



canplay

Music Tech Academy

音楽データサイエンティスト 第1回

Staff Machine Learning Engineer – Free Mission

[Data & Analytics](#) / [Engineering & IT](#) / [Machine Learning](#) / [Software Engineering](#) in [New York, USA](#)

At Spotify, we're proud of our ambitious mission of having 1 billion fans enjoying music around the world, and are seeking a lead machine learning engineer to join us in pursuit of this goal. Our team leverages machine learning to drive both advertiser outcomes and listener satisfaction in Spotify Free, helping democratize music access globally while positively impacting Spotify's bottom line. We develop innovative solutions around the core problem of getting the right ad to the right user at the right time, while embodying Spotify's values and taking our users' best interest at heart.

[SHARE](#)[SAVE](#)[APPLY NOW](#)

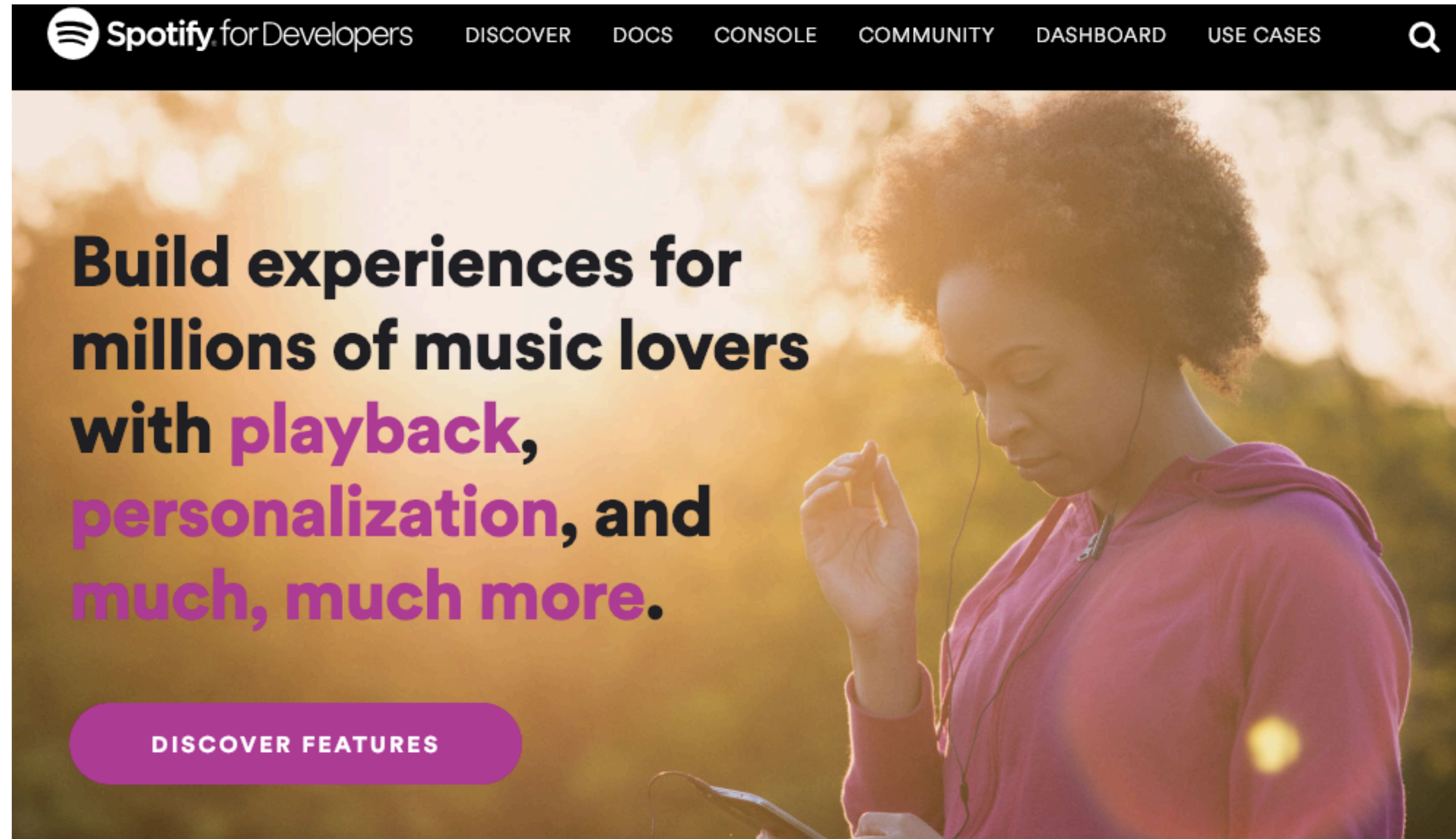
the right music to the right user at the right time

Spotifyの2017年 年間配信データを分析します

Spotifyの2017年配信データ

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	id	name	artists	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalne	liveness	valence	tempo	duration_ms	time_signature
2	7qiZfU4dY1IWl	Shape of You	Ed Sheeran	0.825	0.652	1	-3.183	0	0.0802	0.581	0	0.0931	0.931	95.977	233713	4
3	5CtI0qwDJkDC	Despacito - Re	Luis Fonsi	0.694	0.815	2	-4.328	1	0.12	0.229	0	0.0924	0.813	88.931	228827	4
4	4aWmUDTfIPG	Despacito (Fea	Luis Fonsi	0.66	0.786	2	-4.757	1	0.17	0.209	0	0.112	0.846	177.833	228200	4
5	6RUKPb4LETW	Something Jus	The Chainsmok	0.617	0.635	11	-6.769	0	0.0317	0.0498	1.44E-05	0.164	0.446	103.019	247160	4
6	3DXncPQOG4\	I'm the One	DJ Khaled	0.609	0.668	7	-4.284	1	0.0367	0.0552	0	0.167	0.811	80.924	288600	4
7	7KXjTSCq5nL1	HUMBLE.	Kendrick Lama	0.904	0.611	1	-6.842	0	0.0888	0.000259	2.03E-05	0.0976	0.4	150.02	177000	4
8	3eR23VRfZcc	It Ain't Me (wit	Kygo	0.64	0.533	0	-6.596	1	0.0706	0.119	0	0.0864	0.515	99.968	220781	4
9	3B54sVLJ402z	Unforgettable	French Montan	0.726	0.769	6	-5.043	1	0.123	0.0293	0.0101	0.104	0.733	97.985	233902	4
10	0KKkJNfGyhkC	That's What I L	Bruno Mars	0.853	0.56	1	-4.961	1	0.0406	0.013	0	0.0944	0.86	134.066	206693	4
11	3NdDpSvN911\	I Don't	王冲 Wan ZAYN	0.735	0.451	0	-8.374	1	0.0585	0.0631	1.30E-05	0.325	0.0862	117.973	245200	4
12	7GX5fIRQZVHf	XO TOUR Liif3	Lil Uzi Vert	0.732	0.75	11	-6.366	0	0.231	0.00264	0	0.109	0.401	155.096	182707	4
13	72jbDTw1piOO	Paris	The Chainsmok	0.653	0.658	2	-6.428	1	0.0304	0.0215	1.66E-06	0.0939	0.219	99.99	221507	4
14	0dA2Mk56wEzI	Stay (with Ales	Zedd	0.679	0.634	5	-5.024	0	0.0654	0.232	0	0.115	0.498	102.013	210091	4
15	4iLqG9SeJSntf	Attention	Charlie Puth	0.774	0.626	3	-4.432	0	0.0432	0.0969	3.12E-05	0.0848	0.777	100.041	211475	4
16	0VgkVdmE4gld	Mask Off	Future	0.833	0.434	2	-8.795	1	0.431	0.0102	0.0219	0.165	0.281	150.062	204600	4
17	3a1lNhkSLSkp	Congratulations	Post Malone	0.627	0.812	6	-4.215	1	0.0358	0.198	0	0.212	0.504	123.071	220293	4
18	6kex4EBAj0WF	Swalla (feat. N	Jason Derulo	0.696	0.817	1	-3.862	1	0.109	0.075	0	0.187	0.782	98.064	216409	4
19	6PCUP3dWmTj	Castle on the H	Ed Sheeran	0.461	0.834	2	-4.868	1	0.0989	0.0232	1.14E-05	0.14	0.471	135.007	261154	4
20	5knuzwU65gJK	Rockabye (feat	Clean Bandit	0.72	0.763	9	-4.068	0	0.0523	0.406	0	0.18	0.742	101.965	251088	4
21	0CcQNd8CINk	Believer	Imagine Dragor	0.779	0.787	10	-4.305	0	0.108	0.0524	0	0.14	0.708	124.982	204347	4
22	2rb5MvYT7Zixl	Mi Gente	J Balvin	0.543	0.677	11	-4.915	0	0.0993	0.0148	6.21E-06	0.13	0.294	103.809	189440	4
23	0tKcYR2II1VC	Thunder	Imagine Dragor	0.6	0.81	0	-4.749	1	0.0479	0.00683	0.21	0.155	0.298	167.88	187147	4
24	5uCax9HTNlzC	Say You Won't	James Arthur	0.358	0.557	10	-7.398	1	0.059	0.695	0	0.0902	0.494	85.043	211467	4
25	79cuOz3SPQT	There's Nothin	Shawn Mendes	0.857	0.8	2	-4.035	1	0.0583	0.381	0	0.0913	0.966	121.996	199440	4
26	6De0IHrwBfPfr	Me Reh こそ	Danny Ocean	0.744	0.804	1	-6.327	1	0.0677	0.0231	0	0.0494	0.426	104.823	205715	4
27	6D0b04NJKfEl	Issues	Julia Michaels	0.706	0.427	8	-6.864	1	0.0879	0.413	0	0.0609	0.42	113.804	176320	4
28	0afhq8XCEXxp	Galway Girl	Ed Sheeran	0.624	0.876	9	-3.374	1	0.1	0.0735	0	0.327	0.781	99.943	170827	4
29	3ebXMykcMXC	Scared to Be L	Martin Garrix	0.584	0.54	1	-7.786	0	0.0576	0.0895	0	0.261	0.195	137.972	220883	4
30	7BKLCZ1jbUB\	Closer	The Chainsmok	0.748	0.524	8	-5.599	1	0.0338	0.414	0	0.111	0.661	95.01	244960	4
31	1x5sYLZiu9r5E	Symphony (fea	Clean Bandit	0.707	0.629	0	-4.581	0	0.0563	0.259	1.60E-05	0.138	0.457	122.863	212459	4
32	5GXAXm5YOm	I Feel It Comin	The Weeknd	0.768	0.813	0	-5.94	0	0.128	0.427	0	0.102	0.579	92.994	269187	4
33	5aAx2yezTd8z	Starboy	The Weeknd	0.681	0.594	7	-7.028	1	0.282	0.165	3.49E-06	0.134	0.535	186.054	230453	4
34	10Ah8uOEovT	Wild Thoughts	DJ Khaled	0.671	0.672	0	-3.094	0	0.0688	0.0329	0	0.118	0.632	97.98	204173	4
35	7tr2za8SQg2C	Slide	Calvin Harris	0.736	0.795	1	-3.299	0	0.0545	0.498	1.21E-06	0.254	0.511	104.066	230813	4

Spotifyのデータ収集方法



Spotify for developerに登録してWeb API経由で取得
<https://developer.spotify.com/>

データ分析に必要なモジュールのインポート

必要なモジュールのインポート

```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.ticker as ticker
```

```
from sklearn import datasets, linear_model
from sklearn.decomposition import PCA
from sklearn.preprocessing import MinMaxScaler
from sklearn.manifold import TSNE
```

```
%matplotlib inline
```

```
import os
```


Pythonの基本的数値取り扱い

pythonの基本的数値取り扱い

足し算

```
: 1 + 1
```

```
: 2
```

掛け算

```
: 2 * -3
```

```
: -6
```

除算

```
: 4 / 3
```

```
: 1.3333333333333333
```

除算（切り捨て）

```
: 4 // 3
```

```
: 1
```

剰余

```
: 4 % 3
```

```
: 1
```

pythonの基本的数値取り扱い

べき乗

```
3 ** 2
```

9

比較演算子

```
2 < 3
```

True

変数

```
a = 2
```

```
a
```

2

```
a = a + 3
```

```
a
```

5

```
a /= 2
```

```
a
```

2.5

pythonの基本的文字列取り扱い

pythonの基本的文字列取り扱い

文字列はシングルクォーテーションで囲みます

```
'Hello World!'
```

```
'Hello World!'
```

ダブルクォーテーションでも同じです

```
"Hello World!"
```

```
'Hello World!'
```

日本語も同様に扱うことができます

```
'こんにちは世界!'
```

```
'こんにちは世界!'
```

print文でも表示できます

```
print('Hello', 'World')
```

```
Hello World
```


pythonの基本的文字列取り扱い（文字列の演算）

```
a = 'Hello'  
b = 'World'
```

文字列は+で結合できます

```
a + b
```

'HelloWorld'

スライスで文字列の一部を抽出できます
2文字目まで抽出してみます

```
a[:2]
```

'He'

3文字目から4文字目まで

```
a[2:4]
```

'll'

pythonの基本的配列の取り扱い

pythonの基本的リストの取り扱い

```
a = [1, 2, 3]  
b = [3, 4, 5, 6]
```

リストを結合

```
a + b
```

```
[1, 2, 3, 3, 4, 5, 6]
```

リストへ追加

```
a.append(3)  
a
```

```
[1, 2, 3, 3]
```

スライス

```
b[:3]
```

```
[3, 4, 5]
```

```
b[1:3]
```

```
[4, 5]
```

pythonの基本的辞書の取り扱い

```
d = {'python': 2, '統計': 3}  
d
```

```
{'python': 2, '統計': 3}
```

```
d['python']
```

```
2
```

```
d.keys()
```

```
dict_keys(['python', '統計'])
```

```
d.values()
```

```
dict_values([2, 3])
```


numpy

numpy

**numpyとはpythonで数値計算を行う時に使用するライブラリー
大容量の数値計算や多次元計算などを効率的に行う事ができる**

numpyの基本操作

numpyのインストール

まだインストールの方はpipでインストールしてください

```
!pip install numpy
```

numpyのインポート

numpyはnpという名前インポートされることがほとんどです

```
[ ] import numpy as np
```

これでnumpyを以降npで呼び出す事ができます

numpyの基本操作

arrayメソッドで配列の作成

numpyではarrayというメソッドで配列（ベクトル）を作成し計算を行う事ができます。

ここでは[1, 2, 3, 4, 5]という要素をもった配列をnumpyのarrayメソッドを使用し作成してみましょう

```
[ ] arr = np.array([1, 2, 3, 4, 5])
```

numpyの基本操作

shapeメソッドで大きさ（要素数）を表示

大きさ（要素数）はshapeメソッドで表示できます。

```
[ ] arr = np.array([1, 2, 3, 4, 5])
```

```
[ ] arr.shape
```

```
(5,)
```

numpyの基本操作

配列に対して足し算をする事ができます

```
[ ] arr = np.array([1, 2, 3, 4, 5])
```

```
[ ] arr + 2
```

```
 array([3, 4, 5, 6, 7])
```


すべての要素に+2されました

numpyの基本操作

足し算だけでなく、他の演算子でも同じようなことができます
ルート（平方根）を表示してみましょう

```
[ ] arr = np.array([1, 2, 3, 4, 5])
```

```
[ ] np.sqrt(arr)
```

```
 array([1.      , 1.41421356, 1.73205081, 2.      , 2.23606798])
```

numpyの基本操作

リストと同様にスライスで要素を抽出できます

```
[ ] arr = np.array([1, 2, 3, 4, 5])
```

```
[ ] arr[:2]
```

```
array([1, 2])
```

```
[ ] arr[3:5]
```

```
array([4, 5])
```

numpyの基本操作

arangeメソッド

arangeメソッドを使って[0, 1, 2, 3, 4]という要素をもったarrayを簡単に作ることができます

```
[ ] np.arange(5)
```

```
array([0, 1, 2, 3, 4])
```

arangeメソッドにはstart, end, stepを指定できます

[2, 4, 6, 8]という2ずつ数字が増えていくarrayは次のように作ることができます

```
[ ] np.arange(2, 10, 2)
```

```
array([2, 4, 6, 8])
```

numpyの基本操作

linspaceメソッド

linspaceメソッドもstartからendまでのarrayを作る関数ですが、stepではなく要素の数を指定します

```
[ ] np.linspace(0, 3, 11)
```

```
array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3. ])
```

第3引数（分割数）はnum=でも指定できる。初期値は50

その他の引数

endpoint= 終点を含むかどうか 初期値はTrue

dtype= データタイプ 初期値は"float64"

numpyの基本操作

二次元のarrayも簡単に作れます

```
[ ] arr2 = np.array([[1, 3],  
                    [5, 7]])  
arr2
```

```
array([[1, 3],  
       [5, 7]])
```

```
[ ] arr2.shape
```

```
(2, 2)
```

numpyの基本操作

saveメソッド

saveメソッドでarrayを保存できます

第一引数が保存する名前で、第二引数が保存するarrayです

```
[ ] np.save('test.npy', arr2)
```

loadメソッド

読み込みはloadメソッドでできます

```
[ ] np.load('test.npy')
```

```
array([[1, 3],  
       [5, 7]])
```

pandas

pandas

PandasとはPythonでデータ分析を効率的に行うためのライブラリです。
Pandasを使うと、データの読み込みや統計量の表示、グラフ化など、データ分析に関する作業をより容易に行うことができるようになります。
また主要なコードはCythonまたはC言語で書かれており、Pythonだけでデータ分析を行うのと比較して、非常に高速に処理を行うことができます。

pandasの基本操作

pandasのインストール

まだインストールの方はpipでインストールしてください

```
!pip install pandas
```

pandasのインポート

pandasはpdという名前インポートされることがほとんどです
これでpandasを以降pdで呼び出す事ができます

```
import pandas as pd
```

pandasの基本操作

read_csvメソッド

read_csvメソッドを使うことでcsvをpandasのDataFrameという形式で読み込むことができます

```
df = pd.read_csv('pandas_test.csv')  
#jupyternotebookの場合は添付のcsvファイルがあるディレクトリーを指定  
#google colabの場合はファイルをロード  
df
```

添付のpandas_test.csvを

jupyternotebookの場合は添付のcsvファイルがあるディレクトリーを指定

google colabの場合はファイルをロード

してご使用ください

pandasの基本操作

`read_csv`メソッドで読み込んだ`pandas_test.csv`

	CD番号	曲の長さ	BPM	リリース年	人気ランキング
1	1	3:46	133	2001	4
2	1	3:42	122	2001	10
3	1	2:55	156	2001	7
4	2	3:44	111	2009	5
5	2	4:33	96	2009	2
6	2	2:55	166	2009	9
7	3	3:25	144	2019	6
8	3	3:12	150	2019	1
9	3	4:01	88	2019	8
10	3	4:05	79	2019	3

pandasの基本操作

headメソッド

DataFrameの先頭の数行を抽出できます

() 内に引数として行数指定もできます。

初期値は5のため引数指定なければ最初の5行が表示されます。

```
df.head()
```

	CD番号	曲の長さ	BPM	リリース年	人気ランキング
1	1	3:46	133	2001	4
2	1	3:42	122	2001	10
3	1	2:55	156	2001	7
4	2	3:44	111	2009	5
5	2	4:33	96	2009	2

pandasの基本操作

tailメソッド

DataFrameの末尾の数行を抽出できます

() 内に引数として行数指定もできます

初期値は5のため引数指定なければ最後の5行が表示されます。

```
df.tail()
```

	CD番号	曲の長さ	BPM	リリース年	人気ランキング
6	2	2:55	166	2009	9
7	3	3:25	144	2019	6
8	3	3:12	150	2019	1
9	3	4:01	88	2019	8
10	3	4:05	79	2019	3

pandasの基本操作

スライス（箇所指定）して抜き出しもできます

```
df[2:5]
```

	CD番号	曲の長さ	BPM	リリース年	人気ランキング
3	1	2:55	156	2001	7
4	2	3:44	111	2009	5
5	2	4:33	96	2009	2

pandasの基本操作

列指定（特定の要素指定）して抜き出しもできます

```
df['BPM']  
  
1    133  
2    122  
3    156  
4    111  
5     96  
6    166  
7    144  
8    150  
9     88  
10    79  
Name: BPM, dtype: int64
```

pandasの基本操作

複数の列指定（特定の要素指定）して抜き出しもできます

```
df[['曲の長さ', 'リリース年']]
```

	曲の長さ	リリース年
1	3:46	2001
2	3:42	2001
3	2:55	2001
4	3:44	2009
5	4:33	2009
6	2:55	2009
7	3:25	2019
8	3:12	2019
9	4:01	2019
10	4:05	2019

pandasの基本操作

行と列の両方指定し抜き出す場合はlocメソッドを使用します

```
df.loc[1, 'BPM']
```

```
133
```

スライスによる抜き出し箇所指定もできます

```
df.loc[1:3, 'BPM']
```

```
1    133
```

```
2    122
```

```
3    156
```

```
Name: BPM, dtype: int64
```

pandasの基本操作

条件に合致する行を抽出したい場合は次のように書きます
ここではBPMが100より下の行を抽出しています

df[df['BPM'] < 100]

	CD番号	曲の長さ	BPM	リリース年	人気ランキング
5	2	4:33	96	2009	2
9	3	4:01	88	2019	8
10	3	4:05	79	2019	3

pandasの基本操作

データの大きさはshapeメソッドで表示できます

```
df.shape
```

```
(10, 5)
```

列の抜き出しはcolumnsです

```
df.columns
```

```
Index(['CD番号', '曲の長さ', 'BPM', 'リリース年', '人気ランキング'], dtype='object')
```

matplotlib

matplotlib

Matplotlibは、Pythonでグラフ表示をするための描画ライブラリです。
簡単に高度な線グラフや棒グラフ、3Dグラフなどの表示が可能です。

matplotlibのインストール

まだインストールしていない方はpipでインストールします。

```
!pip install matplotlib
```

matplotlibのインポート

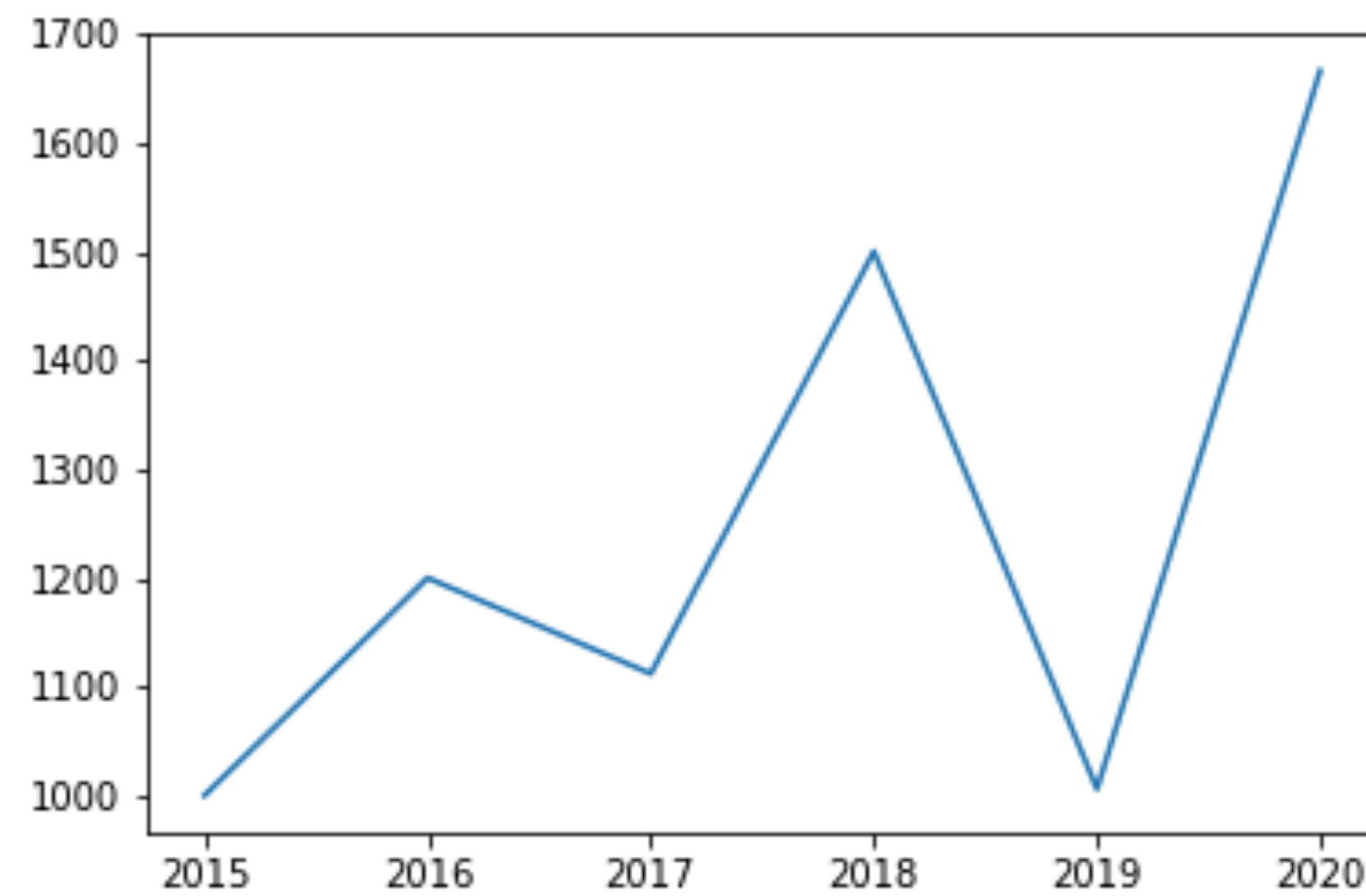
```
%matplotlib inline  
  
import matplotlib.pyplot as plt
```

matplotlib

matplotlibでグラフ表示

簡単なグラフ表示をします

```
year = [2015, 2016, 2017, 2018, 2019, 2020]  
price = [1000, 1200, 1112, 1500, 1006, 1666]  
plt.plot(year, price)  
plt.show()
```



matplotlib

コード	内容
import matplotlib.pyplot as plt	Matplotlibを使う宣言文
year = [2015, 2016, 2017, 2018, 2019, 2020]	横軸の値、今回は年数。Matplotlibは、リストで表します。
price = [1000, 1200, 1112, 1500, 1006, 1666]	縦軸の値、今回は価格。リストで表します。
plt.plot(year, price)	matplotlib.pyplot の plot() メソッドでグラフの描画。year と price を折れ線グラフで描画します。
plt.show()	作成したグラフの表示命令（これがないと表示されません）

pyplot

および他のmatplotlib

モジュールについて

pyplotおよび他のmatplotlib APIについて

pyplotは

matplotlibに用意された描画用のモジュールの一つで、最も活用される基本的なものの一つです。

matplotlibではpyplot以外にモジュールが40種類以上用意されています。

それぞれのモジュールはそれぞれの機能を有し、例えばpyplotでは基本的な機能として、アニメーションに関する `animation` や カラーリングに関する `color` などが用意されています。

各機能はそれぞれ100種類以上も用意されているため公式サイトを参考にしながらグラフ描画を実践してください。

公式サイト

<https://matplotlib.org/index.html>

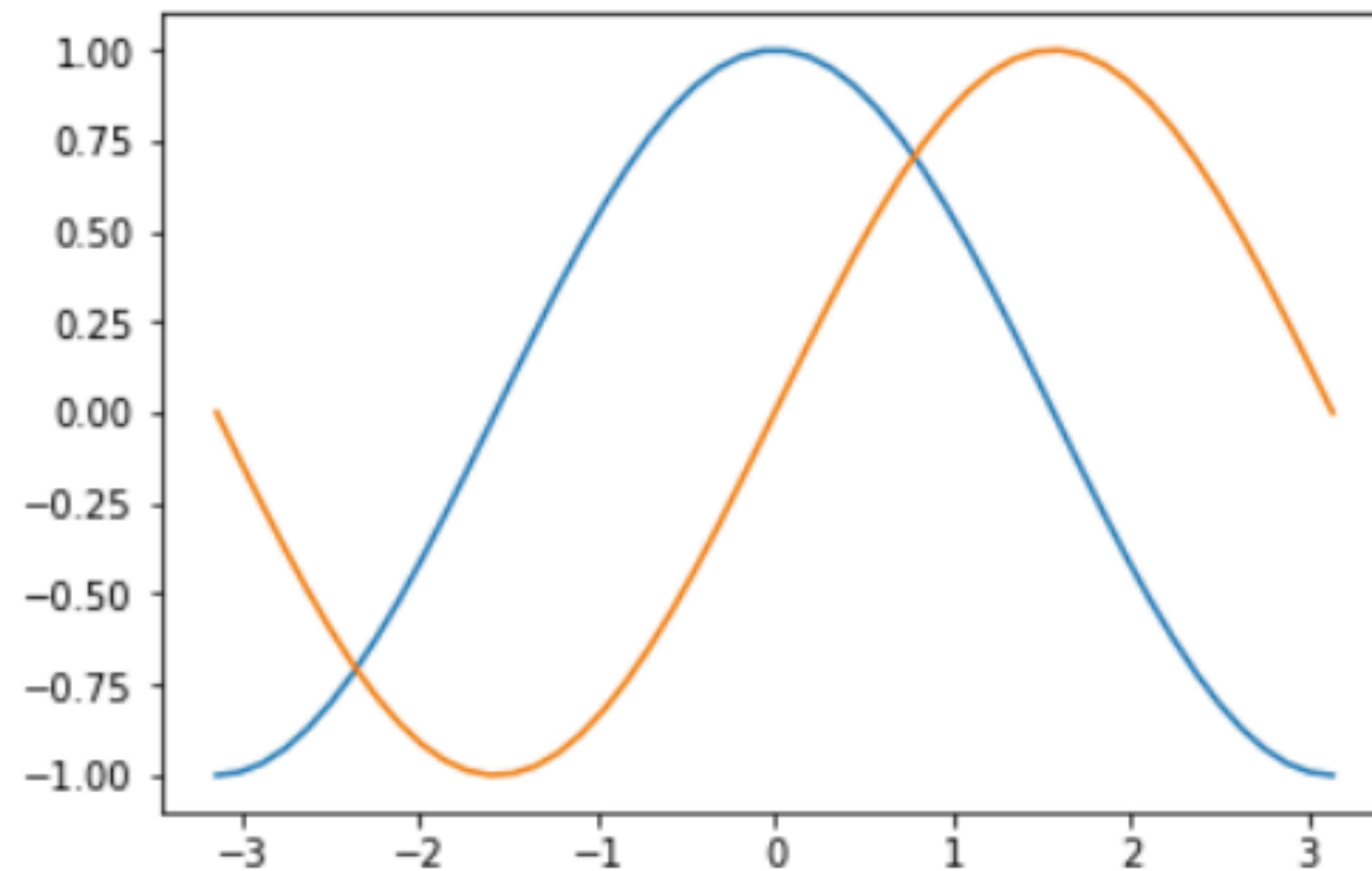
モジュール一覧

<https://matplotlib.org/api/index.html>

**pyplotとnumpyなど
他のモジュールを組み合わせる**

pyplotとnumpyなど他のモジュールを組み合わせる

```
%matplotlib inline  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
x = np.linspace(-np.pi, np.pi)  
plt.plot(x, np.cos(x))  
plt.plot(x, np.sin(x))  
plt.show()
```



他のモジュールを組み合わせて
高度なグラフ表示も可能です

pyplotとnumpyなど他のモジュールを組み合わせる

グラフ表示の実践

```
%matplotlib inline
```

matplotlibをJupyter Notebook
で使用する定型文

```
import numpy as np
```

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

計算に必要なモジュールnumpyと
グラフ表示に必要なモジュール
matplotlib.pyplotのインポート

```
x = np.linspace(-np.pi, np.pi)
```

linspaceで-のpiから+のpiまでの
配列を作る

```
plt.plot(x, np.cos(x))
```

```
plt.plot(x, np.sin(x))
```

```
plt.show()
```

グラフの計算
ここではxのcosとsin

グラフの表示

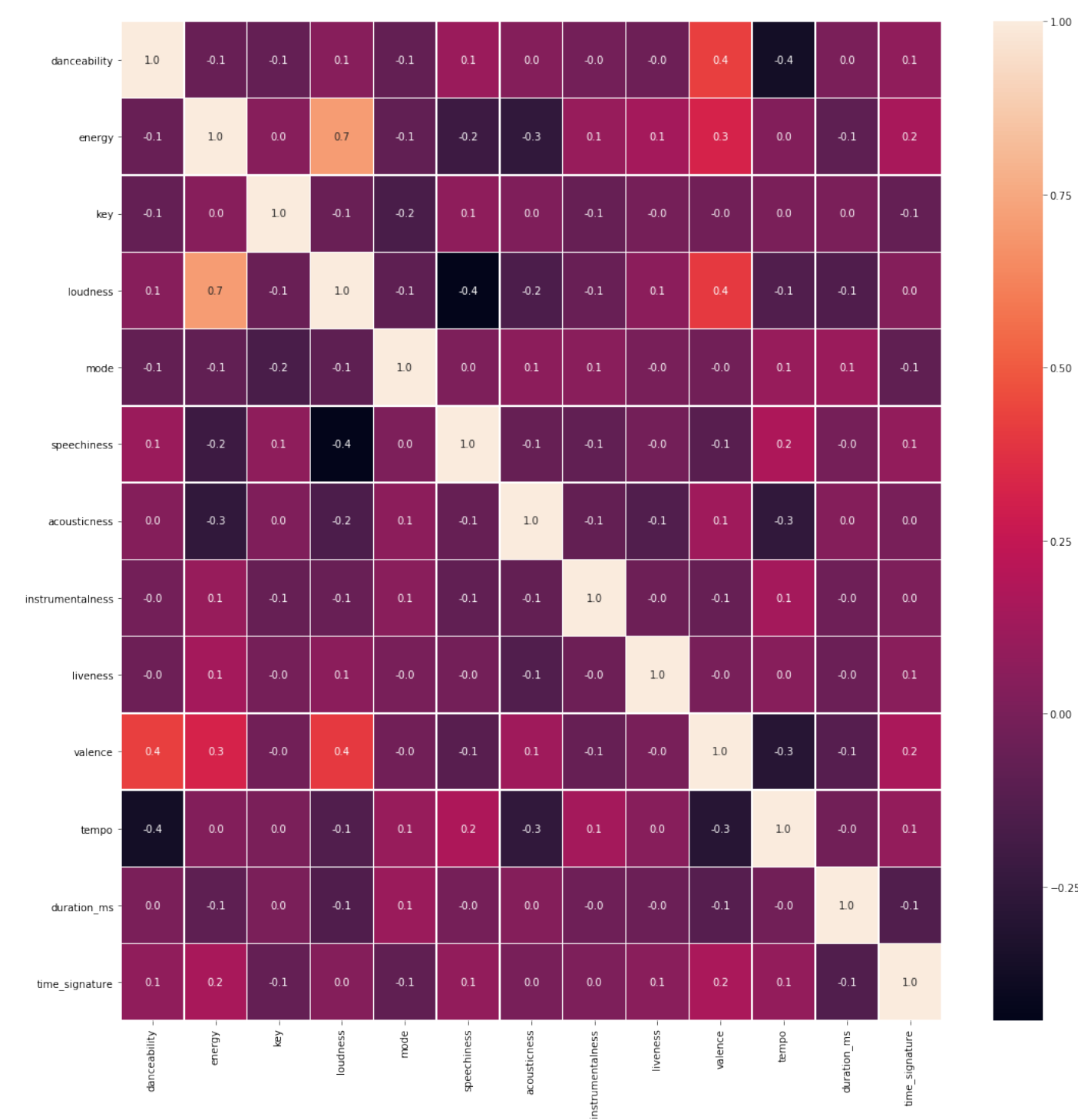
spotifyデータ分析

データセットの読み込み

```
data = pd.read_csv("../任意/spotify2017data.csv")
data.head(10)
```

	id	name	artists	danceability	energy	key	loudness	mode	speechiness
0	7qiZfU4dY1IWlzx7mPBI	Shape of You	Ed Sheeran	0.825	0.652	1.0	-3.183	0.0	0.0802
1	5CtI0qwDJkDQGwXD1H1cL	Despacito - Remix	Luis Fonsi	0.694	0.815	2.0	-4.328	1.0	0.1200
2	4aWmUDTfIPGksMNLV2rQP	Despacito (Featuring Daddy Yankee)	Luis Fonsi	0.660	0.786	2.0	-4.757	1.0	0.1700
3	6RUKPb4LETWmmr3iAEQkt	Something Just Like This	The Chainsmokers	0.617	0.635	11.0	-6.769	0.0	0.0317
4	3DXncPQOG4VBw3QHh3S81	I'm the One	DJ Khaled	0.609	0.668	7.0	-4.284	1.0	0.0367
5	7KXjTSCq5nL1LoYtL7XAw	HUMBLE.	Kendrick Lamar	0.904	0.611	1.0	-6.842	0.0	0.0888
6	3eR23VReFzcdmS7TYCrhC	It Ain't Me (with Selena Gomez)	Kygo	0.640	0.533	0.0	-6.596	1.0	0.0706
7	3B54sVLJ402zGa6Xm4YGN	Unforgettable	French Montana	0.726	0.769	6.0	-5.043	1.0	0.1230
8	OKKkJNfGyhkQ5aFogxQAP	That's What I Like	Bruno Mars	0.853	0.560	1.0	-4.961	1.0	0.0406

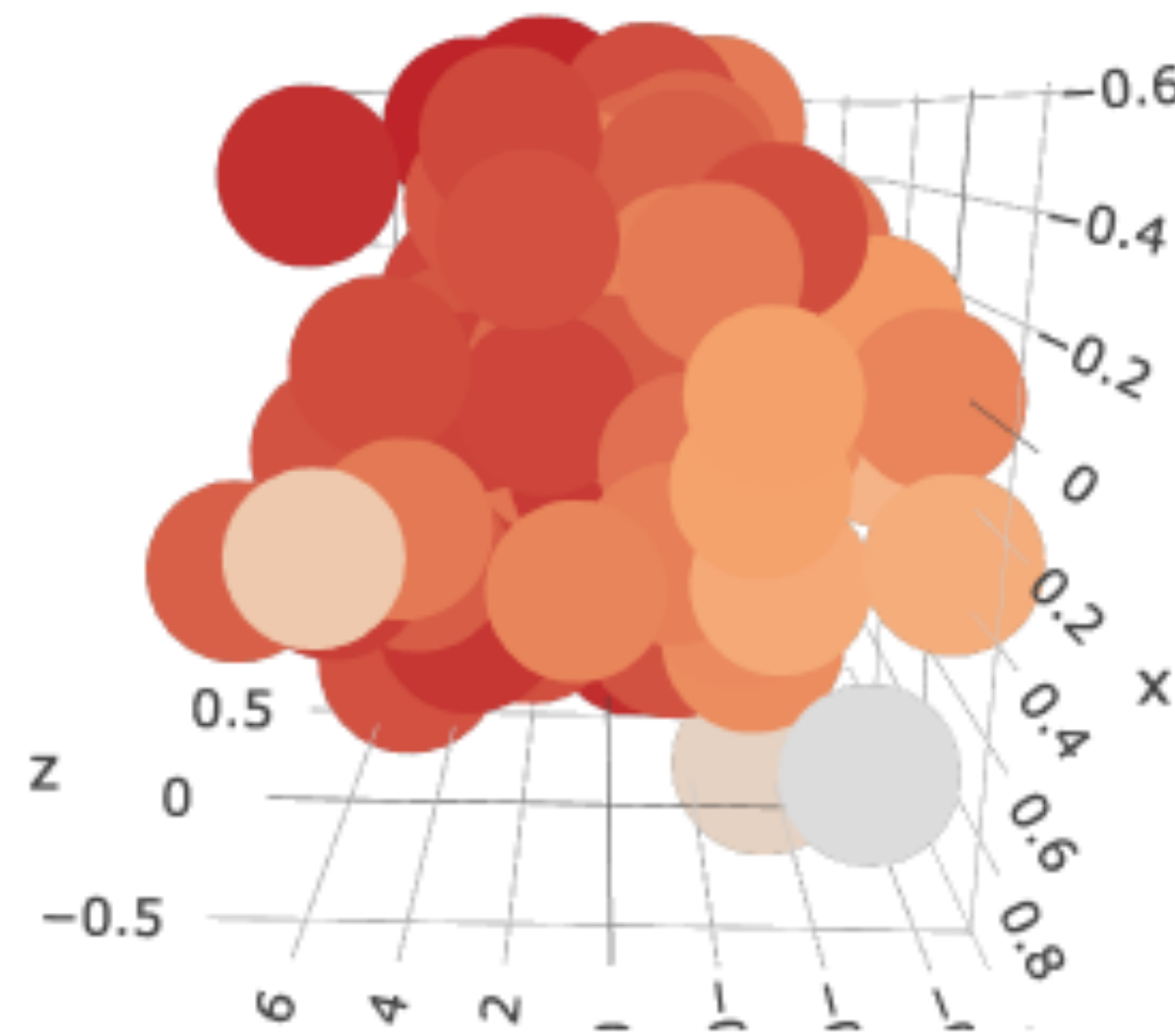
分析はグラフ化できます



各要素同士の相関性グラフ

matplotlibやseabornといった
データグラフ化用のライブラリーを使用

分析はグラフ化できます



異なる楽曲間の各要素の
相関性グラフ

matplotlibやseabornといった
データグラフ化用のライブラリーを使用

分析してみましょう

```
print("Mean value for danceability:", data['danceability'].mean())
sns.distplot(data['danceability'])
plt.show()
print("Mean value for energy:", data['energy'].mean())
sns.distplot(data['energy'])
plt.show()
print("Mean value for mode:", data['mode'].mean())
sns.distplot(data['mode'])
plt.show()
print("Mean value for speechiness:", data['speechiness'].mean())
sns.distplot(data['speechiness'])
plt.show()
print("Mean value for acousticness:", data['acousticness'].mean())
sns.distplot(data['acousticness'])
plt.show()
print("Mean value for instrumentalness:", data['instrumentalness'].mean())
sns.distplot(data['instrumentalness'])
plt.show()
print("Mean value for liveness:", data['liveness'].mean())
sns.distplot(data['liveness'])
plt.show()
print("Mean value for valence:", data['valence'].mean())
sns.distplot(data['valence'])
plt.show()
```

分析してみましょう

```
print("Mean value for danceability:", data['danceability'].mean())  
sns.distplot(data['danceability'])  
plt.show()
```

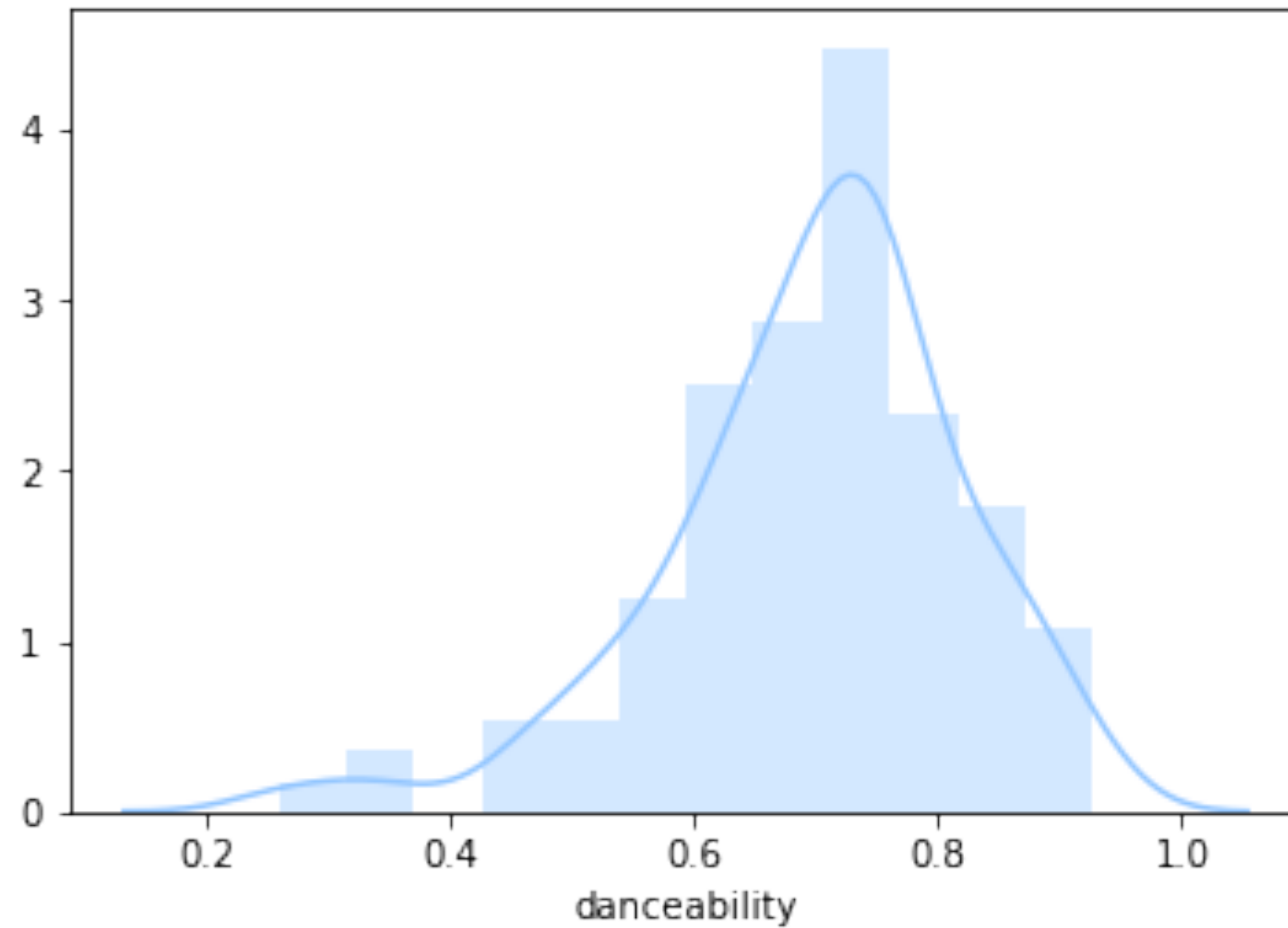
```
Mean value for danceability: 0.6968199999999999
```

danceability

danceability(踊りやすさ)

テンポ、リズムの安定性、ビートの強さ、全体的な規則性などの音楽要素の組み合わせに基づいて、ダンスのための曲であることを示す。0.0の値は、最も踊りづらいことを、1.0は、最も踊りやすいことを示す。

Danceability（踊れる曲かどうか）



リスナーは踊れる曲を好む傾向にあると見て取れます。

分析してみましょう

```
print("Mean value for energy:", data['energy'].mean())  
sns.distplot(data['energy'])  
plt.show()
```

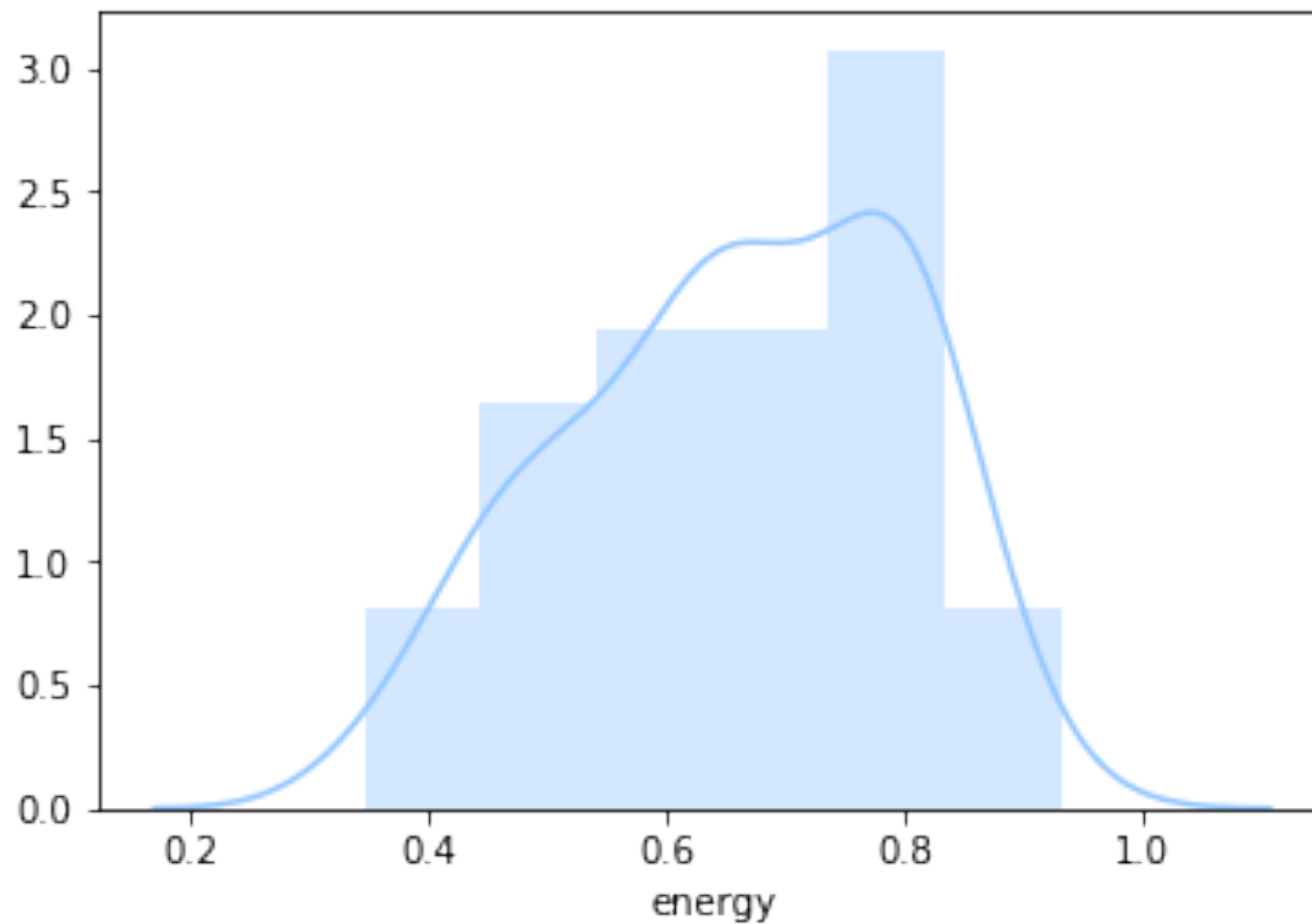
```
Mean value for energy: 0.6606899999999999
```

energy

エネルギーは、0.0から1.0の指標で、強度およびアクティブ度を表す。
一般的には、エネルギッシュな楽曲は、速く、音が大きく、騒々しい傾向となる。
例えば、デスメタルは高いエネルギーを持っているが、バッハのプレリュードは低い値になるなど。

この属性に寄与する知覚的属性は、ダイナミックレンジ、聴覚が感じる音量、音色、開始時点のレート、および一般的なエントロピーを含む。

Energy（楽曲のエネルギー度）



こちらもエネルギー度が高い曲が好まれる傾向が明確なデータとなっています。

分析してみましょう

```
print("Mean value for mode:", data['mode'].mean())  
sns.distplot(data['mode'])  
plt.show()
```

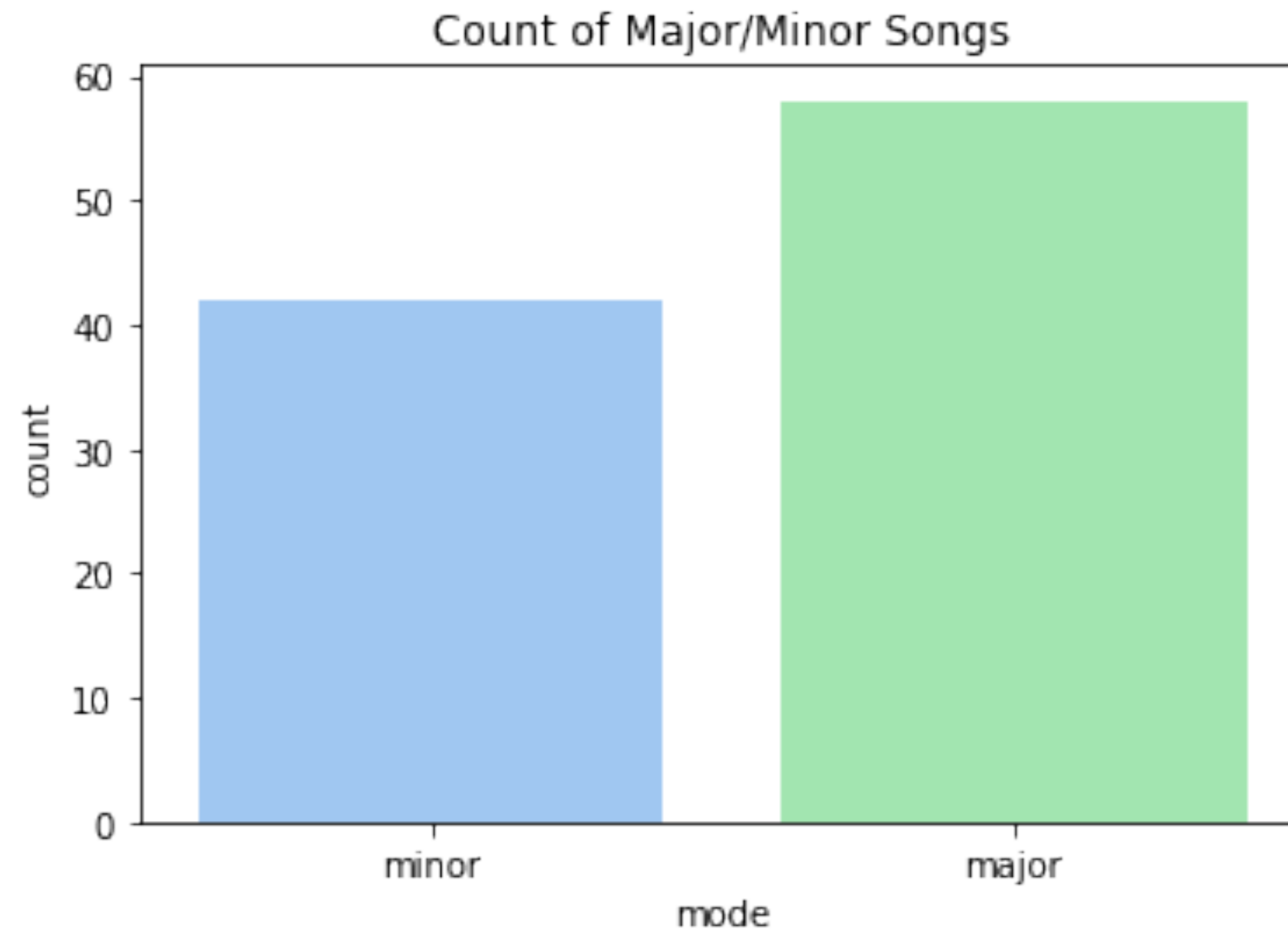
```
Mean value for mode: 0.58
```

mode

その楽曲のキーが長調であるか短調であるかを示す指標。

データの分類として単純に2種類しか属性がないため、出力されたデータの数値はある一方のデータの構成割合（今回は長調）を示す事となる。

Mode、長調か短調か？



こちらは60%弱が長調で、短調よりも人々が好む傾向が伺えます。

分析してみましょう

```
print("Mean value for speechiness:",  
data['speechiness'].mean())  
sns.distplot(data['speechiness'])  
plt.show()
```

```
Mean value for speechiness: 0.103968999999999995
```

speechiness

スピーチネスは、曲の中で話された単語の存在を検出する。

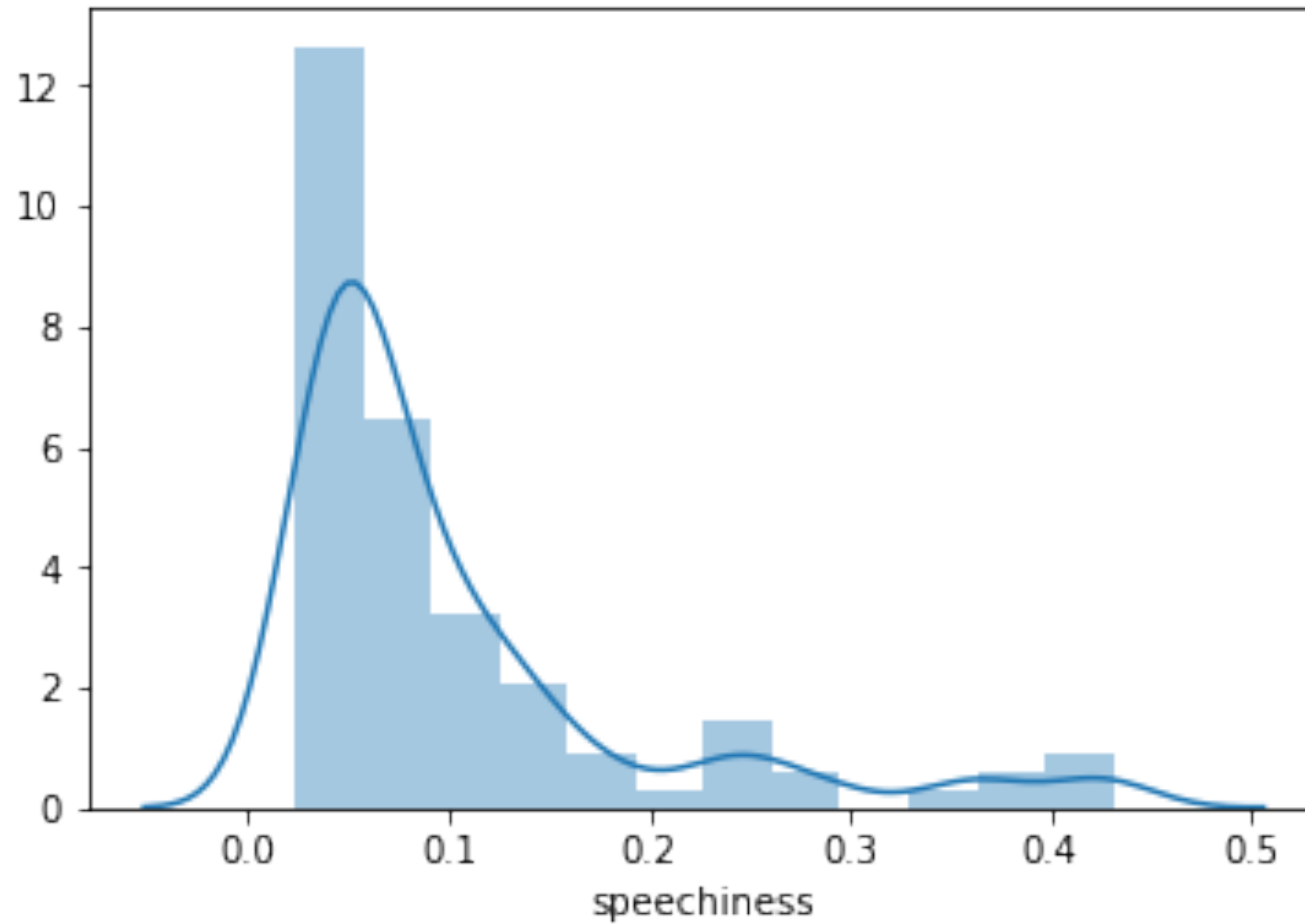
録音（例えば、トークショー、オーディオブック、詩）のように、音声占める割合が大きくなるほど、値は1.0に近くなる。

0.66を超える値は、ほぼ完全に発声された単語で構成されている曲を表す。

0.33と0.66の間の値は、ラップ音楽などのセクションまたはレイヤーのいずれかで、音楽とスピーチの両方を含む可能性がある曲を表す。

0.33未満の値は、音楽やその他の非音声のような曲を表す可能性がかなり高いことを示す。

Speechness、人の話声がどのくらい含まれているか



話声の占める割合は少ないものの0.3以上の割合（ラップなど）もある

分析してみましょう

```
print("Mean value for acousticness:",  
data['acousticness'].mean())  
sns.distplot(data['acousticness'])  
plt.show()
```

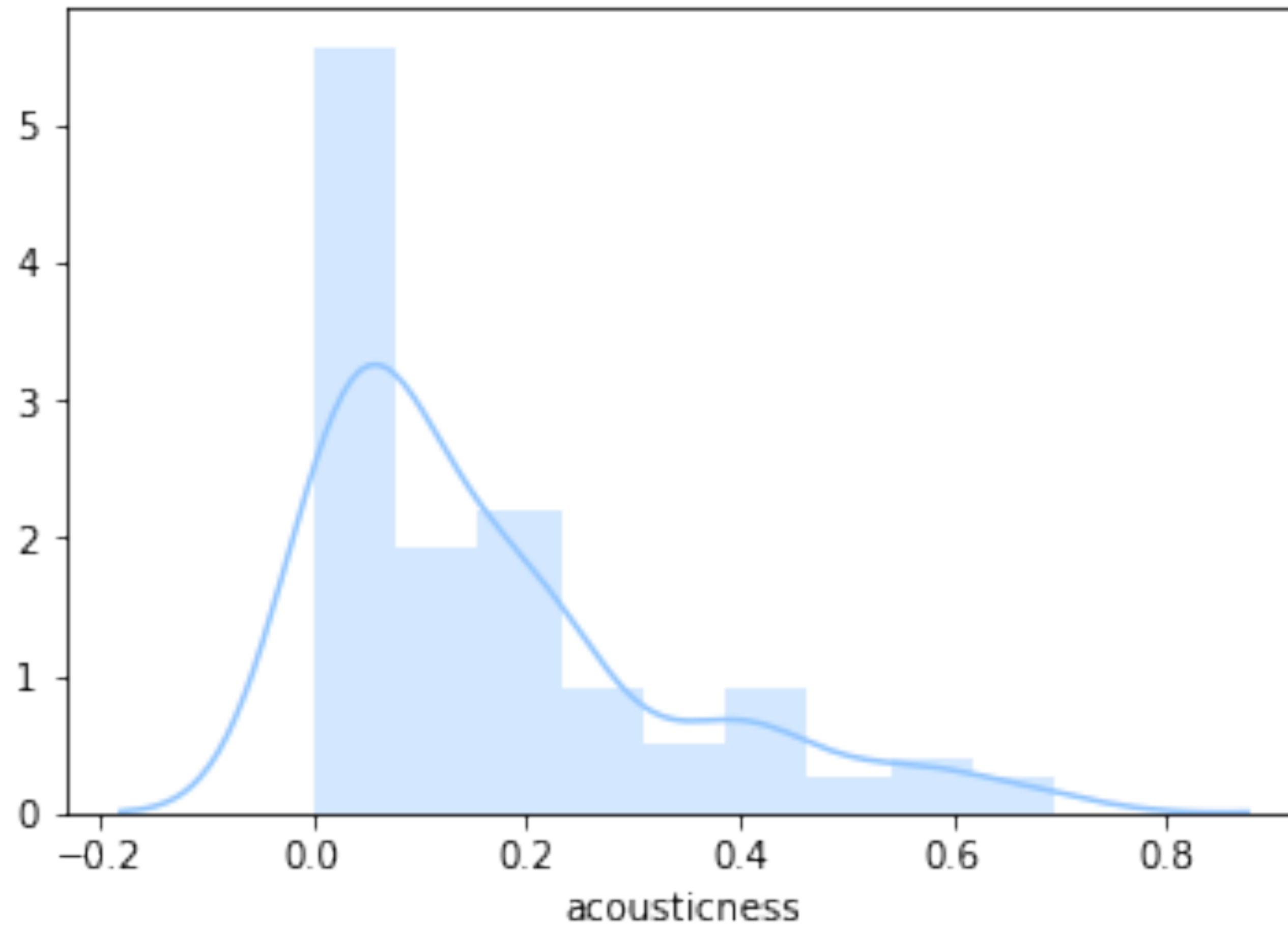
```
Mean value for acousticness: 0.166306490000000006
```

acousticness

曲がアコースティックかどうかを示す0.0から1.0の指標。

1.0は曲がアコースティックであるということが高いということを意味します。

Acousticness、楽曲のアコースティック度です



現代のデジタル時代を象徴するかの様なアコースティック率の低さ、、、

分析してみましょう

```
print("Mean value for instrumentality:",  
data['instrumentality'].mean())  
sns.distplot(data['instrumentality'])  
plt.show()
```

Mean value for instrumentality: 0.0047961442

instrumentality

曲にボーカルがないかどうかを示す。

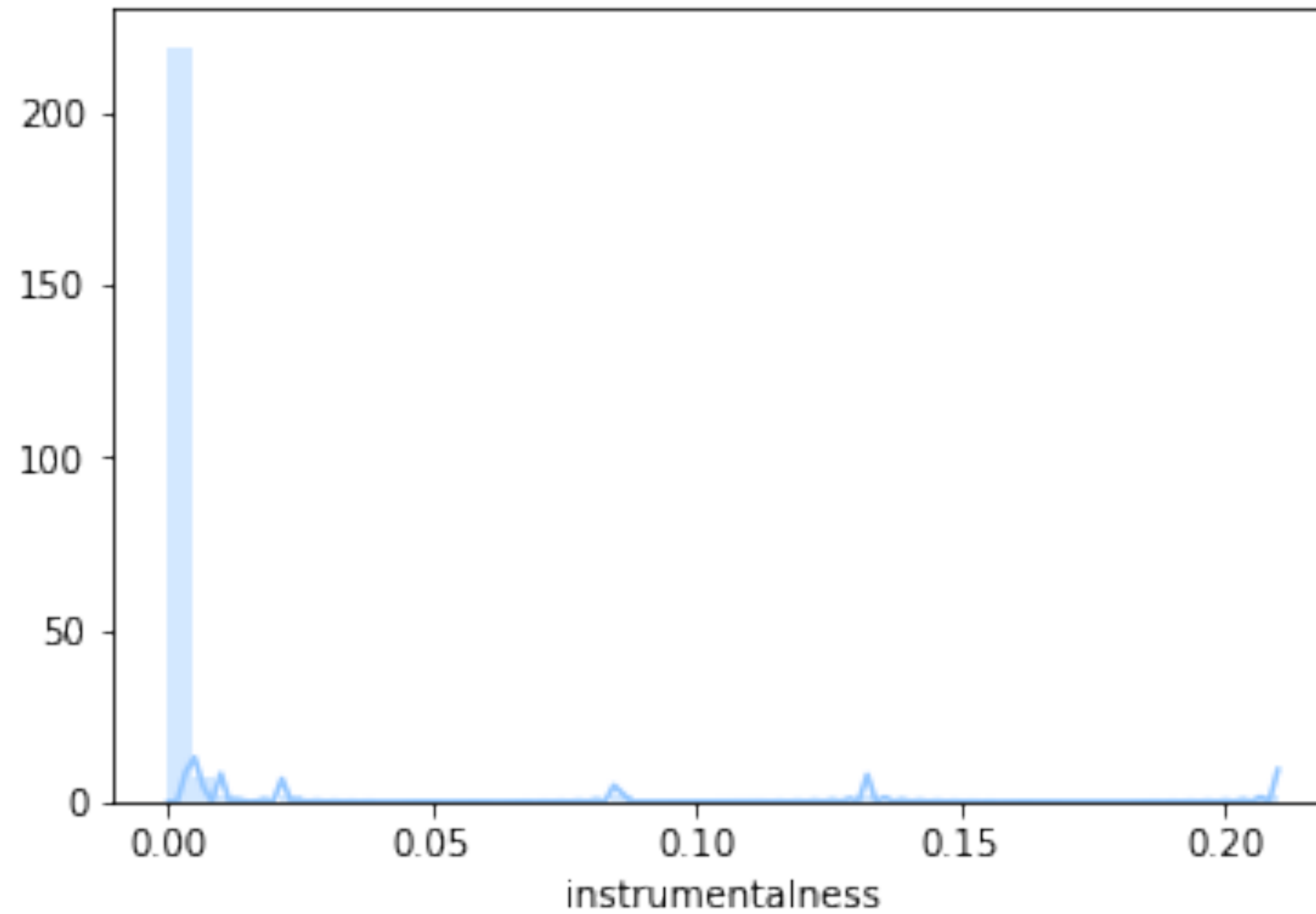
この指標では、“オー(Ooh)”とか“アー(aah)”の音は楽器の出した音として扱われる。

ラップや話し言葉はボーカルとして扱われる。

インストゥルメンタルネスの値が1.0に近いほど、曲にはボーカル・コンテンツが含まれていない可能性が高くなる。

0.5を超える値は、インストゥルメンタルの曲が通常示す値ですが、値が1.0に近づくほど信頼度が高くなります。

Instrumentness、楽曲の中でインストパートが占める割合です



インストパートは、、、残念ながら（喜ぶ人もいるかも？）全くヒット曲には必要ない様です、、、

分析してみましょう

```
print("Mean value for liveness:", data['liveness'].mean())  
sns.distplot(data['liveness'])  
plt.show()
```

```
Mean value for liveness: 0.15060700000000002
```

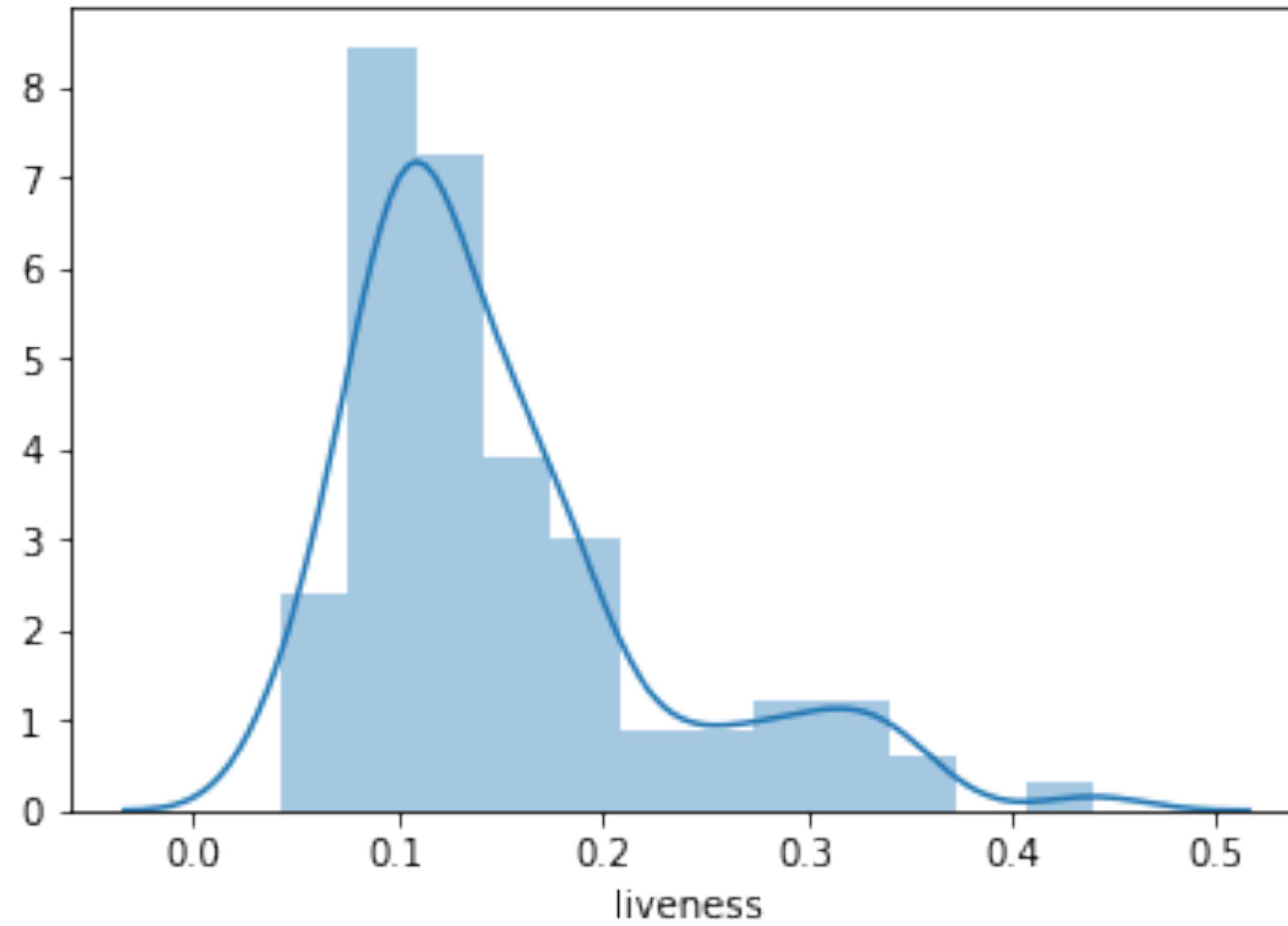
liveness

録音中に聴衆（の声）が存在したかを検出する。

この値が高いほど、曲がライブで実行された可能性が高くなる。

値が0.8を超えると、曲がライブである可能性が高い。

Liveness、楽曲の中で聴衆の声援が占める割合です



ライブ盤は現代は少ない

分析してみましょう

```
print("Mean value for valence:", data['valence'].mean())  
sns.distplot(data['valence'])  
plt.show()
```

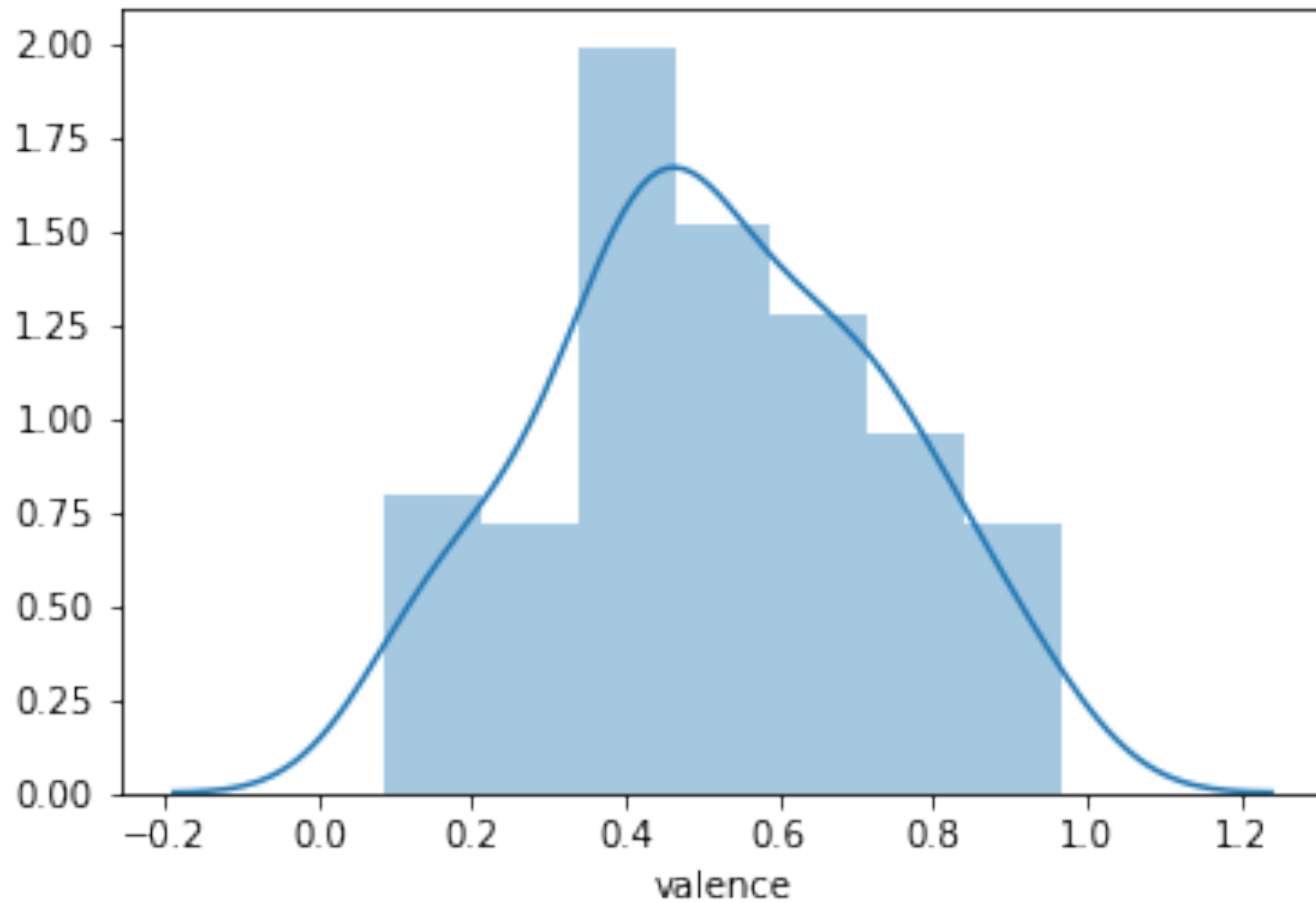
```
Mean value for valence: 0.5170489999999999
```

valence

曲が伝える音楽のポジティブ性を表す0.0から1.0の尺度。

この指数の高い値の曲はより陽性（例えば、幸せ、陽気、陶酔）であり、低い指数の曲はより陰性となる（例：悲しい、落ち込んだ、怒る）。

Valence、曲の明るさ、暗さ



平均 0.517

明るさ暗さは中間くらいというデータ

分析してみましょう

```
print("Mean value for tempo:", data['tempo'].mean())  
sns.distplot(data['tempo'])  
plt.show()
```

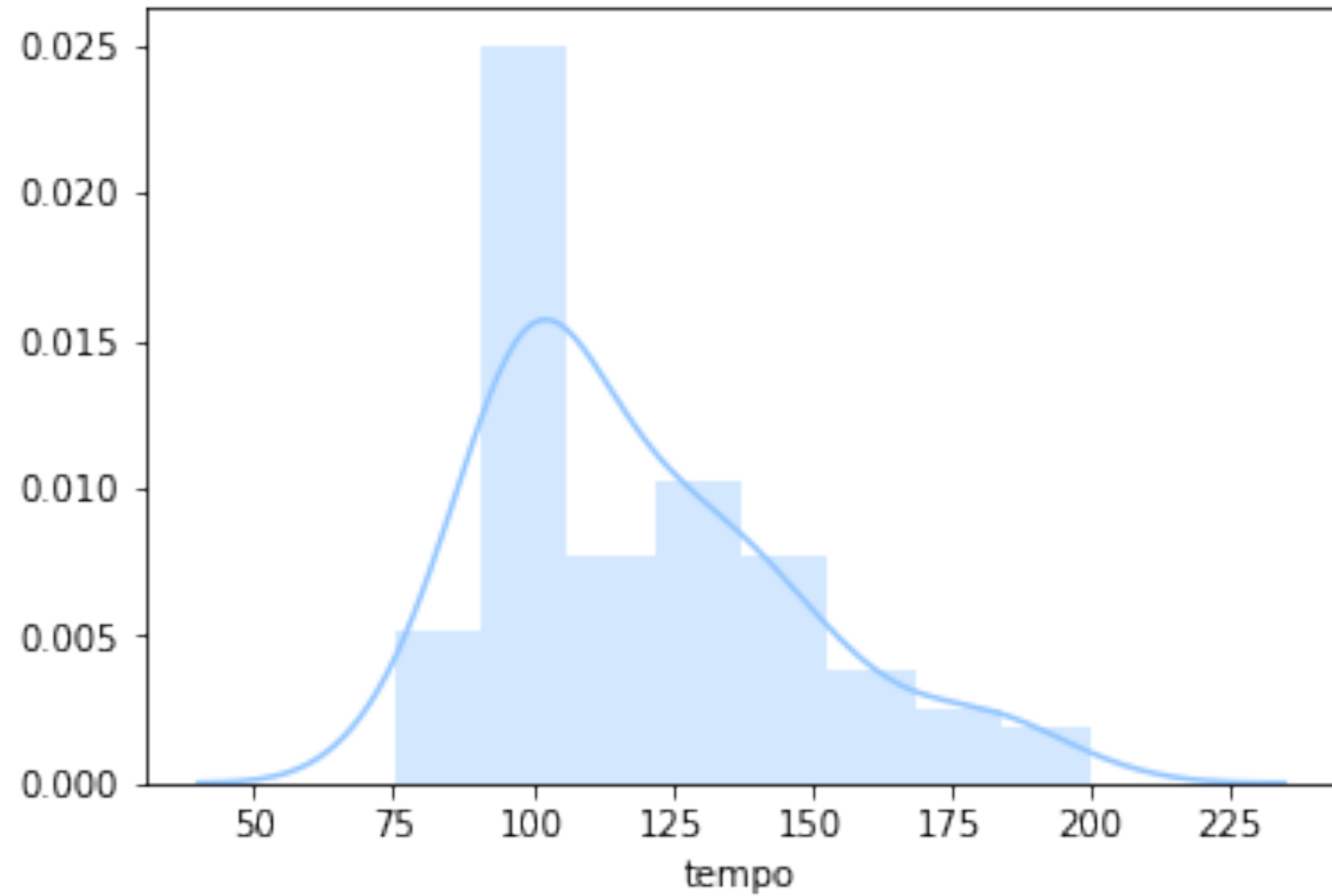
```
Mean value for tempo: 120.8489
```

tempo

曲の全体的な推定テンポ。

1分あたりのビート(BPM)から算出。

Tempoです



平均テンポはBPM約 1 2 0。

一方もっとも多いテンポはBPM 9 0 ～ 1 0 0 あり、他のテンポよりも圧倒的に多いです。

分析してみましょう

```
print("Mean value for duration_ms:",  
data['duration_ms'].mean())  
sns.distplot(data['duration_ms'])  
plt.show()
```

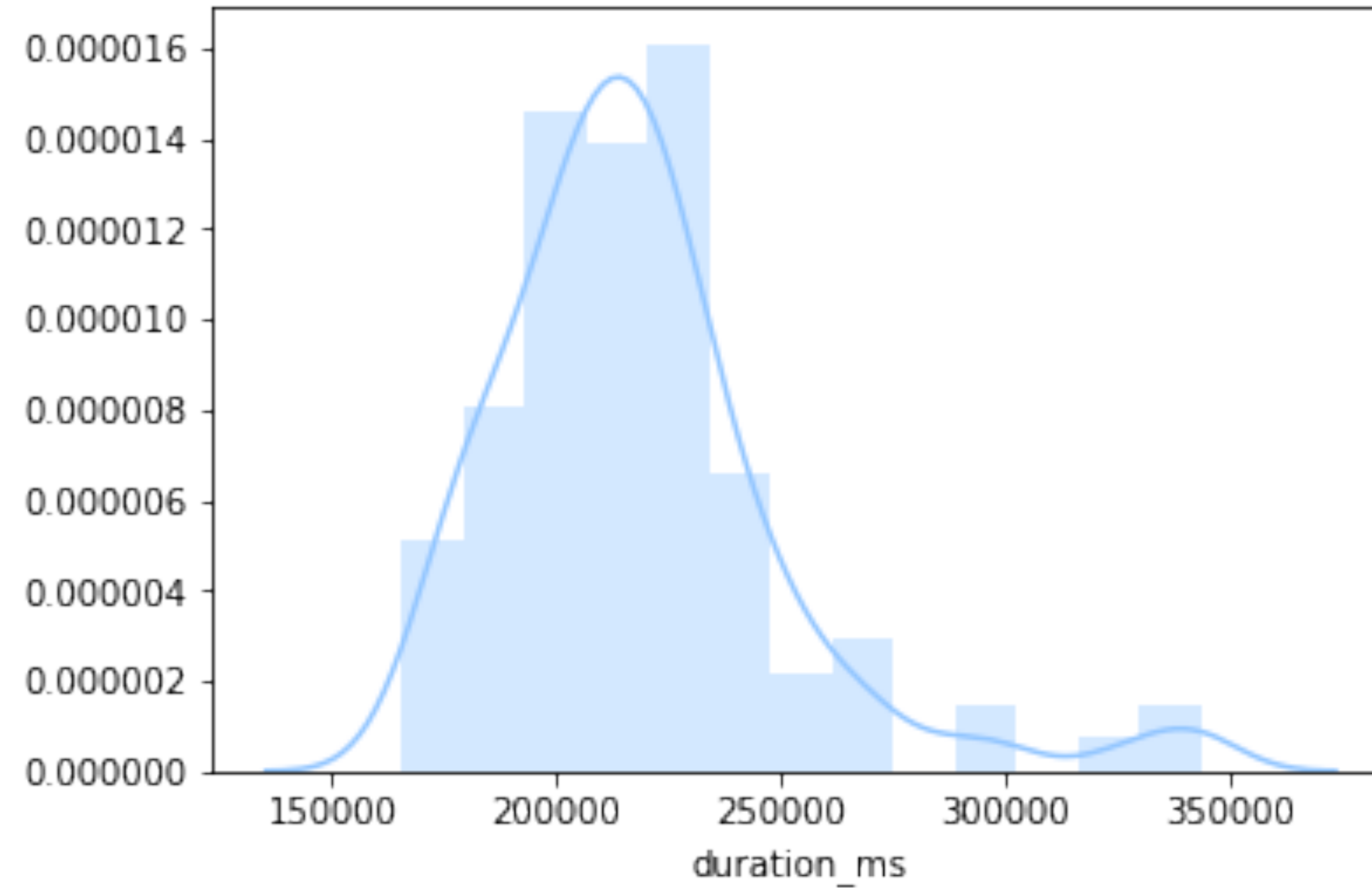
```
Mean value for duration_ms: 218387
```

duration

曲の長さ。

単位はmsec(1000分の1秒) となる。

Duration、曲の長さです



平均 218387msecは約 3 分 3 8 秒。

そのくらいの長さから少し短めくらいが好まれる一方、長い曲は好まれない傾向を示しています。

分析してみましょう

```
numeric = data.drop(['id', 'name', 'artists'], axis=1)
small =
numeric.drop(['tempo', 'duration_ms', 'key', 'loudness', 'time_signature'],
axis=1)
sns.set_palette('pastel')
small.mean().plot.bar()
plt.title('Mean Values of Audio Features')
plt.show()
```

Mean Values of Audio Features

