeric(df['streams'], errors='coerce')
plt.hist([df['artist_count'], df['released_year'], df['released_month'], df['in_spotify_playlists'], df['streams'], df['in_apple_playli
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```




```
df.columns
```

```
Index(['track_name', 'artist(s)_name', 'artist_count', 'released_year',
       'released_month', 'released_day', 'in_spotify_playlists',
       'in_spotify_charts', 'streams', 'in_apple_playlists', 'in_apple_charts',
       'in_deezer_playlists', 'in_deezer_charts', 'in_shazam_charts', 'bpm',
       'key', 'mode', 'danceability_%', 'valence_%', 'energy_%',
       'acousticness_%', 'instrumentalness_%', 'liveness_%', 'speechiness_%'],
      dtype='object')
```

```
df[['artist(s)_name', 'in_spotify_playlists']].groupby(['artist(s)_name'], as_index = False).mean().sort_values(by='in_spotify_playlist
```

	artist(s)_name	in_spotify_playlists
453	Pharrell Williams, Nile Rodgers, Daft Punk	52898.0
566	The Killers	51979.0
636	a-ha	44927.0
150	Drake, WizKid, Kyla	43257.0
210	Gotye, Kimbra	42798.0
...	...	...
474	RM, Colde	105.0
28	Arijit Singh, Sachin-Jigar	86.0
410	Natanael Cano	86.0
526	Shubh	67.0
243	Jack Black	34.0

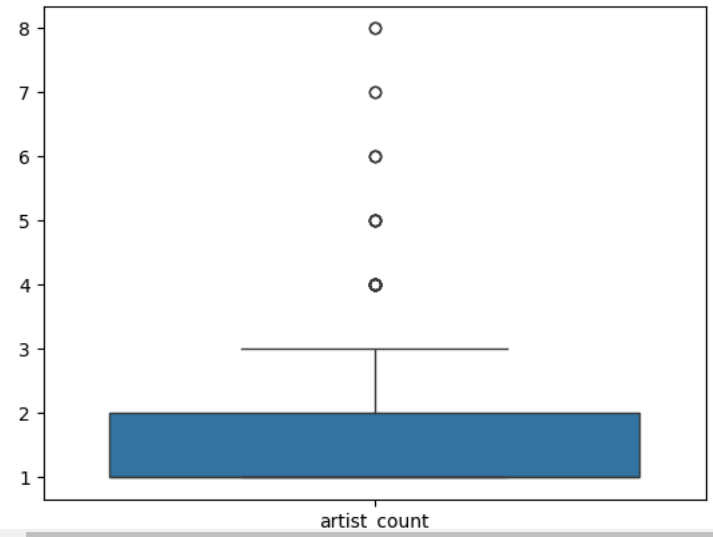
645 rows × 2 columns

```
df.columns
```

```
Index(['track_name', 'artist(s)_name', 'artist_count', 'released_year',
       'released_month', 'released_day', 'in_spotify_playlists',
```

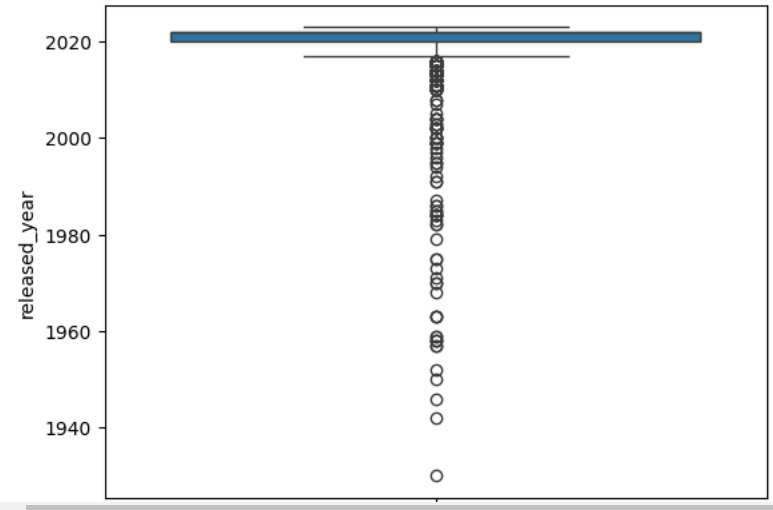
```
       'in_spotify_charts', 'streams', 'in_apple_playlists', 'in_apple_charts',
       'in_deezer_playlists', 'in_deezer_charts', 'in_shazam_charts', 'bpm',
       'key', 'mode', 'danceability_%', 'valence_%', 'energy_%',
       'acousticness_%', 'instrumentalness_%', 'liveness_%', 'speechiness_%'],
      dtype='object')
```

```
sns.boxplot(data=df[['artist_count']])
plt.show()
```



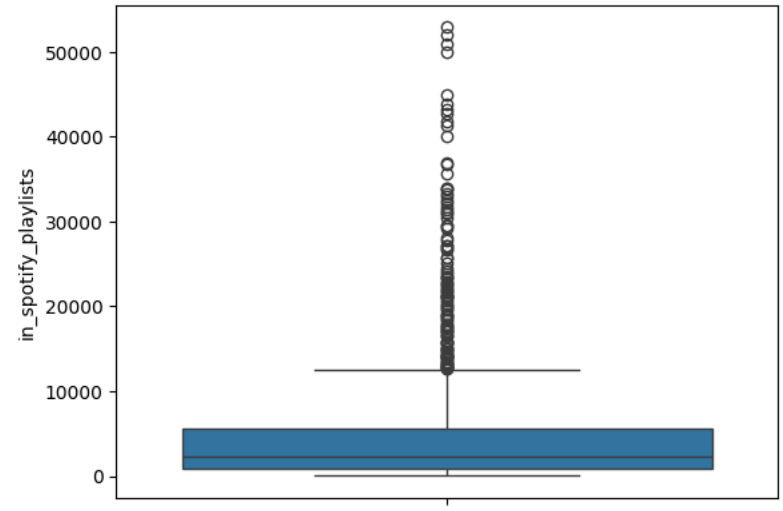

```
sns.boxplot(df['released_year'])
```

```
<Axes: ylabel='released_year'>
```




```
sns.boxplot(df['in_spotify_playlists'])
```

```
<Axes: ylabel='in_spotify_playlists'>
```



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

	0
track_name	0
artist(s)_name	0
artist_count	0
released_year	0
released_month	0
released_day	0
in_spotify_playlists	0
in_spotify_charts	0
streams	1
in_apple_playlists	0
in_apple_charts	0
in_deezer_playlists	0
in_deezer_charts	0
in_shazam_charts	50
bpm	0
key	95
mode	0
danceability_%	0
valence_%	0
energy_%	0
acousticness_%	0
instrumentalness_%	0
liveness_%	0
speechiness_%	0

dtype: int64

```
df['in_shazam_charts'].mode()
```

	in_shazam_charts
0	0

```
df['key'].mode()
```

	key
0	C#

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

	0
track_name	0
artist(s)_name	0
artist_count	0
released_year	0
released_month	0
released_day	0
in_spotify_playlists	0
in_spotify_charts	0
streams	1
in_apple_playlists	0
in_apple_charts	0
in_deezer_playlists	0
in_deezer_charts	0
in_shazam_charts	50
bpm	0
key	95
mode	0
danceability_%	0
valence_%	0
energy_%	0
acousticness_%	0
instrumentalness_%	0
liveness_%	0
speechiness_%	0

dtype: int64

```
df['key'] = df['key'].fillna(df['key'].mode()[0])
```

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

	0
track_name	0
artist(s)_name	0
artist_count	0
released_year	0
released_month	0
released_day	0
in_spotify_playlists	0
in_spotify_charts	0
streams	1
in_apple_playlists	0
in_apple_charts	0
in_deezer_playlists	0
in_deezer_charts	0
in_shazam_charts	50
bpm	0
key	0
mode	0
danceability_%	0
valence_%	0
energy_%	0
acousticness_%	0
instrumentalness_%	0
liveness_%	0
speechiness_%	0

```
df['streams'].mode()
```

	streams
0	1.563386e+08
1	3.955914e+08
2	7.238945e+08
3	1.223481e+09

```
df.columns
```

```
Index(['track_name', 'artist(s)_name', 'artist_count', 'released_year',
       'released_month', 'released_day', 'in_spotify_playlists',
       'in_spotify_charts', 'streams', 'in_apple_playlists', 'in_apple_charts',
       'in_deezer_playlists', 'in_deezer_charts', 'in_shazam_charts', 'bpm',
       'key', 'mode', 'danceability_%', 'valence_%', 'energy_%',
       'acousticness_%', 'instrumentalness_%', 'liveness_%', 'speechiness_%'],
      dtype='object')
```

```
df.drop(labels=['track_name', 'artist(s)_name','released_month', 'released_day','in_spotify_charts','in_apple_charts','in_deezer_charts
```

```
df.head()
```

	artist_count	released_year	in_spotify_playlists	streams	in_apple_playlists	in_deezer_playlists	danceability_%	valence_
0	2	2023	553	141381703.0	43	45	80	8
1	1	2023	1474	133716286.0	48	58	71	6
2	1	2023	1397	140003974.0	94	91	51	3
3	1	2019	7858	800840817.0	116	125	55	5
4	1	2023	3133	303236322.0	84	87	65	2

```
df.columns
```

```
Index(['artist_count', 'released_year', 'in_spotify_playlists', 'streams',
       'in_apple_playlists', 'in_deezer_playlists', 'danceability_%',
       'valence_%', 'energy_%', 'acousticness_%', 'instrumentalness_%'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 953 entries, 0 to 952
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   artist_count          953 non-null    int64
 1   released_year         953 non-null    int64
 2   in_spotify_playlists  953 non-null    int64
 3   streams               952 non-null    float64
 4   in_apple_playlists    953 non-null    int64
 5   in_deezer_playlists   953 non-null    object
 6   danceability_%        953 non-null    int64
 7   valence_%             953 non-null    int64
 8   energy_%              953 non-null    int64
 9   acousticness_%        953 non-null    int64
 10  instrumentalness_%    953 non-null    int64
dtypes: float64(1), int64(9), object(1)
memory usage: 82.0+ KB
```

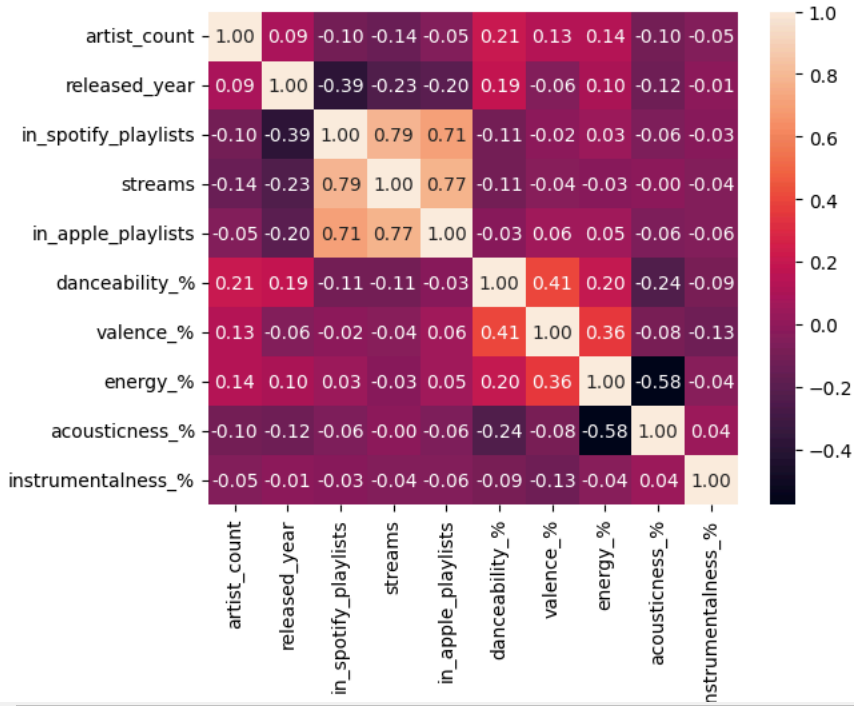
```
print(df.dtypes)
```

```
artist_count              int64
released_year             int64
in_spotify_playlists      int64
streams                 float64
in_apple_playlists        int64
in_deezer_playlists      object
danceability_%            int64
valence_%                 int64
energy_%                  int64
acousticness_%            int64
instrumentalness_%        int64
dtype: object
```

```
df.drop(labels=['in_deezer_playlists'], axis=1, inplace=True)
```

```
sns.heatmap(df.corr(), annot=True, fmt='.2f')
```

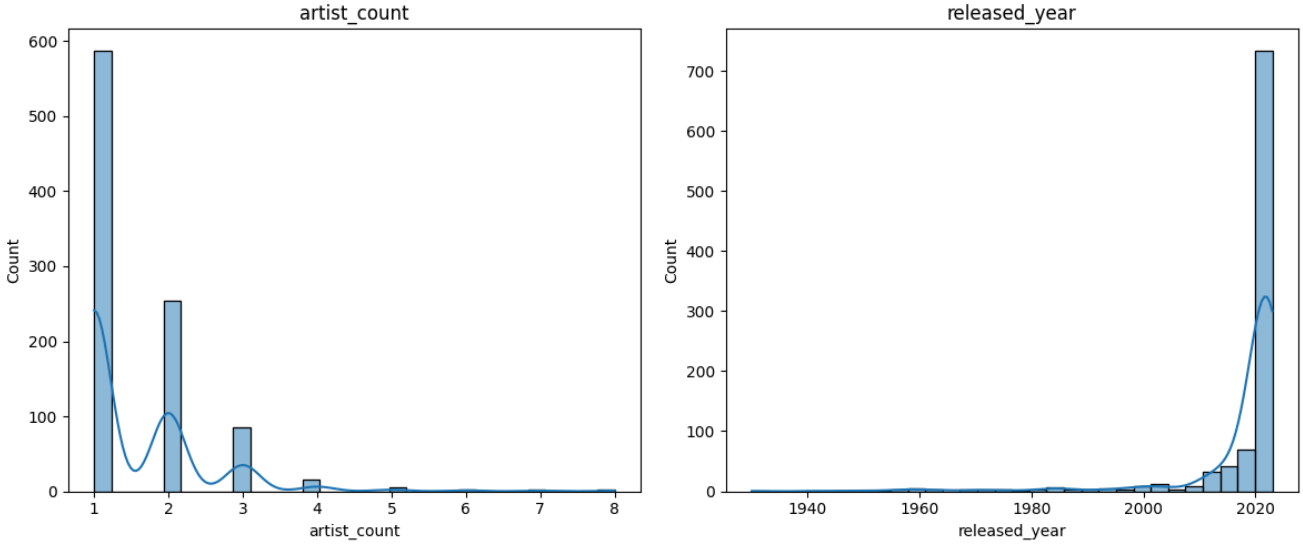
```
<Axes: >
```

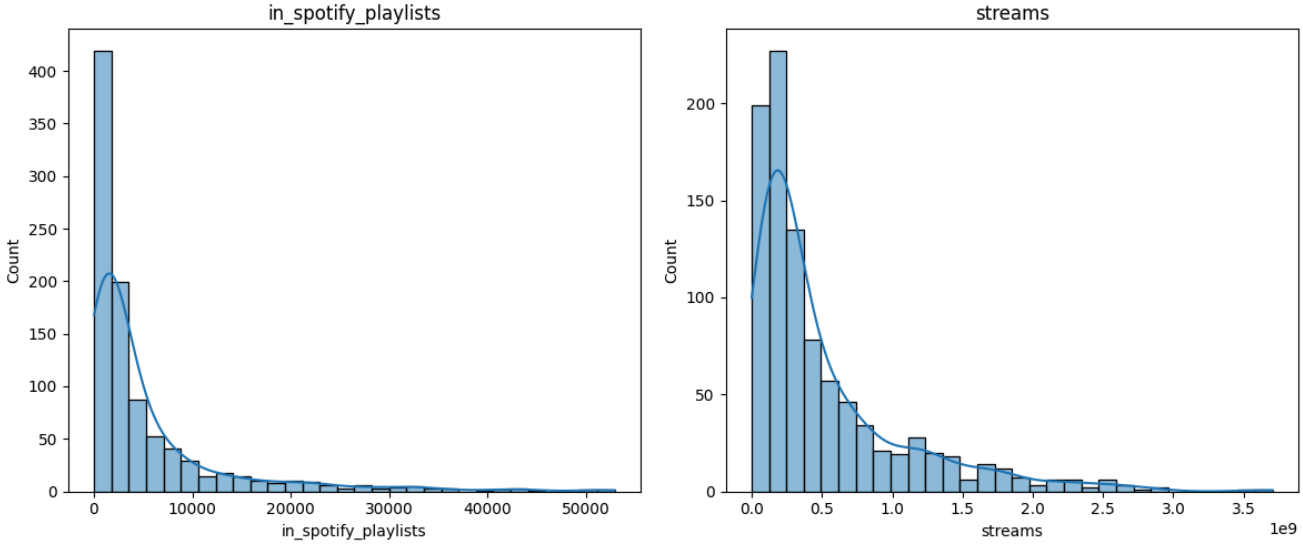


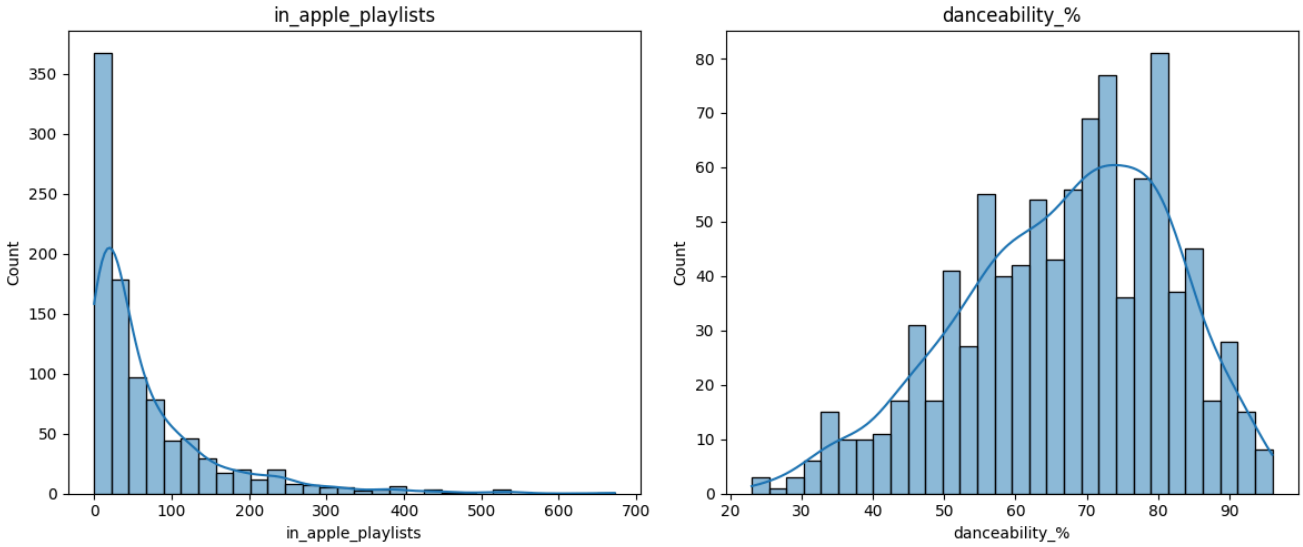

```
df.columns
```

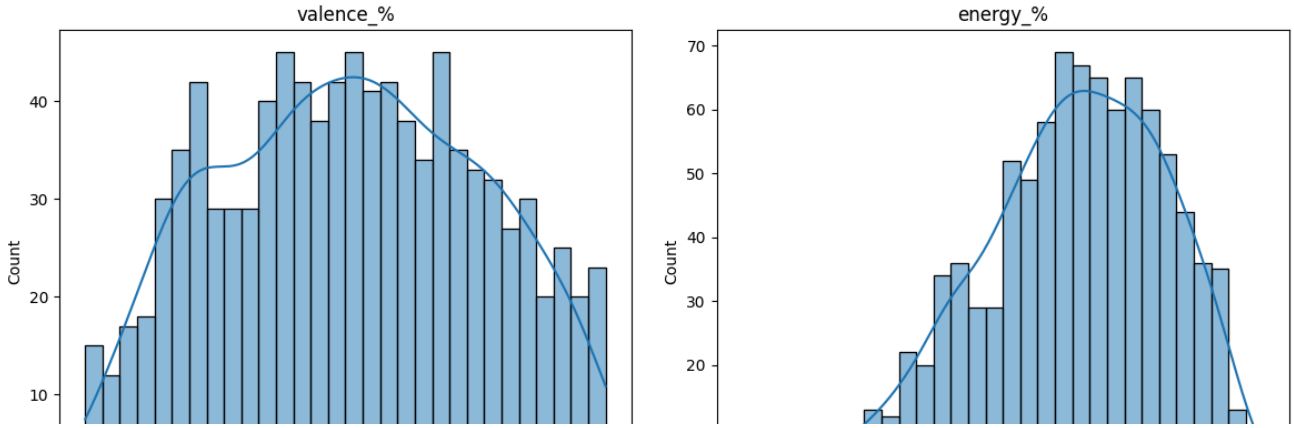
```
Index(['artist_count', 'released_year', 'in_spotify_playlists', 'streams',
       'in_apple_playlists', 'danceability_%', 'valence_%', 'energy_%',
       'acousticness_%', 'instrumentalness_%'],
      dtype='object')
```

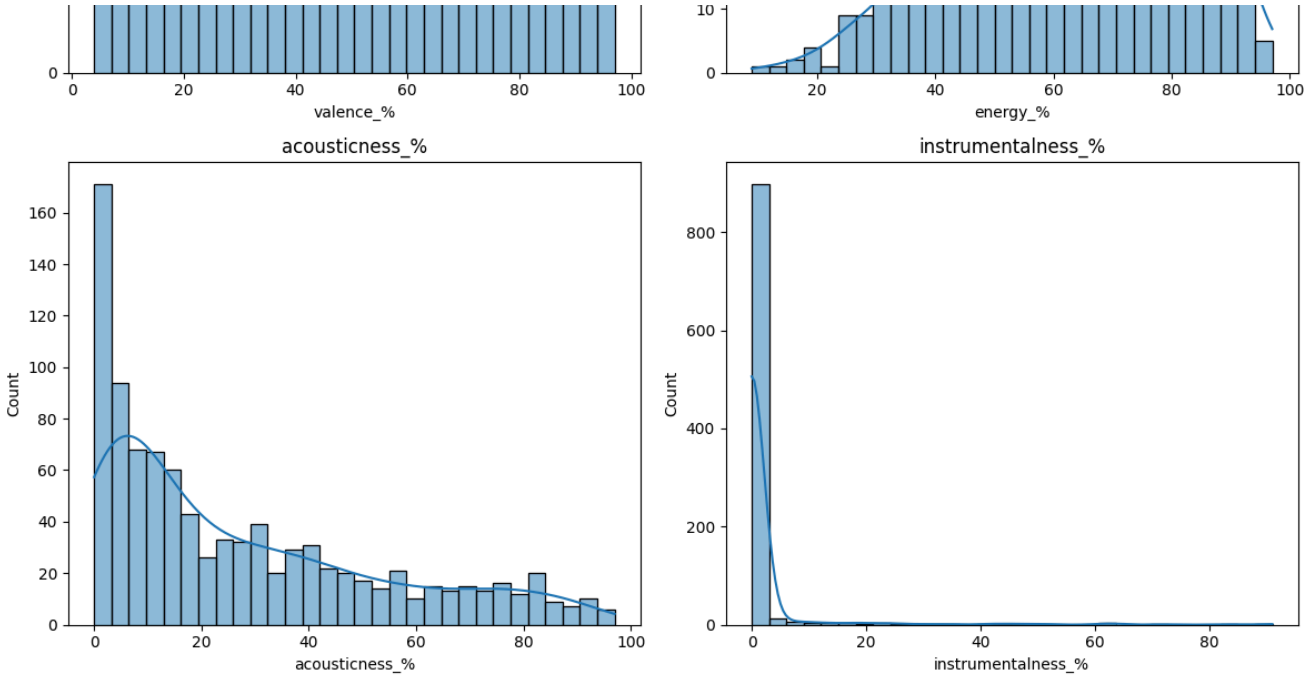
```
fig, axes = plt.subplots(5, 2, figsize=(12, 25))
for ax, col in zip(axes.flat, df.columns):
    sns.histplot(df[col], bins=30, kde=True, ax=ax)
    ax.set_title(col)
plt.tight_layout()
plt.show()
```


artist_count
Count
artist_count
released_year
Count
released_year


in_spotify_playlists
Count
in_spotify_playlists
streams
Count
streams
1e9


in_apple_playlists
Count
in_apple_playlists
danceability_%
Count
danceability_%


valence_%
Count
energy_%
Count


valence_%
energy_%
acousticness_%
Count
instrumentalness_%
Count

```
for col in df.columns:
    value_counts = df[col].value_counts()
    threshold = 0.02 * sum(value_counts)
    value_counts_filtered = value_counts[value_counts > threshold]
    others = sum(value_counts[value_counts <= threshold])
    if others > 0:
        value_counts_filtered["Others"] = others
    plt.figure(figsize=(7, 7))
    value_counts_filtered.plot.pie(autopct='%1.1f%%', startangle=140, wedgeprops={'edgecolor': 'black'})
    plt.title(col)
    plt.ylabel('')
    plt.show()
```