**1. Data Import and Initial Validation**

Index(['number\_row', 'name', 'album', 'release\_date', 'track\_number', 'id',

'uri', 'acousticness', 'danceability', 'energy', 'instrumentalness',

'liveness', 'loudness', 'speechiness', 'tempo', 'valence', 'popularity',

'duration\_ms'],

dtype='object')

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1610 entries, 0 to 1609

Data columns (total 18 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 number\_row 1610 non-null int64

1 name 1610 non-null string

2 album 1610 non-null string

3 release\_date 1610 non-null datetime64[ns]

4 track\_number 1610 non-null int64

5 id 1610 non-null string

6 uri 1610 non-null string

7 acousticness 1610 non-null float64

8 danceability 1610 non-null float64

9 energy 1610 non-null float64

10 instrumentalness 1610 non-null float64

11 liveness 1610 non-null float64

12 loudness 1610 non-null float64

13 speechiness 1610 non-null float64

14 tempo 1610 non-null float64

15 valence 1610 non-null float64

16 popularity 1610 non-null int64

17 duration\_ms 1610 non-null int64

dtypes: datetime64[ns](1), float64(9), int64(4), string(4)

memory usage: 226.5 KB

None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Number of duplicated rows in the data is : 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Number of missing values in the data is :

number\_row 0

name 0

album 0

release\_date 0

track\_number 0

id 0

uri 0

acousticness 0

danceability 0

energy 0

instrumentalness 0

liveness 0

loudness 0

speechiness 0

tempo 0

valence 0

popularity 0

duration\_ms 0

dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Number of duplicated \*\*id\*\* in the data is : 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 2. Check possible outliers using Interquartile Range (IQR) method

Possible outliers:

Number of outlier for acousticness: 4

Number of outlier for danceability: 0

Number of outlier for energy: 8

Number of outlier for instrumentalness: 272

Number of outlier for liveness: 0

Number of outlier for loudness: 14

Number of outlier for speechiness: 92

Number of outlier for tempo: 34

Number of outlier for valence: 0

Number of outlier for popularity: 50

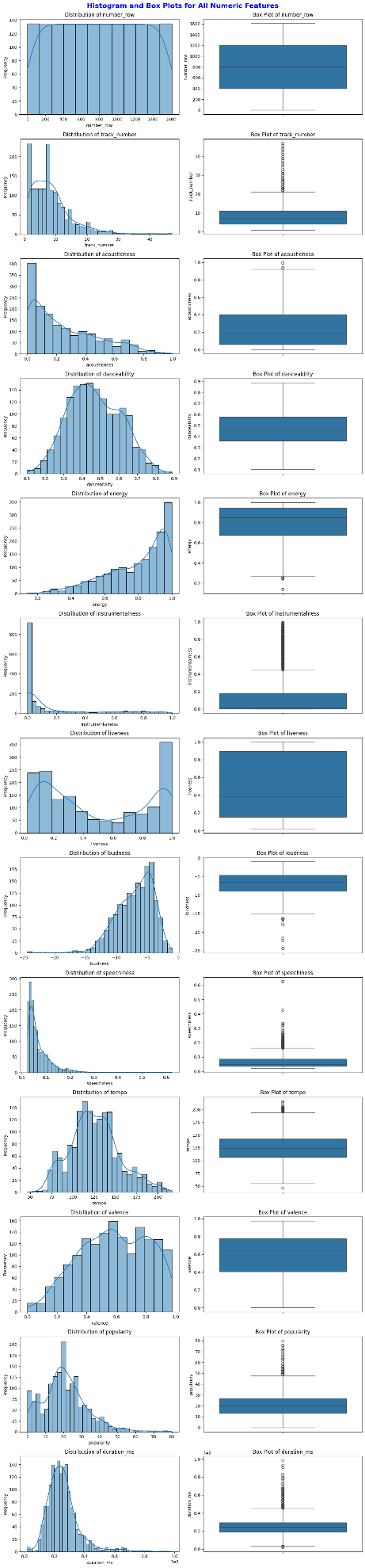
Number of outlier for duration\_ms: 76

# 2. Check for outliers using histogram plot and box plot

Number of numeric columns: 13

Numeric columns: ['number\_row', 'track\_number', 'acousticness', 'danceability', 'energy', 'instrumentalness', 'liveness', 'loudness', 'speechiness', 'tempo', 'valence', 'popularity', 'duration\_ms']

{The plot has saved in Plots folder}



# Sorting dataset based on each variable to learn more about outliers (top or bottom 10 data points shown using above plots)

Sorted values of acousticness:

807 0.994

818 0.994

997 0.939

165 0.935

280 0.920

308 0.920

113 0.914

255 0.900

81 0.898

989 0.892

Name: acousticness, dtype: float64

Sorted values of energy:

807 0.141

818 0.142

1333 0.242

1313 0.246

1325 0.248

1369 0.250

1118 0.252

1115 0.261

1186 0.272

1268 0.274

Name: energy, dtype: float64

Sorted values of loudness:

818 -24.408

807 -24.351

739 -22.301

1118 -21.548

546 -18.104

756 -17.577

1186 -16.572

300 -16.556

328 -16.556

1310 -16.415

Name: loudness, dtype: float64

Sorted values of speechiness:

308 0.624

280 0.624

81 0.425

553 0.335

558 0.330

312 0.323

284 0.323

213 0.306

1176 0.286

217 0.280

Name: speechiness, dtype: float64

Sorted values of popularity:

1403 80

1472 76

862 76

1248 76

1257 73

901 72

1023 71

893 69

1122 69

1256 67

Name: popularity, dtype: int64

Sorted values of duration\_ms:

220 981866

460 929457

537 907373

482 831960

325 815887

297 815887

1175 789240

8 781173

183 773226

111 734706

Name: duration\_ms, dtype: int64

# 3. Drop significant outliers from the dataset (listed above)

number\_row name album release\_date \

0 0 Concert Intro Music - Live Licked Live In NYC 2022-06-10

1 1 Street Fighting Man - Live Licked Live In NYC 2022-06-10

2 2 Start Me Up - Live Licked Live In NYC 2022-06-10

3 3 If You Can't Rock Me - Live Licked Live In NYC 2022-06-10

4 4 Don’t Stop - Live Licked Live In NYC 2022-06-10

track\_number id uri \

0 1 2IEkywLJ4ykbhi1yRQvmsT spotify:track:2IEkywLJ4ykbhi1yRQvmsT

1 2 6GVgVJBKkGJoRfarYRvGTU spotify:track:6GVgVJBKkGJoRfarYRvGTU

2 3 1Lu761pZ0dBTGpzxaQoZNW spotify:track:1Lu761pZ0dBTGpzxaQoZNW

3 4 1agTQzOTUnGNggyckEqiDH spotify:track:1agTQzOTUnGNggyckEqiDH

4 5 7piGJR8YndQBQWVXv6KtQw spotify:track:7piGJR8YndQBQWVXv6KtQw

acousticness danceability energy instrumentalness liveness loudness \

0 0.0824 0.463 0.993 0.996000 0.932 -12.913

1 0.4370 0.326 0.965 0.233000 0.961 -4.803

2 0.4160 0.386 0.969 0.400000 0.956 -4.936

3 0.5670 0.369 0.985 0.000107 0.895 -5.535

4 0.4000 0.303 0.969 0.055900 0.966 -5.098

speechiness tempo valence popularity duration\_ms

0 0.1100 118.001 0.0302 33 48640

1 0.0759 131.455 0.3180 34 253173

2 0.1150 130.066 0.3130 34 263160

3 0.1930 132.994 0.1470 32 305880

4 0.0930 130.533 0.2060 32 305106