

exploratory-data-analysis

April 22, 2024

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

```
[2]: df=pd.read_csv("tracks.csv")
```

```
[3]: df.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
```

```
[4]: df.describe().transpose()
```

```
[4]:
```

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

	50%	75%	max
popularity	27.000000	41.00000	100.000
duration_ms	214893.000000	263867.00000	5621218.000
explicit	0.000000	0.00000	1.000
danceability	0.577000	0.68600	0.991
energy	0.549000	0.74800	1.000
key	5.000000	8.00000	11.000
loudness	-9.243000	-6.48200	5.376
mode	1.000000	1.00000	1.000
speechiness	0.044300	0.07630	0.971
acousticness	0.422000	0.78500	0.996
instrumentalness	0.000024	0.00955	1.000
liveness	0.139000	0.27800	1.000
valence	0.564000	0.76900	1.000
tempo	117.384000	136.32100	246.381
time_signature	4.000000	4.00000	5.000

```
[5]: df.set_index("release_date", inplace=True)
```

```
[6]: df.index=pd.to_datetime(df.index, format='mixed')
```

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

```
[7]:
```

	id	name \
release_date		
1922-02-22	35iwgR4jXetI318WEWsa1Q	Carve
1922-06-01	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista
1922-03-21	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado
1922-03-21	08FmqUhxyLTn6pAh6bk45	El Prisionero - Remasterizado
1922-01-01	08y9GfoqCWf0GsKdwojr5e	Lady of the Evening

	popularity	duration_ms	explicit	artists \
release_date				
1922-02-22	6	126903	0	['Uli']
1922-06-01	0	98200	0	['Fernando Pessoa']
1922-03-21	0	181640	0	['Ignacio Corsini']
1922-03-21	0	176907	0	['Ignacio Corsini']
1922-01-01	0	163080	0	['Dick Haymes']

	id_artists	danceability	energy	key	loudness \
release_date					
1922-02-22	['45tIt06XoIOIio4LBEVpls']	0.645	0.4450	0	-13.338
1922-06-01	['14jtPCOoNZwqk5wd9DxrY']	0.695	0.2630	0	-22.136
1922-03-21	['5Li0oJbxVSAMkBS2fUm3X2']	0.434	0.1770	1	-21.180
1922-03-21	['5Li0oJbxVSAMkBS2fUm3X2']	0.321	0.0946	7	-27.961
1922-01-01	['3BiJGZsyX9sJchTqcSA7Su']	0.402	0.1580	3	-16.900

	mode	speechiness	acousticness	instrumentalness	liveness \
release_date					
1922-02-22	1	0.4510	0.674	0.7440	0.151
1922-06-01	1	0.9570	0.797	0.0000	0.148
1922-03-21	1	0.0512	0.994	0.0218	0.212
1922-03-21	1	0.0504	0.995	0.9180	0.104
1922-01-01	0	0.0390	0.989	0.1300	0.311

	valence	tempo	time_signature
release_date			
1922-02-22	0.127	104.851	3
1922-06-01	0.655	102.009	1
1922-03-21	0.457	130.418	5
1922-03-21	0.397	169.980	3
1922-01-01	0.196	103.220	4

```
[8]: df[["artists"]].iloc[-1]
```

```
[8]: artists      ['Afrosound']
      Name: 2015-07-01 00:00:00, dtype: object
```

```
[9]: df[["artists"]].iloc[0]
```

```
[9]: artists      ['Uli']
      Name: 1922-02-22 00:00:00, dtype: object
```

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

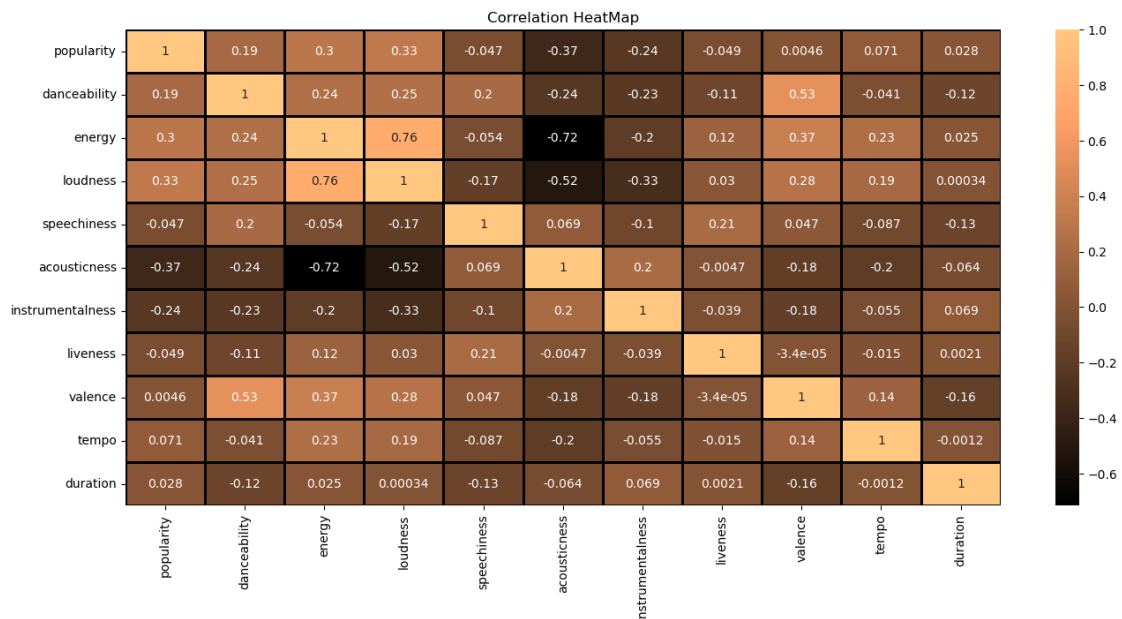
```
[11]: df.duration.head()
```

```
[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
```

1 Feature Analysis

```
[12]: corr_df=df.
      ↪drop(["key","mode","id","name","artists","time_signature","id_artists","explicit"],axis=1).
      ↪corr(method="pearson")
      plt.figure(figsize=(16,7))
      heatmap=sns.heatmap(corr_df,annot=True,cmap="copper", linewidths=1,
      ↪linecolor="Black")
      heatmap.set_title("Correlation HeatMap")
```

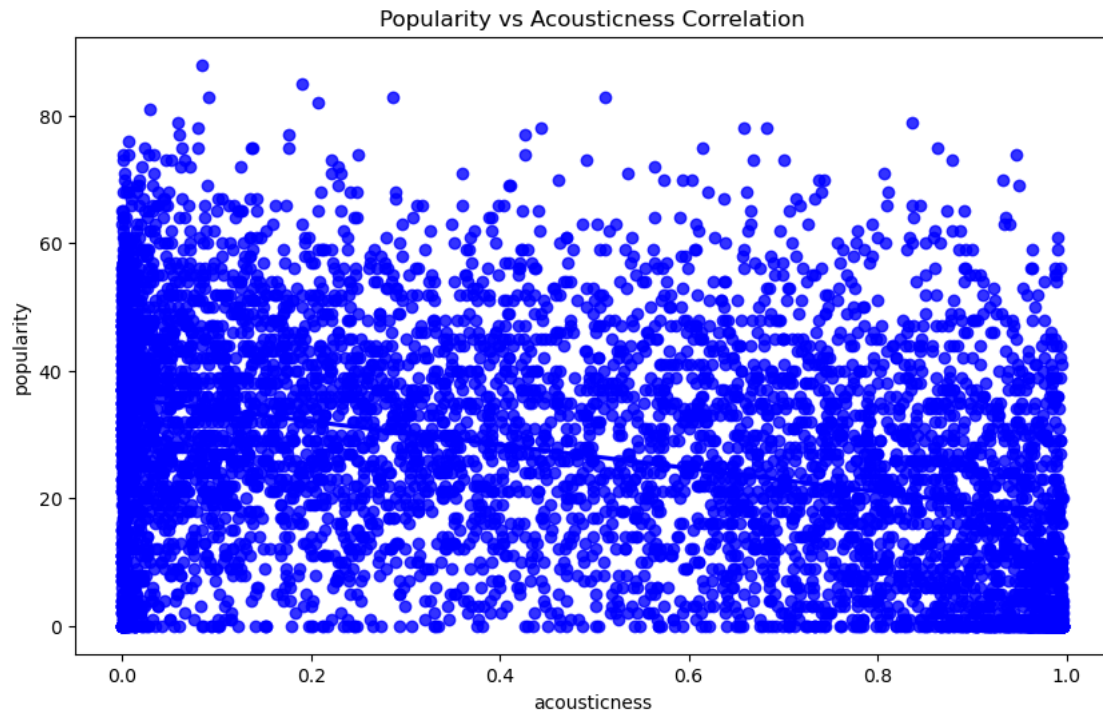
```
[12]: Text(0.5, 1.0, 'Correlation HeatMap')
```



```
[13]: sample_df=df.sample(int(0.01*len(df)))
```

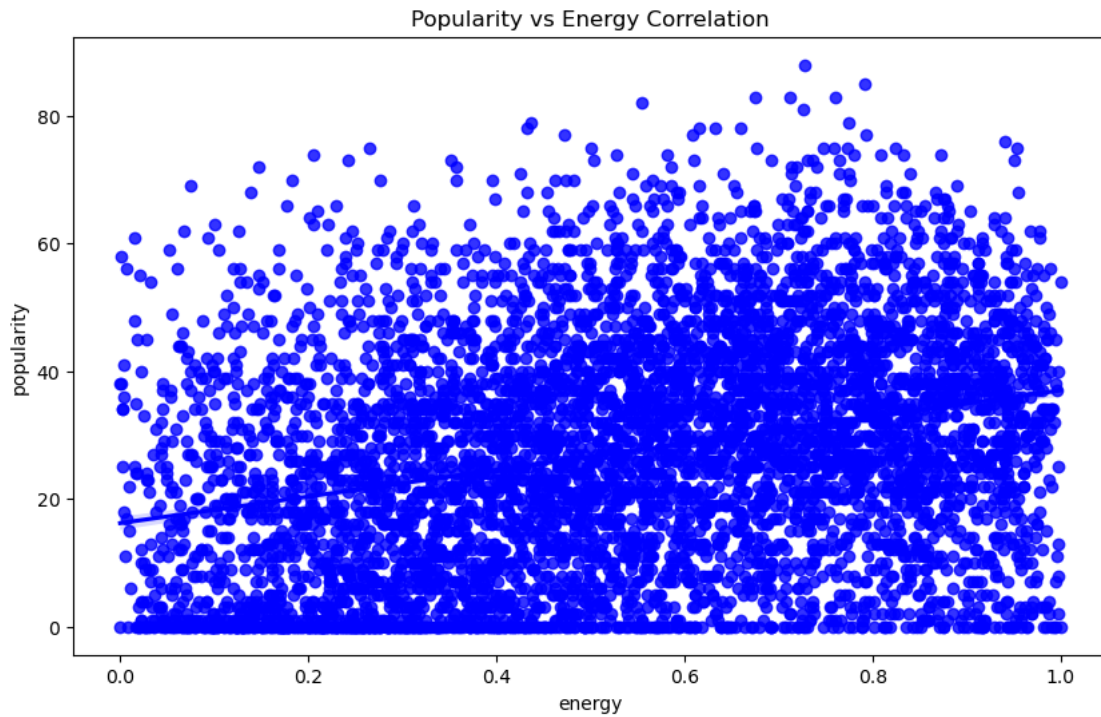
```
[14]: plt.figure(figsize=(10,6))
      sns.regplot(data=sample_df, y="popularity", x="acousticness",color="b").
      ↪set(title="Popularity vs Acousticness Correlation")
```

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



```
[15]: plt.figure(figsize=(10,6))
sns.regplot(data=sample_df, y="popularity", x="energy", color="b").
    ↪set(title="Popularity vs Energy Correlation")
```

```
[15]: [Text(0.5, 1.0, 'Popularity vs Energy Correlation')]
```

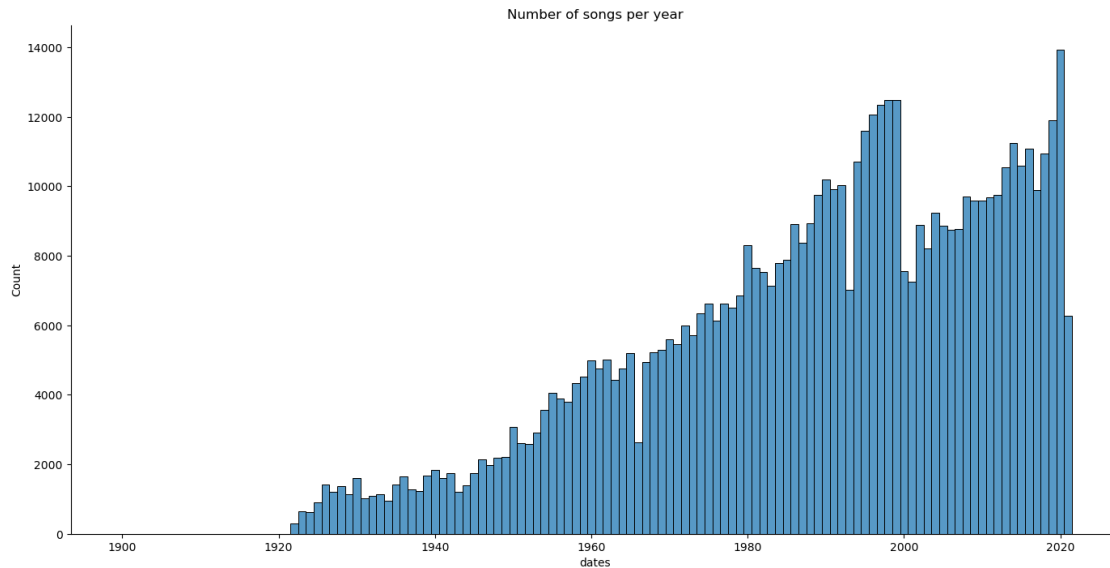


```
[16]: df['dates']=df.index.get_level_values('release_date')
      df.dates=pd.to_datetime(df.dates)
      years=df.dates.dt.year
```

```
[17]: sns.displot(years,discrete=True,aspect=2,height=7 ,kind="hist").
      ↪set(title="Number of songs per year")
```

C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)

```
[17]: <seaborn.axisgrid.FacetGrid at 0x22882f1c390>
```

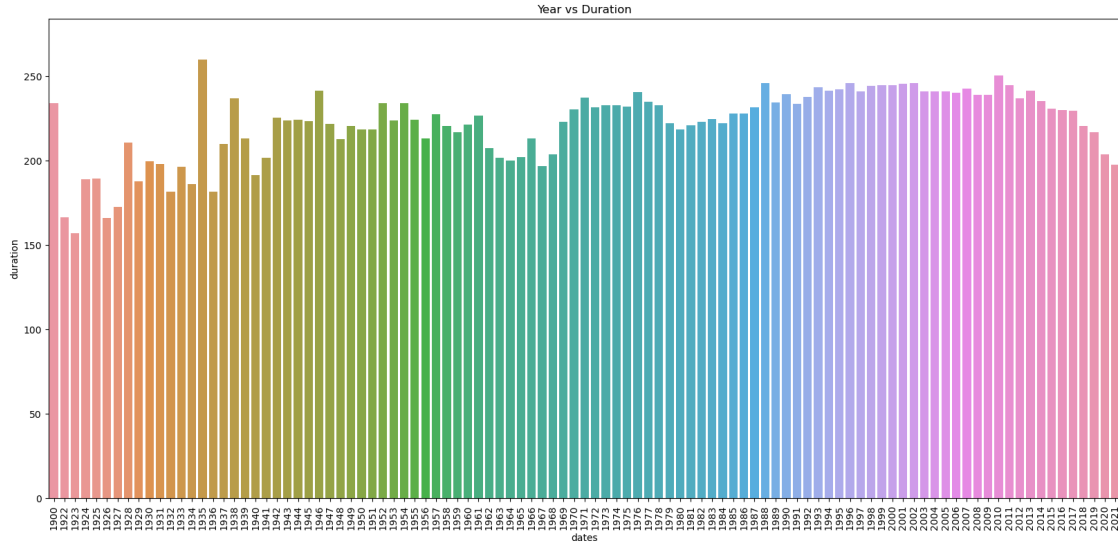


```
[18]: dr=df.duration
fig_dims = (20, 9)
fig, ax = plt.subplots(figsize=fig_dims)
fig=sns.barplot(x=years, y=dr,ax=ax,errwidth=False).set(title="Year vs
↳Duration")
plt.xticks(rotation=90)
```

```
[18]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12,
               13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
               26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
               39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
               52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
               65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77,
               78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90,
               91, 92, 93, 94, 95, 96, 97, 98, 99, 100]),
       [Text(0, 0, '1900'),
        Text(1, 0, '1922'),
        Text(2, 0, '1923'),
        Text(3, 0, '1924'),
        Text(4, 0, '1925'),
        Text(5, 0, '1926'),
        Text(6, 0, '1927'),
        Text(7, 0, '1928'),
        Text(8, 0, '1929'),
        Text(9, 0, '1930'),
        Text(10, 0, '1931'),
        Text(11, 0, '1932'),
        Text(12, 0, '1933'),
```

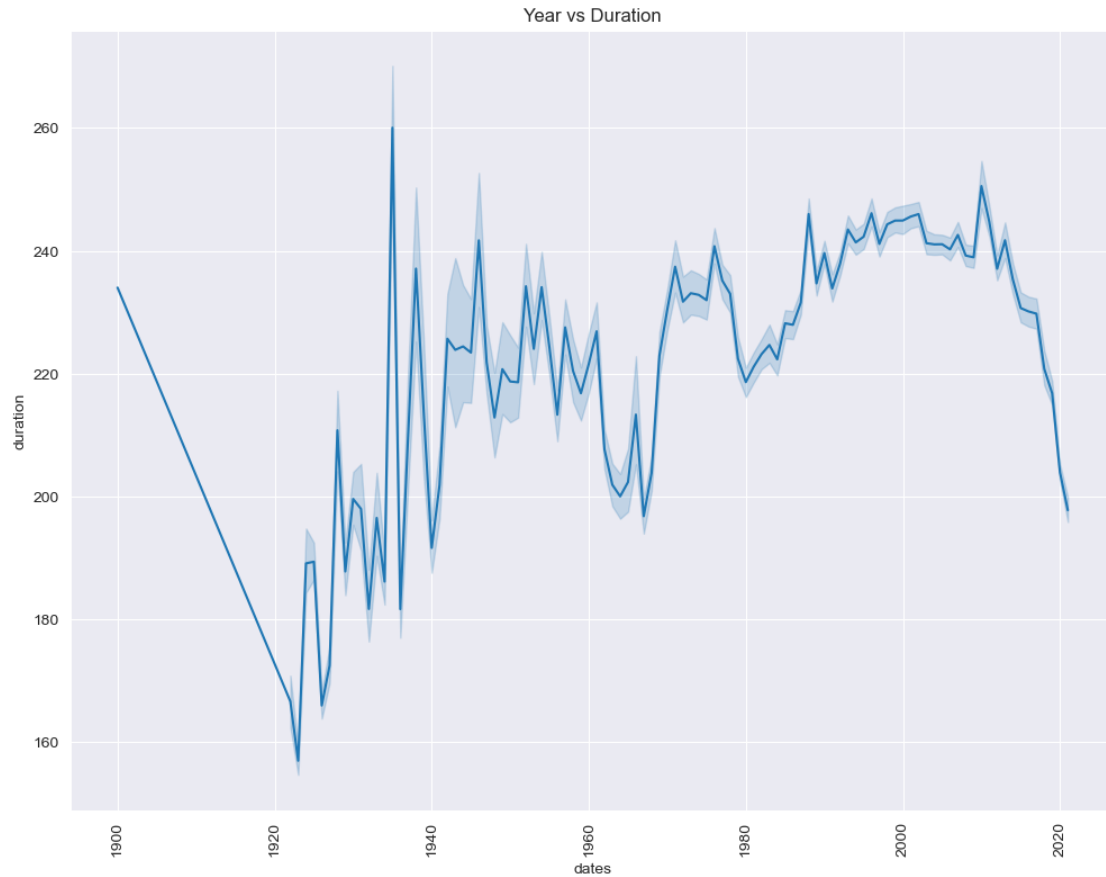
Text(13, 0, '1934'),
Text(14, 0, '1935'),
Text(15, 0, '1936'),
Text(16, 0, '1937'),
Text(17, 0, '1938'),
Text(18, 0, '1939'),
Text(19, 0, '1940'),
Text(20, 0, '1941'),
Text(21, 0, '1942'),
Text(22, 0, '1943'),
Text(23, 0, '1944'),
Text(24, 0, '1945'),
Text(25, 0, '1946'),
Text(26, 0, '1947'),
Text(27, 0, '1948'),
Text(28, 0, '1949'),
Text(29, 0, '1950'),
Text(30, 0, '1951'),
Text(31, 0, '1952'),
Text(32, 0, '1953'),
Text(33, 0, '1954'),
Text(34, 0, '1955'),
Text(35, 0, '1956'),
Text(36, 0, '1957'),
Text(37, 0, '1958'),
Text(38, 0, '1959'),
Text(39, 0, '1960'),
Text(40, 0, '1961'),
Text(41, 0, '1962'),
Text(42, 0, '1963'),
Text(43, 0, '1964'),
Text(44, 0, '1965'),
Text(45, 0, '1966'),
Text(46, 0, '1967'),
Text(47, 0, '1968'),
Text(48, 0, '1969'),
Text(49, 0, '1970'),
Text(50, 0, '1971'),
Text(51, 0, '1972'),
Text(52, 0, '1973'),
Text(53, 0, '1974'),
Text(54, 0, '1975'),
Text(55, 0, '1976'),
Text(56, 0, '1977'),
Text(57, 0, '1978'),
Text(58, 0, '1979'),
Text(59, 0, '1980'),


```
Text(60, 0, '1981'),
Text(61, 0, '1982'),
Text(62, 0, '1983'),
Text(63, 0, '1984'),
Text(64, 0, '1985'),
Text(65, 0, '1986'),
Text(66, 0, '1987'),
Text(67, 0, '1988'),
Text(68, 0, '1989'),
Text(69, 0, '1990'),
Text(70, 0, '1991'),
Text(71, 0, '1992'),
Text(72, 0, '1993'),
Text(73, 0, '1994'),
Text(74, 0, '1995'),
Text(75, 0, '1996'),
Text(76, 0, '1997'),
Text(77, 0, '1998'),
Text(78, 0, '1999'),
Text(79, 0, '2000'),
Text(80, 0, '2001'),
Text(81, 0, '2002'),
Text(82, 0, '2003'),
Text(83, 0, '2004'),
Text(84, 0, '2005'),
Text(85, 0, '2006'),
Text(86, 0, '2007'),
Text(87, 0, '2008'),
Text(88, 0, '2009'),
Text(89, 0, '2010'),
Text(90, 0, '2011'),
Text(91, 0, '2012'),
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']]
```



```
[19]: dr=df.duration
sns.set_style(style="darkgrid")
fig_dims = (12, 9)
fig, ax = plt.subplots(figsize=fig_dims)
fig=sns.lineplot(x=years, y=dr,ax=ax).set(title="Year vs Duration")
plt.xticks(rotation=90)
```

```
[19]: (array([1880., 1900., 1920., 1940., 1960., 1980., 2000., 2020., 2040.]),
      [Text(1880.0, 0, '1880'),
       Text(1900.0, 0, '1900'),
       Text(1920.0, 0, '1920'),
       Text(1940.0, 0, '1940'),
       Text(1960.0, 0, '1960'),
       Text(1980.0, 0, '1980'),
       Text(2000.0, 0, '2000'),
       Text(2020.0, 0, '2020'),
       Text(2040.0, 0, '2040')])
```



```
[20]: df[df["duration"]==df["duration"].min()].iloc[0]
```

```
[20]: id                2s6e7KLoQ5hie3Cnh73v2v
      name                Pause Track
      popularity                0
      explicit                0
      artists              ['Louis Armstrong']
      id_artists            ['19eLuQmk9aCobbVDHc6eek']
      danceability            0.0
      energy                  0.0
      key                    0
      loudness                -60.0
      mode                    0
      speechiness             0.0
      acousticness            0.0
      instrumentalness         0.0
      liveness                0.0
      valence                  0.0
      tempo                   0.0
```

```

time_signature      0
duration            3
dates              1925-01-01 00:00:00
Name: 1925-01-01 00:00:00, dtype: object

```

```
[21]: df[df["duration"]==df["duration"].max()].iloc[0]
```

```

[21]: id          3EEv9UCeZdn4MVFv8ts01E
name
popularity      3
explicit        0
artists         ['   ']
id_artists      ['2ySk9zib3PuomvMGmCqdTA']
danceability    0.638
energy          0.537
key             8
loudness       -13.365
mode            1
speechiness     0.775
acousticness    0.825
instrumentalness 0.0
liveness        0.345
valence         0.401
tempo          131.446
time_signature  3
duration        5621
dates           1979-07-28 00:00:00
Name: 1979-07-28 00:00:00, dtype: object

```

```

[22]: decades_df=df.drop(["id","name","id_artists","artists"],axis=1).resample("10A").
      ↪mean()
decades_df.index=[f"{date_index-1}'s " for date_index in decades_df.index.year]
decades_df

```

```

[22]:
      popularity  explicit  danceability  energy  key  loudness \
1899's    19.000000  0.000000    0.659000  0.791000  2.000000 -4.895000
1909's         NaN         NaN         NaN         NaN         NaN         NaN
1919's         NaN         NaN         NaN         NaN         NaN         NaN
1929's     1.154622  0.003038    0.591392  0.290079  5.061849 -14.467812
1939's     2.101966  0.001054    0.546994  0.307452  5.129904 -13.437494
1949's     2.076927  0.001298    0.478537  0.266274  5.266078 -14.400397
1959's     9.366773  0.000429    0.482315  0.305697  5.045977 -14.267081
1969's    19.129777  0.000752    0.497264  0.417778  5.077728 -12.330709
1979's    24.321556  0.002076    0.530289  0.509039  5.129180 -11.389069
1989's    25.706795  0.012112    0.563001  0.552254  5.184684 -11.325755
1999's    30.286927  0.033560    0.574754  0.580257  5.251843 -10.089910
2009's    36.797246  0.050843    0.591064  0.651000  5.315793  -7.465232

```

2019's	40.032942	0.135694	0.615645	0.656647	5.335338	-7.319048
2029's	35.191848	0.260627	0.671531	0.617998	5.269543	-7.898172

	mode	speechiness	acousticness	instrumentalness	liveness	\
1899's	1.000000	0.029500	0.139000	0.000002	0.161000	
1909's	NaN	NaN	NaN	NaN	NaN	
1919's	NaN	NaN	NaN	NaN	NaN	
1929's	0.731011	0.262685	0.896172	0.326337	0.207478	
1939's	0.713457	0.187095	0.870396	0.280580	0.216754	
1949's	0.705580	0.114451	0.901684	0.357791	0.214518	
1959's	0.704834	0.099794	0.829466	0.236285	0.211956	
1969's	0.710525	0.074259	0.657810	0.151731	0.214114	
1979's	0.690587	0.112379	0.487428	0.098135	0.223204	
1989's	0.666956	0.130598	0.402808	0.080092	0.224767	
1999's	0.662785	0.099696	0.360872	0.072077	0.216725	
2009's	0.637965	0.079916	0.311532	0.055901	0.210008	
2019's	0.595547	0.094608	0.284220	0.094913	0.203696	
2029's	0.520936	0.130416	0.278645	0.119411	0.172572	

	valence	tempo	time_signature	duration	\
1899's	0.956000	141.999000	4.000000	234.000000	
1909's	NaN	NaN	NaN	NaN	
1919's	NaN	NaN	NaN	NaN	
1929's	0.598686	112.810796	3.782227	185.289280	
1939's	0.570885	112.393005	3.782438	206.058513	
1949's	0.498483	107.313873	3.743213	221.555982	
1959's	0.496949	111.064630	3.778960	222.952709	
1969's	0.565004	114.394391	3.801876	210.820786	
1979's	0.582910	117.787538	3.848624	231.113428	
1989's	0.577633	118.906802	3.868787	230.590793	
1999's	0.569688	119.713682	3.889851	242.100170	
2009's	0.562085	121.572760	3.924630	242.592185	
2019's	0.509203	122.129494	3.934927	227.986914	
2029's	0.506228	120.758488	3.942843	197.759911	

	dates
1899's	1900-01-01 00:00:00.000000000
1909's	NaT
1919's	NaT
1929's	1927-02-10 20:41:05.624999936
1939's	1936-03-17 06:59:01.170268800
1949's	1946-05-11 10:31:41.188684160
1959's	1956-03-28 06:42:41.239645056
1969's	1965-11-14 14:30:05.754403568
1979's	1976-01-06 02:55:57.605985024
1989's	1986-01-15 23:27:33.758282432
1999's	1995-11-30 22:58:45.342420736

```
2009's 2006-01-06 10:45:07.503291008
2019's 2016-04-02 22:44:44.666697216
2029's 2021-03-06 22:53:44.582073088
```

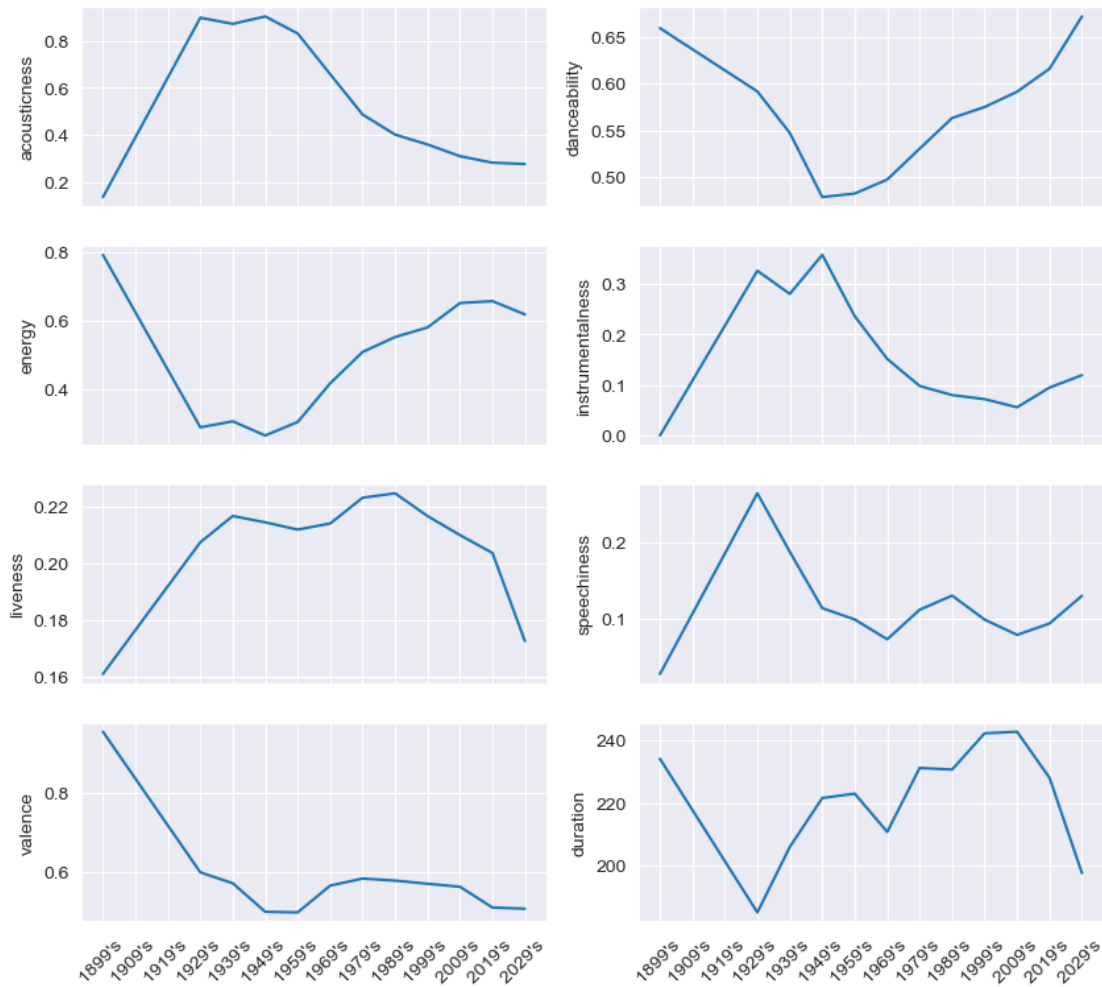
```
[23]: trend_df=decades_df[["acousticness", "danceability", "energy",
    ↪ "instrumentalness", "liveness", "speechiness", "valence", "duration"]]
trend_df
```

```
[23]:
```

	acousticness	danceability	energy	instrumentalness	liveness	\
1899's	0.139000	0.659000	0.791000	0.000002	0.161000	
1909's	NaN	NaN	NaN	NaN	NaN	
1919's	NaN	NaN	NaN	NaN	NaN	
1929's	0.896172	0.591392	0.290079	0.326337	0.207478	
1939's	0.870396	0.546994	0.307452	0.280580	0.216754	
1949's	0.901684	0.478537	0.266274	0.357791	0.214518	
1959's	0.829466	0.482315	0.305697	0.236285	0.211956	
1969's	0.657810	0.497264	0.417778	0.151731	0.214114	
1979's	0.487428	0.530289	0.509039	0.098135	0.223204	
1989's	0.402808	0.563001	0.552254	0.080092	0.224767	
1999's	0.360872	0.574754	0.580257	0.072077	0.216725	
2009's	0.311532	0.591064	0.651000	0.055901	0.210008	
2019's	0.284220	0.615645	0.656647	0.094913	0.203696	
2029's	0.278645	0.671531	0.617998	0.119411	0.172572	

	speechiness	valence	duration
1899's	0.029500	0.956000	234.000000
1909's	NaN	NaN	NaN
1919's	NaN	NaN	NaN
1929's	0.262685	0.598686	185.289280
1939's	0.187095	0.570885	206.058513
1949's	0.114451	0.498483	221.555982
1959's	0.099794	0.496949	222.952709
1969's	0.074259	0.565004	210.820786
1979's	0.112379	0.582910	231.113428
1989's	0.130598	0.577633	230.590793
1999's	0.099696	0.569688	242.100170
2009's	0.079916	0.562085	242.592185
2019's	0.094608	0.509203	227.986914
2029's	0.130416	0.506228	197.759911

```
[24]: i=0
f, axes = plt.subplots(4, 2, figsize=(10, 9), sharex=True)
for ax in f.axes:
    sns.lineplot(x=trend_df.index, y=trend_df.iloc[:,i], data=trend_df, ax=ax)
    plt.sca(ax)
    plt.xticks(rotation=45)
    i=i+1
```



```
[25]: key={0:"C",1:"C ",2:"D",3:"D ",4:"E",5:"F",6:"F ",7:"G",8:"G ",9:"A",10:"A ",11:
      ↪ "B"}
      key_df=pd.DataFrame(df["key"].value_counts())
```

```
[26]: key_df.head()
```

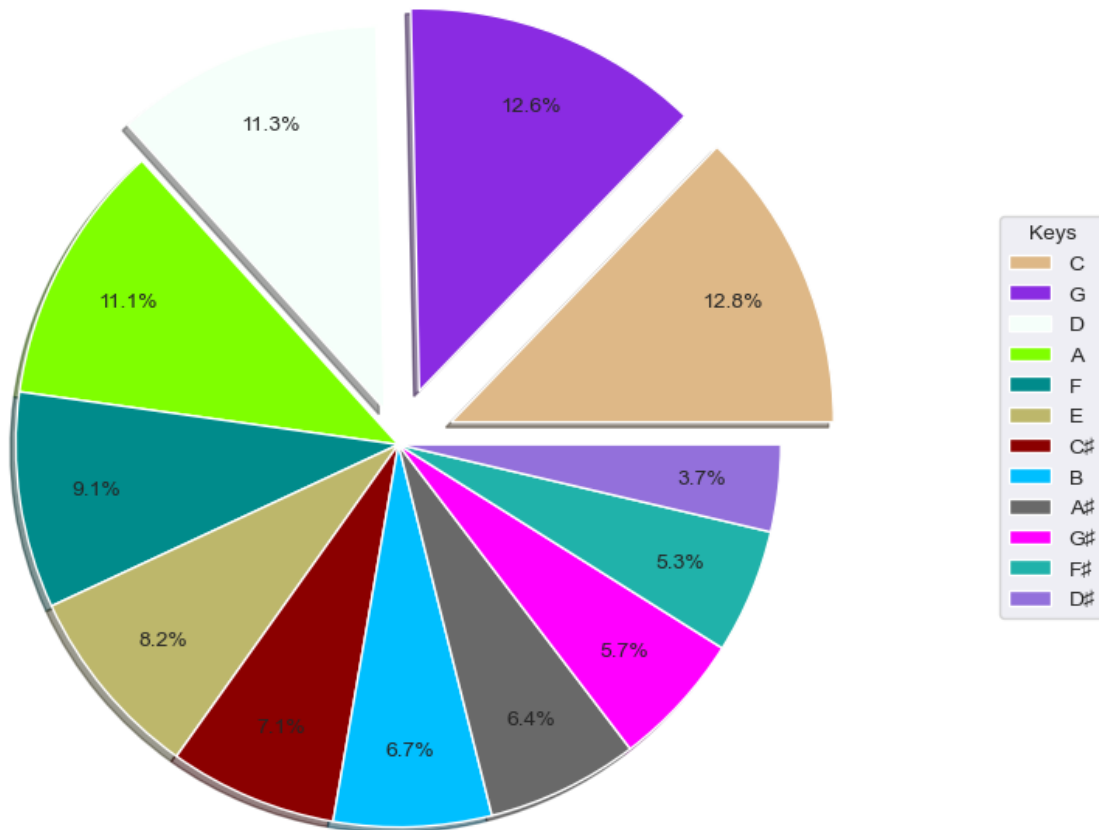
```
[26]:      count
key
0      74950
7      73779
2      66552
9      65128
5      53614
```

```
[27]: key_df["key names"]= key_df.index.to_series().map(key)
      key_df
```

```
[27]:      count key names
      key
0      74950      C
7      73779      G
2      66552      D
9      65128      A
5      53614      F
4      48220      E
1      41736      C
11     39132      B
10     37710      A
8      33460      G
6      30856      F
3      21535      D
```

```
[28]: colors=["#DEB887", "#8A2BE2", "#F5FFFA", "#7FFF00", "#008B8B", "#BDB76B",
      ↪ "#8B0000", "#00BFFF", "#696969", "#FF00FF", "#20B2AA", "#9370DB"]
plt.figure(figsize=(12,4))
key_labels=key_df["key names"].values
key_values=key_df["count"].values
plt.pie(key_values,shadow=True,pctdistance=0.8, autopct="%.
  ↪ 1f%%",radius=2,explode=(0.3,0.3,0.
  ↪ 2,0,0,0,0,0,0,0,0,0),center=(2,3),colors=colors)
plt.legend(labels=key_labels, bbox_to_anchor=(2,1),title="Keys")
```

```
[28]: <matplotlib.legend.Legend at 0x2289897e250>
```

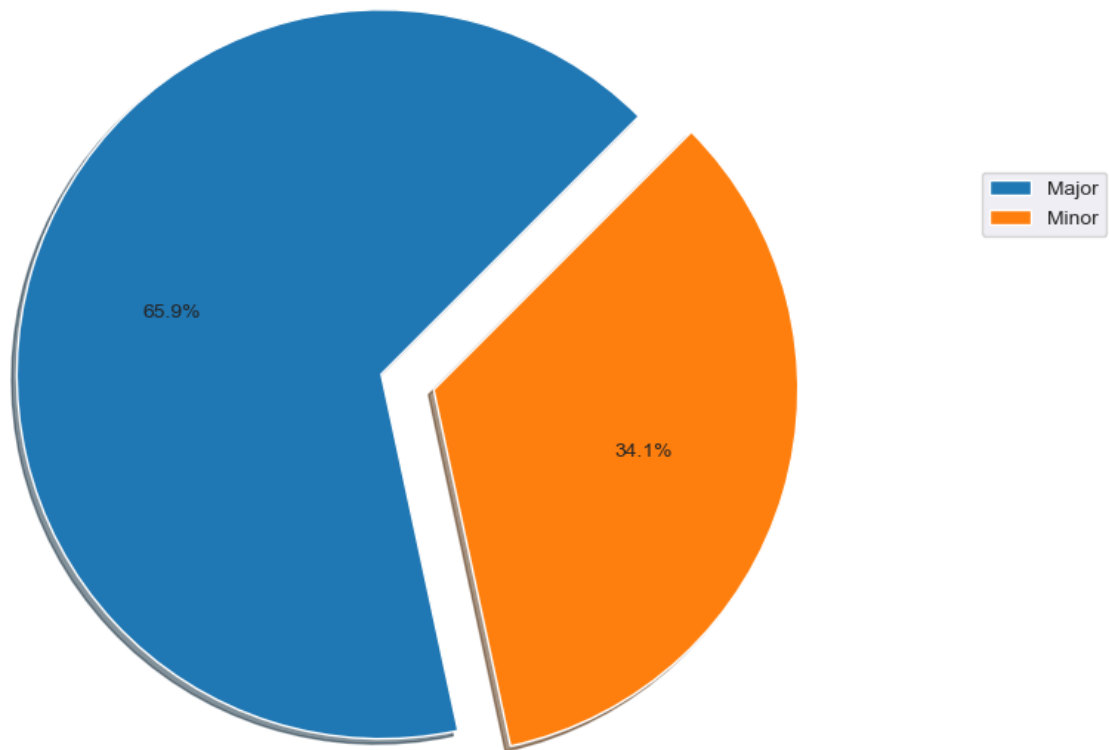



```
[29]: mode={0:"Minor", 1:"Major"}
mode_df=pd.DataFrame(df["mode"].value_counts())
mode_df["Mode names"]=mode_df.index.to_series().map(mode)
mode_df
```

```
[29]:      count Mode names
mode
1      386498      Major
0       200174      Minor
```

```
[30]: plt.figure(figsize=(10,4))
mode_labels=mode_df["Mode names"].values
mode_values=mode_df["count"].values
plt.pie(mode_values,shadow=True,autopct="%.1f%%",radius=2,explode=(0.
↪3,0),center=(2,1), startangle=45)
plt.legend(labels=mode_labels, bbox_to_anchor=(2,1))
```

```
[30]: <matplotlib.legend.Legend at 0x22898a9b490>
```



2 Genre Based Analysis

```
[31]: artists_df = pd.read_csv("artists.csv")
      tracks_df = pd.read_csv("tracks.csv")
      df = pd.concat([tracks_df, artists_df["genres"]], axis=1)
```

```
[32]: df.
      ↪drop(["id", "name", "explicit", "artists", "id_artists", "release_date", "time_signature"], axis=1)
```

```
[32]:
```

	popularity	duration_ms	danceability	energy	key	loudness	mode	\
0	6.0	126903.0	0.645	0.4450	0.0	-13.338	1.0	
1	0.0	98200.0	0.695	0.2630	0.0	-22.136	1.0	
2	0.0	181640.0	0.434	0.1770	1.0	-21.180	1.0	
3	0.0	176907.0	0.321	0.0946	7.0	-27.961	1.0	
4	0.0	163080.0	0.402	0.1580	3.0	-16.900	0.0	
...	
1162090	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1162091	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1162092	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1162093	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1162094	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

	speechiness	acousticness	instrumentalness	liveness	valence	\
0	0.4510	0.674	0.7440	0.151	0.127	
1	0.9570	0.797	0.0000	0.148	0.655	
2	0.0512	0.994	0.0218	0.212	0.457	
3	0.0504	0.995	0.9180	0.104	0.397	
4	0.0390	0.989	0.1300	0.311	0.196	
...	
1162090	NaN	NaN	NaN	NaN	NaN	
1162091	NaN	NaN	NaN	NaN	NaN	
1162092	NaN	NaN	NaN	NaN	NaN	
1162093	NaN	NaN	NaN	NaN	NaN	
1162094	NaN	NaN	NaN	NaN	NaN	

	tempo	genres
0	104.851	[]
1	102.009	[]
2	130.418	[]
3	169.980	[]
4	103.220	[]
...
1162090	NaN	['black comedy']
1162091	NaN	[]
1162092	NaN	[]
1162093	NaN	['black comedy']
1162094	NaN	['new comedy']

[1162095 rows x 14 columns]

```
[33]: df.isna().any()
```

```
[33]: id                True
      name              True
      popularity        True
      duration_ms       True
      explicit          True
      artists           True
      id_artists        True
      release_date      True
      danceability      True
      energy             True
      key               True
      loudness           True
      mode              True
      speechiness       True
      acousticness      True
      instrumentalness  True
      liveness           True
```

```

valence          True
tempo            True
time_signature   True
genres           False
dtype: bool

```

```
[34]: genre_df=df.dropna()
```

```
[37]: genre_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 586601 entries, 0 to 586671
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   id                    586601 non-null object
 1   name                  586601 non-null object
 2   popularity            586601 non-null float64
 3   duration_ms          586601 non-null float64
 4   explicit              586601 non-null float64
 5   artists               586601 non-null object
 6   id_artists            586601 non-null object
 7   release_date          586601 non-null object
 8   danceability          586601 non-null float64
 9   energy                586601 non-null float64
10   key                   586601 non-null float64
11   loudness              586601 non-null float64
12   mode                  586601 non-null float64
13   speechiness           586601 non-null float64
14   acousticness          586601 non-null float64
15   instrumentalness      586601 non-null float64
16   liveness              586601 non-null float64
17   valence               586601 non-null float64
18   tempo                 586601 non-null float64
19   time_signature        586601 non-null float64
20   genres                586601 non-null object
dtypes: float64(15), object(6)
memory usage: 98.5+ MB

```

```

[38]: genre_df['duration'] = genre_df['duration_ms'].apply(lambda x:round(x/1000))
genre_df.drop(['genres','key','mode'],axis=1).describe().transpose().
      ↪sort_index()

```

```

C:\Users\Admin\AppData\Local\Temp\ipykernel_10560\2083305419.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
genre_df['duration'] = genre_df['duration_ms'].apply(lambda x:round(x/1000))
```

```
[38]:
```

	count	mean	std	min	25%	\
acousticness	586601.0	0.449803	0.348812	0.0	0.0969	
danceability	586601.0	0.563612	0.166101	0.0	0.4530	
duration	586601.0	230.054333	126.532822	3.0	175.0000	
duration_ms	586601.0	230054.852626	126532.824981	3344.0	175083.0000	
energy	586601.0	0.542071	0.251910	0.0	0.3430	
explicit	586601.0	0.044091	0.205298	0.0	0.0000	
instrumentalness	586601.0	0.113425	0.266843	0.0	0.0000	
liveness	586601.0	0.213933	0.184328	0.0	0.0983	
loudness	586601.0	-10.205789	5.089422	-60.0	-12.8910	
popularity	586601.0	27.573212	18.369417	0.0	13.0000	
speechiness	586601.0	0.104870	0.179902	0.0	0.0340	
tempo	586601.0	118.467930	29.762942	0.0	95.6060	
time_signature	586601.0	3.873410	0.473112	0.0	4.0000	
valence	586601.0	0.552306	0.257673	0.0	0.3460	

	50%	75%	max
acousticness	0.422000	0.78400	0.996
danceability	0.577000	0.68600	0.991
duration	215.000000	264.00000	5621.000
duration_ms	214907.000000	263867.00000	5621218.000
energy	0.549000	0.74800	1.000
explicit	0.000000	0.00000	1.000
instrumentalness	0.000024	0.00955	1.000
liveness	0.139000	0.27800	1.000
loudness	-9.242000	-6.48100	5.376
popularity	27.000000	41.00000	100.000
speechiness	0.044300	0.07630	0.971
tempo	117.387000	136.32400	246.381
time_signature	4.000000	4.00000	5.000
valence	0.564000	0.76900	1.000

```
[39]: genre_df.head()
```

```
[39]:
```

	id	name	popularity	\
0	35iwgR4jXetI318WEWsa1Q	Carve	6.0	
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0.0	
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0.0	
3	08FmqUhxytLTn6pAh6bk45	El Prisionero - Remasterizado	0.0	
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0.0	

duration_ms	explicit	artists	id_artists	\
-------------	----------	---------	------------	---

0	126903.0	0.0	['Uli']	['45tIt06XoIOIio4LBEVpls']
1	98200.0	0.0	['Fernando Pessoa']	['14jtPC0oNZwquk5wd9DxrY']
2	181640.0	0.0	['Ignacio Corsini']	['5Li0oJbxVSAMkBS2fUm3X2']
3	176907.0	0.0	['Ignacio Corsini']	['5Li0oJbxVSAMkBS2fUm3X2']
4	163080.0	0.0	['Dick Haymes']	['3BiJGZsyX9sJchTqcSA7Su']

	release_date	danceability	energy	...	mode	speechiness	acousticness	\
0	1922-02-22	0.645	0.4450	...	1.0	0.4510	0.674	
1	1922-06-01	0.695	0.2630	...	1.0	0.9570	0.797	
2	1922-03-21	0.434	0.1770	...	1.0	0.0512	0.994	
3	1922-03-21	0.321	0.0946	...	1.0	0.0504	0.995	
4	1922	0.402	0.1580	...	0.0	0.0390	0.989	

	instrumentalness	liveness	valence	tempo	time_signature	genres	\
0	0.7440	0.151	0.127	104.851	3.0	[]	
1	0.0000	0.148	0.655	102.009	1.0	[]	
2	0.0218	0.212	0.457	130.418	5.0	[]	
3	0.9180	0.104	0.397	169.980	3.0	[]	
4	0.1300	0.311	0.196	103.220	4.0	[]	

	duration
0	127
1	98
2	182
3	177
4	163

[5 rows x 22 columns]

```
[40]: genre_df[genre_df['genres']=="[]"]
```

```
[40]:
```

	id	name	\
0	35iwgR4jXetI318WEWsa1Q	Carve	
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	
3	08FmqUhxtlyLTn6pAh6bk45	El Prisionero - Remasterizado	
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	
...	
586667	5rgu12WBIHQQtvej2MdHSH0		
586668	0NuWgxEp51CutD2pJoF40M	blind	
586669	27Y1N4Q4U3EfDU5Ubw8ws2	What They'll Say About Us	
586670	45XJsGpFTyzbzeWK8VzR8S	A Day At A Time	
586671	50cn6dZ3BJFPWh4ylwFXtn	Mar de Emociones	

	popularity	duration_ms	explicit	artists	\
0	6.0	126903.0	0.0	['Uli']	
1	0.0	98200.0	0.0	['Fernando Pessoa']	

2	0.0	181640.0	0.0	['Ignacio Corsini']
3	0.0	176907.0	0.0	['Ignacio Corsini']
4	0.0	163080.0	0.0	['Dick Haymes']
...
586667	50.0	258267.0	0.0	['YueYue']
586668	72.0	153293.0	0.0	['ROLE MODEL']
586669	70.0	187601.0	0.0	['FINNEAS']
586670	58.0	142003.0	0.0	['Gentle Bones', 'Clara Benin']
586671	38.0	214360.0	0.0	['Afrosound']

	id_artists	release_date	\
0	['45tIt06XoI0Iio4LBEVpls']	1922-02-22	
1	['14jtPC0oNZwqk5wd9DxrY']	1922-06-01	
2	['5Li0oJbxVSAMkBS2fUm3X2']	1922-03-21	
3	['5Li0oJbxVSAMkBS2fUm3X2']	1922-03-21	
4	['3BiJGZsyX9sJchTqcSA7Su']	1922	
...	
586667	['1QLBXXM5GCpyQQSVMNZqrZ']	2020-09-26	
586668	['1dy5WNgIKQU6ezkpZs4y8z']	2020-10-21	
586669	['37M5pPGs6V1fchFJSgCguX']	2020-09-02	
586670	['4jGPdu95icCKVF31CcFKbS', '5ebPSE9YI5aLeZ1Z2g...']	2021-03-05	
586671	['0i4Qda0k4nf7jnNHmSNpYv']	2015-07-01	

	danceability	energy	...	mode	speechiness	acousticness	\
0	0.645	0.4450	...	1.0	0.4510	0.674	
1	0.695	0.2630	...	1.0	0.9570	0.797	
2	0.434	0.1770	...	1.0	0.0512	0.994	
3	0.321	0.0946	...	1.0	0.0504	0.995	
4	0.402	0.1580	...	0.0	0.0390	0.989	
...	
586667	0.560	0.5180	...	0.0	0.0292	0.785	
586668	0.765	0.6630	...	1.0	0.0652	0.141	
586669	0.535	0.3140	...	0.0	0.0408	0.895	
586670	0.696	0.6150	...	1.0	0.0345	0.206	
586671	0.686	0.7230	...	1.0	0.0363	0.105	

	instrumentalness	liveness	valence	tempo	time_signature	genres	\
0	0.744000	0.1510	0.1270	104.851	3.0	['']	
1	0.000000	0.1480	0.6550	102.009	1.0	['']	
2	0.021800	0.2120	0.4570	130.418	5.0	['']	
3	0.918000	0.1040	0.3970	169.980	3.0	['']	
4	0.130000	0.3110	0.1960	103.220	4.0	['']	
...	
586667	0.000000	0.0648	0.2110	131.896	4.0	['']	
586668	0.000297	0.0924	0.6860	150.091	4.0	['']	
586669	0.000150	0.0874	0.0663	145.095	4.0	['']	
586670	0.000003	0.3050	0.4380	90.029	4.0	['']	

```
586671          0.000000    0.2640    0.9750   112.204          4.0      []
```

```

      duration
0          127
1           98
2          182
3          177
4          163
...
586667      258
586668      153
586669      188
586670      142
586671      214
```

```
[399952 rows x 22 columns]
```

```
[41]: genre_df=genre_df[genre_df['genres']!="[]"]
```

```
[42]: genre_df.head()
```

```
[42]:
```

	id	name	popularity	\
45	1kXWSsJkBVZ1jSoI8NnEDm	Marta	0.0	
46	1l1Wk0n0kuMCz10N6l2yfJ	Carol of the Bells	0.0	
47	1lia44teZBfbv0PnPkc5dK	Machinalment	0.0	
48	1pGB0fY0PvpArBZT7GaUVK	Capítulo 2.19 - Banquero Anarquista	0.0	
136	6tl1bsB5SXbsLj0ndGiM7Q	El Poder de Tus Ojos - Remasterizado	0.0	

	duration_ms	explicit	artists	id_artists	\
45	177693.0	0.0	['Dick Haymes']	['3BiJGZsyX9sJchTqcSA7Su']	
46	286370.0	0.0	['Grandcubby Trio']	['4XVZpokXbUzg6QeomBANY9']	
47	145400.0	0.0	['Victor Boucher']	['7vVR02JJYvsEAEPNHQMx0Q']	
48	106000.0	0.0	['Fernando Pessoa']	['14jtPC0oNZwquk5wd9DxrY']	
136	172800.0	0.0	['Ignacio Corsini']	['5Li0oJbxVSAMkBS2fUm3X2']	

	release_date	danceability	energy	...	mode	speechiness	acousticness	\
45	1922	0.255	0.343	...	1.0	0.0305	0.987	
46	1922	0.382	0.750	...	1.0	0.0742	0.123	
47	1922	0.457	0.193	...	1.0	0.0655	0.996	
48	1922-06-01	0.729	0.211	...	1.0	0.9570	0.769	
136	1922-03-21	0.357	0.244	...	1.0	0.0478	0.993	

	instrumentalness	liveness	valence	tempo	time_signature	\
45	0.0153	0.3260	0.313	101.096	4.0	
46	0.7510	0.7250	0.346	149.787	3.0	
47	0.3480	0.0809	0.671	177.097	4.0	
48	0.0000	0.5560	0.728	110.862	5.0	


```
136          0.0455    0.2740    0.559    72.510          5.0
```

```
          genres  duration
45    ['carnaval cadiz']    178
46    ['carnaval cadiz']    286
47    ['carnaval cadiz']    145
48    ['carnaval cadiz']    106
136  ['classical harp', 'harp']    173
```

```
[5 rows x 22 columns]
```

```
[43]: genre_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 186649 entries, 45 to 586664
Data columns (total 22 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    186649 non-null  object
1   name                  186649 non-null  object
2   popularity            186649 non-null  float64
3   duration_ms           186649 non-null  float64
4   explicit              186649 non-null  float64
5   artists               186649 non-null  object
6   id_artists            186649 non-null  object
7   release_date          186649 non-null  object
8   danceability          186649 non-null  float64
9   energy                186649 non-null  float64
10  key                   186649 non-null  float64
11  loudness              186649 non-null  float64
12  mode                  186649 non-null  float64
13  speechiness           186649 non-null  float64
14  acousticness          186649 non-null  float64
15  instrumentalness       186649 non-null  float64
16  liveness              186649 non-null  float64
17  valence               186649 non-null  float64
18  tempo                 186649 non-null  float64
19  time_signature         186649 non-null  float64
20  genres                 186649 non-null  object
21  duration              186649 non-null  int64
dtypes: float64(15), int64(1), object(6)
memory usage: 32.8+ MB
```

```
[44]: genre_df.nunique()
```

```
[44]: id                    186649
      name                  160670
```

```

popularity          93
duration_ms        62556
explicit            2
artists            53602
id_artists         53918
release_date       16249
danceability       1161
energy             2176
key                12
loudness           24148
mode               2
speechiness        1645
acousticness       4779
instrumentalness    5401
liveness           1750
valence            1709
tempo             85136
time_signature      5
genres             39688
duration           1356
dtype: int64

```

```

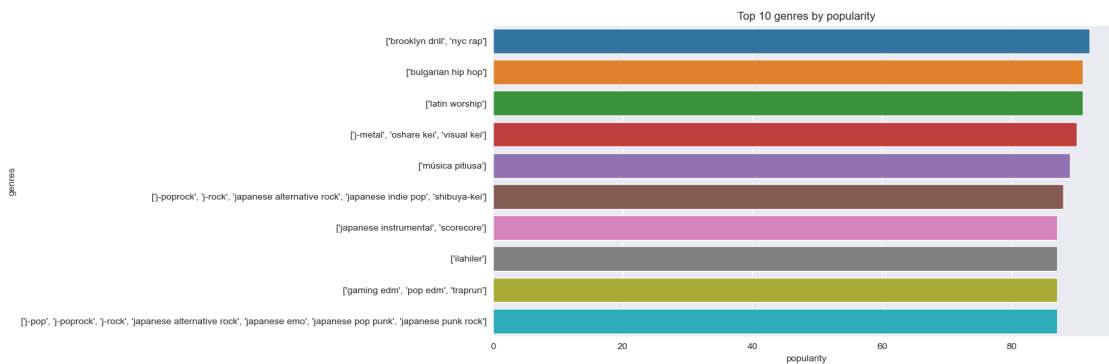
[45]: sns.set_style(style="darkgrid")
plt.figure(figsize=(12,6))
popu=genre_df.sort_values("popularity", ascending=False).head(10)
sns.barplot(y="genres", x="popularity", data=popu, ).set(title="Top 10 genres_
↳by popularity")

```

```

[45]: [Text(0.5, 1.0, 'Top 10 genres by popularity')]

```



```

[46]: genre_df.sort_values("popularity", ascending=True).head(10)[["popularity",
↳"genres"]]

```

```
[46]:
```

	popularity	genres
45	0.0	['carnaval cadiz']
85715	0.0	['arabesk']
85717	0.0	['turkish classical', 'turkish instrumental']
512205	0.0	['dark clubbing']
85719	0.0	['karadeniz pop']
512203	0.0	['canadian electronic']
512195	0.0	['deep darkpsy']
512193	0.0	['goa psytrance']
85725	0.0	['ilahiler', 'turkish classical']
512188	0.0	['dark psytrance']

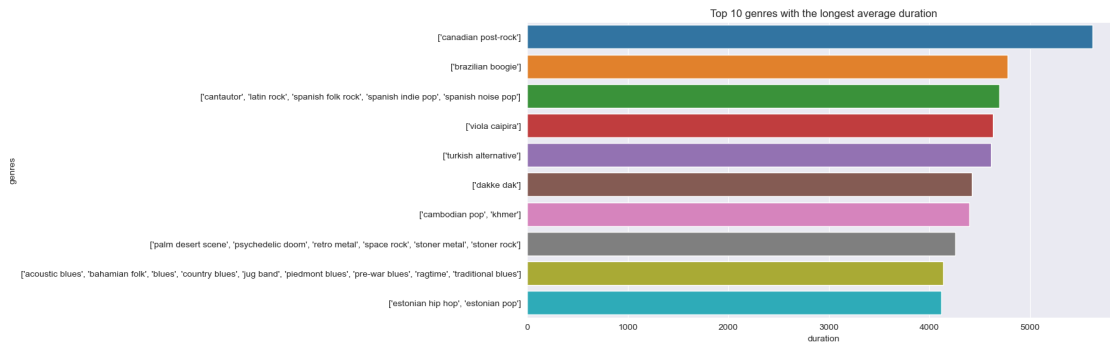
3 Duration of Genres

```
[47]: genre_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 186649 entries, 45 to 586664
Data columns (total 22 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                     186649 non-null object
1   name                   186649 non-null object
2   popularity             186649 non-null float64
3   duration_ms           186649 non-null float64
4   explicit               186649 non-null float64
5   artists               186649 non-null object
6   id_artists             186649 non-null object
7   release_date           186649 non-null object
8   danceability           186649 non-null float64
9   energy                 186649 non-null float64
10  key                    186649 non-null float64
11  loudness               186649 non-null float64
12  mode                   186649 non-null float64
13  speechiness            186649 non-null float64
14  acousticness           186649 non-null float64
15  instrumentalness        186649 non-null float64
16  liveness               186649 non-null float64
17  valence                 186649 non-null float64
18  tempo                  186649 non-null float64
19  time_signature          186649 non-null float64
20  genres                  186649 non-null object
21  duration                186649 non-null int64
dtypes: float64(15), int64(1), object(6)
memory usage: 32.8+ MB
```

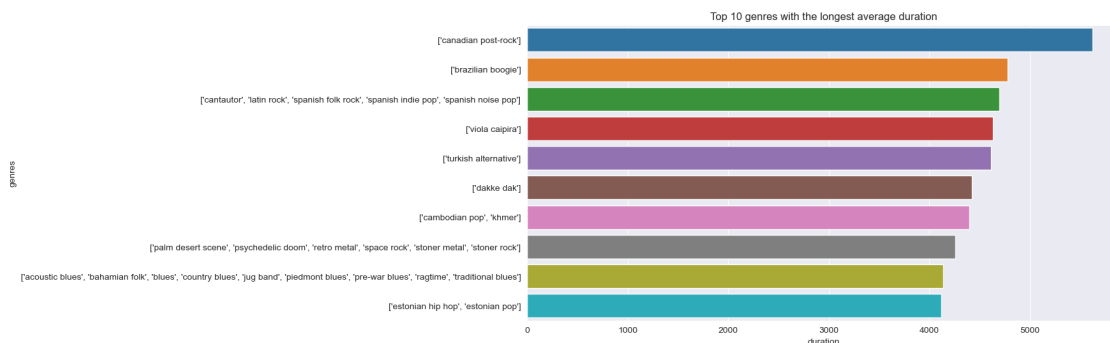
```
[48]: sns.set_style(style="darkgrid")
plt.figure(figsize=(12,6))
dur=genre_df.sort_values("duration", ascending=False).head(10)
sns.barplot(y="genres", x="duration", data=dur ).set(title="Top 10 genres with_
↳the longest average duration")
```

```
[48]: [Text(0.5, 1.0, 'Top 10 genres with the longest average duration')]
```



```
[49]: sns.set_style(style="darkgrid")
plt.figure(figsize=(12,6))
dur=genre_df.sort_values("duration", ascending=False).head(10)
sns.barplot(y="genres", x="duration", data=dur ).set(title="Top 10 genres with_
↳the longest average duration")
```

```
[49]: [Text(0.5, 1.0, 'Top 10 genres with the longest average duration')]
```



```
[50]: genre_df.sort_values("duration").head(10)
```

```
[50]:
```

	id \
1620	52qf3kN9pExT1HdSlh3ZeR
1575	4WeyR22Ax2fF9dY0NxxgjFV
1570	4SjlyAejCNUB4MrGM1KuVp

1580	4ZyFcBGN2aU9vgX1nD9d38
12550	3IcXTeq902dpsSXsDj9naH
352701	4A21jLdaDYg6N4HXzxJhbn
12627	523qs4UcG1Q6ycdha1VGqs
15601	4LBPCzMSeWb6UalrMJatBb
158610	1N4jTBZnj2M3SOTLB8FXPs
210767	4H1JR8xpuwQH5EMEOQoyL4

	name	popularity \
1620	Pause Track	0.0
1575	Pause Track	0.0
1570	Pause Track	0.0
1580	Pause Track	0.0
12550	Pause Track - Live	0.0
352701	Spectacle - Live 2019	0.0
12627	Pause Track - Live	0.0
15601	Rhapsody on a Theme of Paganini, Op. 43: Intro...	0.0
158610	Happy New Year 2019	0.0
210767	Skit - Happy New Year MAPARA!!!	0.0

	duration_ms	explicit	artists \
1620	3344.0	0.0	['Louis Armstrong']
1575	3344.0	0.0	['Louis Armstrong']
1570	3344.0	0.0	['Louis Armstrong']
1580	4000.0	0.0	['Louis Armstrong']
12550	5991.0	0.0	['Benny Goodman']
352701	6373.0	0.0	['The Dark Tenor']
12627	6362.0	0.0	['Benny Goodman']
15601	7523.0	0.0	['Sergei Rachmaninoff', 'Leopold Stokowski']
158610	12000.0	1.0	['Corey-G']
210767	13284.0	0.0	['Skeem Sa 2015']

	id_artists	release_date \
1620	['19eLuQmk9aCobbVDHc6eek']	1925
1575	['19eLuQmk9aCobbVDHc6eek']	1925
1570	['19eLuQmk9aCobbVDHc6eek']	1925
1580	['19eLuQmk9aCobbVDHc6eek']	1925
12550	['1pBuKaLHJlIlqYxQQaflve']	1938
352701	['2GIKcsT0xfxspYUZ5yX2tL']	2019-10-11
12627	['1pBuKaLHJlIlqYxQQaflve']	1938
15601	['0Kekt6CKSoOm5mivKcoH51', '52sDxFX9DvIxUupTy8...']	1941
158610	['3VSvWHcT0eMCQ4pT1mxF9b']	2019-01-01
210767	['5WJS8w04yAX3kC75bkWB9Z']	2018-03-30

	danceability	energy	...	mode	speechiness	acousticness \
1620	0.0	0.000	...	0.0	0.0	0.000000
1575	0.0	0.000	...	0.0	0.0	0.000000

1570	0.0	0.000	...	0.0	0.0	0.000000
1580	0.0	0.000	...	0.0	0.0	0.000000
12550	0.0	0.000	...	0.0	0.0	0.000000
352701	0.0	0.951	...	1.0	0.0	0.000003
12627	0.0	0.000	...	0.0	0.0	0.000000
15601	0.0	0.177	...	0.0	0.0	0.992000
158610	0.0	0.101	...	0.0	0.0	0.227000
210767	0.0	0.553	...	0.0	0.0	0.827000

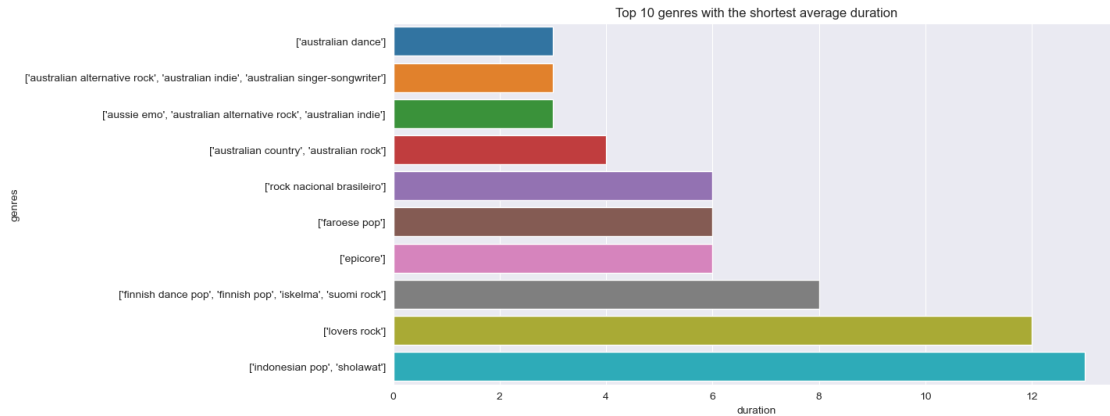
	instrumentalness	liveness	valence	tempo	time_signature	\
1620	0.000	0.000	0.0	0.0	0.0	
1575	0.000	0.000	0.0	0.0	0.0	
1570	0.000	0.000	0.0	0.0	0.0	
1580	0.000	0.000	0.0	0.0	0.0	
12550	0.000	0.000	0.0	0.0	0.0	
352701	0.990	0.000	0.0	0.0	0.0	
12627	0.000	0.000	0.0	0.0	0.0	
15601	0.412	0.000	0.0	0.0	0.0	
158610	0.000	0.264	0.0	0.0	0.0	
210767	0.000	0.278	0.0	0.0	0.0	

	genres	duration
1620	['australian dance']	3
1575	['australian alternative rock', 'australian in...']	3
1570	['aussie emo', 'australian alternative rock', ...]	3
1580	['australian country', 'australian rock']	4
12550	['rock nacional brasileiro']	6
352701	['faroesse pop']	6
12627	['epicore']	6
15601	['finnish dance pop', 'finnish pop', 'iskelma'...]	8
158610	['lovers rock']	12
210767	['indonesian pop', 'sholawat']	13

[10 rows x 22 columns]

```
[51]: sns.set_style(style="darkgrid")
plt.figure(figsize=(12,6))
dur_min=genre_df.sort_values("duration", ascending=True).head(10)
sns.barplot(y="genres", x="duration", data=dur_min ).set(title="Top 10 genres_
↳with the shortest average duration")
```

```
[51]: [Text(0.5, 1.0, 'Top 10 genres with the shortest average duration')]
```



```
[52]: from collections import Counter
genre_names = " ".join(genre_df['genres'].tolist()).split(" ")
column_names = ["word", "count"]
most_common_words_in_genres_df = pd.
↳ DataFrame([dict(zip(column_names, word_count)) for word_count in
↳ Counter(genre_names).most_common(30)])
```

```
[53]: most_common_words_in_genres_df
```

```
[53]:
```

	word	count
0	pop']	18642
1	pop',	12664
2	hip	12569
3	rock',	11098
4	rock']	10485
5	['classic	7232
6	indie']	6818
7	hop']	6703
8	hop',	6596
9	metal',	6557
10	['deep	5384
11	metal']	4531
12	indie',	3846
13	folk']	3738
14	house']	3713
15	rap']	3362
16	house',	3358
17	jazz']	3337
18	'pop	2848
19	'indie	2765
20	['german	2759
21	punk']	2713

```

22     folk',      2512
23     ['musica    2465
24     rap',       2388
25     alternative 2241
26     ['indie    2219
27     'new        2190
28     jazz',      2185
29     ['japanese  2167

```

```

[54]: pop_df=genre_df[genre_df["genres"].str.contains("pop")].
      ↪sort_values("popularity", ascending=False).head(10)
metal_df=genre_df[genre_df["genres"].str.contains("metal")].
      ↪sort_values("popularity", ascending=False).head(10)
indie_df=genre_df[genre_df["genres"].str.contains("indie")].
      ↪sort_values("popularity", ascending=False).head(10)
rock_df=genre_df[genre_df["genres"].str.contains("rock")].
      ↪sort_values("popularity", ascending=False).head(10)

```

```

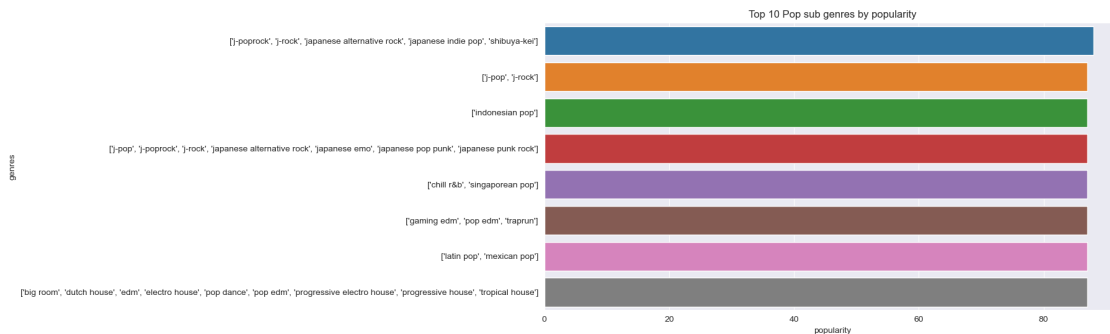
[55]: plt.figure(figsize=(12,6))
      sns.barplot(y="genres", x="popularity", data=pop_df).set(title="Top 10 Pop sub_
      ↪genres by popularity")

```

```

[55]: [Text(0.5, 1.0, 'Top 10 Pop sub genres by popularity')]

```



```

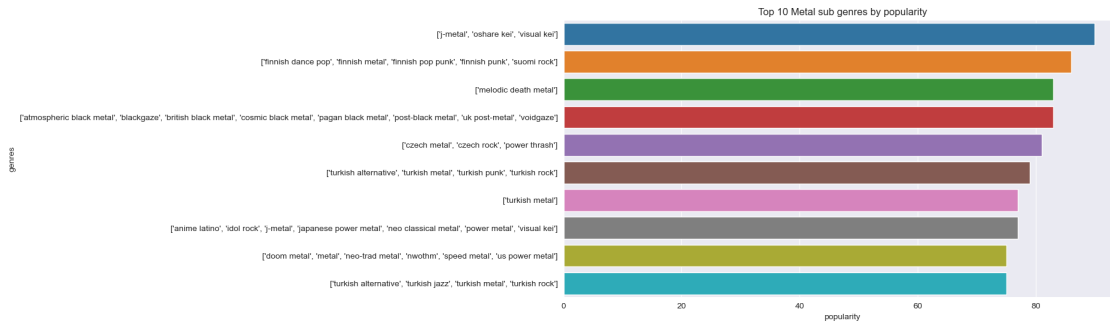
[56]: plt.figure(figsize=(12,6))
      sns.barplot(y="genres", x="popularity", data=metal_df).set(title="Top 10 Metal_
      ↪sub genres by popularity")

```

```

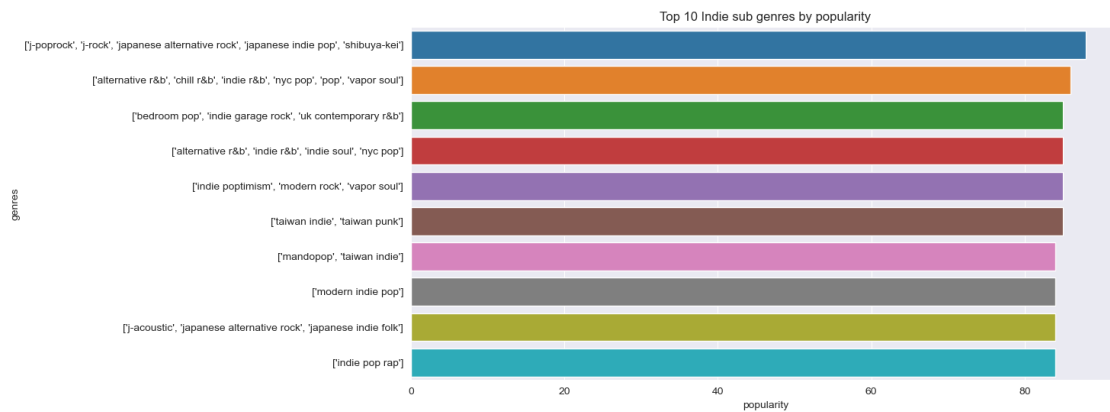
[56]: [Text(0.5, 1.0, 'Top 10 Metal sub genres by popularity')]

```

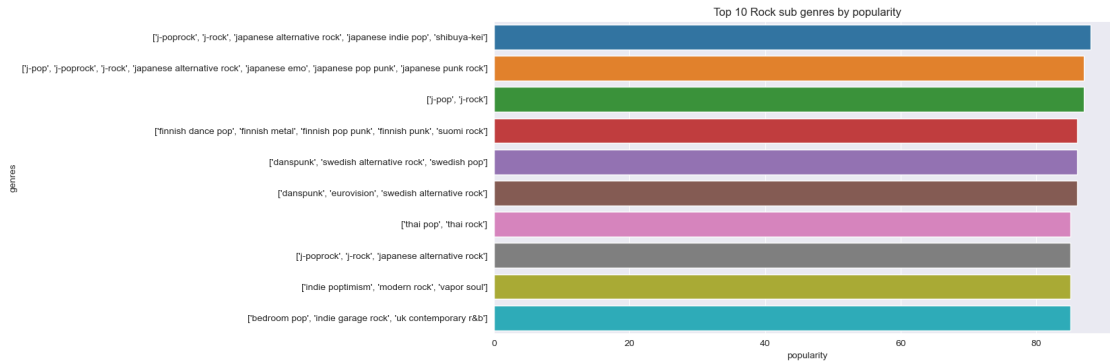
```
[57]: plt.figure(figsize=(12,6))
sns.barplot(y="genres", x="popularity", data=indie_df).set(title="Top 10 Indie_
↳sub genres by popularity")
```

```
[57]: [Text(0.5, 1.0, 'Top 10 Indie sub genres by popularity')]
```



```
[58]: plt.figure(figsize=(12,6))
sns.barplot(y="genres", x="popularity", data=rock_df).set(title="Top 10 Rock_
↳sub genres by popularity")
```

```
[58]: [Text(0.5, 1.0, 'Top 10 Rock sub genres by popularity')]
```



4 Artist based Analysis

```
[84]: artists= pd.read_csv("artists.csv")
```

```
[85]: artists.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1162095 entries, 0 to 1162094
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   id           1162095 non-null  object
1   followers    1162084 non-null  float64
2   genres       1162095 non-null  object
3   name         1162092 non-null  object
4   popularity   1162095 non-null  int64
dtypes: float64(1), int64(1), object(3)
memory usage: 44.3+ MB
```

```
[94]: art=artists.drop(["id","name","popularity"],axis=1)
```

```
[95]: tracks = pd.read_csv("tracks.csv")
artists_df = pd.concat([tracks_df, art], axis=1)
```

```
[96]: artists_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1162095 entries, 0 to 1162094
Data columns (total 22 columns):
#   Column      Non-Null Count  Dtype
---  -
0   id           586672 non-null  object
1   name         586601 non-null  object
```

```

2  popularity      586672 non-null  float64
3  duration_ms     586672 non-null  float64
4  explicit        586672 non-null  float64
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  float64
11 loudness        586672 non-null  float64
12 mode           586672 non-null  float64
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  float64
20 followers       1162084 non-null  float64
21 genres          1162095 non-null  object
dtypes: float64(16), object(6)
memory usage: 195.1+ MB

```

```
[97]: artists_df["artists"].value_counts()
```

```

[97]: artists
['Die drei ???']      3856
['TKKG Retro-Archiv'] 2006
['Benjamin Blümchen'] 1503
['Bibi Blocksberg']   1472
['Lata Mangeshkar']    1373
...
['IU', 'Jang Yi-jeong'] 1
[' ' ]                  1
['Vincy Chan', ' ' ]    1
['Dough-Boy']           1
['Gentle Bones', 'Clara Benin'] 1
Name: count, Length: 114030, dtype: int64

```

```

[113]: artists_df['release_date'] = pd.
        ↳to_datetime(artists_df['release_date'],format='mixed')
artists_df['year'] = artists_df['release_date'].dt.year
artists_df.dropna

```

```

[113]: <bound method DataFrame.dropna of                                id
name \
0      35iwgR4jXetI318WEWsa1Q                                Carve

```

1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado
3	08FmqUhxyLTn6pAh6bk45	El Prisionero - Remasterizado
4	08y9GfoqCWf0GsKdwojr5e	Lady of the Evening

...
1162090	NaN	NaN
1162091	NaN	NaN
1162092	NaN	NaN
1162093	NaN	NaN
1162094	NaN	NaN

	popularity	duration_ms	explicit	artists \
0	6.0	126903.0	0.0	['Uli']
1	0.0	98200.0	0.0	['Fernando Pessoa']
2	0.0	181640.0	0.0	['Ignacio Corsini']
3	0.0	176907.0	0.0	['Ignacio Corsini']
4	0.0	163080.0	0.0	['Dick Haymes']
...
1162090	NaN	NaN	NaN	NaN
1162091	NaN	NaN	NaN	NaN
1162092	NaN	NaN	NaN	NaN
1162093	NaN	NaN	NaN	NaN
1162094	NaN	NaN	NaN	NaN

	id_artists	release_date	danceability	energy	...	\
0	['45tIt06XoIOIio4LBEVpls']	1922-02-22	0.645	0.4450
1	['14jtPC0oNZwqk5wd9DxrY']	1922-06-01	0.695	0.2630
2	['5Li0oJbxVSAMkBS2fUm3X2']	1922-03-21	0.434	0.1770
3	['5Li0oJbxVSAMkBS2fUm3X2']	1922-03-21	0.321	0.0946
4	['3BiJGZsyX9sJchTqcSA7Su']	1922-01-01	0.402	0.1580
...
1162090	NaN	NaT	NaN	NaN
1162091	NaN	NaT	NaN	NaN
1162092	NaN	NaT	NaN	NaN
1162093	NaN	NaT	NaN	NaN
1162094	NaN	NaT	NaN	NaN

	acousticness	instrumentalness	liveness	valence	tempo \
0	0.674	0.7440	0.151	0.127	104.851
1	0.797	0.0000	0.148	0.655	102.009
2	0.994	0.0218	0.212	0.457	130.418
3	0.995	0.9180	0.104	0.397	169.980
4	0.989	0.1300	0.311	0.196	103.220
...
1162090	NaN	NaN	NaN	NaN	NaN
1162091	NaN	NaN	NaN	NaN	NaN
1162092	NaN	NaN	NaN	NaN	NaN

1162093	NaN	NaN	NaN	NaN	NaN
1162094	NaN	NaN	NaN	NaN	NaN

	time_signature	followers	genres	year	years
0	3.0	0.0	[]	1922.0	1922.0
1	1.0	5.0	[]	1922.0	1922.0
2	5.0	0.0	[]	1922.0	1922.0
3	3.0	0.0	[]	1922.0	1922.0
4	4.0	2.0	[]	1922.0	1922.0
...
1162090	NaN	4831.0	['black comedy']	NaN	NaN
1162091	NaN	46.0	[]	NaN	NaN
1162092	NaN	257.0	[]	NaN	NaN
1162093	NaN	2357.0	['black comedy']	NaN	NaN
1162094	NaN	40.0	['new comedy']	NaN	NaN

[1162095 rows x 24 columns]>

```
[114]: years=artists_df.year.unique()
years
```

```
[114]: array([1922., 1923., 1924., 1925., 1926., 1927., 1928., 1929., 1930.,
1931., 1932., 1933., 1934., 1935., 1936., 1937., 1938., 1939.,
1940., 1941., 1942., 1943., 1944., 1945., 1946., 1947., 1948.,
1949., 1950., 1951., 1952., 1953., 1954., 1955., 1956., 1957.,
1958., 1959., 1960., 1961., 1962., 1963., 1964., 1965., 1966.,
1968., 2008., 2020., 2018., 1997., 2006., 1991., 2012., 2015.,
2011., 1992., 2007., 1996., 2021., 2013., 2014., 2017., 1967.,
1969., 1970., 1971., 1972., 1973., 1974., 1975., 1976., 1977.,
1978., 1979., 1980., 1981., 1982., 1983., 1984., 1985., 1986.,
1987., 1988., 1989., 1990., 1993., 1994., 1995., 1998., 1999.,
2000., 2019., 2016., 2010., 2009., 2004., 2003., 2005., 2001.,
2002., 1900., nan])
```

```
[125]: artists_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1162095 entries, 0 to 1162094
Data columns (total 25 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    586672 non-null object
1   name                  586601 non-null object
2   popularity            586672 non-null float64
3   duration_ms          586672 non-null float64
4   explicit              586672 non-null float64
5   artists              586672 non-null object
```

```

6   id_artists      586672 non-null  object
7   release_date    586672 non-null  datetime64[ns]
8   danceability    586672 non-null  float64
9   energy          586672 non-null  float64
10  key             586672 non-null  float64
11  loudness        586672 non-null  float64
12  mode            586672 non-null  float64
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  float64
20  followers       1162084 non-null  float64
21  genres          1162095 non-null  object
22  year            586672 non-null  float64
23  years           586672 non-null  float64
24  count           0 non-null      float64
dtypes: datetime64[ns](1), float64(19), object(5)
memory usage: 221.7+ MB

```

```

[137]: artists_df["count"] = artists_df.count(axis=1)
artists_df.drop(["followers", "genres"], axis=1)
artists_df.dropna

```

```

[137]: <bound method DataFrame.dropna of                                id
name \
0      35iwgR4jXetI318WEWsa1Q                                     Carve
1      021ht4sdgPcrDgSk7JTbKY  Capítulo 2.16 - Banquero Anarquista
2      07A5yehtSnoedViJAZkNnc  Vivo para Quererte - Remasterizado
3      08FmqUhxyLTn6pAh6bk45    El Prisionero - Remasterizado
4      08y9GfoqCWfOGsKdwojr5e    Lady of the Evening
...
1162090      NaN      NaN
1162091      NaN      NaN
1162092      NaN      NaN
1162093      NaN      NaN
1162094      NaN      NaN

      popularity  duration_ms  explicit  artists \
0              6.0    126903.0      0.0      ['Uli']
1              0.0     98200.0      0.0  ['Fernando Pessoa']
2              0.0    181640.0      0.0  ['Ignacio Corsini']
3              0.0    176907.0      0.0  ['Ignacio Corsini']
4              0.0    163080.0      0.0    ['Dick Haymes']
...           ...           ...      ...

```

1162090	NaN	NaN	NaN	NaN
1162091	NaN	NaN	NaN	NaN
1162092	NaN	NaN	NaN	NaN
1162093	NaN	NaN	NaN	NaN
1162094	NaN	NaN	NaN	NaN

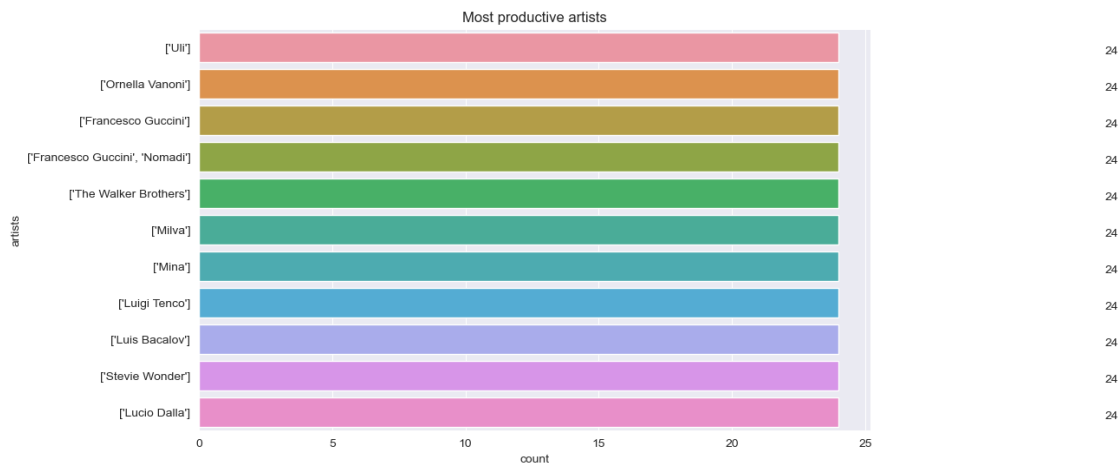
	id_artists	release_date	danceability	energy	...	\
0	['45tIt06XoIOIio4LBEVpls']	1922-02-22	0.645	0.4450	...	
1	['14jtPC0oNZwqk5wd9DxrY']	1922-06-01	0.695	0.2630	...	
2	['5Li0oJbxVSAMkBS2fUm3X2']	1922-03-21	0.434	0.1770	...	
3	['5Li0oJbxVSAMkBS2fUm3X2']	1922-03-21	0.321	0.0946	...	
4	['3BiJGZsyX9sJchTqcSA7Su']	1922-01-01	0.402	0.1580	...	
...
1162090	NaN	NaT	NaN	NaN	...	
1162091	NaN	NaT	NaN	NaN	...	
1162092	NaN	NaT	NaN	NaN	...	
1162093	NaN	NaT	NaN	NaN	...	
1162094	NaN	NaT	NaN	NaN	...	

	instrumentalness	liveness	valence	tempo	time_signature	\
0	0.7440	0.151	0.127	104.851	3.0	
1	0.0000	0.148	0.655	102.009	1.0	
2	0.0218	0.212	0.457	130.418	5.0	
3	0.9180	0.104	0.397	169.980	3.0	
4	0.1300	0.311	0.196	103.220	4.0	
...
1162090	NaN	NaN	NaN	NaN	NaN	
1162091	NaN	NaN	NaN	NaN	NaN	
1162092	NaN	NaN	NaN	NaN	NaN	
1162093	NaN	NaN	NaN	NaN	NaN	
1162094	NaN	NaN	NaN	NaN	NaN	

	followers	genres	year	years	count
0	0.0	[]	1922.0	1922.0	24
1	5.0	[]	1922.0	1922.0	24
2	0.0	[]	1922.0	1922.0	24
3	0.0	[]	1922.0	1922.0	24
4	2.0	[]	1922.0	1922.0	24
...
1162090	4831.0	['black comedy']	NaN	NaN	2
1162091	46.0	[]	NaN	NaN	2
1162092	257.0	[]	NaN	NaN	2
1162093	2357.0	['black comedy']	NaN	NaN	2
1162094	40.0	['new comedy']	NaN	NaN	2

[1162095 rows x 25 columns]>

```
[138]: plt.figure(figsize=(10,6))
plot1=sns.barplot(data=artists_df.sort_values("count", ascending=False).
↳head(20),x="count" ,y="artists")
plt.title("Most productive artists")
for p in plot1.patches:
    width = p.get_width()
    plt.text(10+p.get_width(), p.get_y()+0.65*p.get_height(),
            '{:1.0f}'.format(width),
            ha='left', va='center')
```



```
[150]: tracks = pd.read_csv("tracks.csv")
genre_w_df = pd.concat([tracks_df, art], axis=1)
genre_w_df.dropna(axis=1)
```

```
[150]:
```

	genres
0	['']
1	['']
2	['']
3	['']
4	['']
...	...
1162090	['black comedy']
1162091	['']
1162092	['']
1162093	['black comedy']
1162094	['new comedy']

[1162095 rows x 1 columns]

```
[ ]:
```