

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

```
In [2]: df_tracks=pd.read_csv("spotify.csv")
df_tracks.head(10)
```

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	duration_ms
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.61100	0.389	99373
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOUsryehmNudP	1	0.24600	0.590	137373
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.95200	0.663	170267
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tlvf	0	0.70300	0.240	152427
4	Movie	Fabien Nataf	Ouverture	0lusIXpMROHdEPvSI1fTQK	4	0.95000	0.331	82625
5	Movie	Henri Salvador	Le petit souper aux chandelles	0Mf1jKa8eNAf1a4PwTbizj	0	0.74900	0.578	160627
6	Movie	Martin & les fées	Premières recherches (par Paul Ventimila, Lori...)	0NUiKYRd6jt1LKMVGkUdnZ	2	0.34400	0.703	212293
7	Movie	Laura Mayne	Let Me Let Go	0PbIF9YVD505GutwotpB5C	15	0.93900	0.416	240067
8	Movie	Chorus	Helka	0ST6uPfvaPpJLtQwhE6KfC	0	0.00104	0.734	226200
9	Movie	Le Club des Juniors	Les bisous des bisounours	0VSqZ3KStsjcfERGdcWpFO	10	0.31900	0.598	152694

```
In [3]: #null values  
pd.isnull(df_tracks)
```

232720	False									
232721	False									
232722	False									
232723	False									
232724	False									

232725 rows × 18 columns

In [4]: `pd.isnull(df_tracks).sum()`

Out[4]:

genre	0
artist_name	0
track_name	0
track_id	0
popularity	0
acousticness	0
danceability	0
duration_ms	0
energy	0
instrumentalness	0
key	0
liveness	0
loudness	0
mode	0
speechiness	0
tempo	0
time_signature	0
valence	0
dtype: int64	

In [5]: `#information
df_tracks.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 232725 entries, 0 to 232724
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   genre            232725 non-null   object 
 1   artist_name      232725 non-null   object 
 2   track_name       232725 non-null   object 
 3   track_id         232725 non-null   object 
 4   popularity       232725 non-null   int64  
 5   acousticness     232725 non-null   float64
 6   danceability     232725 non-null   float64
 7   duration_ms      232725 non-null   int64  
 8   energy            232725 non-null   float64
 9   instrumentalness 232725 non-null   float64
 10  key               232725 non-null   object 
 11  liveness          232725 non-null   float64
 12  loudness          232725 non-null   float64
 13  mode              232725 non-null   object 
 14  speechiness       232725 non-null   float64
 15  tempo              232725 non-null   float64
 16  time_signature    232725 non-null   object 
 17  valence            232725 non-null   float64
dtypes: float64(9), int64(2), object(7)
memory usage: 32.0+ MB
```

In [6]: `#popularity of the songs`

```
sorted_df=df_tracks.sort_values("popularity", ascending=True).head(10)  
sorted_df
```

Out[6]:

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	du
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.6110	0.389	
74954	Children's Music	Sing n Play	See, See My Playmate	3WaCwwpGoxLEkFmd6cpZO5	0	0.1390	0.556	
74958	Children's Music	Children Songs Company	By the God - Instrumental	5yDehr9ccZo3vBO8hZXFcK	0	0.2130	0.736	
74959	Children's Music	Children Songs Company	Interruption Please - Instrumental	6NEULw7AmTOAYRunPWcFvn	0	0.0131	0.639	
74961	Children's Music	Children Songs Company	Breeze	6v126eNn1A79yFIKGfvqlz	0	0.0280	0.607	
74962	Children's Music	Children Songs Company	Woodland	7qR2PWbDeuhh1Vzd0anb25	0	0.9950	0.529	
74963	Children's Music	Children Songs Company	Cool Me - Instrumental	0V3Q9RYUaiLKTJlvg4KCST	0	0.2500	0.623	
74964	Children's Music	Sing n Play	The Tortoise and the Hare	0oX193rf5y87RipsZnH8Pq	0	0.4960	0.563	
56153	Movie	Bruno Pelletier	Lié par le sang	3GO9Wo14FvMXvOKE4Lltjg	0	0.0294	0.656	
74966	Children's Music	Children Songs Company	Forsaken	1rRNU87xfqvxiUsEXXyi8k	0	0.9940	0.429	

In [7]:

```
sorted_df=df_tracks.sort_values("popularity", ascending=True)  
sorted_df
```

Out[7]:

107802	Pop	Ariana Grande	break up with your girlfriend, i'm bored	4kV4N9D1iKVxx1KLvtTpjS	99	0.0421	0.726
86951	Rap	Post Malone	Wow.	6MWtB6iiXylwun0YzU6DFP	99	0.1630	0.833
107803	Pop	Post Malone	Wow.	6MWtB6iiXylwun0YzU6DFP	99	0.1630	0.833
9027	Dance	Ariana Grande	7 rings	14msK75pk3pA33pzPVNtBF	100	0.5780	0.725
107804	Pop	Ariana Grande	7 rings	14msK75pk3pA33pzPVNtBF	100	0.5780	0.725

232725 rows × 18 columns

In [8]:

```
#descriptive statistics
df_tracks.describe().transpose()
```

Out[8]:

	count	mean	std	min	25%	50%	75%
popularity	232725.0	41.127502	18.189948	0.00000	29.0000	43.000000	55.0000
acousticness	232725.0	0.368560	0.354768	0.00000	0.0376	0.232000	0.7220
danceability	232725.0	0.554364	0.185608	0.05690	0.4350	0.571000	0.6920
duration_ms	232725.0	235122.339306	118935.909299	15387.00000	182857.0000	220427.000000	265768.0000
energy	232725.0	0.570958	0.263456	0.00002	0.3850	0.605000	0.7870
instrumentalness	232725.0	0.148301	0.302768	0.00000	0.0000	0.000044	0.0358
liveness	232725.0	0.215009	0.198273	0.00967	0.0974	0.128000	0.2640
loudness	232725.0	-9.569885	5.998204	-52.45700	-11.7710	-7.762000	-5.5010
speechiness	232725.0	0.120765	0.185518	0.02220	0.0367	0.050100	0.1050
tempo	232725.0	117.666585	30.898907	30.37900	92.9590	115.778000	139.0540
valence	232725.0	0.454917	0.260065	0.00000	0.2370	0.444000	0.6600

In [9]:

```
# most popular songs
mp=df_tracks.query("popularity>90", inplace=False).sort_values("popularity", ascending=False)
mp.head(15)
```

Out[9]:

	genre	artist_name	track_name		track_id	popularity	acousticness	danceability	duration_ms
107804	Pop	Ariana Grande	7 rings	14msK75pk3pA33pzPVNtBF	100	0.5780	0.725		
9027	Dance	Ariana Grande	7 rings	14msK75pk3pA33pzPVNtBF	100	0.5780	0.725		
9026	Dance	Ariana Grande	break up with your girlfriend, i'm bored	4kV4N9D1iKVxx1KLvtTpjS	99	0.0421	0.726		
86951	Rap	Post Malone	Wow.	6MWtB6iiXylwun0YzU6DFP	99	0.1630	0.833		
107803	Pop	Post Malone	Wow.	6MWtB6iiXylwun0YzU6DFP	99	0.1630	0.833		
107802	Pop	Ariana Grande	break up with your	4kV4N9D1iKVxx1KLvtTpjS	99	0.0421	0.726		

girlfriend, i'm bored							
138918	Reggaeton	Daddy Yankee	Con Calma	5w9c2J52mkdntKOmRLeM2m	98	0.1100	0.737
66643	Hip-Hop	Daddy Yankee	Con Calma	5w9c2J52mkdntKOmRLeM2m	98	0.1100	0.737
107909	Pop	Daddy Yankee	Con Calma	5w9c2J52mkdntKOmRLeM2m	98	0.1100	0.737
86953	Rap	Post Malone	Sunflower - Spider-Man: Into the Spider- Verse	3KkXRkHbMCARz0aVfEt68P	97	0.5560	0.760
92824	Indie	Halsey	Without Me	5p7ujcrUXASCNwRaWNHR1C	97	0.2970	0.752
9037	Dance	Ava Max	Sweet but Psycho	25sgk305KZfyuqVBQIahim	97	0.0691	0.719
9028	Dance	Halsey	Without Me	5p7ujcrUXASCNwRaWNHR1C	97	0.2970	0.752
107875	Pop	Pedro Capó	Calma - Remix	5iwz1NiezX7WWjnCgY5TH4	97	0.3230	0.826
107806	Pop	Post Malone	Sunflower - Spider-Man: Into the Spider- Verse	3KkXRkHbMCARz0aVfEt68P	97	0.5560	0.760

```
In [10]: # setting one of the columns as index and convert to date time
df_tracks.set_index("energy", inplace=False)
df_tracks.index=pd.to_datetime(df_tracks.index)
df_tracks.head()
```

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability
1970-01-01 00:00:00.000000000	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.611	0
1970-01-01 00:00:00.000000001	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOUsryehmNudP	1	0.246	0
1970-01-01 00:00:00.000000002	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.952	0
1970-01-01 00:00:00.000000003	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tlfv	0	0.703	0
1970-01-01 00:00:00.000000004	Movie	Fabien Nataf	Ouverture	0lusIXpMROHdEPvSI1fTQK	4	0.950	0

```
In [11]: music=pd.read_csv("spotify.csv")
music.head(10)
```

Out[11]:

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	duration_ms
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.61100	0.389	99373
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOUsryehmNudP	1	0.24600	0.590	137373
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.95200	0.663	170267
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tlfv	0	0.70300	0.240	152427
4	Movie	Fabien Nataf	Ouverture	0lusIXpMROHdEPvSI1fTQK	4	0.95000	0.331	82625
5	Movie	Henri Salvador	Le petit souper aux chandelles	0Mf1jKa8eNAf1a4PwTbizj	0	0.74900	0.578	160627
6	Movie	Martin & les fées	Premières recherches (par Paul Ventimila, Lori...)	0NUiKYRd6jt1LKMVGkUdnZ	2	0.34400	0.703	212293
7	Movie	Laura Mayne	Let Me Let Go	0PbIF9YVD505GutwotpB5C	15	0.93900	0.416	240067
8	Movie	Chorus	Helka	0ST6uPfvaPpjLtQwhE6KfC	0	0.00104	0.734	226200
9	Movie	Le Club des Juniors	Les bisous des bisounours	0VSqZ3KStsjcfERGdcWpFO	10	0.31900	0.598	152694

In [12]:

```
#iloc function
music[["artist_name", "track_name"]].iloc[2330]
```

Out[12]:

```
artist_name      Brett Eldredge
track_name      Waited Too Long
Name: 2330, dtype: object
```

In [13]:

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

In [14]:

```
music.drop("duration_ms", inplace=True, axis=1)
```

In [15]:

```
music.head()
```

Out[15]:

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	energy	inst
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.611	0.389	0.910	
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOUsryehmNudP	1	0.246	0.590	0.737	

2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.952	0.663	0.131
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tlvf	0	0.703	0.240	0.326
4	Movie	Fabien Nataf	Ouverture	0lusIXpMROHdEPvSI1ftQK	4	0.950	0.331	0.225

In [16]: `music.duration.head()`

Out[16]:

```
0      99
1     137
2     170
3     152
4      83
Name: duration, dtype: int64
```

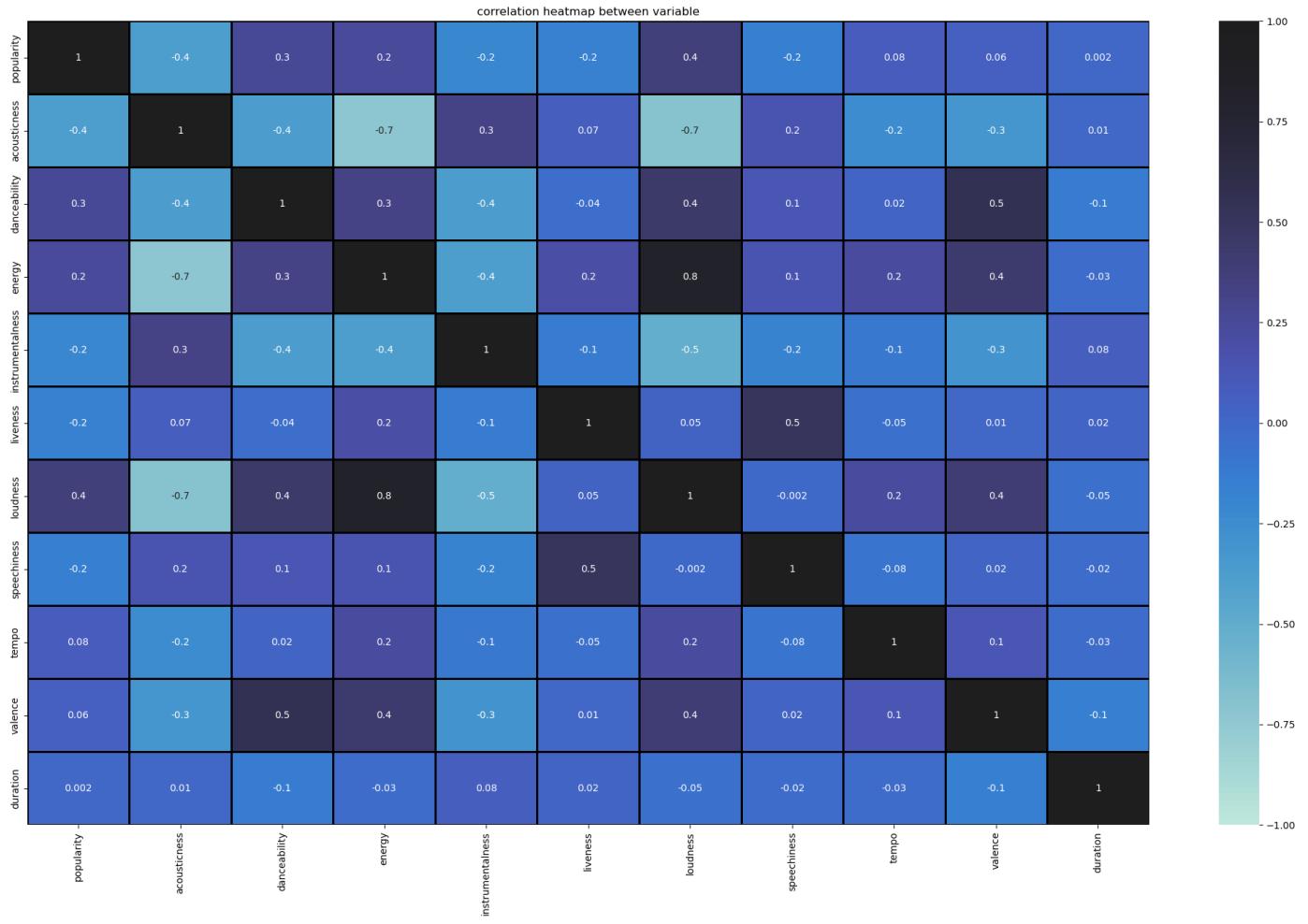
In [17]: `music.corr()`

	popularity	acousticness	danceability	energy	instrumentalness	liveness	loudness	speech
popularity	1.000000	-0.381295	0.256564	0.248922	-0.210983	-0.167995	0.363011	-0.15
acousticness	-0.381295	1.000000	-0.364546	-0.725576	0.316154	0.069004	-0.690202	0.15
danceability	0.256564	-0.364546	1.000000	0.325807	-0.364941	-0.041684	0.438668	0.13
energy	0.248922	-0.725576	0.325807	1.000000	-0.378957	0.192801	0.816088	0.14
instrumentalness	-0.210983	0.316154	-0.364941	-0.378957	1.000000	-0.134198	-0.506320	-0.17
liveness	-0.167995	0.069004	-0.041684	0.192801	-0.134198	1.000000	0.045686	0.51
loudness	0.363011	-0.690202	0.438668	0.816088	-0.506320	0.045686	1.000000	-0.00
speechiness	-0.151076	0.150935	0.134560	0.145120	-0.177147	0.510147	-0.002273	1.00
tempo	0.081039	-0.238247	0.021939	0.228774	-0.104133	-0.051355	0.228364	-0.08
valence	0.060076	-0.325798	0.547154	0.436771	-0.307522	0.011804	0.399901	0.02
duration	0.002347	0.011200	-0.125781	-0.030545	0.076017	0.023791	-0.047614	-0.01

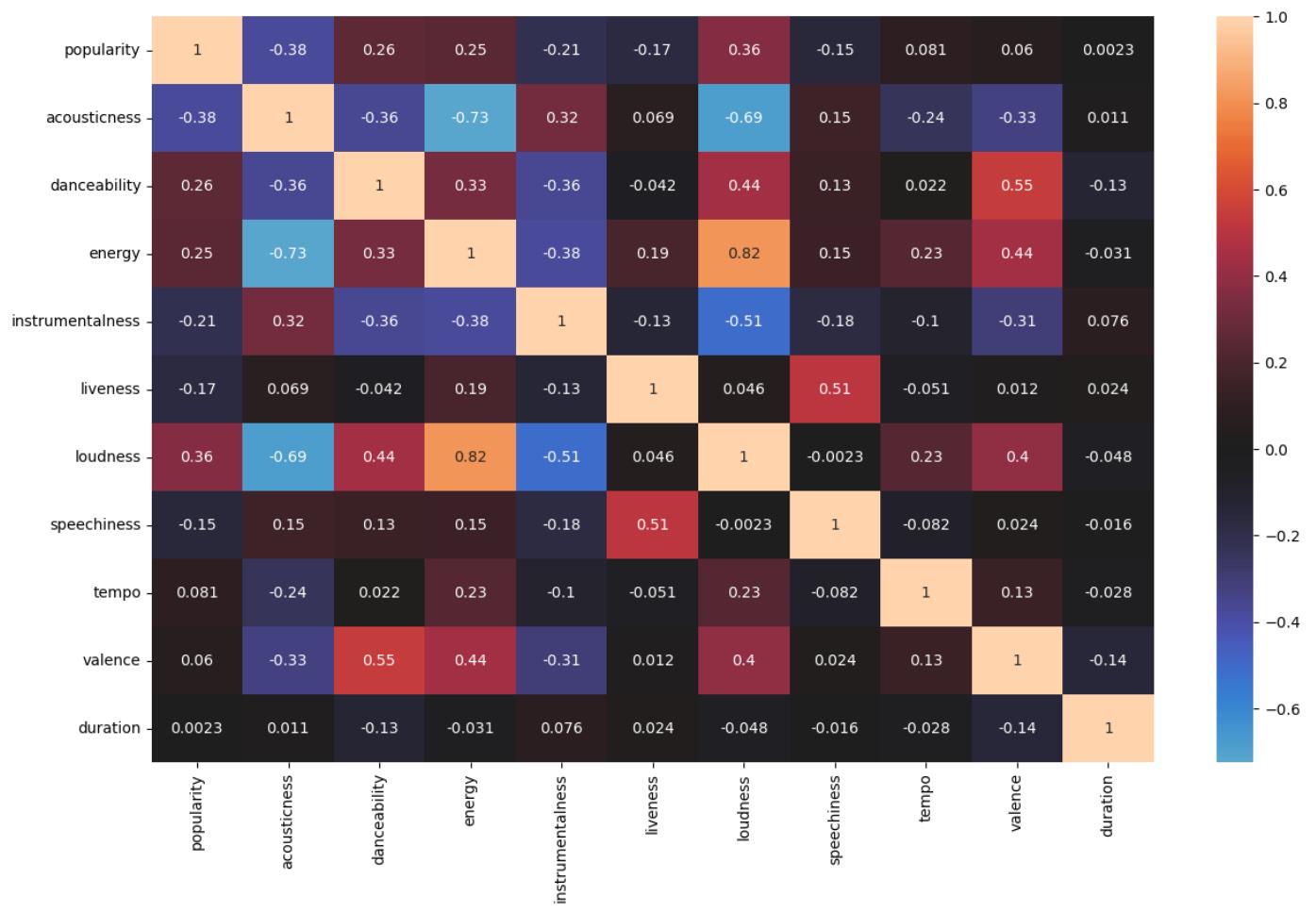
In [18]: `#visualization`
`corr_df=music.drop(["mode", "key"], axis=1).corr(method="pearson")`
`plt.figure(figsize=(26,15))`
`heatmap=sns.heatmap(corr_df, annot=True, fmt=".1g", vmin=-1, vmax=1, center=1, linewidths=1, li`
`heatmap.set_title("correlation heatmap between variable")`
`heatmap.set_xticklabels(heatmap.get_xticklabels(), rotation=90)`

Out[18]:

```
[Text(0.5, 0, 'popularity'),
 Text(1.5, 0, 'acousticness'),
 Text(2.5, 0, 'danceability'),
 Text(3.5, 0, 'energy'),
 Text(4.5, 0, 'instrumentalness'),
 Text(5.5, 0, 'liveness'),
 Text(6.5, 0, 'loudness'),
 Text(7.5, 0, 'speechiness'),
 Text(8.5, 0, 'tempo'),
 Text(9.5, 0, 'valence'),
 Text(10.5, 0, 'duration')]
```



```
In [19]: corr=music.corr()
plt.figure(figsize=(15,9))
heatmap=sns.heatmap(corr,center=0,annot=True)
```

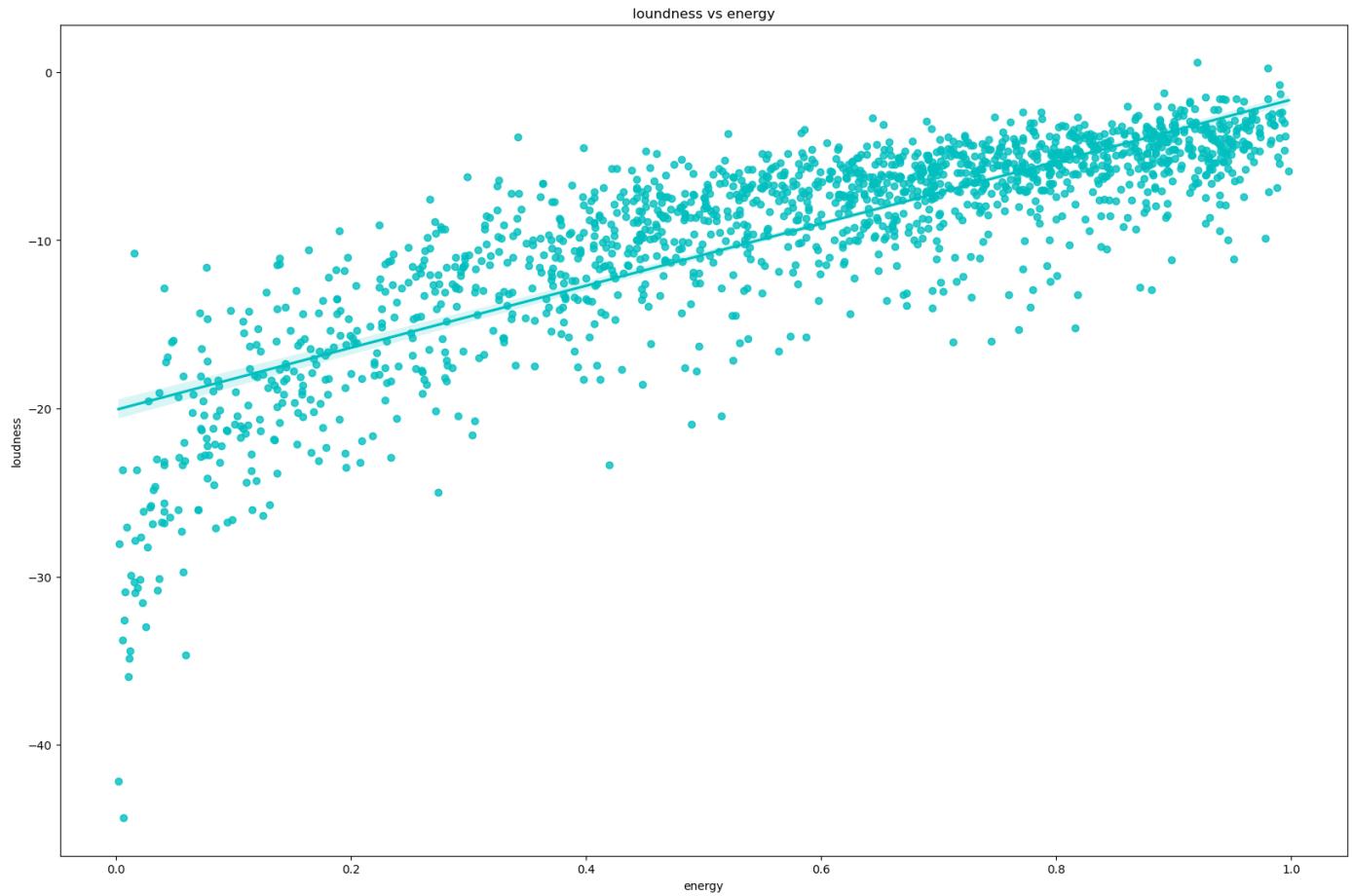


```
In [20]: sample_df=music.sample(int(0.007*len(music)))
print(len(sample_df))
```

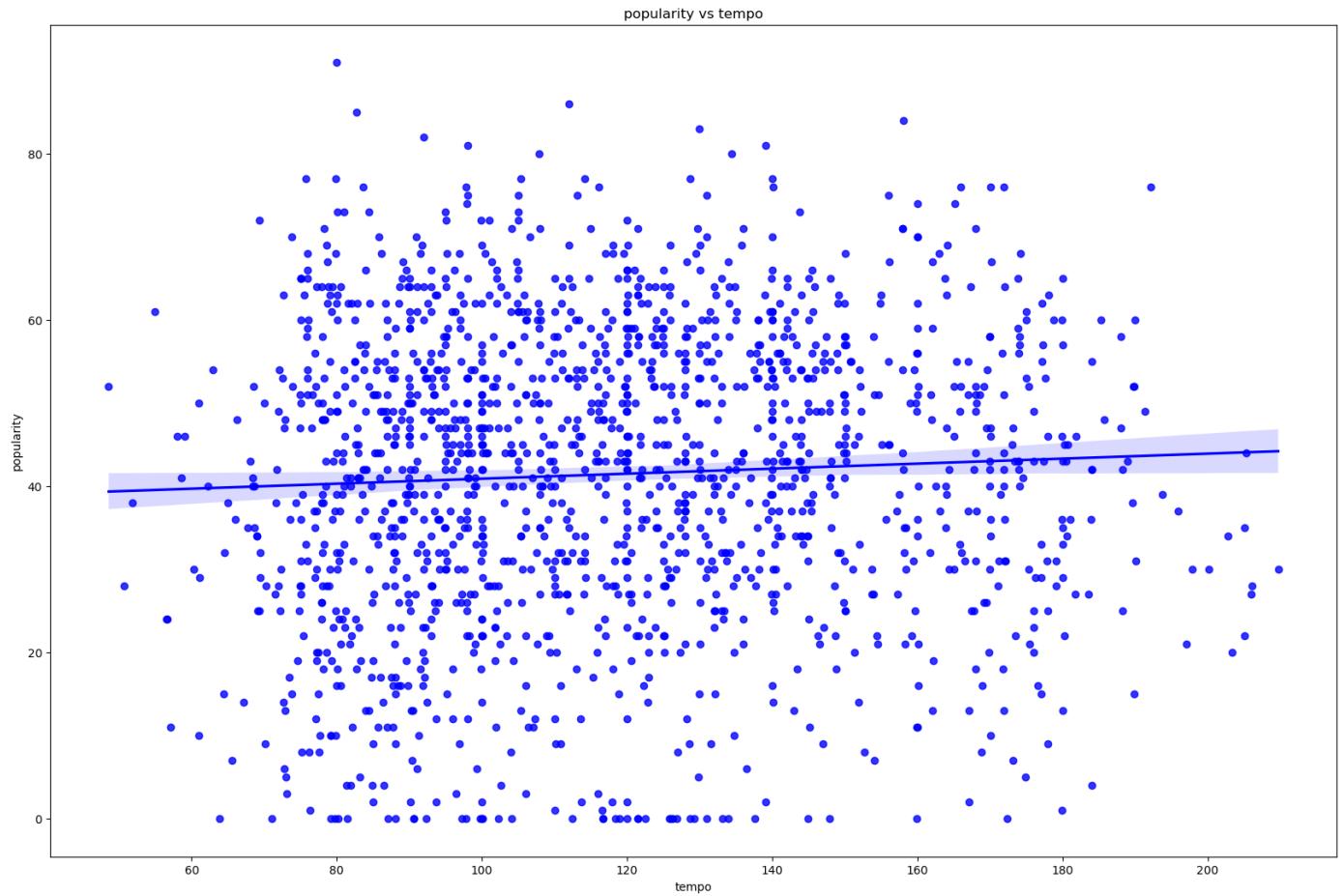
1629

```
In [21]: plt.figure(figsize=(20,13))
sns.regplot(data=sample_df,y="loudness",x="energy",color="c").set(title="loudness vs en
```

```
Out[21]: [Text(0.5, 1.0, 'loudness vs energy')]
```



```
In [22]: plt.figure(figsize=(20,13))
sns.regplot(data=sample_df, y="popularity", x="tempo", color="b").set(title="popularity vs tempo")
Out[22]: [Text(0.5, 1.0, 'popularity vs tempo')]
```



```
#music["date"] = music["tempo"].index.get_level_values("energy")
```

```
In [23]: #music.dates=pd.to_datetime(music.dates)
#years=music.dates.dt.year
```

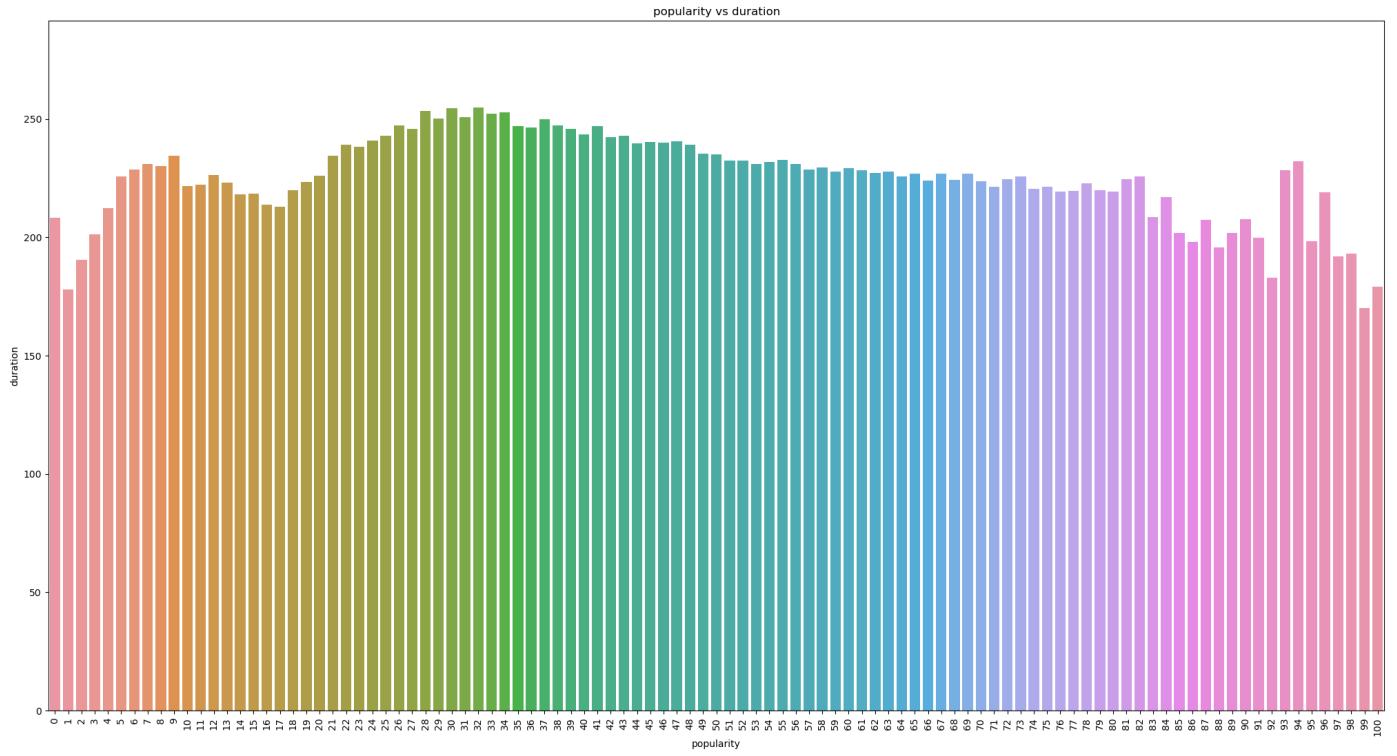
```
In [ ]:
```

```
In [24]: music1=music["duration"]
fig_dim=(25,13)
fig,ax=plt.subplots(figsize=fig_dim)
fig=sns.barplot(y=music1,x=music["popularity"],ax=ax,errwidth=False).set(title="popularity")
plt.xticks(rotation=90)
```

```
Out[24]: (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, '0'),
 Text(1, 0, '1'),
 Text(2, 0, '2'),
 Text(3, 0, '3'),
 Text(4, 0, '4'),
 Text(5, 0, '5'),
 Text(6, 0, '6'),
 Text(7, 0, '7'),
 Text(8, 0, '8'),
 Text(9, 0, '9'),
 Text(10, 0, '10'),
 Text(11, 0, '11'),
 Text(12, 0, '12'),
 Text(13, 0, '13'),
 Text(14, 0, '14'),
 Text(15, 0, '15'),
 Text(16, 0, '16'),
 Text(17, 0, '17'),
 Text(18, 0, '18'),
 Text(19, 0, '19'),
 Text(20, 0, '20'),
 Text(21, 0, '21'),
 Text(22, 0, '22'),
 Text(23, 0, '23'),
 Text(24, 0, '24'),
 Text(25, 0, '25'),
 Text(26, 0, '26'),
 Text(27, 0, '27'),
 Text(28, 0, '28'),
 Text(29, 0, '29'),
 Text(30, 0, '30'),
 Text(31, 0, '31'),
 Text(32, 0, '32'),
 Text(33, 0, '33'),
 Text(34, 0, '34'),
 Text(35, 0, '35'),
 Text(36, 0, '36'),
 Text(37, 0, '37'),
 Text(38, 0, '38'),
 Text(39, 0, '39'),
 Text(40, 0, '40'),
 Text(41, 0, '41'),
 Text(42, 0, '42'),
 Text(43, 0, '43'),
 Text(44, 0, '44'),
 Text(45, 0, '45'),
```

```
Text(46, 0, '46'),
Text(47, 0, '47'),
Text(48, 0, '48'),
Text(49, 0, '49'),
Text(50, 0, '50'),
Text(51, 0, '51'),
Text(52, 0, '52'),
Text(53, 0, '53'),
Text(54, 0, '54'),
Text(55, 0, '55'),
Text(56, 0, '56'),
Text(57, 0, '57'),
Text(58, 0, '58'),
Text(59, 0, '59'),
Text(60, 0, '60'),
Text(61, 0, '61'),
Text(62, 0, '62'),
Text(63, 0, '63'),
Text(64, 0, '64'),
Text(65, 0, '65'),
Text(66, 0, '66'),
Text(67, 0, '67'),
Text(68, 0, '68'),
Text(69, 0, '69'),
Text(70, 0, '70'),
Text(71, 0, '71'),
Text(72, 0, '72'),
Text(73, 0, '73'),
Text(74, 0, '74'),
Text(75, 0, '75'),
Text(76, 0, '76'),
Text(77, 0, '77'),
Text(78, 0, '78'),
Text(79, 0, '79'),
Text(80, 0, '80'),
Text(81, 0, '81'),
Text(82, 0, '82'),
Text(83, 0, '83'),
Text(84, 0, '84'),
Text(85, 0, '85'),
Text(86, 0, '86'),
Text(87, 0, '87'),
Text(88, 0, '88'),
Text(89, 0, '89'),
Text(90, 0, '90'),
Text(91, 0, '91'),
Text(92, 0, '92'),
Text(93, 0, '93'),
Text(94, 0, '94'),
Text(95, 0, '95'),
Text(96, 0, '96'),
Text(97, 0, '97'),
Text(98, 0, '98'),
Text(99, 0, '99'),
Text(100, 0, '100'))
```

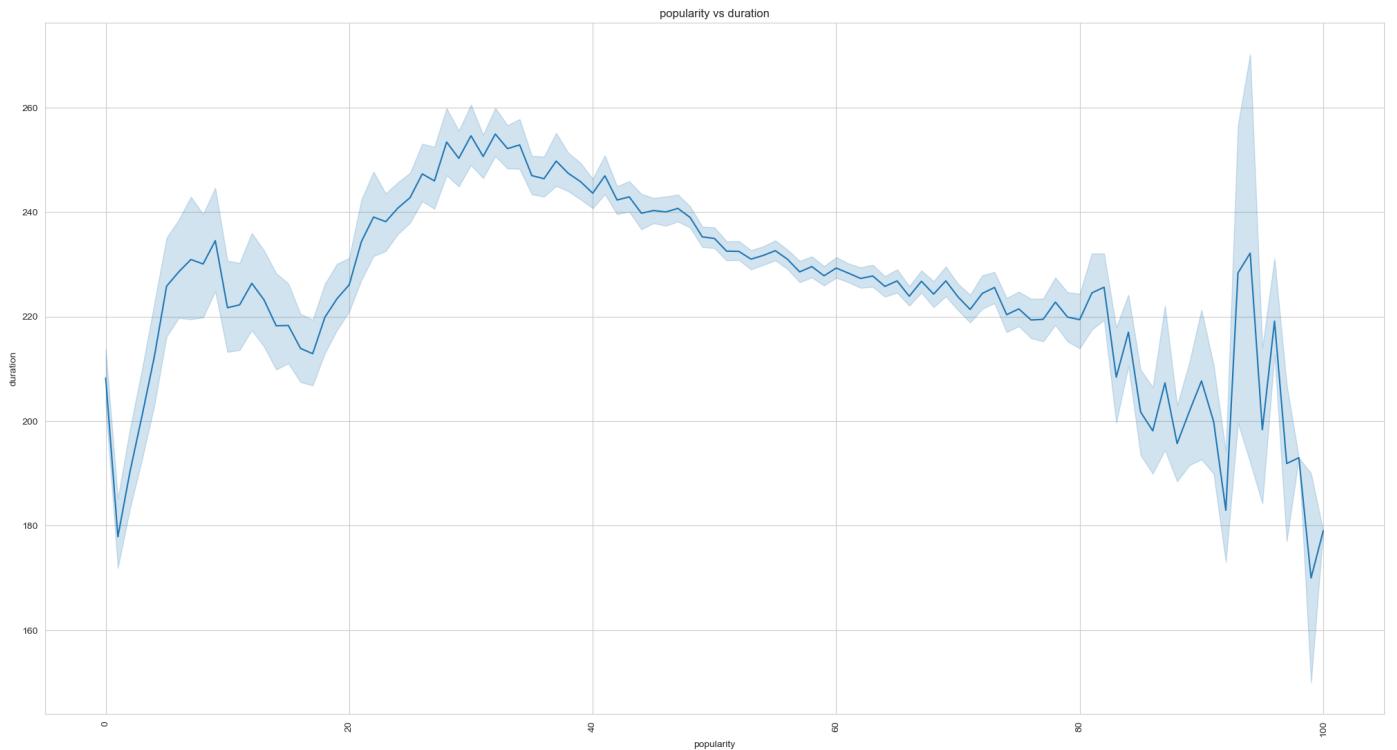


In [25]:

```
music1=music["duration"]
sns.set_style(style="whitegrid")
fig_dim=(25,13)
fig,ax=plt.subplots(figsize=fig_dim)
fig=sns.lineplot(y=music1,x=music["popularity"],ax=ax).set(title="popularity vs duration")
plt.xticks(rotation=90)
```

Out[25]:

```
(array([-20., 0., 20., 40., 60., 80., 100., 120.]),
 [Text(0, 0, ''),
 Text(0, 0, '')])
```



```
In [26]: music.head(10)
```

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	energy	inst
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.61100	0.389	0.9100	
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0bjC1NfoEOUsryehmNudP	1	0.24600	0.590	0.7370	
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.95200	0.663	0.1310	
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tlvf	0	0.70300	0.240	0.3260	
4	Movie	Fabien Nataf	Ouverture	0luslXpMROHdEPvSI1fTQK	4	0.95000	0.331	0.2250	
5	Movie	Henri Salvador	Le petit souper aux chandelles	0Mf1jKa8eNAf1a4PwTbizj	0	0.74900	0.578	0.0948	
6	Movie	Martin & les fées	Premières recherches (par Paul Ventimila, Lori...)	0NUiKYRd6jt1LKMVGkUdnZ	2	0.34400	0.703	0.2700	
7	Movie	Laura Mayne	Let Me Let Go	0PbIF9YVD505GutwotpB5C	15	0.93900	0.416	0.2690	
8	Movie	Chorus	Helka	0ST6uPfvaPpJLqwhE6KfC	0	0.00104	0.734	0.4810	
9	Movie	Le Club des Juniors	Les bisous des bisounours	0VSqZ3KStsjcfERGdcWpFO	10	0.31900	0.598	0.7050	

```
In [27]: #x=music.artist_name  
#y=music.popularity
```

```
In [ ]:
```

```
In [28]: artist=music.sort_values("popularity", ascending=False).head(1000)
```

```
In [29]: artist
```

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	energy
9027	Dance	Ariana Grande	7 rings	14msK75pk3pA33pzPVNtBF	100	0.5780	0.725	0.321
107804	Pop	Ariana Grande	7 rings	14msK75pk3pA33pzPVNtBF	100	0.5780	0.725	0.321
86951	Rap	Post Malone	Wow.	6MWtB6iiXylwun0YzU6DFP	99	0.1630	0.833	0.539
107803	Pop	Post Malone	Wow.	6MWtB6iiXylwun0YzU6DFP	99	0.1630	0.833	0.539

107802	Pop	Ariana Grande	break up with your girlfriend, i'm bored	4kV4N9D1iKVxx1KLvtTpjS	99	0.0421	0.726	0.554
...
191952	Soul	Alicia Keys	No One	6lwKcFdiRQZOWeYNhUiWlv	80	0.0209	0.644	0.548
13838	Dance	James TW	When You Love Someone	1o6YK7phnXgsljfWSQQfWc	80	0.2480	0.675	0.444
191947	Soul	Alicia Keys	If I Ain't Got You	3XVBdLihbNbxBwZosxcGuJ	80	0.6030	0.609	0.445
108122	Pop	NAV	Champion (feat. Travis Scott)	6nO3tr47nr2P7f3hXb8Jlo	80	0.3810	0.642	0.782
108123	Pop	Beyoncé	Halo	4JehYebil9JE8sR8MisGVb	80	0.2720	0.508	0.720

1000 rows × 18 columns

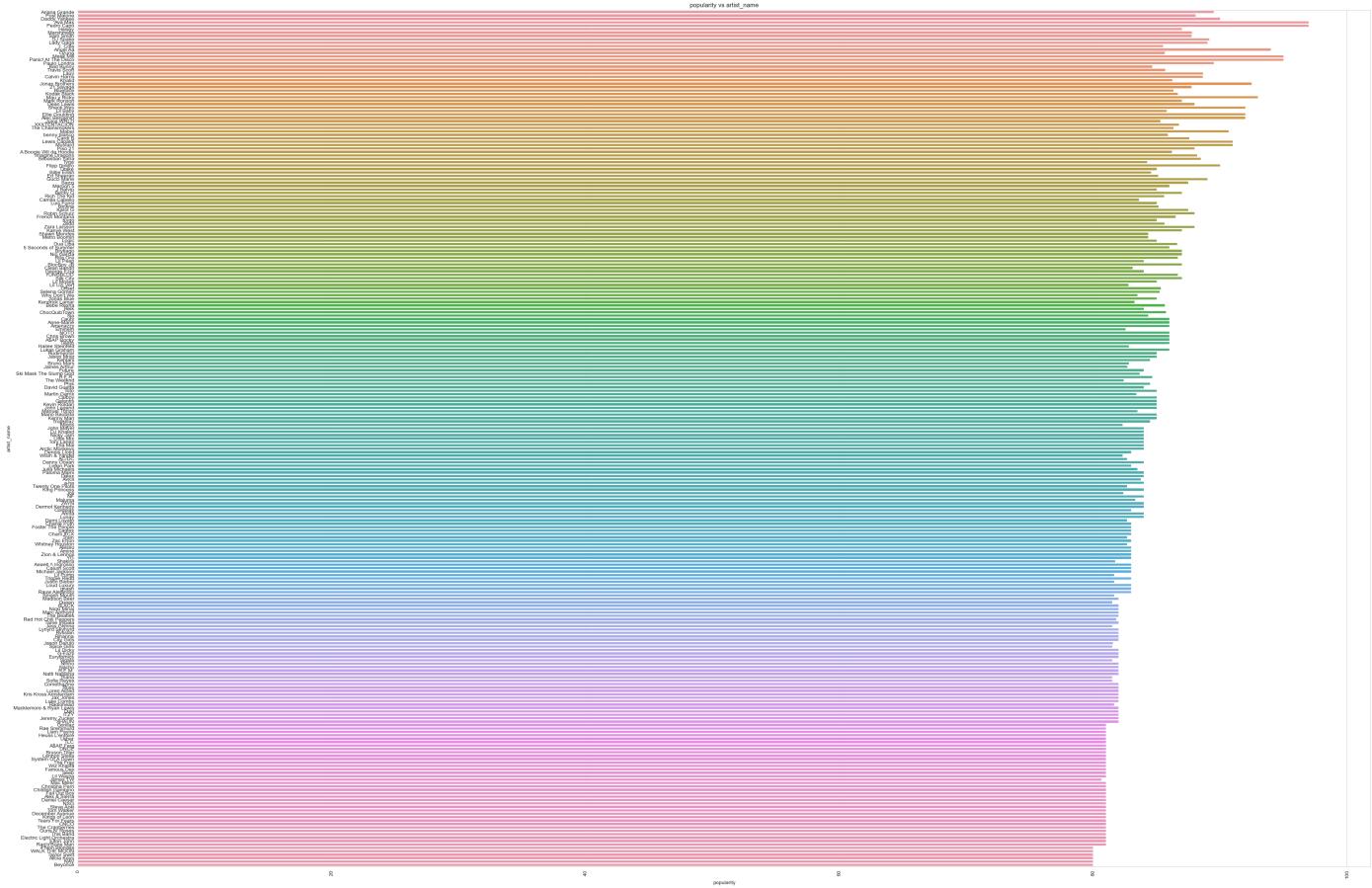
```
In [31]: x=artist["popularity"]
y=artist["artist_name"]
```

```
In [36]: fig_dim=(45,30)
fig,ax=plt.subplots(figsize=fig_dim)
sns.barplot(x,y,ax=ax,errwidth=False).set(title="popularity vs artist_name")
plt.xticks(rotation=90)
```

C:\Users\Lenovo\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

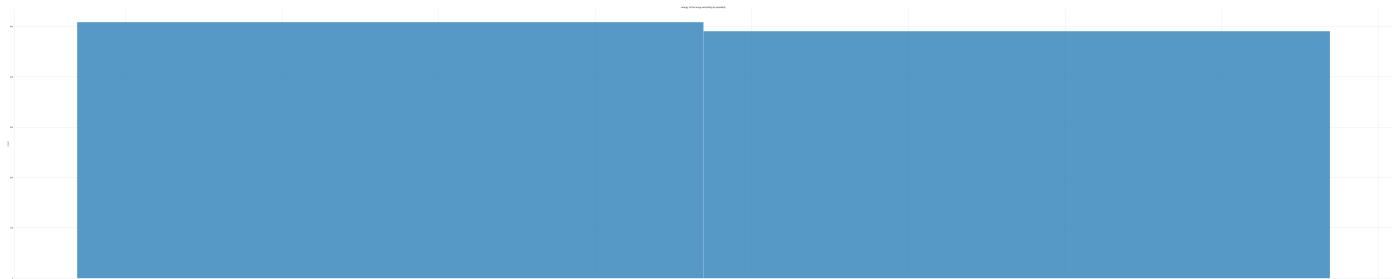
```
warnings.warn(
(array([
  0., 20., 40., 60., 80., 100., 120.]),
 [Text(0, 0, ''),
  Text(0, 0, '')]))
```

Out[36]:

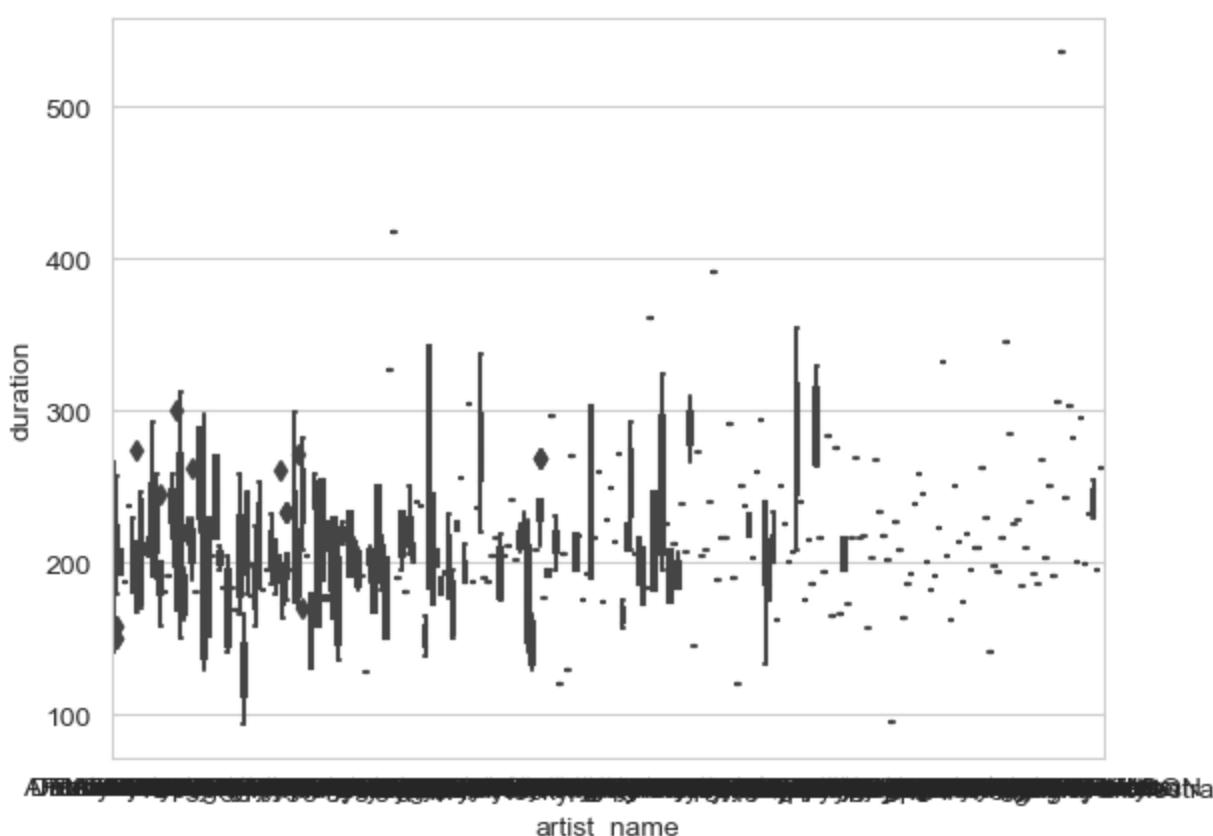


```
In [39]: sns.displot(artist["energy"], discrete=True, aspect=5, height=20, kind="hist").set(title="en
```

```
Out[39]: <seaborn.axisgrid.FacetGrid at 0x1fe18972fd0>
```



```
In [46]: sns.boxplot(x=artist["artist_name"], y=artist["duration"]);
```



```
In [48]: popular=music.sort_values("duration", ascending=False).head(500)
popular
```

Out[48]:

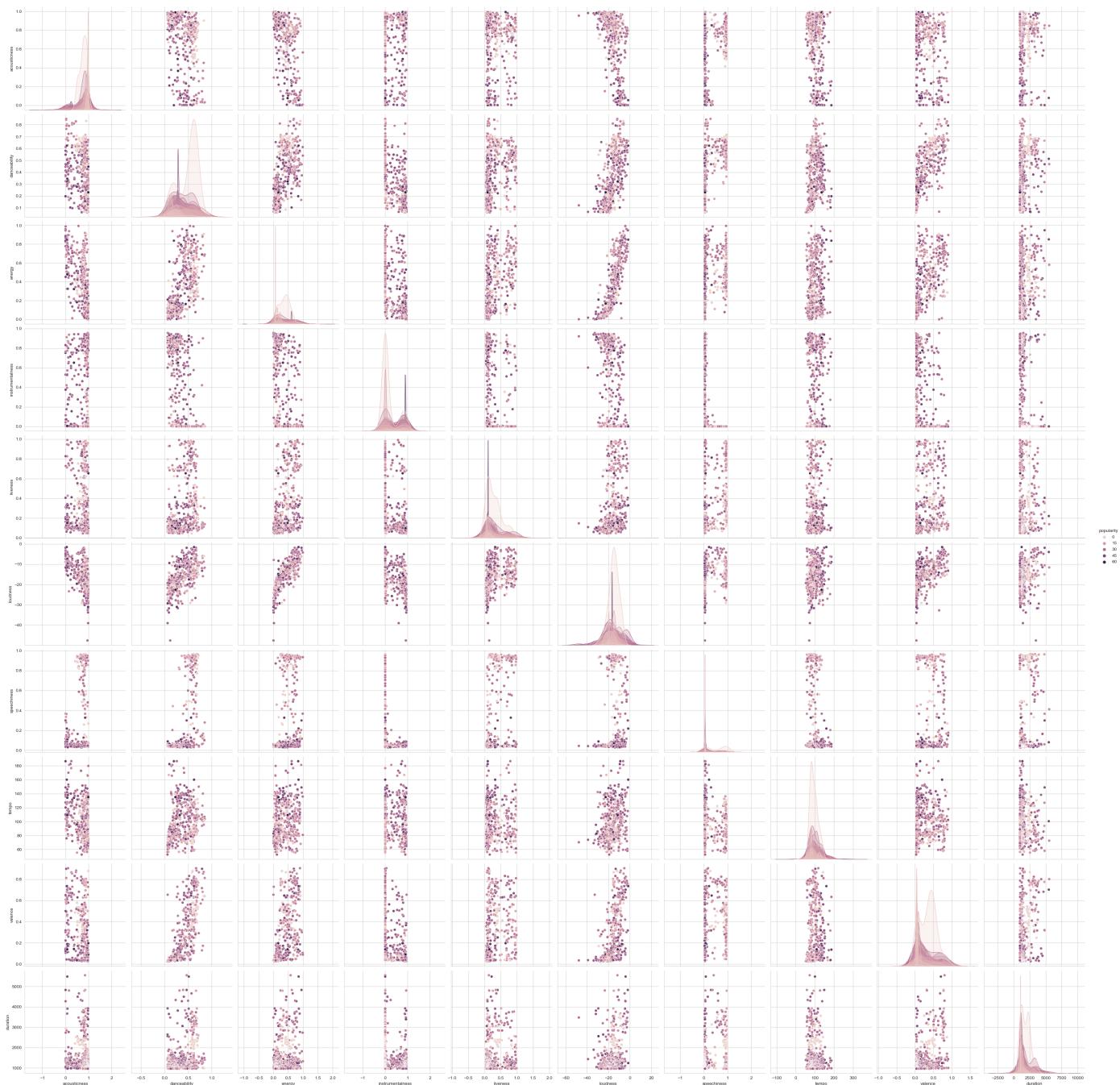
	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	error
212083	Comedy	La Mesa Reñoña	Episodio 15 (Lady Orinoco, Políticas De Youtub...)	76vVk4HCOLP5r7hA53SyXg	22	0.84600	0.4680	0
162671	Reggaeton	DJ Luigi	6 : 00 Am	5p44BE1eYof6B2HSj3LTfQ	44	0.07840	0.5130	0
23593	Electronic	Excision	Lost Lands 2017 Mix	40IBktlvVS4tBpGHm8W44C	41	0.00164	0.4670	0
211969	Comedy	La Mesa Reñoña	Episodio 14 (Machismo, Juanga, Bebé a Bordo)	3Y800wAK7yLS2vckDD4WCz	21	0.85000	0.3920	0
219057	World	Jamie Llewellyn	Nature Sounds for Sleep: Crackling Log Fire wi...	40duG6erWyOVizxE9P8KqV	22	0.15000	0.3160	0
...
104747	Classical	Ludwig van Beethoven	Missa solemnis, Op. 123: Agnus Dei. Adagio (Live)	1DAf0sZlfuBidm1ecvvGwp	14	0.97800	0.0977	0
211553	Comedy	Danny Bhoy	Gecko Story	4qRKe2PGplUJmvPhQ6ocbD	17	0.75200	0.5830	0

168459	Comedy	Ron White	Dr. Phil Story	2tKL6AuloHLRoEABV7jMdp	31	0.93800	0.5830	0
172714	Comedy	David Cross	Diarrhea Moustache	0Sq4PzGaoboM6ozteoAcyM	19	0.87500	0.6010	0
125953	Classical	Claude Debussy	Jeux, poème dansé, L.126	5nmywZVxuRHekpXFHkgxi	36	0.96800	0.1280	0

500 rows × 18 columns

In [53]: `sns.pairplot(popular,hue="popularity",height=3.5)`

Out[53]: `<seaborn.axisgrid.PairGrid at 0x1fe3e474c70>`



In [56]: `#3d animation`

```
import numpy as np
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
plt.style.use('seaborn-poster')
```

```
In [57]: fig = plt.figure(figsize = (12,10))
```

```
ax = plt.axes(projection='3d')

x = np.arange(-5, 5.1, 0.2)
y = np.arange(-5, 5.1, 0.2)
```

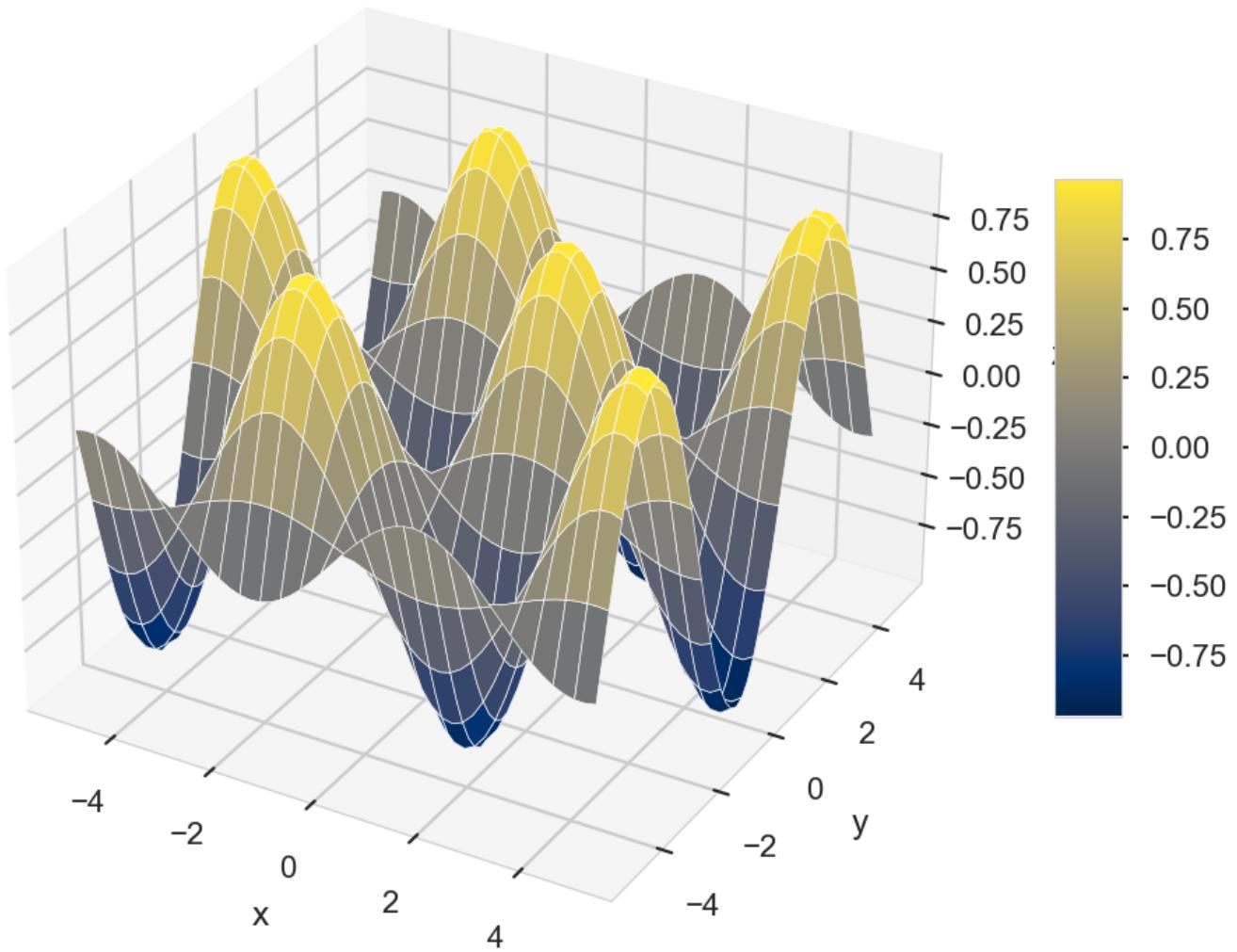
```
X, Y = np.meshgrid(x, y)
Z = np.sin(X)*np.cos(Y)
```

```
surf = ax.plot_surface(X, Y, Z, cmap = plt.cm.cividis)
```

```
# Set axes label
ax.set_xlabel('x', labelpad=20)
ax.set_ylabel('y', labelpad=20)
ax.set_zlabel('z', labelpad=20)
```

```
fig.colorbar(surf, shrink=0.5, aspect=8)
```

```
plt.show()
```



```
In [59]: from numpy import linspace
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from mpl_toolkits import mplot3d
```

```
from scipy import signal
```

```
# Creating 3D figure
```

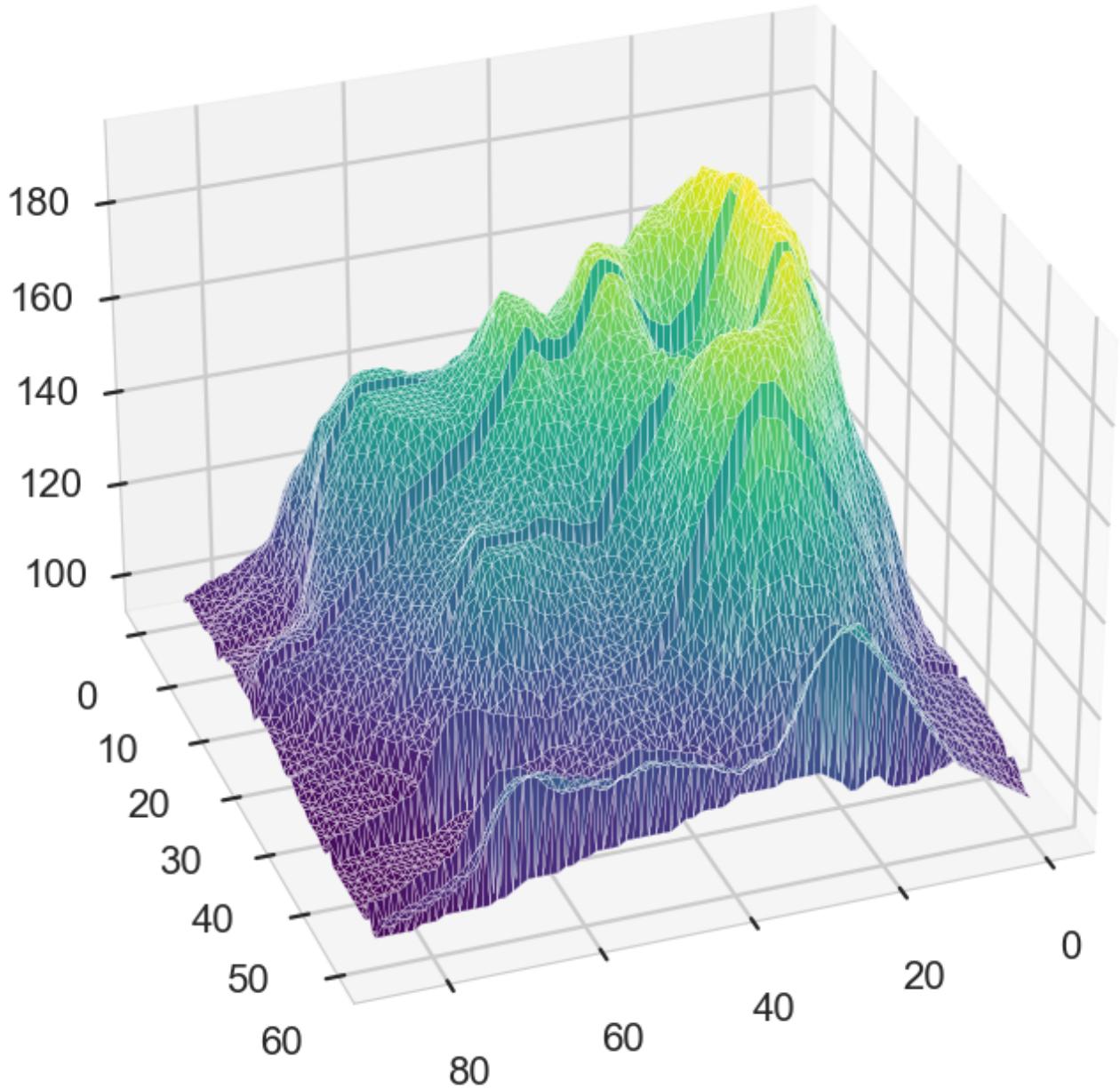
```
fig = plt.figure(figsize = (8, 8))
```

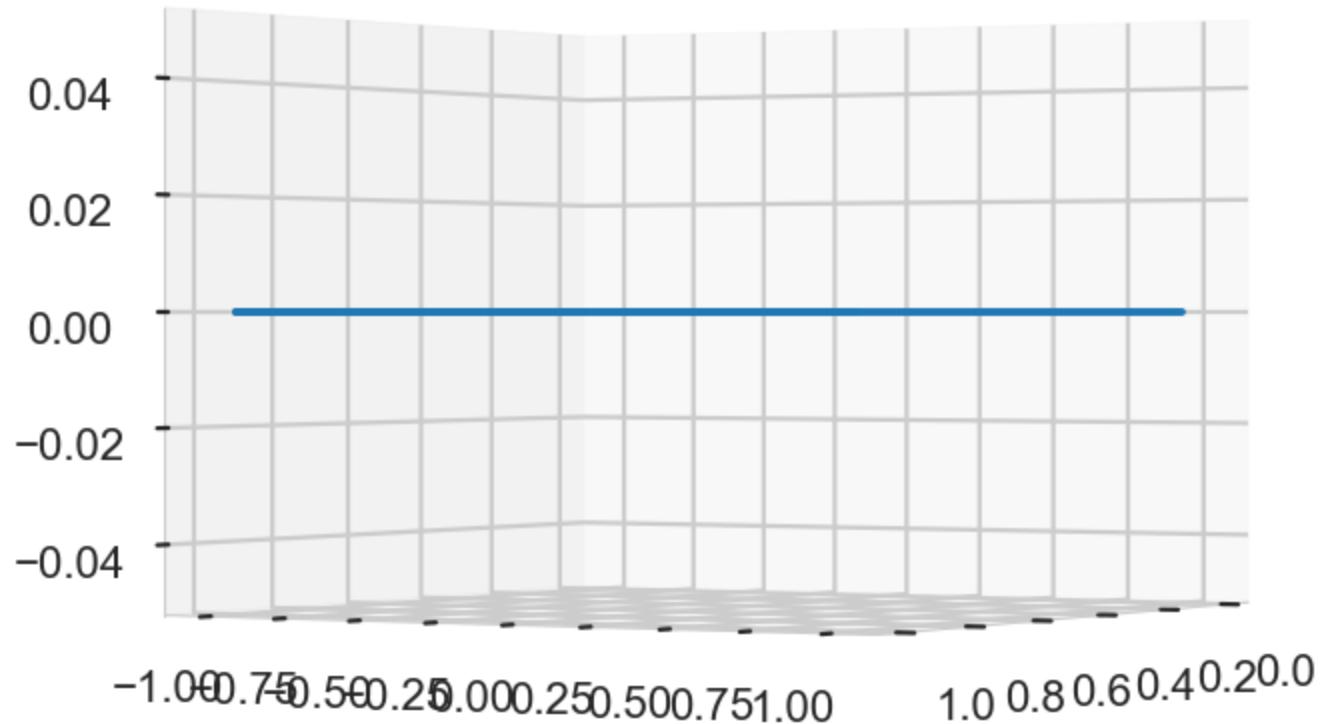
```
ax = plt.axes(projection = '3d')

# Creating Dataset
t = np.linspace(0, 1, 1000, endpoint = True)
ax.plot3D(t, signal.square(2 * np.pi * 5 * t))

# 360 Degree view
for angle in range(0, 360):
    ax.view_init(angle, 30)
    plt.draw()
    plt.pause(.001)

plt.show()
```






```
<Figure size 1280x880 with 0 Axes>
```

In []: