Analiza piosenek rockowych

March 27, 2025

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:
```
df = pd.read_csv("80s/archive/UltimateClassicRock.csv")
df['Decade'] = (df['Year'] // 10) * 10
df['Decade'] = df['Decade'].astype(str) + 's'
order = ['1960s', '1970s', '1980s', '1990s', '2000s', '2010s', '2020s']
df['Decade'] = pd.Categorical(df['Decade'], categories=order, ordered=True)


def convert_duration(duration):
    minutes, seconds = map(int, duration.split(':'))
    return minutes * 60 + seconds

df['Duration'] = df['Duration'].apply(convert_duration)
```

0.1 Opis danych

Zbior danych zawiera ponad 14000 wpisow o piosenkach rockowych z okresu 1962 - 2024. Dane sa wziete z API Spotify. Wartoscia target jest Popularity.

1. Track - tytul utworu
2. Artist - wykonawca
3. Album - album
4. Year - rok wydania
5. Duration - dlugosc trwania w sekundach
6. Time-Signature - metrum
7. Danceability - jak dobrze utwor nadaje sie do tanca
8. Energy - Intensywnosc utworu
9. Key - tonacja
10. Loudness - Srednia glosnosc
11. Mode - tryb
12. Speechiness - ilosc mowy w utworze
13. Acousticness - jak bardzo utwor jest akustyczny
14. Instrumentalness - jak bardzo jest instrumentalny
15. Valence - poziom pozytywnosci/nastroju
16. Tempo - temp utworu (BPM)

17. Populrity - wskaznik popularnosci
18. Decade - dekada wydania

Targetem jest cecha Popularity.

[27]:
```
df.columns
```

[27]:
```
Index(['Track', 'Artist', 'Album', 'Year', 'Duration', 'Time_Signature',
       'Danceability', 'Energy', 'Key', 'Loudness', 'Mode', 'Speechiness',
       'Acousticness', 'Instrumentalness', 'Liveness', 'Valence', 'Tempo',
       'Popularity', 'Decade'],
      dtype='object')
```

[4]:
```
df.head()
```

[4]:

	Track	Artist	Album	Year	Duration	Time_Signature	Danceability	Energy	Key	Loudness	Mode	Speechiness	Acousticness	Instrumentalness	Liveness	Valence	Tempo	Popularity	Decade
0	Play A Simple Song	38 Special	38 Special	1977	193	4	0.521	0.367	0	-13.866	1	0.0278	0.6920	0.000003	0.1080	0.789	83.412	16	1970s
1	Four Wheels	38 Special	38 Special	1977	283	4	0.535	0.710	2	-12.287	1	0.0428	0.0100	0.023000	0.0495	0.445	160.361	10	1970s
2	Fly Away	38 Special	38 Special	1977	313	4	0.563	0.563	2	-10.781	1	0.0263	0.0357	0.001850	0.1400	0.564	106.739	13	1970s
3	Tell Everybody	38 Special	38 Special	1977	249	4	0.638	0.694	11	-10.206	0	0.0310	0.1610	0.000034	0.0908	0.936	124.962	10	1970s
4	Just Wanna Rock & Roll	38 Special	38 Special	1977	357	4	0.388	0.701	2	-9.984	1	0.0360	0.0130	0.042200	0.1150	0.769	126.769	11	1970s

[5]:
```
df.describe()
```

[5]:

	Year	Duration	Time_Signature	Danceability	Energy
count	14418.000000	14418.000000	14418.000000	14418.000000	14418.000000
mean	1987.634693	260.399986	3.917811	0.503063	0.656563

```
std       15.318819    104.113755        0.356628      0.142619      0.229607
min     1962.000000      4.000000        0.000000      0.000000      0.000000
25%     1975.000000    206.000000        4.000000      0.405000      0.493000
50%     1983.000000    248.000000        4.000000      0.509000      0.690500
75%     1999.000000    296.000000        4.000000      0.603000      0.854000
max     2024.000000   2018.000000        5.000000      0.987000      0.998000

                Key      Loudness          Mode   Speechiness  Acousticness  \
count  14418.000000  14418.000000  14418.000000  14418.000000  14418.000000
mean       5.166597     -9.438675      0.730129      0.051354      0.226924
std        3.503423      4.179623      0.443908      0.046291      0.268857
min        0.000000    -60.000000      0.000000      0.000000      0.000000
25%        2.000000    -11.914250      0.000000      0.031800      0.013100
50%        5.000000     -8.810500      1.000000      0.039000      0.104000
75%        9.000000     -6.341500      1.000000      0.054300      0.370000
max       11.000000     -0.203000      1.000000      0.952000      0.995000

       Instrumentalness      Liveness       Valence         Tempo  \
count      14418.000000  14418.000000  14418.000000  14418.000000
mean           0.089682      0.210653      0.528818    122.641620
std            0.215783      0.180537      0.243557     27.940743
min            0.000000      0.000000      0.000000      0.000000
25%            0.000013      0.093800      0.338000    102.196500
50%            0.000737      0.140000      0.529000    121.859000
75%            0.028375      0.278000      0.724750    139.228000
max            0.992000      1.000000      0.991000    238.895000

         Popularity
count  14418.000000
mean      25.394438
std       15.090860
min        0.000000
25%       14.000000
50%       23.000000
75%       34.000000
max       91.000000
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14418 entries, 0 to 14417
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Track             14418 non-null  object
 1   Artist            14418 non-null  object
 2   Album             14418 non-null  object
 3   Year              14418 non-null  int64
```

```
 4   Duration          14418 non-null  int64
 5   Time_Signature    14418 non-null  int64
 6   Danceability      14418 non-null  float64
 7   Energy            14418 non-null  float64
 8   Key               14418 non-null  int64
 9   Loudness          14418 non-null  float64
 10  Mode              14418 non-null  int64
 11  Speechiness       14418 non-null  float64
 12  Acousticness      14418 non-null  float64
 13  Instrumentalness  14418 non-null  float64
 14  Liveness          14418 non-null  float64
 15  Valence           14418 non-null  float64
 16  Tempo             14418 non-null  float64
 17  Popularity        14418 non-null  int64
 18  Decade            14418 non-null  category
dtypes: category(1), float64(9), int64(6), object(3)
memory usage: 2.0+ MB
```

0.2 Export statystyk numerycznych i kategorialnych

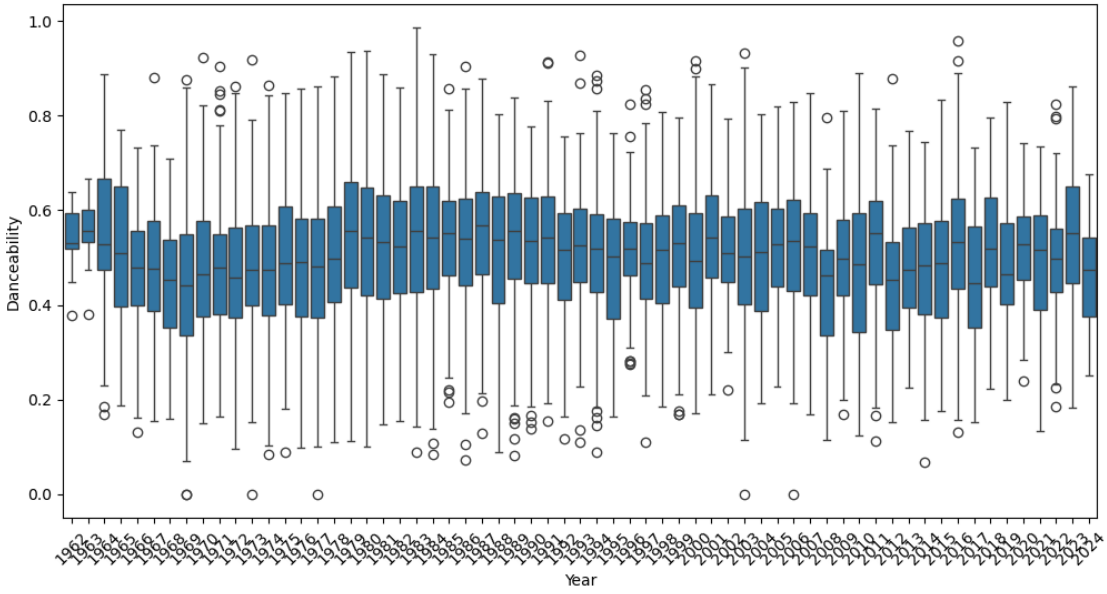
[7]:
```
data = df.describe(percentiles=[0.05, 0.95])

missing_values = df.isna().sum()
data = data.drop("count", axis=0)
data.loc["missing_values"] = missing_values

data.to_csv("numericalStatistics.csv", index=False)
```

[8]:
```
categorical_cols = df.select_dtypes(include=['object', 'category']).columns

desc = df[categorical_cols].describe()

desc.loc['missing'] = df[categorical_cols].isnull().sum()

desc.loc['class_proportions'] = [
    df[col].value_counts(normalize=True).round(2).to_dict()
    for col in categorical_cols
]

desc.drop(index=["top", "freq", "count"])
data.to_csv("categoricalStatistics.csv", index=False)
```

0.3 Boxplots

[9]:
```
    plt.figure(figsize=(12, 6))
    sns.boxplot(x=df["Year"], y=df["Danceability"])
    plt.xticks(rotation=45)
```

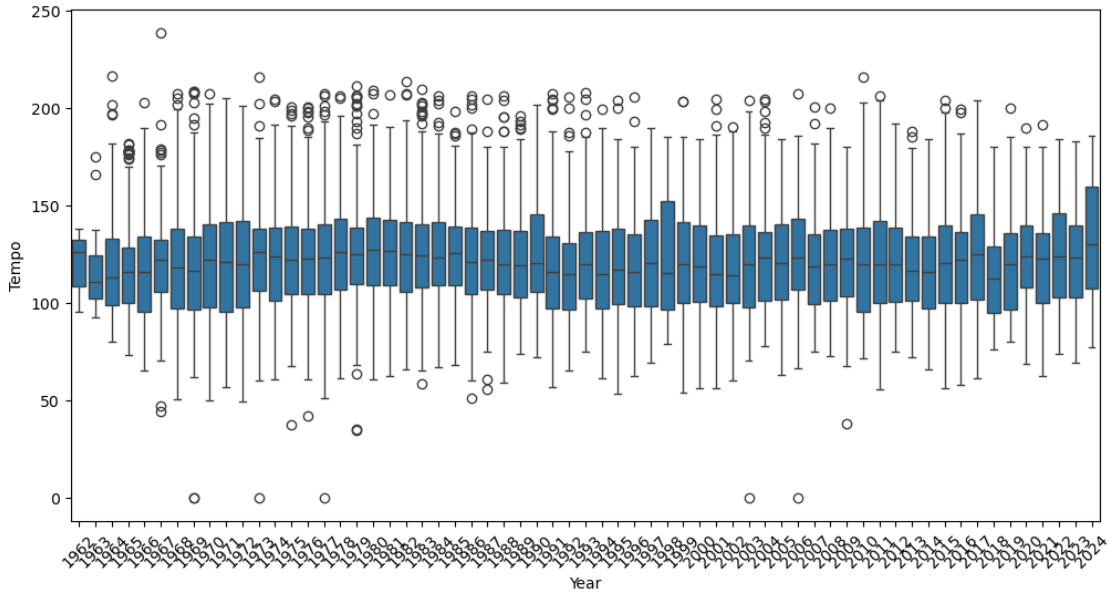
```
plt.show()
```



0.3.1 Wnioski

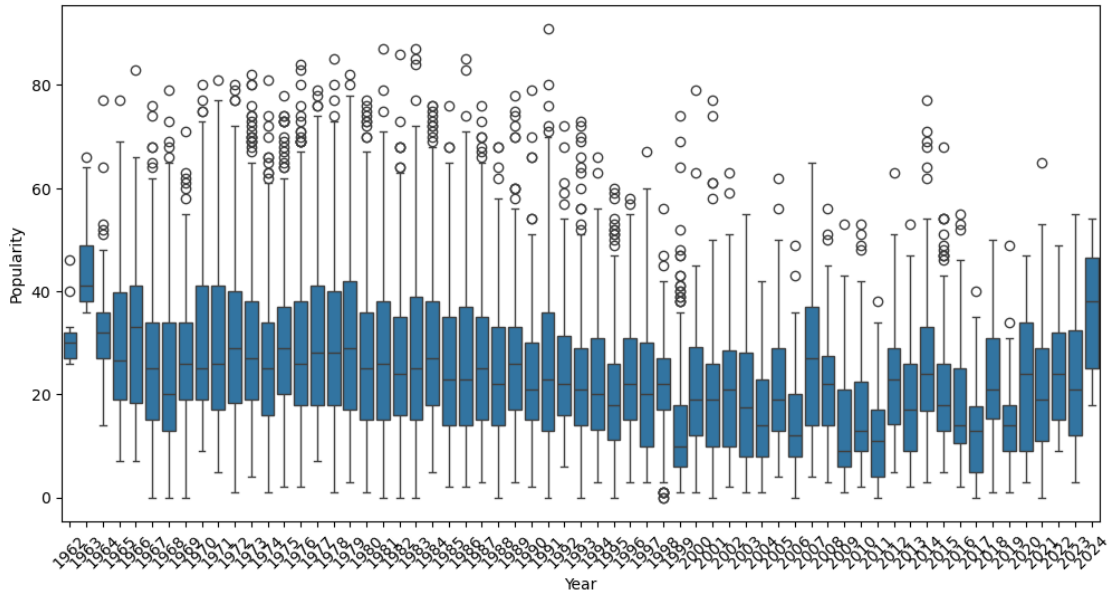
Danceability na przestrzeni lat utrzymywala sie na dosc stabilnym poziomie, w wiekszosci w zakresie 0.5- 0.6. W latach 60. i 70. pudelka sa dosyc wysokie, co moze wskazywac na duze zroznicowanie muzyki w tamtym czase. W latach 90. pudelka sa troche krotsze, co moze wskazywac na mniejsze zroznicowanie. Niemniej jednak dla kazdego roku wystepuja przypadki skrajne.

[10]:
```
plt.figure(figsize=(12, 6))
sns.boxplot(x=df["Year"], y=df["Tempo"])
plt.xticks(rotation=45)

plt.show()
```



Tempo ogolnie utrzymuje sie w przedziale 100-130, jednak widac ze w latach 1975 - 1990 bylo bardzo podobne, natomiast juz od 1990 roku pudelka sa troche wyzsze, co wskazuje na wieksza roznorodnosc.

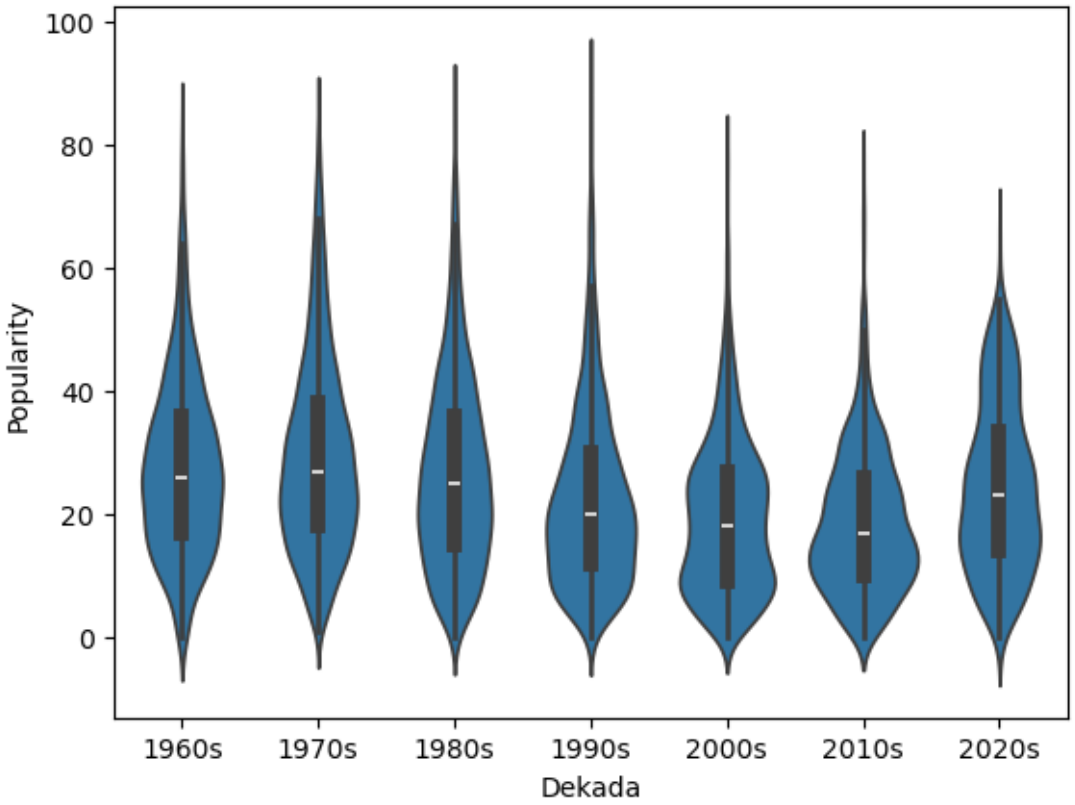
[11]:
```
plt.figure(figsize=(12, 6))
sns.boxplot(x=df["Year"], y=df["Popularity"])
plt.xticks(rotation=45)

plt.show()
```



W latach 1960 - 1990. mediana byla dosyc wysoka - w okolicach 35. W latach 1990 - 2018 mediana utrzymywala sie na poziomie ok. 20, co wskazuje na spadek popularnosci w porownaniu do lat 1960-1990. W okolicach lat 2018 - 2020 widac znaczy wzrost popularnosci, mediana w ostatnich latach osiagala nawet poziom 40.

0.4 Violin plots

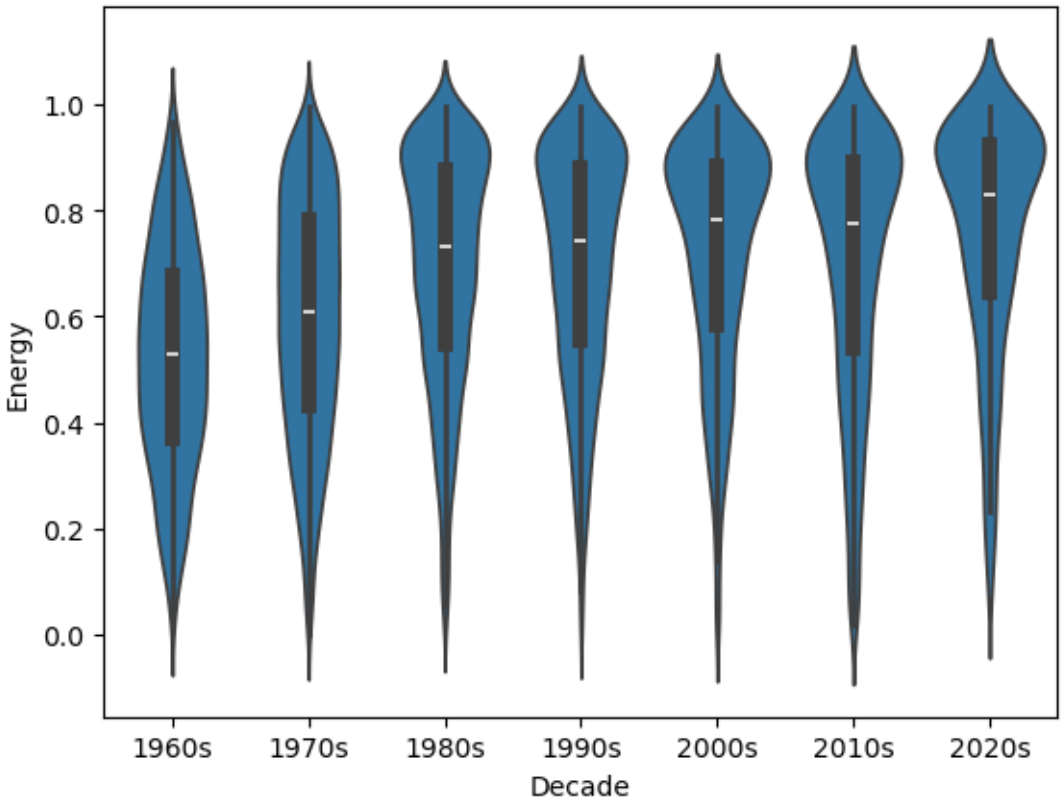
[12]:
```
sns.violinplot(x='Decade', y='Popularity', data=df)

plt.xlabel("Dekada")
plt.ylabel("Popularity")
plt.show()
```



Widac ze w latach 70 i 80 Popularity bylo w bardzo szerokim zakresie, natomiast w latach 2000 i 2010 bylo duzo utworow w okolicach wartosci 15. W 2020s powrocila duza roznorodnosc

[13]:
```
sns.violinplot(x='Decade', y='Energy', data=df)

plt.xlabel("Decade")
```

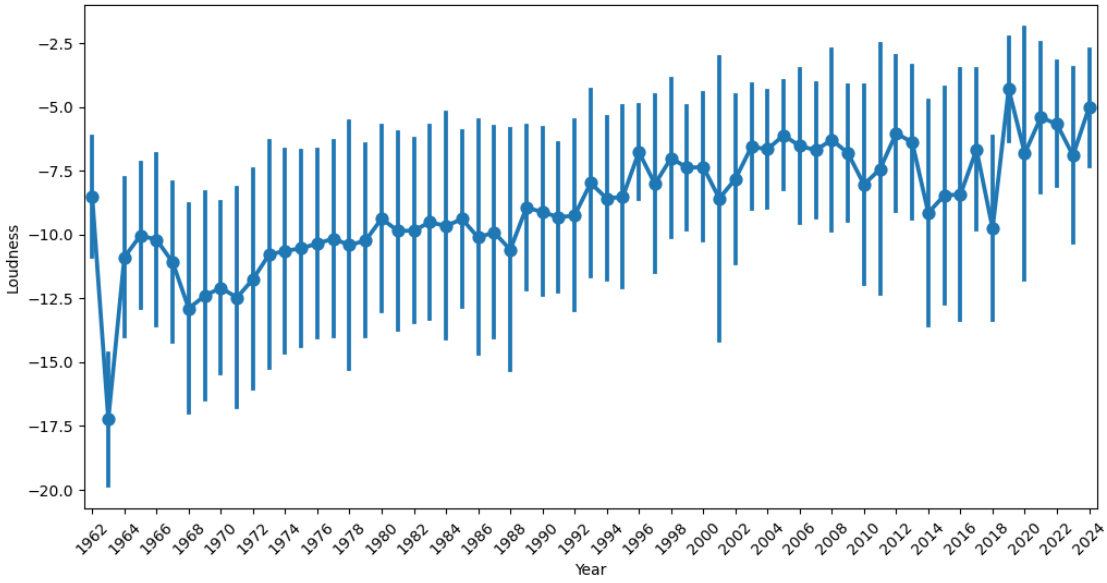
```
plt.ylabel("Energy")
plt.show()
```



Wriac wyraznie ze z kazda dekada wskaznik energii rosl coraz bardziej - mediana podniosla sie z ok. 55 az do 80. Wskazuje to na bardziej energiczna muzyke na przestrzeni lat, jest to jakis trend. W latach 60 i 70 wartosci byly bardziej rozrzucone, za to od lat 80 skupialy sie glownie w okolicach 0.9

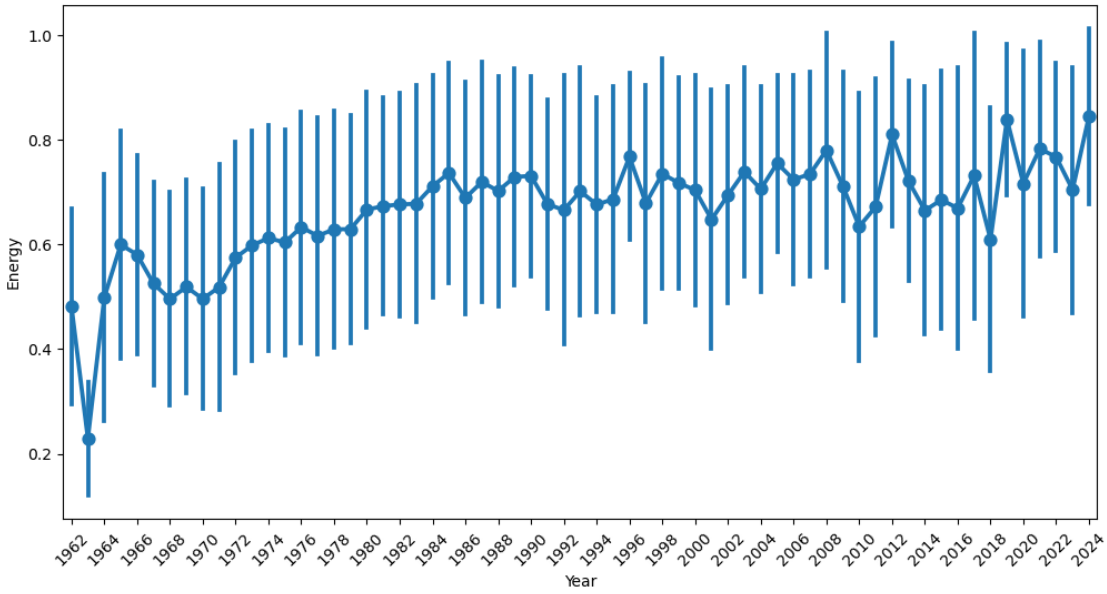
0.5 Error bars

[14]:
```
plt.figure(figsize=(12, 6))
sns.pointplot(x='Year', y='Loudness', data=df, errorbar='sd')
plt.xlabel("Year")
plt.ylabel("Loudness")

years = sorted(df['Year'].unique())
plt.xticks(ticks=range(0, len(years), 2), rotation=45, labels=years[::2])

plt.show()
```



Widac na wykresie ze Loudness stopniowo rosla, az do ok. roku 2008 zaczela sie bardziej wahac. Slupki bledow w okresie 1970 - 1990 sa dosyc stalej dlugosci, ale pozniej sie znacznie bardziej wahaja. Male dlugosci slupkow bledow wystepuja w latach 2020 - 2024, co moze wskazywac na czesciowe ujednolicenie muzyki rockowej

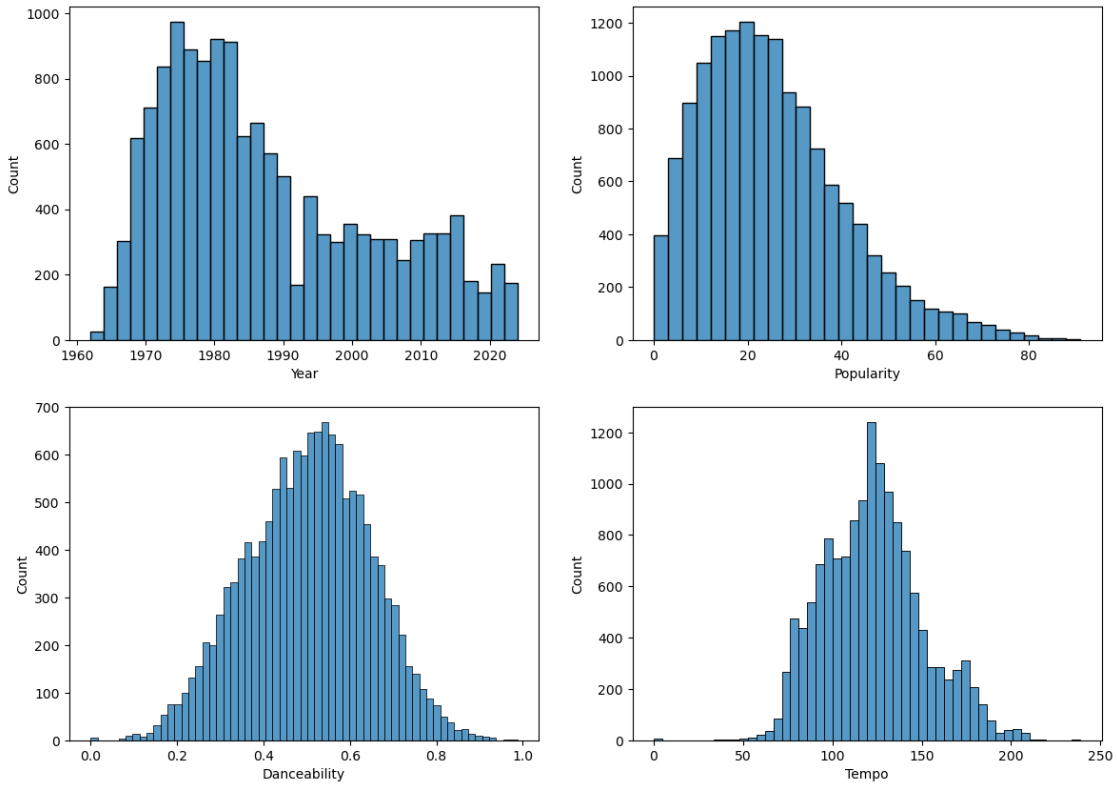
[15]:
```
plt.figure(figsize=(12, 6))
sns.pointplot(x='Year', y='Energy', data=df, errorbar='sd')
plt.xlabel("Year")
plt.ylabel("Energy")

years = sorted(df['Year'].unique())
plt.xticks(ticks=range(0, len(years), 2), rotation=45, labels=years[::2])

plt.show()
```



Widac trend wzorstu energii wraz z latami. Slupki bledow sa mniej wiecej stale, co wskazuje na duza roznorodnosc. W latach 2016 - 2024 slupki sa troche mniejsze - trend jest bardziej skonsolidowany

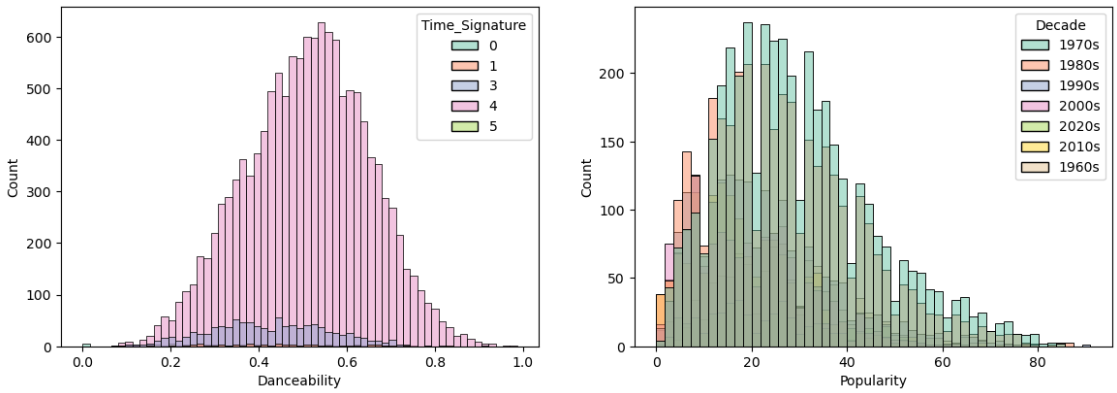
0.6 Histogramy

[188]:
```
hists = plt.figure(figsize=(14, 10))

ax1 = hists.add_subplot(2, 2, 1)
ax2 = hists.add_subplot(2, 2, 2)
ax3 = hists.add_subplot(2, 2, 3)
ax4 = hists.add_subplot(2, 2, 4)


sns.histplot(df, x="Year", ax=ax1)
sns.histplot(df, x="Popularity", bins=30, ax=ax2)
sns.histplot(df, x="Danceability", ax=ax3)
sns.histplot(df, x="Tempo", bins=50, ax=ax4)

plt.show()
```



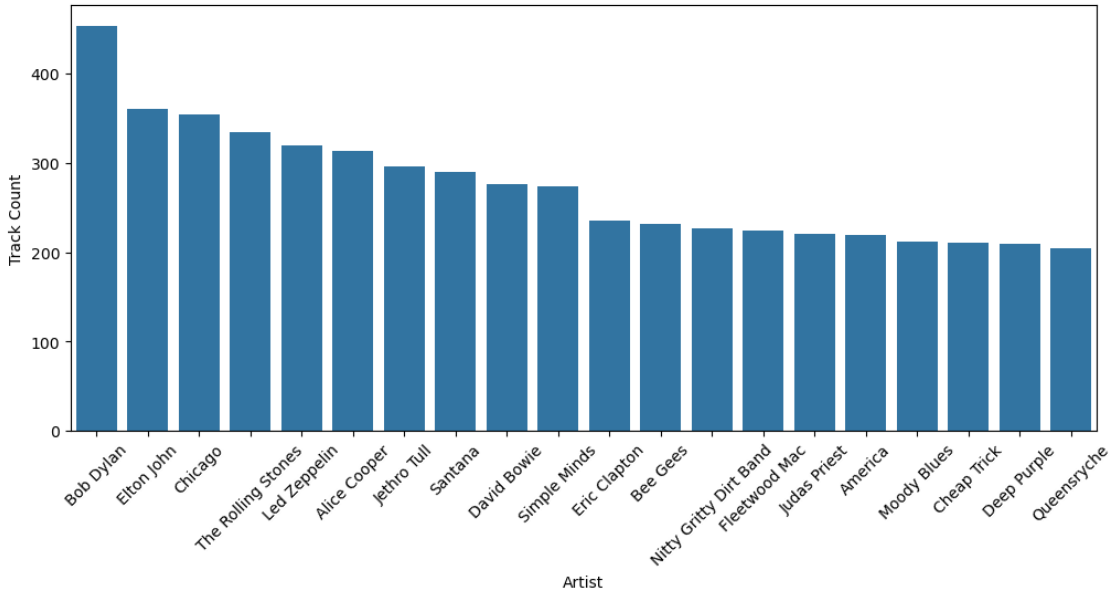
Z wykresow widac ze znaczna wiekszosc piosenek pochodzi z lat 1965 - 1985. Pozniej nastapilo znaczne zmiejszenie ilosci piosenek. Ponadto widac, ze duza wiekszosc piosenek trzyma sie w okolicy 10 - 30 jesli chodzi o popularnosc. Piosenek malo popularnych jest znacznie wiecej niz piosenek o wskazniku popularnosci w przedziale 50 - 80. Z wykresu opisujacego tempo widac ze najwiecej piosenek ma temp ok. 120, jest tez troche bardzo energicznych piosenek o tempie w okolicach 200.

[194]:
```
hists = plt.figure(figsize=(14, 10))

ax1 = hists.add_subplot(2, 2, 1)
ax2 = hists.add_subplot(2, 2, 2)

sns.histplot(df, x="Danceability", hue="Time_Signature", palette="Set2", ax=ax1)
sns.histplot(df, x="Popularity", bins=50, hue="Decade", palette="Set2", ax=ax2)

plt.show()
```



Najwięcej utworów ma metrum 4/4 (wartość 4) — klasyczne metrum większości muzyki pop/rock. Utwory w metrum 4 są najczęściej taneczne, ze szczytem gęstości ok. 0.55. Metrum 3 i inne (0, 1, 5) są znacznie mniej popularne i często mają niższą taneczność. Najwięcej utworów ma niską popularność — szczyt rozkładu to ok. 10–25 pkt. Niewiele utworów osiąga bardzo wysoką popularność (np. powyżej 70). Lata 1970s i 1980s dominują wśród najpopularniejszych utworów.

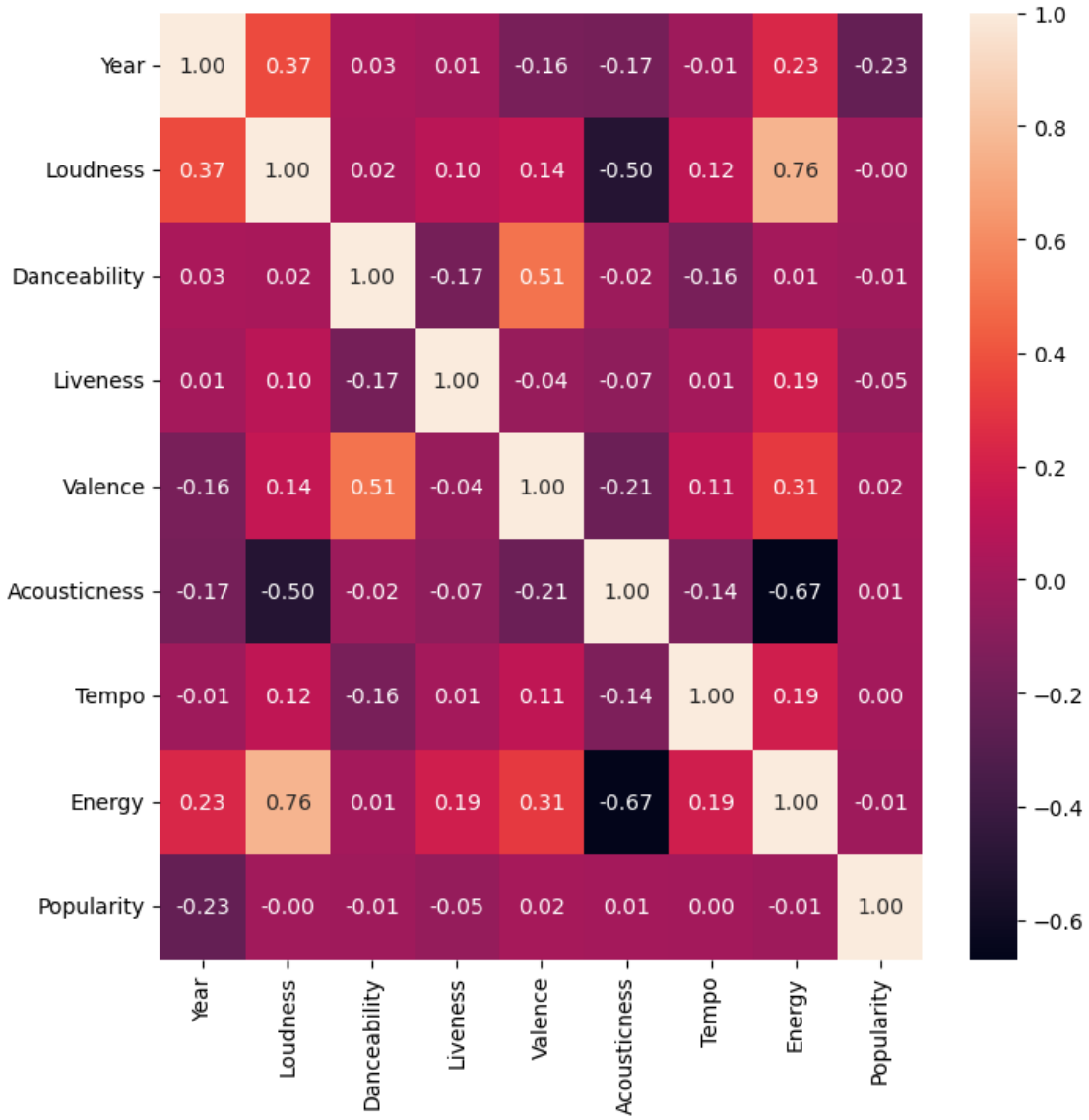
[26]:
```
artist_track_count = df['Artist'].value_counts()
top_20_artists = artist_track_count.nlargest(20)

plt.figure(figsize=(12, 5))
plt.xticks(rotation=45)

top_20_df = top_20_artists.reset_index()
top_20_df.columns = ['Artist', 'Track Count']

sns.barplot(top_20_df, x="Artist", y="Track Count")
plt.show()
```



Bob Dylan ma najwięcej utworów w zestawieniu — ponad 450. W top 5 sa Elton John, Chicago, The Rolling Stones i Led Zeppelin

0.7 Korelacje

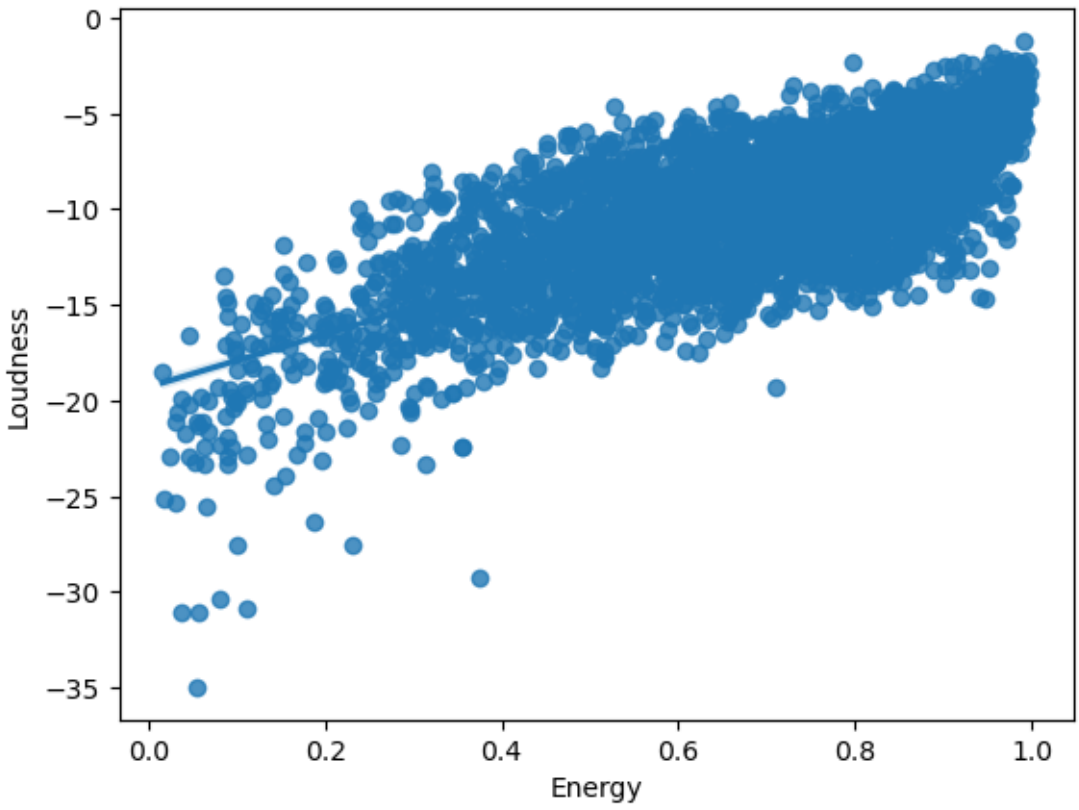
[24]:
```
plt.figure(figsize=(8, 8))
features = ["Year", "Loudness", "Danceability", "Liveness", "Valence",␣
 ↪"Acousticness", "Tempo", "Energy", "Popularity"]
sns.heatmap(df[features].corr(), annot=True, fmt=".2f")
plt.show()
```



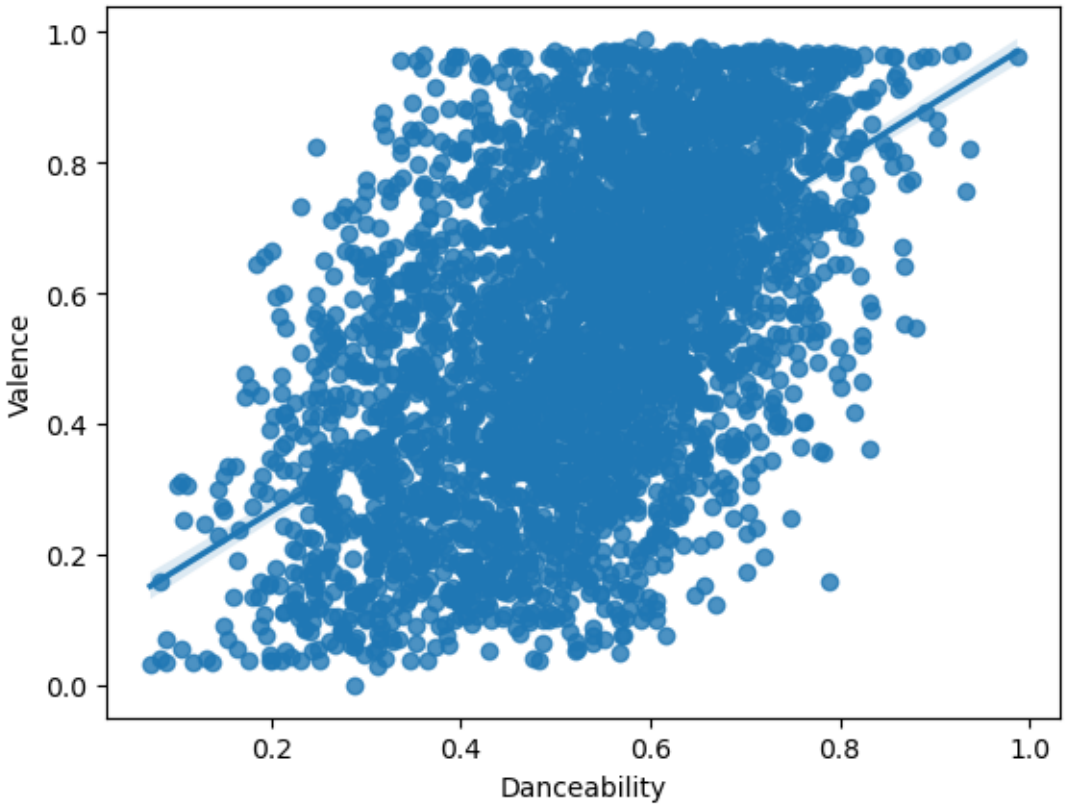
Najsilniejsze dodatnie korelacje zachodza miedzy Energy i Loudness oraz Danceability i Valence. Silne relacje ujemne zachodza miedzy Energy i Acousticness oraz Loudness i Acousticness. Co ciekawe, Popularity ma bardzo niska korelacje ze wszystkimi cechami, co oznacza ze zaden z tych czynnikow nie wplywa silnie na popularnosc utworu. Ciekawa jest tez niska korelacja miedzy Danceability i Energy oraz Tempo i Energy.

0.8 Regresja

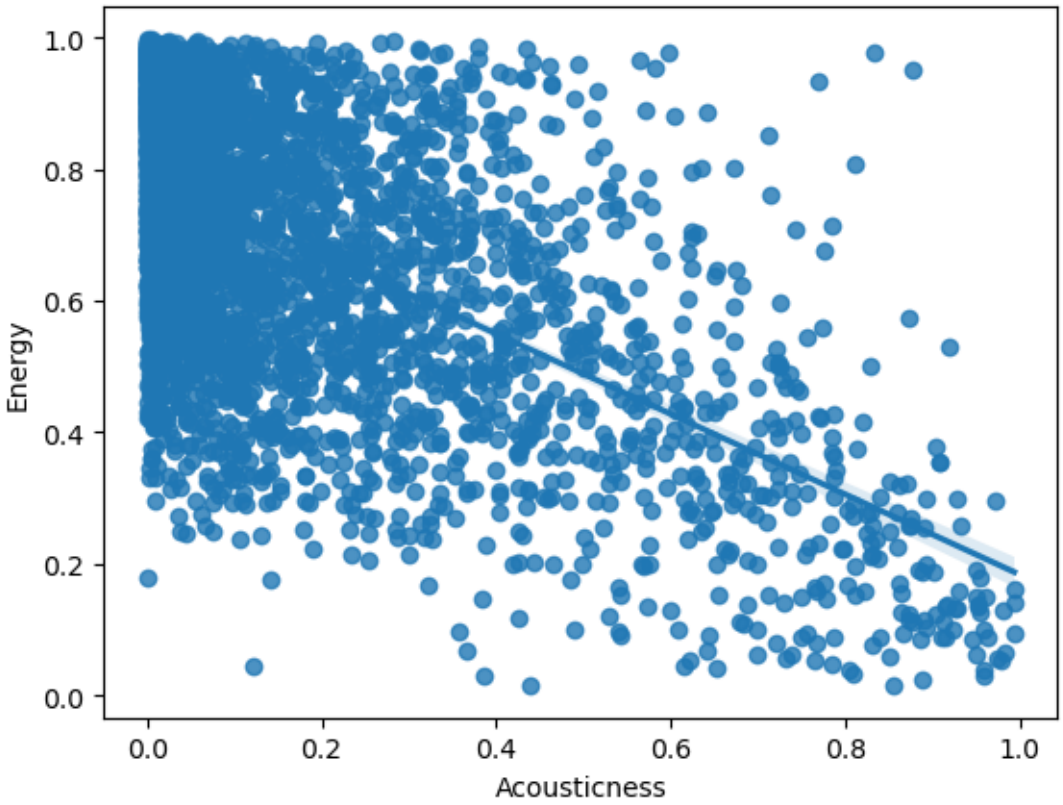
[19]:
```
df_80s = df[df['Decade'] == "1980s"]

sns.regplot(x="Energy", y="Loudness", data=df_80s)
plt.show()
```



[20]:
```
sns.regplot(x="Danceability", y="Valence", data=df_80s)
plt.show()
```



[23]:
```
sns.regplot(x="Acousticness", y="Energy", data=df_80s)
plt.show()
```



0.9 Wnioski

1. Największą liczbę utworów posiadają Bob Dylan, Elton John, The Rolling Stones czy Led Zeppelin
2. Większość utworów w zestawieniu charakteryzuje się stosunkowo niską popularnością, co może wynikać z dużej liczby mniej znanych nagrań
3. Popularność nie koreluje znacząco z żadną cechą dźwiękową, co sugeruje, że sukces utworu zależy od czynników pozamuzycznych
4. Między Energy a Loudness występuje silna dodatnia korelacja (r 0.76). Głośniejsze utwory są zwykle bardziej dynamiczne i intensywne
5. Współczesne utwory (po 1990 r.) mają tendencję do bycia głośniejszymi i bardziej energetycznymi, co potwierdzają zarówno regresje, jak i trend liniowy
6. Utwory rockowe mają najczęściej metrum 4/4, co potwierdza silne skupienie taneczności wokół wartości 0.5–0.6.
7. Akustyczność znacząco spada po latach 70., co jest zgodne z przejściem od instrumentów klasycznych do elektronicznych
8. Dane pokazują, że muzyka rockowa jest zróżnicowana, ale kierunkowo spójna, a jej odbiór (np. popularność) zależy od czegoś więcej niż tylko brzmienia

0.10 Importance

[196]:
```
from sklearn.ensemble import RandomForestRegressor

X = df.drop(columns=['Popularity']).select_dtypes(include='number')
y = df['Popularity']

model = RandomForestRegressor()
model.fit(X, y)

importances = pd.Series(model.feature_importances_, index=X.columns).
 ↪sort_values(ascending=False)

print(importances)
```

```
Year                0.153763
Loudness            0.108264
Danceability        0.092158
Liveness            0.089583
Valence             0.089408
Speechiness         0.088152
Acousticness        0.087473
Tempo               0.082381
Instrumentalness    0.078956
Energy              0.075928
Key                 0.040069
Mode                0.009249
Time_Signature      0.004615
dtype: float64
```