In [1]:

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

In [2]:

```
tracks = pd.read_csv('C:/Users/Bae/Desktop/4th Semester/Data Science Lab/Datasets/tracks.cs
```

In [3]:

```
tracks.head()
```

Out[3]:

	id	name	popularity	duration_ms	explicit	artists	
0	35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0	['Uli']	['4
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']	['14jtP
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']	['5LiOo
3	08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']	['5LiOo
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0	['Dick Haymes']	['3BiJ

In [4]:

```
tracks.head(20)
```

Out[4]:

	id	name	popularity	duration_ms	explicit	artists
0	35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0	['Uli']
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']
3	08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0	['Dick Haymes']
5	0BRXJHRNGQ3W4v9frnSfhu	Ave Maria	0	178933	0	['Dick Haymes']
6	0Dd9ImXtAtGwsmsAD69KZT	La Butte Rouge	0	134467	0	['Francis Marty']
7	0IA0Hju8CAgYfV1hwhidBH	La Java	0	161427	0	['Mistinguett']
8	0Igl1UCz84pYeVetnI1IGP	Old Fashioned Girl	0	310073	0	['Greg Fieler']
9	0JV4iqw2ISKJaHBQZ0e5zK	Martín Fierro - Remasterizado	0	181173	0	['Ignacio Corsini']
10	0OYGe21oScKJfanLyM7daU	Capítulo 2.8 - Banquero Anarquista	0	99100	0	['Fernando Pessoa']
11	0PE42H6tsIQuyMMiGRiqtb	Capítulo 2.25 - Banquero Anarquista	0	132700	0	['Fernando Pessoa']
12	0PH9AACae1f957JAavhOI2	Lazy Boi	0	157333	0	['Uli']
13	0QiT0Oo5QdLXdFw6RDOj7h	Tu Verras Montmartre	1	186800	0	['Lucien Boyer']
14	0TWsNj5iSvbMTtbEDP7A4V	Elle Prend L'boulevard Magenta	0	172027	0	['Félix Mayol']
15	0cC9CYjLRIzwchQ42xVnq6	Capítulo 1.23 - Banquero Anarquista	0	96600	0	['Fernando Pessoa']
16	0eb1PfHxT6HnXvvdUOzmME	Capítulo 1.18 - Banquero Anarquista	0	103200	0	['Fernando Pessoa']
17	0grXU6GKVNCVMJbseA0Uhe	Capítulo 1.10 - Banquero Anarquista	0	95800	0	['Fernando Pessoa']
18	0kCB1bDVBC8gWCFcnJyIZc	Ca C'est Une Chose	0	188000	0	['Victor Boucher'] ['
19	0l3BQsVJ7F76wIN5QhJzaP	El Vendaval - Remasterizado	0	153533	0	['Ignacio Corsini']

In [5]:

```
tracks.shape
```

Out[5]:

```
(586672, 20)
```

In [6]:

```
tracks.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
```

In [7]:

```
tracks.groupby ('popularity').size()
```

Out[7]:

```
popularity
0      44690
1      12024
2       9639
3       8154
4       7733
       ...
96         2
97         2
98         1
99         1
100        1
Length: 101, dtype: int64
```

In [8]:

```
popularity_num=tracks.groupby ('popularity').size()
```

In [9]:

```
popularity_num.tail(30)
```

Out[9]:

```
popularity
71     968
72     846
73     732
74     673
75     571
76     485
77     391
78     314
79     279
80     218
81     158
82     136
83     116
84      91
85      60
86      53
87      38
88      19
89      16
90      12
91      11
92      10
93       2
94       6
95       1
96       2
97       2
98       1
99       1
100      1
dtype: int64
```

In [10]:

```
tracks.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
```

In [11]:

```
tracks.describe()
```

Out[11]:

	popularity	duration_ms	explicit	danceability	energy	k
count	586672.000000	5.866720e+05	586672.000000	586672.000000	586672.000000	586672.0000
mean	27.570053	2.300512e+05	0.044086	0.563594	0.542036	5.2216
std	18.370642	1.265261e+05	0.205286	0.166103	0.251923	3.5194
min	0.000000	3.344000e+03	0.000000	0.000000	0.000000	0.0000
25%	13.000000	1.750930e+05	0.000000	0.453000	0.343000	2.0000
50%	27.000000	2.148930e+05	0.000000	0.577000	0.549000	5.0000
75%	41.000000	2.638670e+05	0.000000	0.686000	0.748000	8.0000
max	100.000000	5.621218e+06	1.000000	0.991000	1.000000	11.0000

In [12]:

```
tracks.isnull().sum()
```

Out[12]:

```
id                  0
name               71
popularity          0
duration_ms         0
explicit            0
artists             0
id_artists          0
release_date        0
danceability        0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
acousticness        0
instrumentalness    0
liveness            0
valence             0
tempo               0
time_signature      0
dtype: int64
```

In [13]:

```
tracks=tracks.dropna(axis=0)
```

In [14]:

```
tracks.isnull().sum()
```

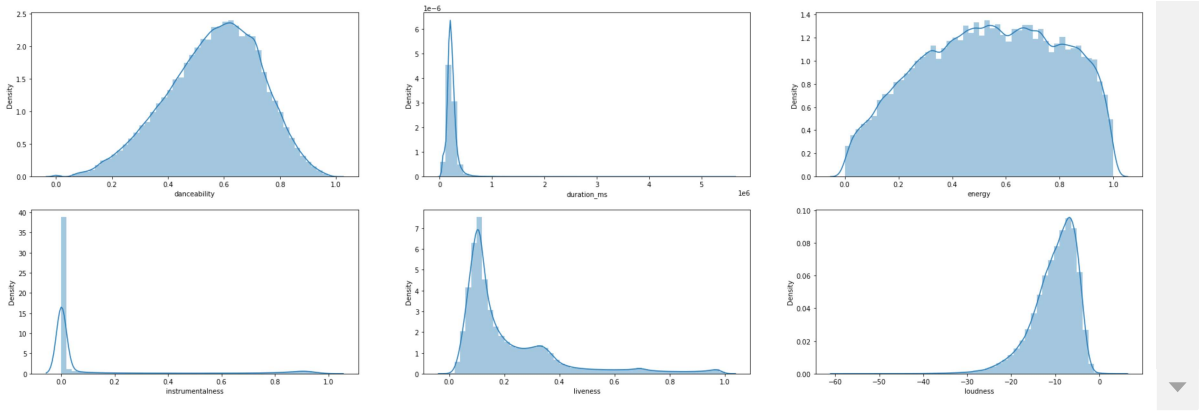
Out[14]:

```
id                  0
name                0
popularity          0
duration_ms         0
explicit            0
artists             0
id_artists          0
release_date        0
danceability        0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
acousticness        0
instrumentalness    0
liveness            0
valence             0
tempo               0
time_signature      0
dtype: int64
```

In [15]:

```
plt.figure(figsize = (30, 10))
plt.subplot(231)
sns.distplot(tracks['danceability'])
plt.subplot(232)
sns.distplot(tracks['duration_ms'])
plt.subplot(233)
sns.distplot(tracks['energy'])
plt.subplot(234)
sns.distplot(tracks['instrumentalness'])
plt.subplot(235)
sns.distplot(tracks['liveness'])
plt.subplot(236)
sns.distplot(tracks['loudness'])
plt.show()
```

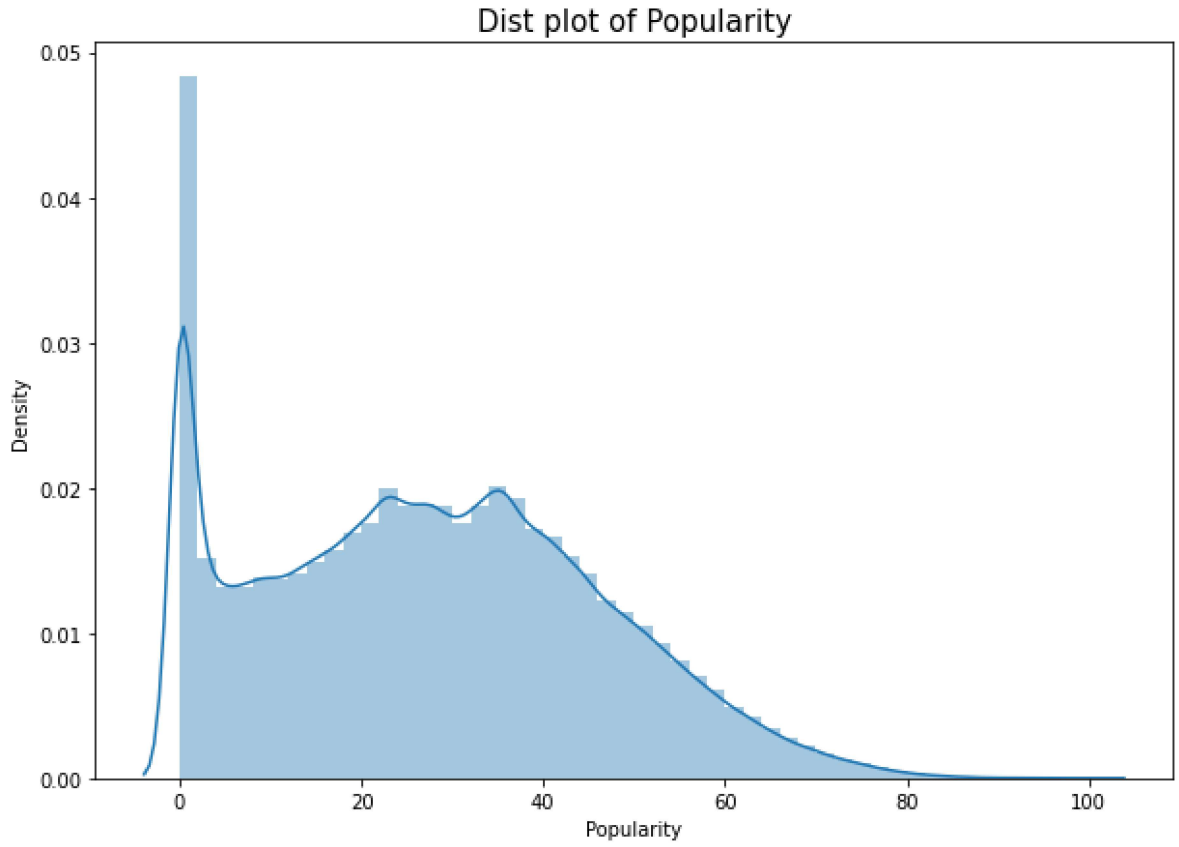
```
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\distributions.py:2557: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use eithe
r `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\distributions.py:2557: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use eithe
r `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\distributions.py:2557: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use eithe
r `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\distributions.py:2557: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use eithe
r `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\distributions.py:2557: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use eithe
r `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\distributions.py:2557: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use eithe
r `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [16]:

```
plt.figure(figsize = (10, 7))
sns.distplot(tracks.popularity)
plt.title("Dist plot of Popularity", fontdict = {'fontsize' : 15})
plt.xlabel('Popularity')
plt.show()
```

```
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\distributions.py:2557: FutureWarning: `distplot` is a deprecated function
and will be removed in a future version. Please adapt your code to use eithe
r `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

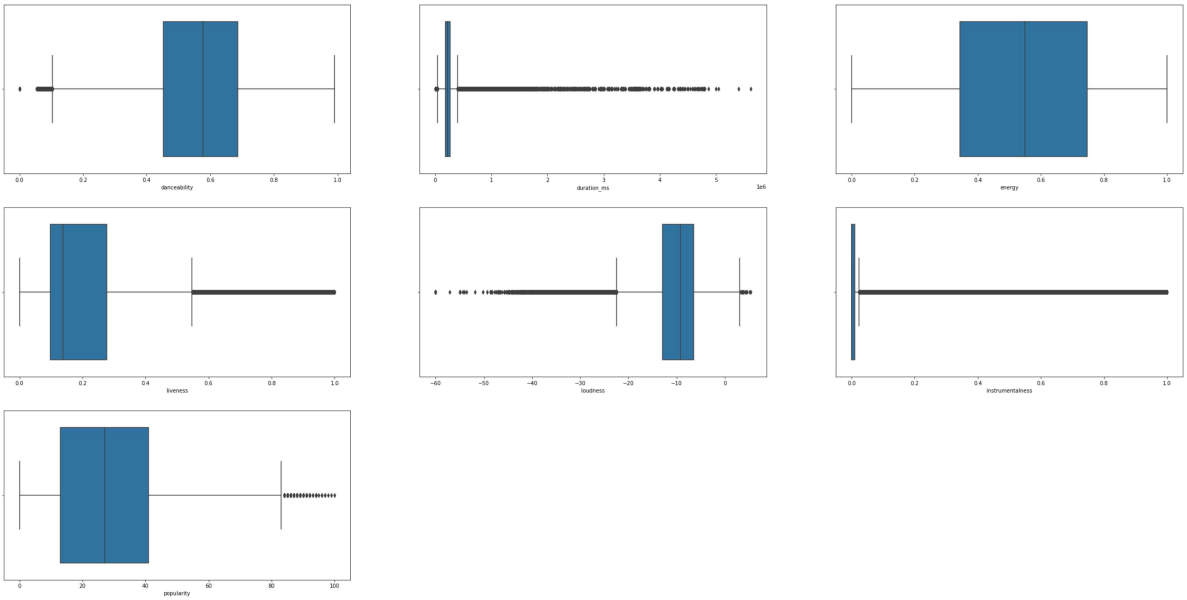


In [17]:

```
plt.figure(figsize = (40, 20))
plt.subplot(331)
sns.boxplot(tracks['danceability'])
plt.subplot(332)
sns.boxplot(tracks['duration_ms'])
plt.subplot(333)
sns.boxplot(tracks['energy'])
plt.subplot(334)
sns.boxplot(tracks['liveness'])
plt.subplot(335)
sns.boxplot(tracks['loudness'])
plt.subplot(336)
sns.boxplot(tracks['instrumentalness'])
plt.subplot(337)
sns.boxplot(tracks['popularity'])
plt.show()
```

```
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seab
orn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seab
orn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seab
orn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seab
orn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seab
orn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seab
orn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seab
orn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will resul
```

```
t in an error or misinterpretation.
  warnings.warn(
```



In [18]:

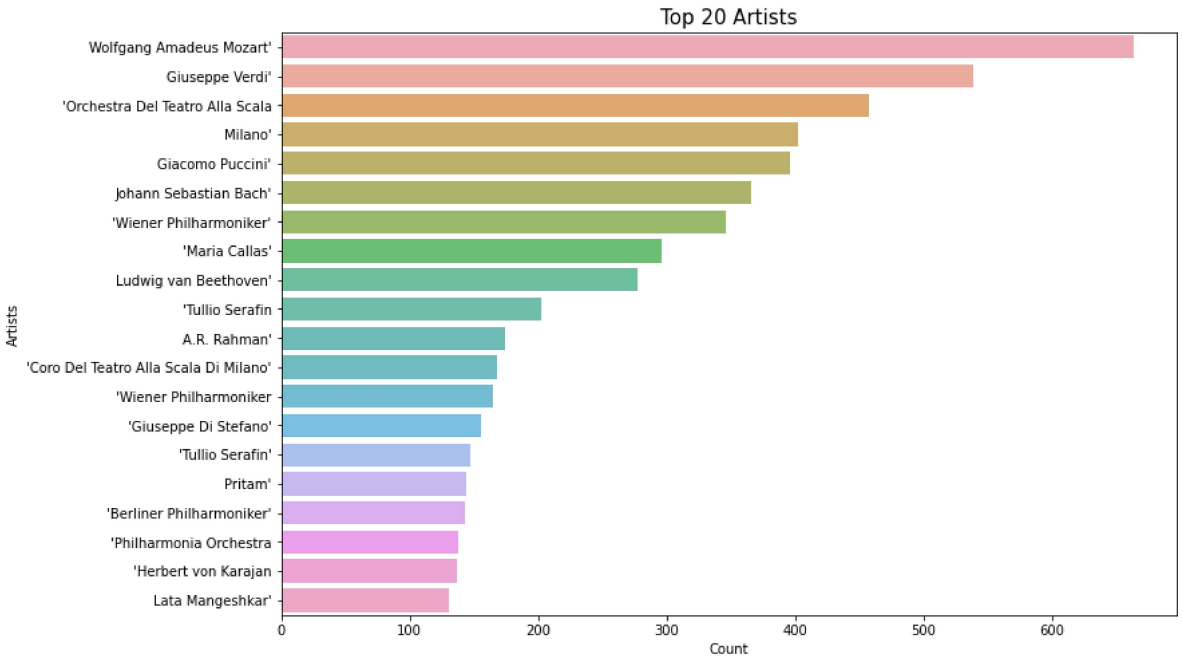
```
print(tracks.groupby('artists').size().sort_values(ascending=False))
```

```
artists
['Die drei ???']                                                    3856
['TKKG Retro-Archiv']                                               2006
['Benjamin Blümchen']                                               1503
['Bibi Blocksberg']                                                 1472
['Lata Mangeshkar']                                                 1373
                                                                    ...
['Jess Penner']                                                        1
['Jess Ingerslev', 'Tritonus Koret']                                   1
['Jess Ingerslev', 'Bendt Reiner', 'Niels Weyde', 'Tritonus Koret']    1
['Jess Glynne', 'Jonas Blue']                                          1
['최진희']
1
Length: 114029, dtype: int64
```

In [19]:

```
l = []
for i in tracks.artists:
    a=i[2:-2]
    l.append(a)
l = list(set(l))
lst = []
for i in l:
    res = i.strip('][').split(', ')
    lst.append(res)
d=[]
for i in lst:
    for j in i:
        d.append(j)
dff=pd.DataFrame(d)
dff = dff[0].value_counts()

dff = dff[:20, ]
plt.figure(figsize = (12, 8))
sns.barplot(dff.values, dff.index, alpha = 0.8)
plt.title("Top 20 Artists", fontdict = {'fontsize' : 15})
plt.ylabel("Artists")
plt.xlabel("Count")
plt.show()
```

```
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\_decorators.py:36: FutureWarning: Pass the following variables as keyword
args: x, y. From version 0.12, the only valid positional argument will be `d
ata`, and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```



In [20]:

```
tracks.head()
```

Out[20]:

	id	name	popularity	duration_ms	explicit	artists	
0	35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0	['Uli']	['4
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']	['14jtP
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']	['5LiOo
3	08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']	['5LiOo
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0	['Dick Haymes']	['3BiJ

In [21]:

```
tracks['release_date'] = pd.to_datetime(tracks['release_date'])
```

In [22]:

```
tracks.head()
```

Out[22]:

	id	name	popularity	duration_ms	explicit	artists	
0	35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0	['Uli']	['4
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']	['14jtP
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']	['5LiOo
3	08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']	['5LiOo
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0	['Dick Haymes']	['3BiJ

In [23]:

```
tracks['year']=tracks['release_date'].dt.year
```

In [24]:

```
tracks.head()
```

Out[24]:

	id	name	popularity	duration_ms	explicit	artists	
0	35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0	['Uli']	['4
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	98200	0	['Fernando Pessoa']	['14jtP
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	181640	0	['Ignacio Corsini']	['5LiOo
3	08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	176907	0	['Ignacio Corsini']	['5LiOo
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	163080	0	['Dick Haymes']	['3BiJ

5 rows × 21 columns

In [25]:

```
latesttracks = tracks[tracks['year'] > 2016]

latesttracks.head()
```

Out[25]:

	id	name	popularity	duration_ms	explicit	artists	
39511	6Pkt6qVikqPBt9bEQy8iTz	A Lover's Concerto	41	159560	0	['The Toys']	
39529	1hx7X9cMXHWJjknb9O6Ava	The September Of My Years - Live At The Sands ...	26	187333	0	['Frank Sinatra']	[''
39533	19oquvXf3bc65GSqtPYA5S	It Was A Very Good Year - Live At The Sands Ho...	25	236800	0	['Frank Sinatra']	[''
39581	55qyghODi24yaDgKBI6Ix0	The Circle Game - Live at The 2nd Fret, Philad...	18	313093	0	['Joni Mitchell']	['5
39583	00xemFYjQNRpOIPhVaLAHa	Urge For Going - Live at The 2nd Fret, Philade...	18	295093	0	['Joni Mitchell']	['5

5 rows × 21 columns

In [26]:

```
latesttracks.groupby('year').size()
```

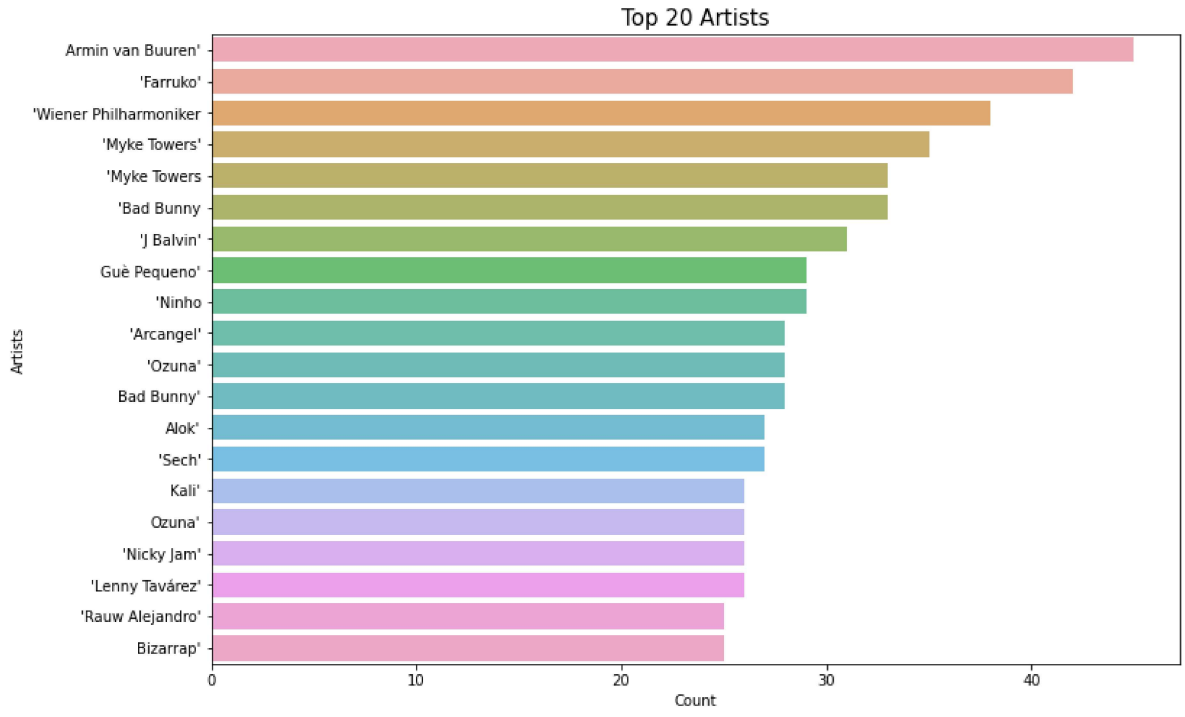
Out[26]:

```
year
2017     9889
2018    10936
2019    11907
2020    13937
2021     6281
dtype: int64
```

In [27]:

```
l = []
for i in latesttracks.artists:
    a=i[2:-2]
    l.append(a)
l = list(set(l))
lst = []
for i in l:
    res = i.strip('][').split(', ')
    lst.append(res)
d=[]
for i in lst:
    for j in i:
        d.append(j)
dff=pd.DataFrame(d)
dff = dff[0].value_counts()

dff = dff[:20, ]
plt.figure(figsize = (12, 8))
sns.barplot(dff.values, dff.index, alpha = 0.8)
plt.title("Top 20 Artists", fontdict = {'fontsize' : 15})
plt.ylabel("Artists")
plt.xlabel("Count")
plt.show()
```

```
c:\users\bae\appdata\local\programs\python\python39\lib\site-packages\seabor
n\_decorators.py:36: FutureWarning: Pass the following variables as keyword
args: x, y. From version 0.12, the only valid positional argument will be `d
ata`, and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```



In [28]:

```
latesttracks.groupby('popularity').size()
```

Out[28]:

```
popularity
0      3984
1      1262
2       900
3       648
4       512
       ...
96        2
97        2
98        1
99        1
100       1
Length: 101, dtype: int64
```

In [29]:

```
latesttracks.popularity.count()
```

Out[29]:

```
52950
```

In [30]:

```
popularlatesttracks=latesttracks[latesttracks['popularity'] > 0]

popularlatesttracks.head()
```

Out[30]:

	id	name	popularity	duration_ms	explicit	artists	
39511	6Pkt6qVikqPBt9bEQy8iTz	A Lover's Concerto	41	159560	0	['The Toys']	['6lI
39529	1hx7X9cMXHWJjknb9O6Ava	The September Of My Years - Live At The Sands ...	26	187333	0	['Frank Sinatra']	['1Mx
39533	19oquvXf3bc65GSqtPYA5S	It Was A Very Good Year - Live At The Sands Ho...	25	236800	0	['Frank Sinatra']	['1Mx
39581	55qyghODi24yaDgKBI6Ix0	The Circle Game - Live at The 2nd Fret, Philad...	18	313093	0	['Joni Mitchell']	['5hW4
39583	00xemFYjQNRpOIPhVaLAHa	Urge For Going - Live at The 2nd Fret, Philade...	18	295093	0	['Joni Mitchell']	['5hW4

5 rows × 21 columns

In [31]:

```
popularlatesttracks.count()
```

Out[31]:

```
id                  48966
name                48966
popularity          48966
duration_ms         48966
explicit            48966
artists             48966
id_artists          48966
release_date        48966
danceability        48966
energy              48966
key                 48966
loudness            48966
mode                48966
speechiness         48966
acousticness        48966
instrumentalness    48966
liveness            48966
valence             48966
tempo               48966
time_signature      48966
year                48966
dtype: int64
```

In [32]:

```
populartracks=tracks[tracks['popularity'] > 0]

populartracks.head()
populartracks.count()
```

Out[32]:

```
id                  541970
name                541970
popularity          541970
duration_ms         541970
explicit            541970
artists             541970
id_artists          541970
release_date        541970
danceability        541970
energy              541970
key                 541970
loudness            541970
mode                541970
speechiness         541970
acousticness        541970
instrumentalness    541970
liveness            541970
valence             541970
tempo               541970
time_signature      541970
year                541970
dtype: int64
```

In [33]:

```
popularlatesttracks.groupby('popularity').size()
```

Out[33]:

```
popularity
1      1262
2       900
3       648
4       512
5       478
       ...
96        2
97        2
98        1
99        1
100       1
Length: 100, dtype: int64
```

In [34]:

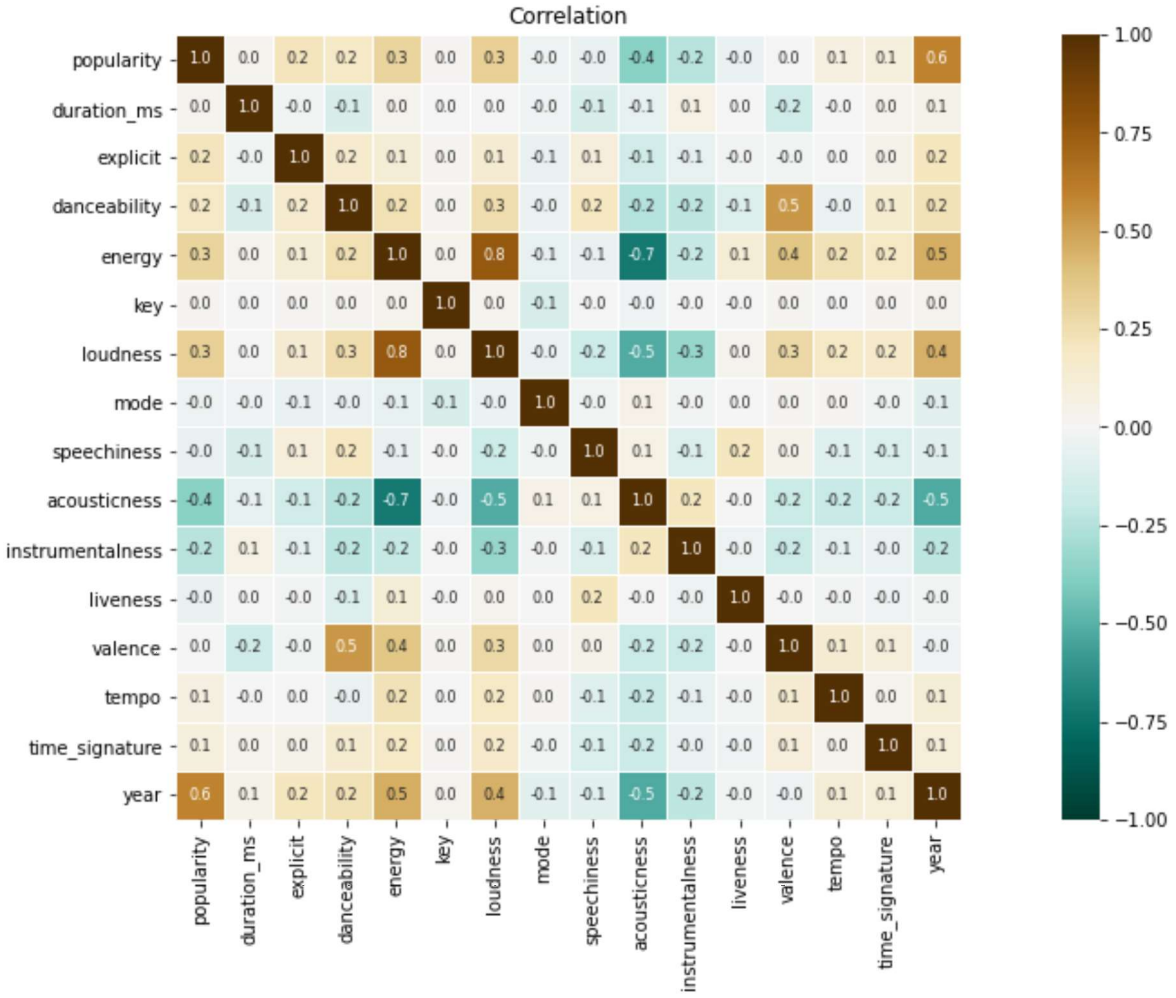
```
populartracks.groupby('popularity').size()
```

Out[34]:

```
popularity
1      12024
2       9634
3       8152
4       7732
5       7730
       ...
96         2
97         2
98         1
99         1
100        1
Length: 100, dtype: int64
```

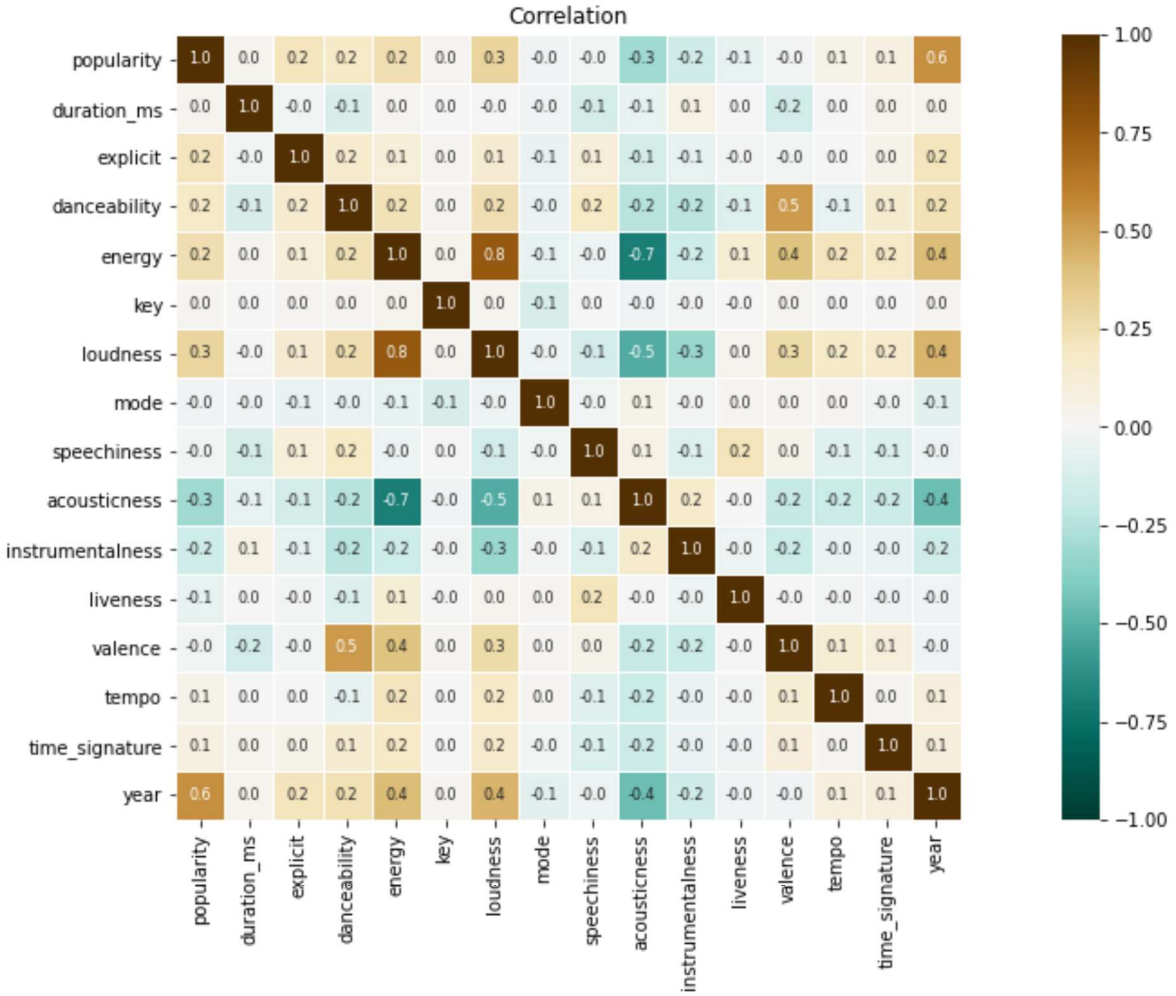
In [35]:

```
corr = tracks.corr()
plt.figure(figsize=(20,8))
sns.heatmap(corr, vmax=1, vmin=-1, center=0,linewidth=.5,square=True, annot = True, annot_k
plt.title('Correlation')
plt.show()
```



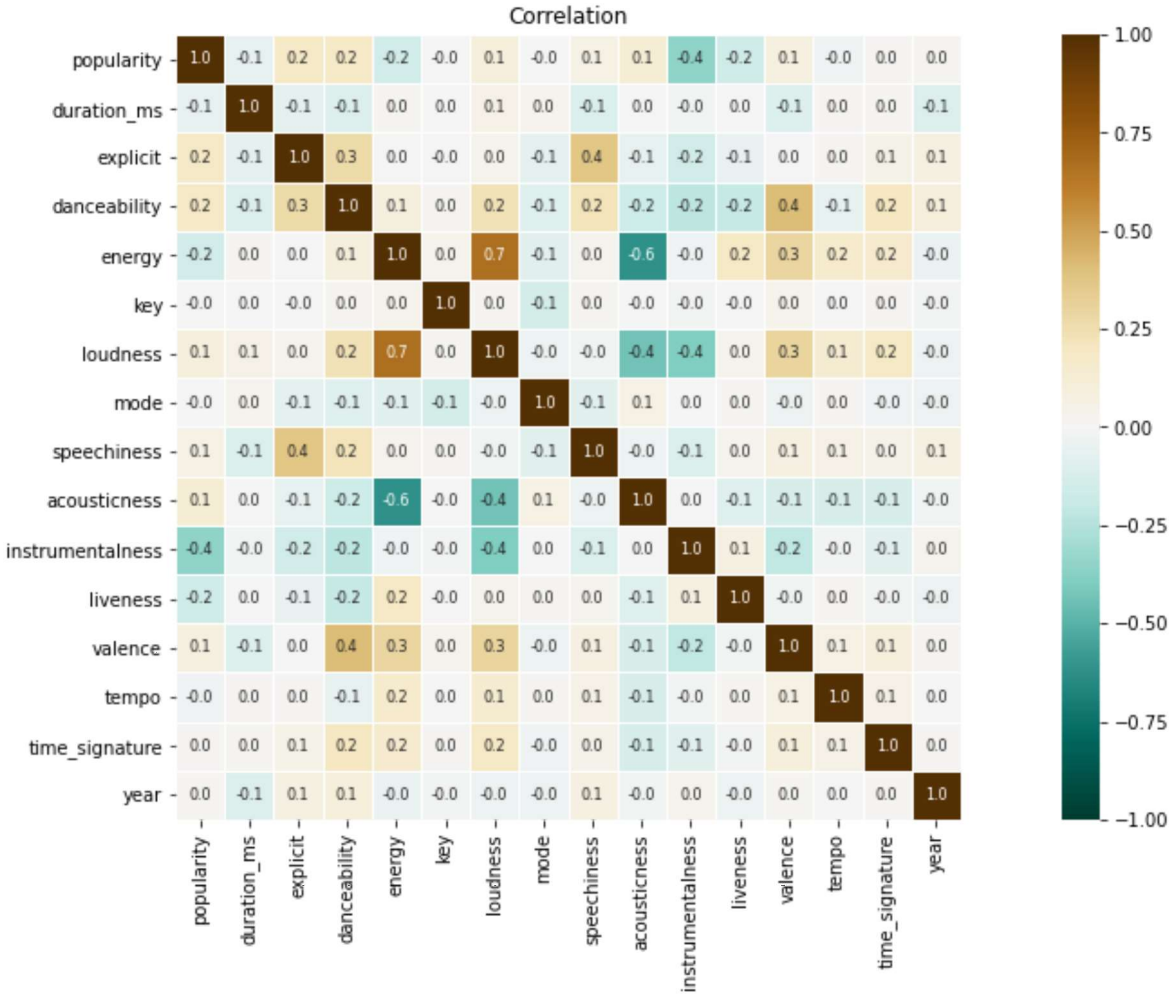
In [36]:

```
corr = popularartracks.corr()
plt.figure(figsize=(20,8))
sns.heatmap(corr, vmax=1, vmin=-1, center=0,linewidth=.5,square=True, annot = True, annot_k
plt.title('Correlation')
plt.show()
```



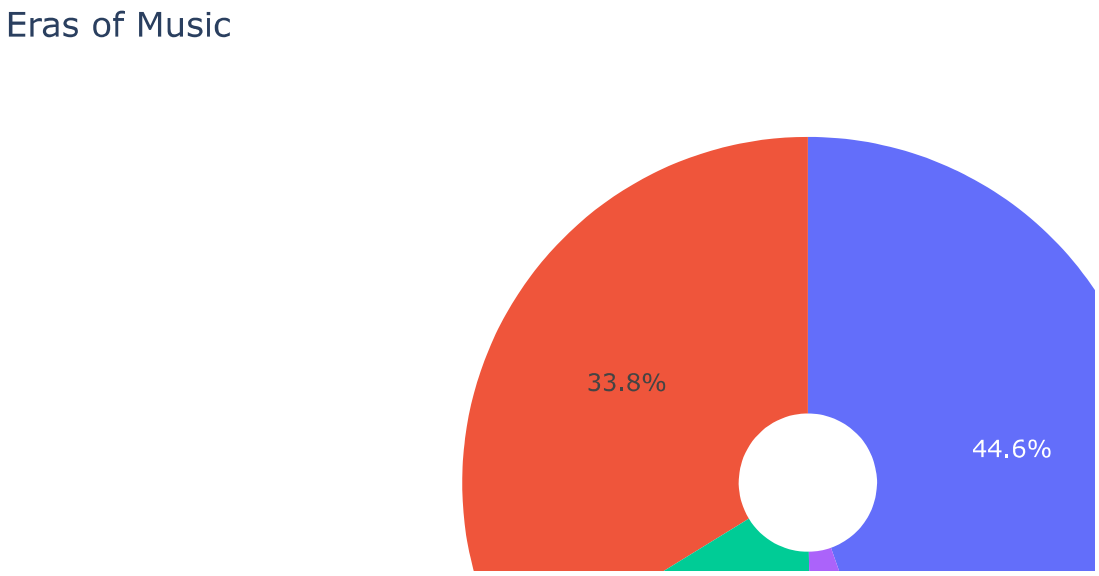
In [37]:

```
corr = popularlatesttracks.corr()
plt.figure(figsize=(20,8))
sns.heatmap(corr, vmax=1, vmin=-1, center=0,linewidth=.5,square=True, annot = True, annot_k
plt.title('Correlation')
plt.show()
```



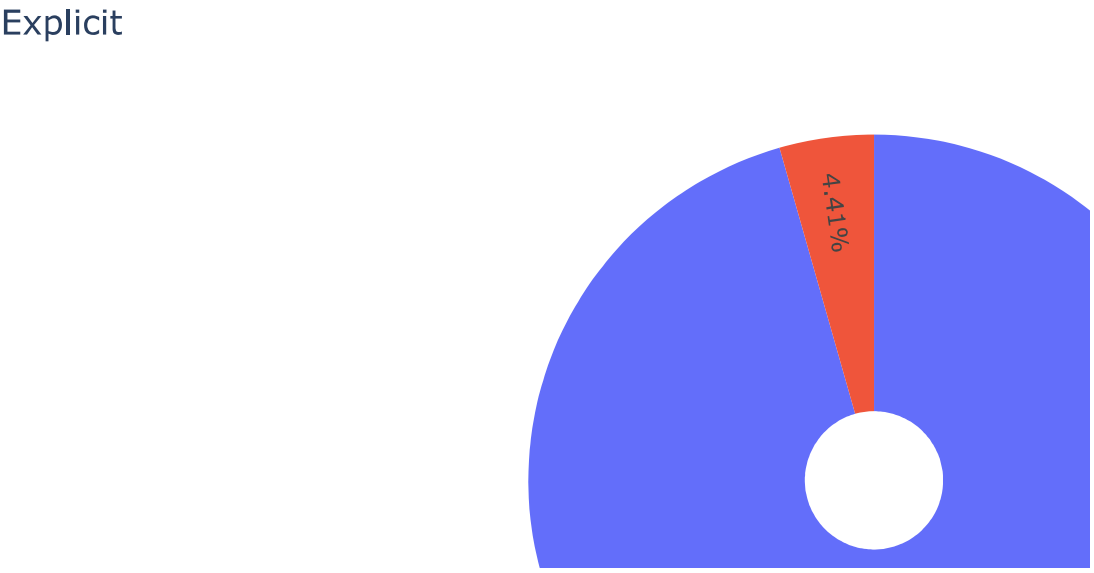
In [38]:

```
import plotly.express as px

def year(yr):
    if yr>1920 and yr<=1945:
        return "Post-Great War"
    if yr>1945 and yr<=1970:
        return "Retro"
    if yr>1970 and yr<=1995:
        return "Modern"
    else:
        return "Post-Modern"
tracks['era'] = tracks['year'].apply(year)
px.pie(data_frame = tracks, names = 'era', hole = 0.2, title = 'Eras of Music')
```



In [39]:

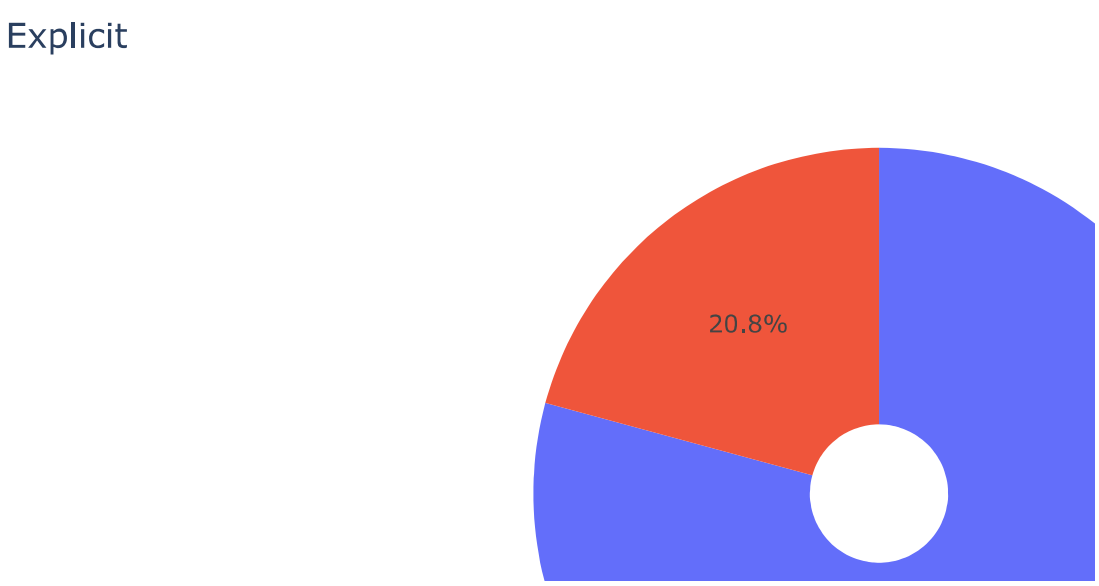
```
def expl(ex):
    if ex == 1:
        return 'Yes'
    else:
        return 'No'

tracks['isExplicit'] = tracks['explicit'].apply(expl)
px.pie(data_frame = tracks, names = 'isExplicit', hole = 0.2, title = 'Explicit')
```



In [40]:

```
latesttracks['isExplicit'] = tracks['explicit'].apply(expl)
px.pie(data_frame = latesttracks, names = 'isExplicit', hole = 0.2, title = 'Explicit')
```

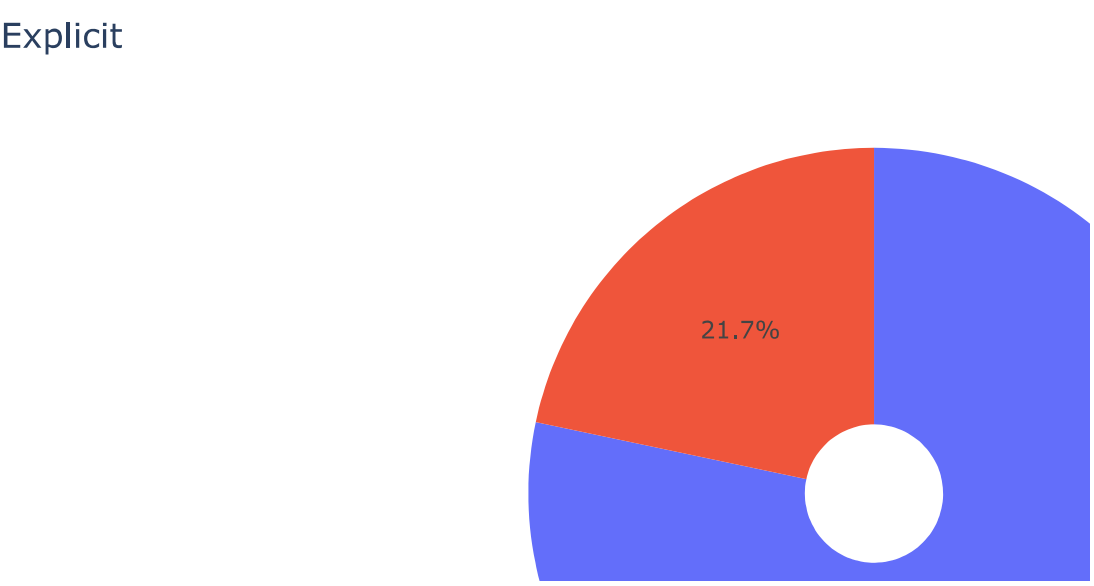
```
<ipython-input-40-cc4a06656881>:1: SettingWithCopyWarning:


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 (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
```



In [41]:

```
popularlatesttracks['isExplicit'] = popularlatesttracks['explicit'].apply(expl)
px.pie(data_frame = popularlatesttracks, names = 'isExplicit', hole = 0.2, title = 'Explici
```

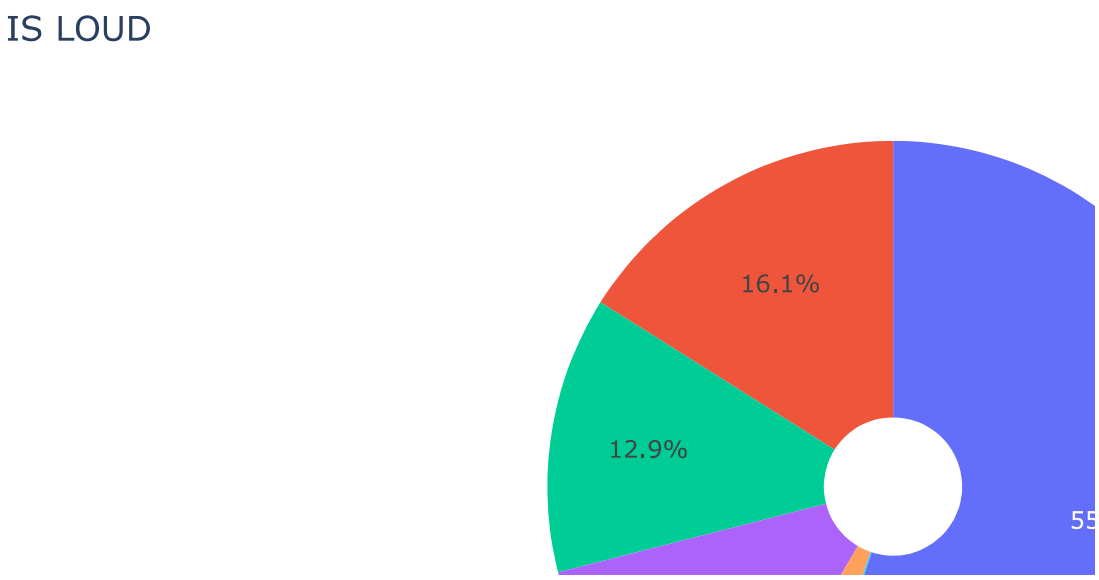
```
<ipython-input-41-e65aa03eb822>:1: SettingWithCopyWarning:


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 (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
```



In [42]:

```
def loud(row):
    m=popularlatesttracks['loudness'].median()
    sd=popularlatesttracks['loudness'].std()
    if row['loudness']>=m+(1.5*sd):
        return "Extreme"
    elif row['loudness']>=m+(sd):
        return "Very Loud"
    elif row['loudness']>=m+(0.5*sd):
        return "Loud"
    elif row['loudness']>=m-(0.5*sd):
        return "Soft"
    elif row['loudness']>=m-(sd):
        return "Very Soft"
    else:
        return "Mellow"
popularlatesttracks['is_loud']=popularlatesttracks.apply(lambda row: loud(row), axis=1)
px.pie(data_frame = popularlatesttracks, names = 'is_loud', hole = 0.2, title = 'IS LOUD')
```

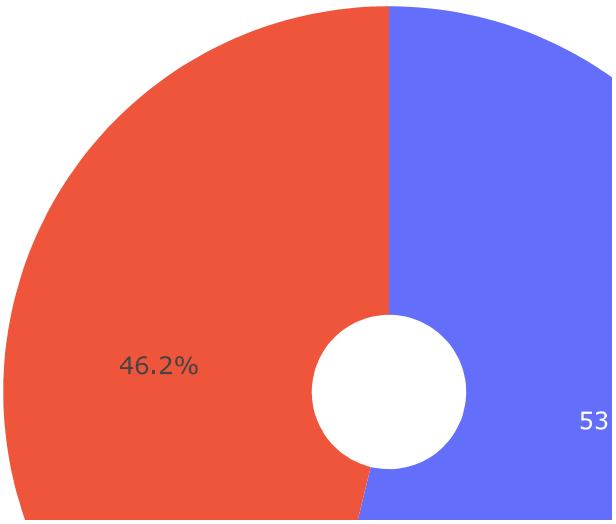
```
<ipython-input-42-05705ead552e>:16: SettingWithCopyWarning:


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 (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
```



In [43]:

```
def energy(row):
    if row['energy']>=popularlatesttracks['energy'].mean():
        return "High"
    else:
        return "Low"
popularlatesttracks['en_type']=popularlatesttracks.apply(lambda row: energy(row),axis=1)
px.pie(names=popularlatesttracks['en_type'],hole=0.2)
```

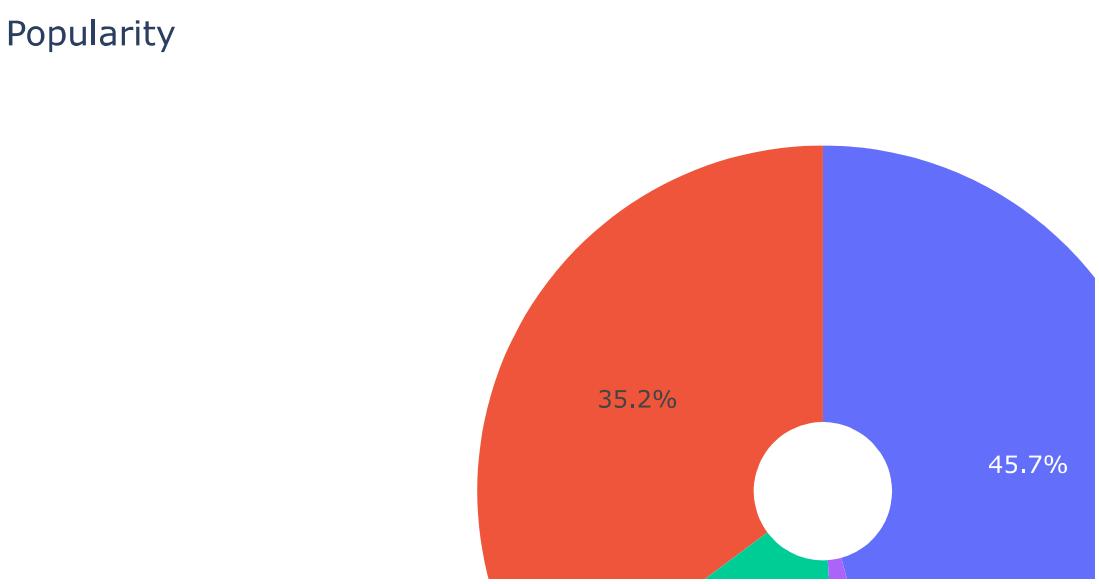
```
<ipython-input-43-a04a82702b0c>:6: SettingWithCopyWarning:


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 (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
```



In [44]:

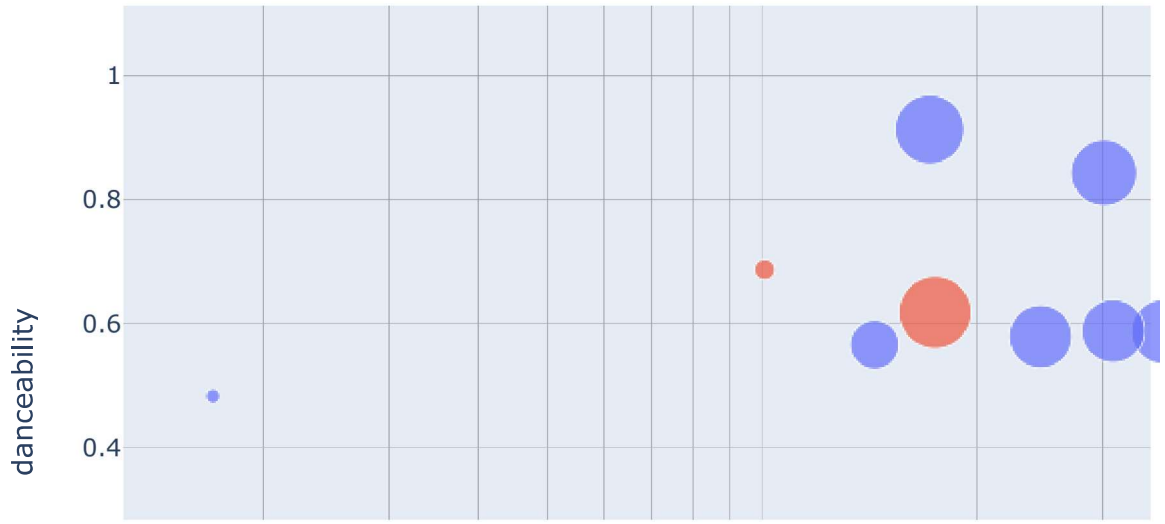
```
def func(df):
    if df > 75:
        return 'Very Popular'
    elif df > 50 and df < 76:
        return 'Popular'
    elif df > 25 and df < 51:
        return 'Average'
    else:
        return 'Not popular'
popularlatesttracks['isPopular'] = popularlatesttracks['popularity'].apply(func)
px.pie(data_frame = popularlatesttracks, names = 'isPopular', hole = 0.2, title = 'Populari
```

```
<ipython-input-44-37419a1483af>:10: 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 (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
```



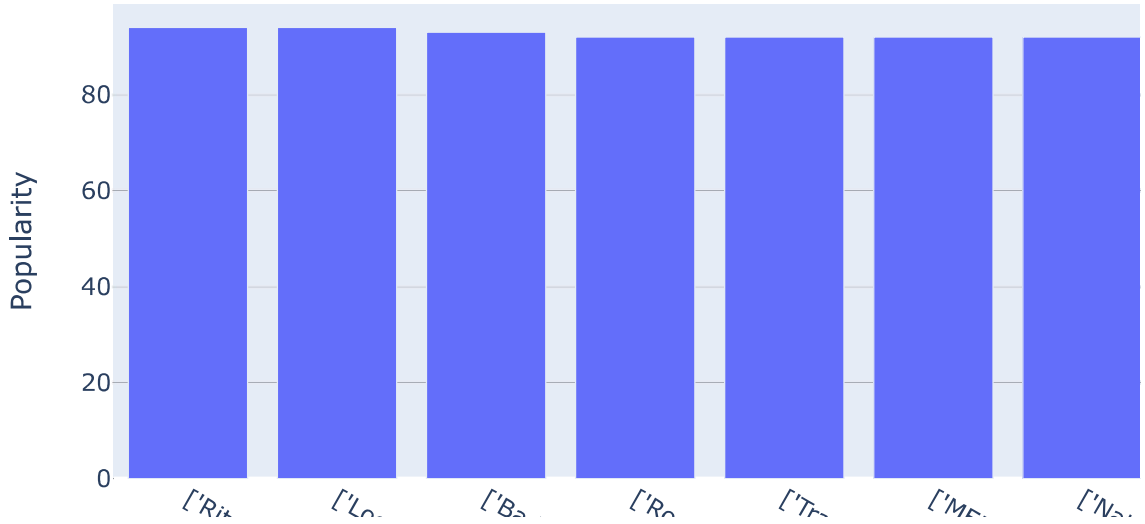
In [45]:

```
fig = px.scatter(popularlatesttracks, x="loudness", y="danceability", size="popularity",
                 color="isExplicit", log_x=True,size_max=30)
fig.show()
```



In [46]:

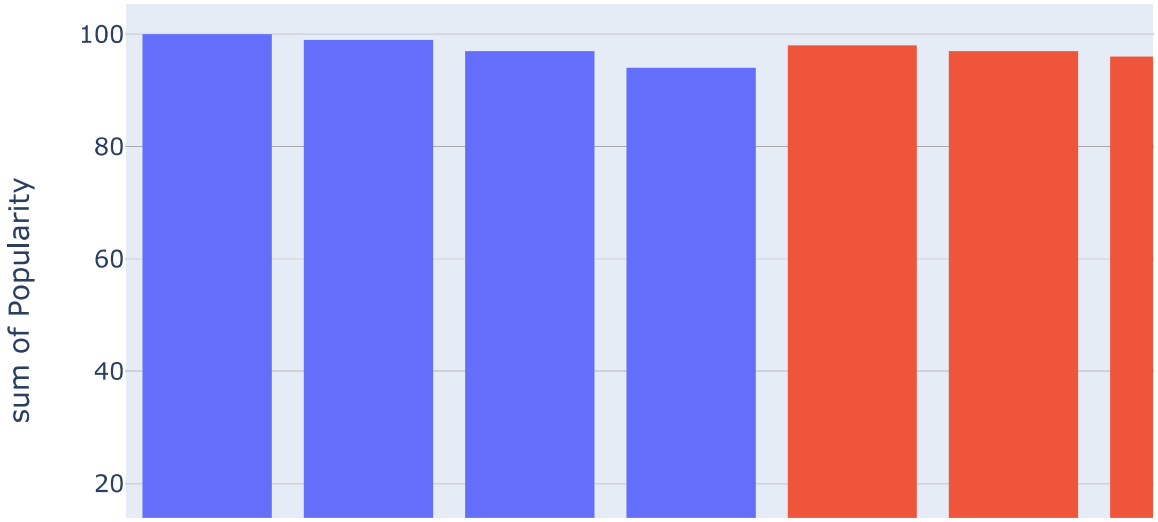
```
art=latesttracks
n=[]
g=[]
for name, group in art.groupby(['artists'])['popularity']:
    n.append(name)
    g.append(group.mean())
artist_pop=pd.DataFrame(n,g)

artist_pop.columns=['Name']
artist_pop['popularity']=artist_pop.index
artist_pop.sort_values(by='popularity',ascending=False,inplace=True)
px.bar(x=artist_pop['Name'].head(10),y=artist_pop['popularity'].head(10)).update_layout(yax
```



In [47]:

```
latesttracks.sort_values(by='popularity',ascending=False,inplace=True)
dff=latesttracks.head(10)
px.histogram(x=dff.name,y=dff.popularity,color=dff.isExplicit,labels={'x':'Top 10 songs','y
```

```
<ipython-input-47-67cd437abf3f>:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
```



In [48]:

```
isPopular= pd.get_dummies(popularlatesttracks['isPopular'], drop_first=True)
isExplicit= pd.get_dummies(popularlatesttracks['isExplicit'], drop_first=True)
isLoud= pd.get_dummies(popularlatesttracks['is_loud'], drop_first=True)
popularlatesttracks = pd.concat([popularlatesttracks, isPopular, isExplicit, isLoud], axis
popularlatesttracks.head()
```

Out[48]:

	id	name	popularity	duration_ms	explicit	artists	
39511	6Pkt6qVikqPBt9bEQy8iTz	A Lover's Concerto	41	159560	0	['The Toys']	['6I
39529	1hx7X9cMXHWJjknb9O6Ava	The September Of My Years - Live At The Sands ...	26	187333	0	['Frank Sinatra']	['1Mx
39533	19oquvXf3bc65GSqtPYA5S	It Was A Very Good Year - Live At The Sands Ho...	25	236800	0	['Frank Sinatra']	['1Mx
39581	55qyghODi24yaDgKBI6Ix0	The Circle Game - Live at The 2nd Fret, Philad...	18	313093	0	['Joni Mitchell']	['5hW4
39583	00xemFYjQNRpOIPhVaLAHa	Urge For Going - Live at The 2nd Fret, Philade...	18	295093	0	['Joni Mitchell']	['5hW4

5 rows × 34 columns

In [49]:

```
popularlatesttracks.head()
```

Out[49]:

	id	name	popularity	duration_ms	explicit	artists	
39511	6Pkt6qVikqPBt9bEQy8iTz	A Lover's Concerto	41	159560	0	['The Toys']	['6I
39529	1hx7X9cMXHWJjknb9O6Ava	The September Of My Years - Live At The Sands ...	26	187333	0	['Frank Sinatra']	['1Mx
39533	19oquvXf3bc65GSqtPYA5S	It Was A Very Good Year - Live At The Sands Ho...	25	236800	0	['Frank Sinatra']	['1Mx
39581	55qyghODi24yaDgKBI6Ix0	The Circle Game - Live at The 2nd Fret, Philad...	18	313093	0	['Joni Mitchell']	['5hW4
39583	00xemFYjQNRpOIPhVaLAHa	Urge For Going - Live at The 2nd Fret, Philade...	18	295093	0	['Joni Mitchell']	['5hW4

5 rows × 34 columns

In [50]:

```
popularlatesttracks.drop(['isPopular', 'isExplicit', 'is_loud', 'id', 'name','en_type','id_
            axis = 1, inplace = True)
```

In [51]:

```
popularlatesttracks.head()
```

Out[51]:

	popularity	explicit	danceability	energy	key	loudness	mode	speechiness	acousticn
39511	41	0	0.671	0.867	2	-2.706	1	0.0571	0.
39529	26	0	0.319	0.201	7	-17.796	1	0.0623	0.
39533	25	0	0.269	0.129	7	-18.168	0	0.0576	0.
39581	18	0	0.644	0.212	11	-14.118	1	0.0347	0.
39583	18	0	0.627	0.184	1	-15.533	1	0.0450	0.

5 rows × 24 columns

In [52]:

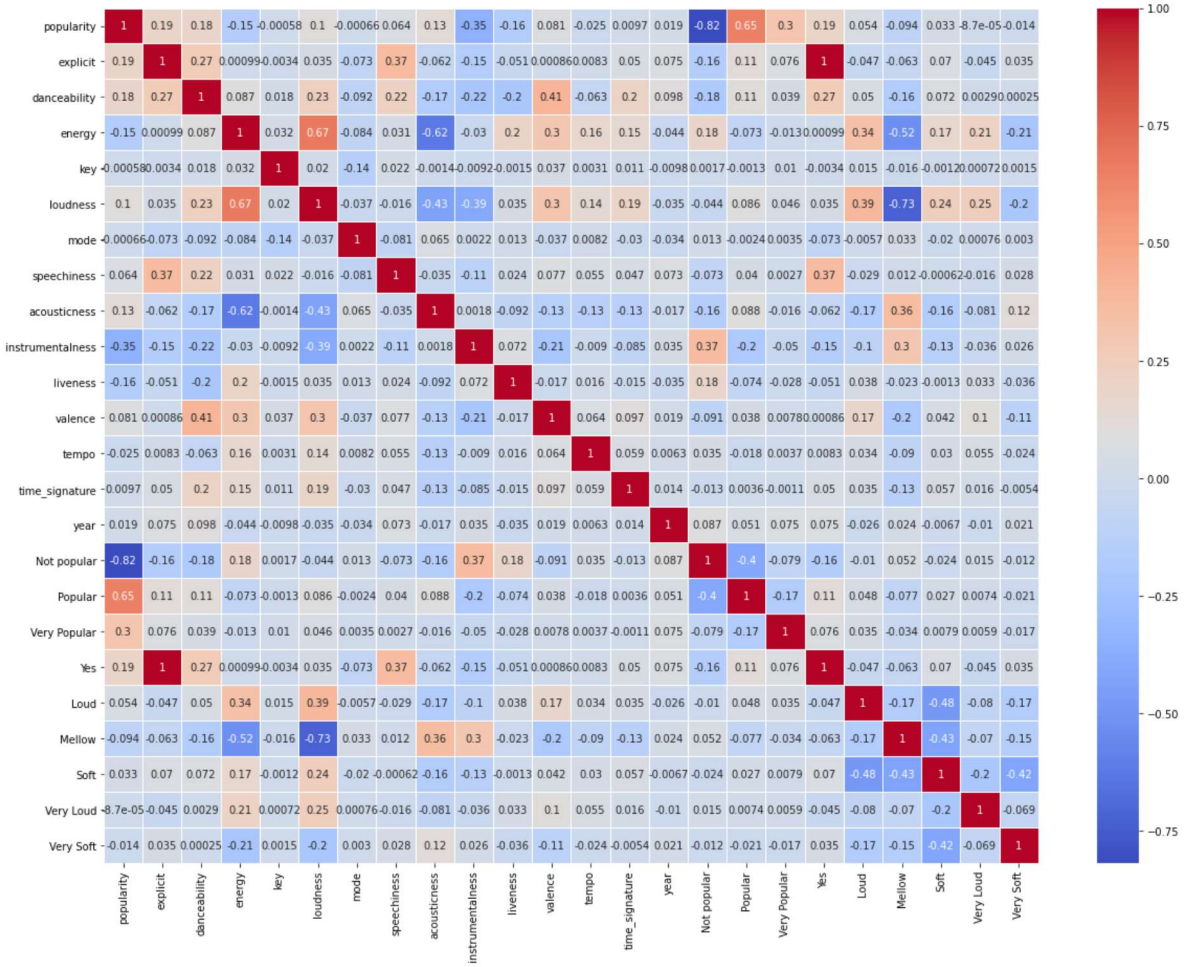
```
popularlatesttracks.columns
```

Out[52]:

```
Index(['popularity', 'explicit', 'danceability', 'energy', 'key', 'loudnes
s',
       'mode', 'speechiness', 'acousticness', 'instrumentalness', 'livenes
s',
       'valence', 'tempo', 'time_signature', 'year', 'Not popular', 'Popula
r',
       'Very Popular', 'Yes', 'Loud', 'Mellow', 'Soft', 'Very Loud',
       'Very Soft'],
      dtype='object')
```

In [53]:

```
plt.figure(figsize=(20,15))
sns.heatmap(popularlatesttracks.corr(),linecolor='white',linewidths=1,cmap='coolwarm',annot
plt.show()
```



In [54]:

```
X= popularlatesttracks.loc[:,popularlatesttracks.columns!='popularity']
y= popularlatesttracks.loc[:,popularlatesttracks.columns=='popularity']
X.head()
```

Out[54]:

	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrun
39511	0	0.671	0.867	2	-2.706	1	0.0571	0.436	
39529	0	0.319	0.201	7	-17.796	1	0.0623	0.887	
39533	0	0.269	0.129	7	-18.168	0	0.0576	0.938	
39581	0	0.644	0.212	11	-14.118	1	0.0347	0.881	
39583	0	0.627	0.184	1	-15.533	1	0.0450	0.955	

5 rows × 23 columns

In [55]:

```
y.head()
```

Out[55]:

	popularity
39511	41
39529	26
39533	25
39581	18
39583	18

In [56]:

```
from sklearn.preprocessing import StandardScaler
sc_X= StandardScaler()
sc_y= StandardScaler()
X=sc_X.fit_transform(X)
y=sc_y.fit_transform(y)
print(X)
print(y)
```

```
[[-0.52616734  0.13316652  1.11410068 ... -1.10598414  5.48179671
  -0.37903114]
 [-0.52616734 -2.12145849 -2.19268299 ... -1.10598414 -0.18242194
  -0.37903114]
 [-0.52616734 -2.44171772 -2.55017312 ... -1.10598414 -0.18242194
  -0.37903114]
 ...
 [-0.52616734  0.73525388  0.10121199 ...  0.9041721  -0.18242194
  -0.37903114]
 [-0.52616734 -0.7379386  -1.6316221  ... -1.10598414 -0.18242194
  -0.37903114]
 [-0.52616734  0.29329613 -0.13711476 ...  0.9041721  -0.18242194
  -0.37903114]]
[[-0.25260478]
 [-0.99588622]
 [-1.04543832]
 ...
 [ 1.28351021]
 [ 1.18440601]
 [ 0.58978086]]
```

In [57]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25, random_state=0)
```

In [58]:

```
from sklearn.linear_model import LinearRegression
regressor_lin=LinearRegression()
regressor_lin.fit(X_train,y_train)
```

Out[58]:

```
LinearRegression()
```

In [59]:

```
y_pred_lin = regressor_lin.predict(X_test)
```

In [60]:

```
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
```

In [61]:

```
print("Training Score of Linear Regression is: {}\n".format(regressor_lin.score(X_train, y_
print("R2 Score of Linear Regression is: {}\n".format(r2_score(y_test, y_pred_lin)))
print("Mean Squared Error of Linear Regression is: {}\n".format(mean_squared_error(y_test,
print("Mean Absolute Error of Linear Regression is: {}\n".format(mean_absolute_error(y_test
```

```
Training Score of Linear Regression is: 0.8974998370297538

R2 Score of Linear Regression is: 0.8987325755443236

Mean Squared Error of Linear Regression is: 0.10290762711672685

Mean Absolute Error of Linear Regression is: 0.2667010322333701
```

In [67]:

```
fig, ax = plt.subplots()
#ax.scatter(y_test, y_pred_lin)
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4)
ax.set_xlabel('Actual')
ax.set_ylabel('Predicted')
#regression line
y_test, y_predicted = y_test.reshape(-1,1), y_pred_lin.reshape(-1,1)
ax.plot(y_test, LinearRegression().fit(y_test, y_pred_lin).predict(y_test))
ax.set_title('R2: ' + str(r2_score(y_test, y_predicted))
plt.show()
```

```
  File "<ipython-input-67-d37124e0b5ef>", line 10
    plt.show()
    ^
SyntaxError: invalid syntax
```

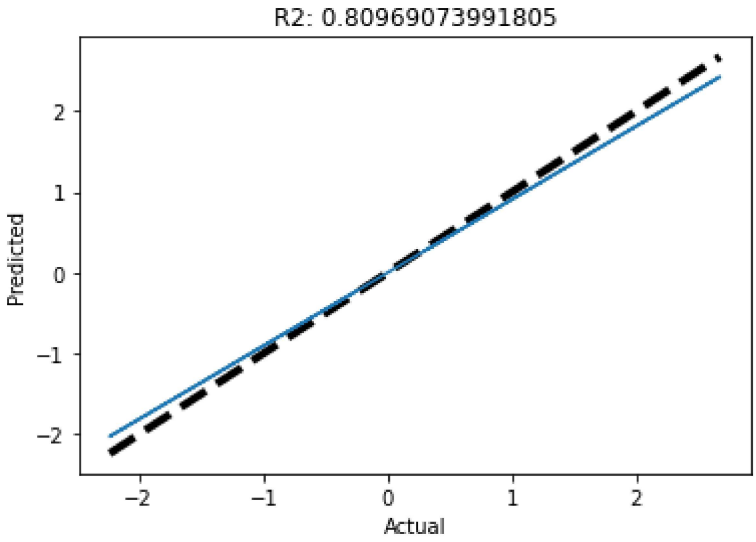
In [68]:

```
from sklearn.tree import DecisionTreeRegressor
regressor_dt= DecisionTreeRegressor(random_state=0)
regressor_dt.fit(X_train,y_train)

y_pred_dt= regressor_dt.predict(X_test)

print("Training Score of Decision Tree Regressor is: {}\n".format(regressor_dt.score(X_trai
print("R2 Score of Decision Tree Regressor is: {}\n".format(r2_score(y_test, y_pred_dt)))
print("Mean Squared Error of Decision Tree Regressor is: {}\n".format(mean_squared_error(y_
print("Mean Absolute Error of Decision Tree Regressor is: {}\n".format(mean_absolute_error(
```

```
Training Score of Decision Tree Regressor is: 0.9974754575813249

R2 Score of Decision Tree Regressor is: 0.80969073991805

Mean Squared Error of Decision Tree Regressor is: 0.19339165065805858

Mean Absolute Error of Decision Tree Regressor is: 0.3434141196652201
```

In [69]:

```
fig, ax = plt.subplots()
#ax.scatter(y_test, y_pred_lin)
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4)
ax.set_xlabel('Actual')
ax.set_ylabel('Predicted')
#regression line
y_test, y_predicted = y_test.reshape(-1,1), y_pred_dt.reshape(-1,1)
ax.plot(y_test, LinearRegression().fit(y_test, y_pred_dt).predict(y_test))
ax.set_title('R2: ' + str(r2_score(y_test, y_predicted)))
plt.show()
```



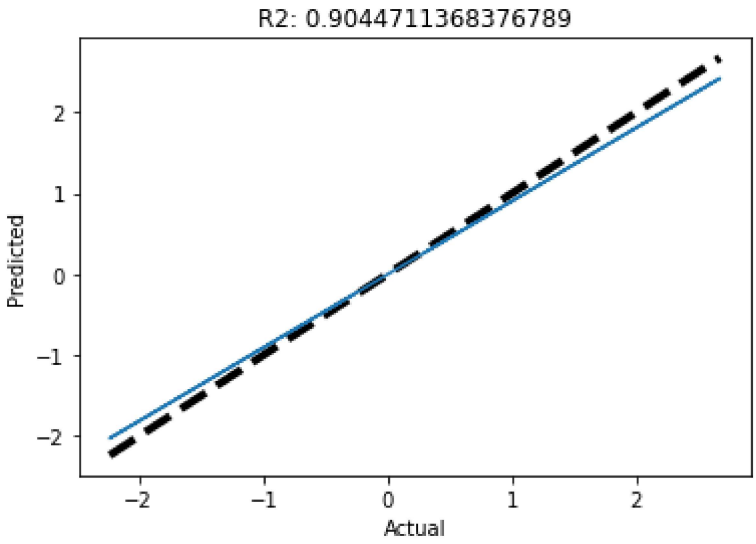
In [70]:

```
from sklearn.ensemble import RandomForestRegressor
regressor_rf=RandomForestRegressor(n_estimators=300, random_state=0)
regressor_rf.fit(X_train,y_train)

y_pred_rf=regressor_rf.predict(X_test)

print("Training Score of Random Forest Regression is: {}\n".format(regressor_rf.score(X_tra
print("R2 Score of Random Forest Regression is: {}\n".format(r2_score(y_test, y_pred_rf)))
print("Mean Squared Error of Random Forest Regression is: {}\n".format(mean_squared_error(y
print("Mean Absolute Error of Random Forest Regression is: {}\n".format(mean_absolute_error
```

```
<ipython-input-70-31d6bd563996>:3: DataConversionWarning:

A column-vector y was passed when a 1d array was expected. Please change the
shape of y to (n_samples,), for example using ravel().


Training Score of Random Forest Regression is: 0.9845976814863735

R2 Score of Random Forest Regression is: 0.9044711368376789

Mean Squared Error of Random Forest Regression is: 0.09707611980885064

Mean Absolute Error of Random Forest Regression is: 0.25704854887588974
```

In [71]:

```
fig, ax = plt.subplots()
#ax.scatter(y_test, y_pred_lin)
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4)
ax.set_xlabel('Actual')
ax.set_ylabel('Predicted')
#regression line
y_test, y_predicted = y_test.reshape(-1,1), y_pred_rf.reshape(-1,1)
ax.plot(y_test, LinearRegression().fit(y_test, y_pred_rf).predict(y_test))
ax.set_title('R2: ' + str(r2_score(y_test, y_predicted)))
plt.show()
```



In [72]:

```
```

In [73]:

```
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-73-cec5639e0e8d> in <module>
     11 test_scores = []
     12
---> 13 from imblearn.over_sampling import SMOTE
     14 oversample = SMOTE()
     15 X_balanced, y_balanced = oversample.fit_resample(X_scaled, y_train)

ModuleNotFoundError: No module named 'imblearn'
```

In [74]:

```
```

In [ ]:

```
```