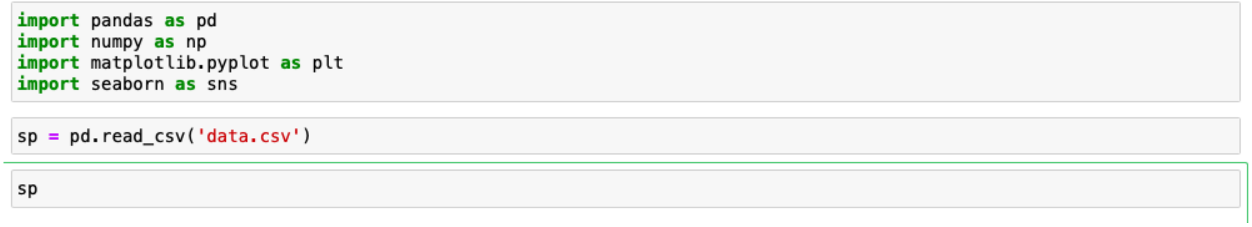
Exploratory Data Analysis on Spotify dataset

The following libraries were imported because they will be used in data preprocessing and exploratory data analysis. I have imported the pandas library, and numpy library. For data analysis, I have imported the seaborn and matplotlib libraries.

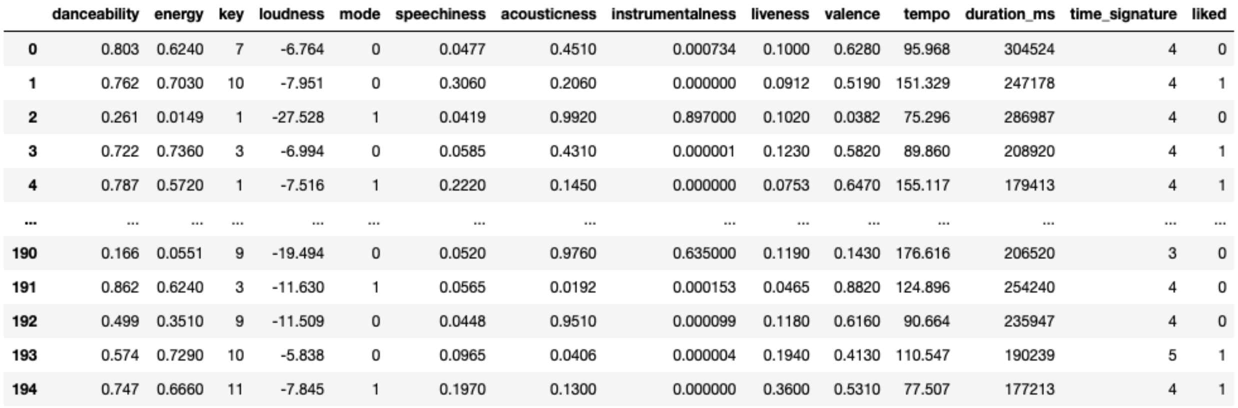
After importing the necessary libraries, we read the dataset file called data.csv



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

```
sp = pd.read_csv('data.csv')
```

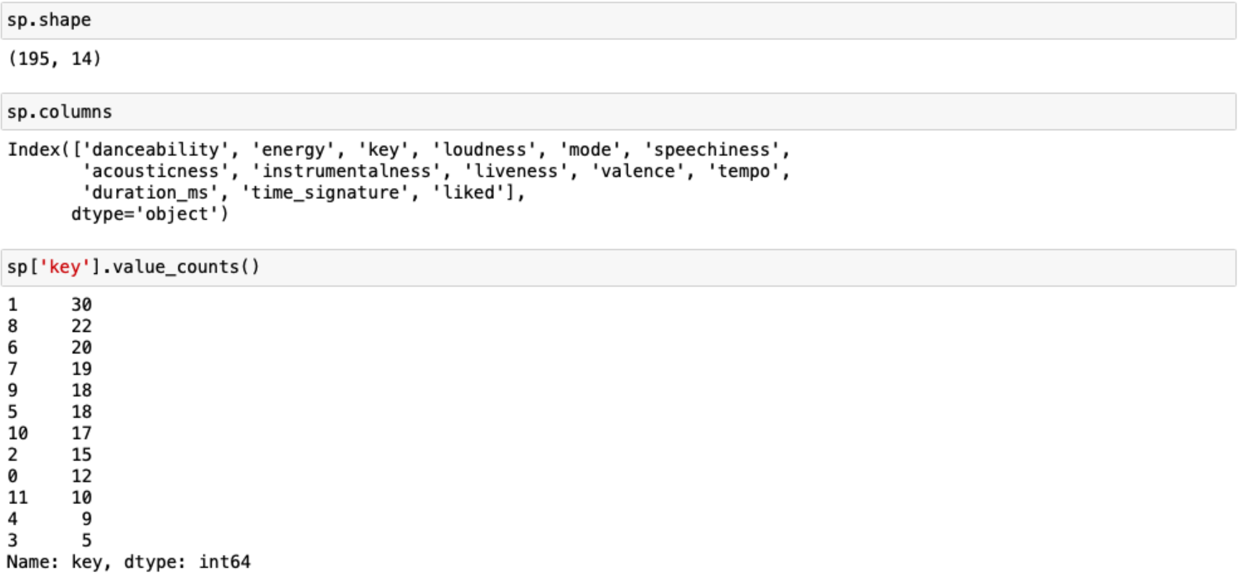
```
sp
```



	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	liked
0	0.803	0.6240	7	-6.764	0	0.0477	0.4510	0.000734	0.1000	0.6280	95.968	304524	4	0
1	0.762	0.7030	10	-7.951	0	0.3060	0.2060	0.000000	0.0912	0.5190	151.329	247178	4	1
2	0.261	0.0149	1	-27.528	1	0.0419	0.9920	0.897000	0.1020	0.0382	75.296	286987	4	0
3	0.722	0.7360	3	-6.994	0	0.0585	0.4310	0.000001	0.1230	0.5820	89.860	208920	4	1
4	0.787	0.5720	1	-7.516	1	0.2220	0.1450	0.000000	0.0753	0.6470	155.117	179413	4	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
190	0.166	0.0551	9	-19.494	0	0.0520	0.9760	0.635000	0.1190	0.1430	176.616	206520	3	0
191	0.862	0.6240	3	-11.630	1	0.0565	0.0192	0.000153	0.0465	0.8820	124.896	254240	4	0
192	0.499	0.3510	9	-11.509	0	0.0448	0.9510	0.000099	0.1180	0.6160	90.664	235947	4	0
193	0.574	0.7290	10	-5.838	0	0.0965	0.0406	0.000004	0.1940	0.4130	110.547	190239	5	1
194	0.747	0.6660	11	-7.845	1	0.1970	0.1300	0.000000	0.3600	0.5310	77.507	177213	4	1

The dataset consists of 195 rows and 14 columns. Here the target variable is the **liked** column.

Then we check the count of unique values for some columns. First, let's check how many unique values are there in the key column. So, there are 11 unique keys, where key 1 is most common in all the songs. Then it is key 8 as the second most common in all the songs.



```
sp.shape
```

```
(195, 14)
```

```
sp.columns
```

```
Index(['danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
       'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_ms', 'time_signature', 'liked'],
      dtype='object')
```

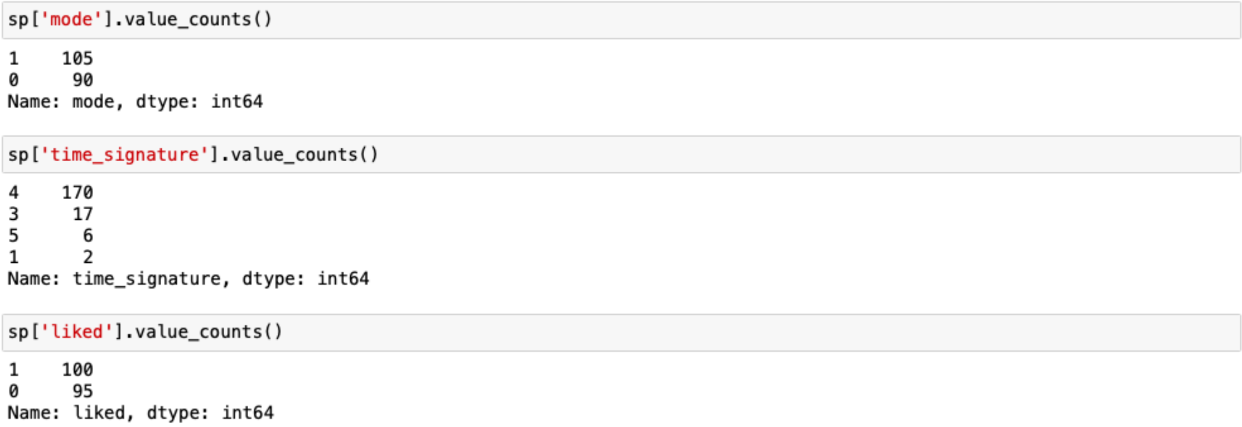
```
sp['key'].value_counts()
```

```
1     30
8     22
6     20
7     19
9     18
5     18
10    17
2     15
0     12
11    10
4      9
3      5
Name: key, dtype: int64
```

Here we see the unique values in the mode column. There are 2 unique values in the mode column, which are 1 and 0. The value 1 is the majority in the mode column.

Then we look into the unique values in the time signature column. Here we can see that there are 4 unique values, which means that there are 4 different unique time signatures such as 1,3,4, and 5. In this column, the time signature 4 is the most common in the songs.

Lastly, we look into the liked column where there are 2 unique values that tell us whether the song is liked or not. Most of the songs are liked only because the majority of the songs are shown as 1, which means they were liked.



```
sp['mode'].value_counts()
```

```
1    105
0     90
Name: mode, dtype: int64
```

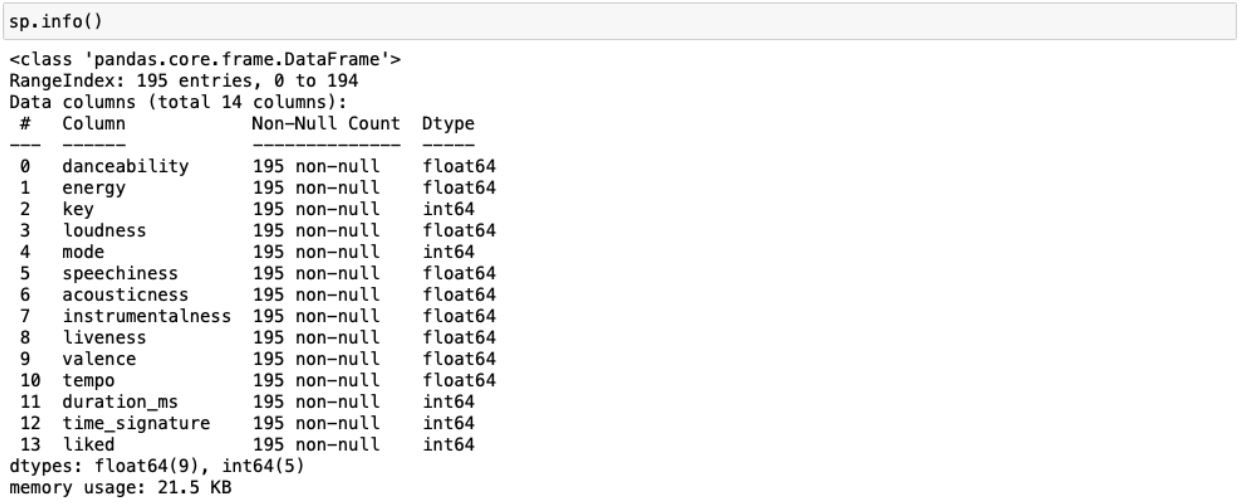
```
sp['time_signature'].value_counts()
```

```
4    170
3     17
5      6
1      2
Name: time_signature, dtype: int64
```

```
sp['liked'].value_counts()
```

```
1    100
0     95
Name: liked, dtype: int64
```

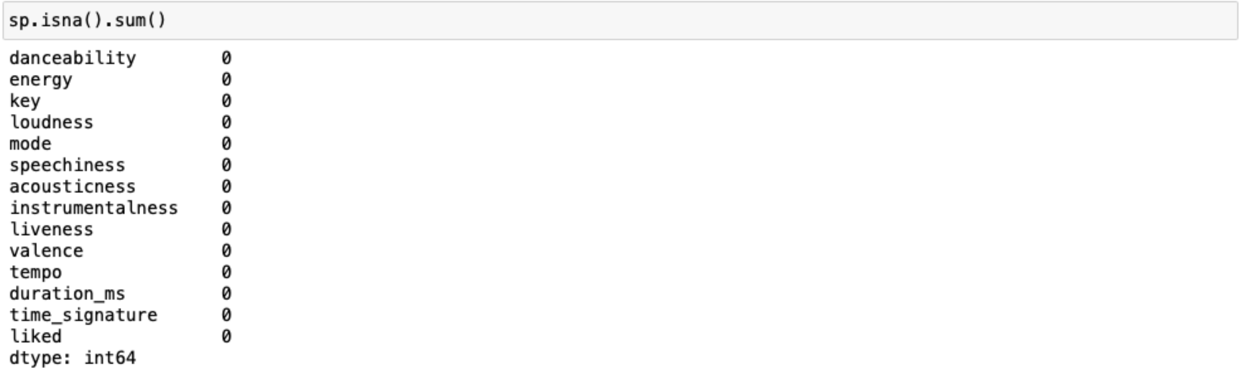
Below you can see the attributes of this dataset and we can see data types of each column. All of them are in numeric form and some of them are integers and some are in float type. However, these variables are different from each other because they can be categorical, discrete or continuous variables. Danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, tempo, duration_ms, and time_signature are all continuous variables. Key, mode, and liked are all categorical variables.



```
sp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   danceability      195 non-null    float64
 1   energy            195 non-null    float64
 2   key               195 non-null    int64
 3   loudness          195 non-null    float64
 4   mode              195 non-null    int64
 5   speechiness       195 non-null    float64
 6   acousticness      195 non-null    float64
 7   instrumentalness  195 non-null    float64
 8   liveness          195 non-null    float64
 9   valence           195 non-null    float64
 10  tempo             195 non-null    float64
 11  duration_ms       195 non-null    int64
 12  time_signature    195 non-null    int64
 13  liked             195 non-null    int64
dtypes: float64(9), int64(5)
memory usage: 21.5 KB
```

It is necessary to check if there are any missing values in the dataset before we begin the preprocessing or exploratory data analysis. Missing values can be a problem when we perform these tasks so it is best to resolve them in the beginning itself.



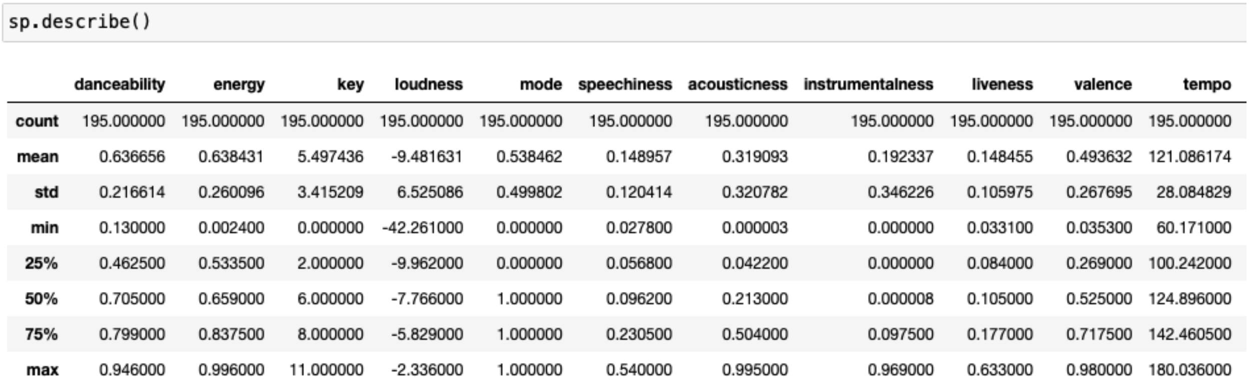
```
sp.isna().sum()
```

```
danceability        0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
acousticness        0
instrumentalness    0
liveness            0
valence             0
tempo               0
duration_ms         0
time_signature      0
liked               0
dtype: int64
```

This is a summary of the dataset where it shows a statistical summary of each column in the dataset. The statistical summary gives insightful information about each column. It gives information like total count, mean, standard deviation, minimum value, maximum value and also the 25%, 50%, 75% of the column values. We can determine if the mean and the 50% value are close to each other or not.

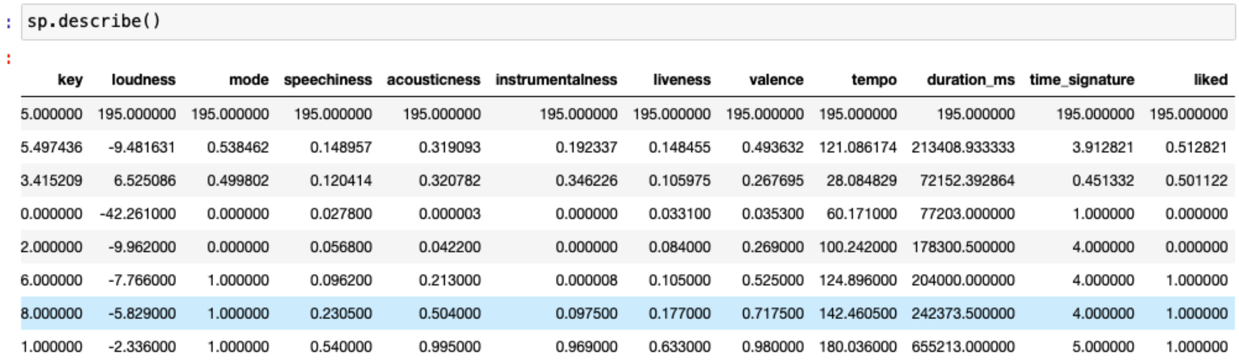
As you can see, in this table we can get the following information

1. Loudness
    a. Maximum loudness is -42 dCB and minimum loudness is -2.3 dCB



```
sp.describe()
```

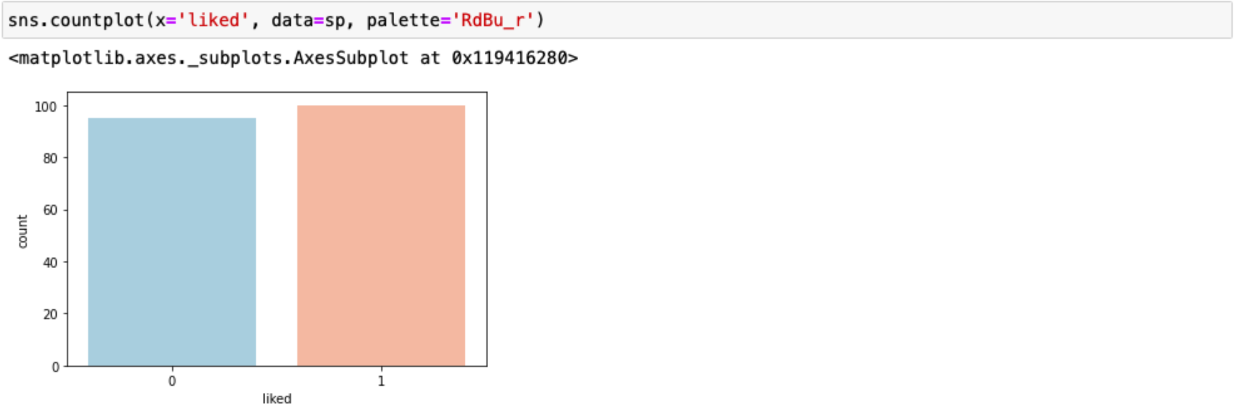
	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000
mean	0.636656	0.638431	5.497436	-9.481631	0.538462	0.148957	0.319093	0.192337	0.148455	0.493632	121.086174
std	0.216614	0.260096	3.415209	6.525086	0.499802	0.120414	0.320782	0.346226	0.105975	0.267695	28.084829
min	0.130000	0.002400	0.000000	-42.261000	0.000000	0.027800	0.000003	0.000000	0.033100	0.035300	60.171000
25%	0.462500	0.533500	2.000000	-9.962000	0.000000	0.056800	0.042200	0.000000	0.084000	0.269000	100.242000
50%	0.705000	0.659000	6.000000	-7.766000	1.000000	0.096200	0.213000	0.000008	0.105000	0.525000	124.896000
75%	0.799000	0.837500	8.000000	-5.829000	1.000000	0.230500	0.504000	0.097500	0.177000	0.717500	142.460500
max	0.946000	0.996000	11.000000	-2.336000	1.000000	0.540000	0.995000	0.969000	0.633000	0.980000	180.036000



```
sp.describe()
```

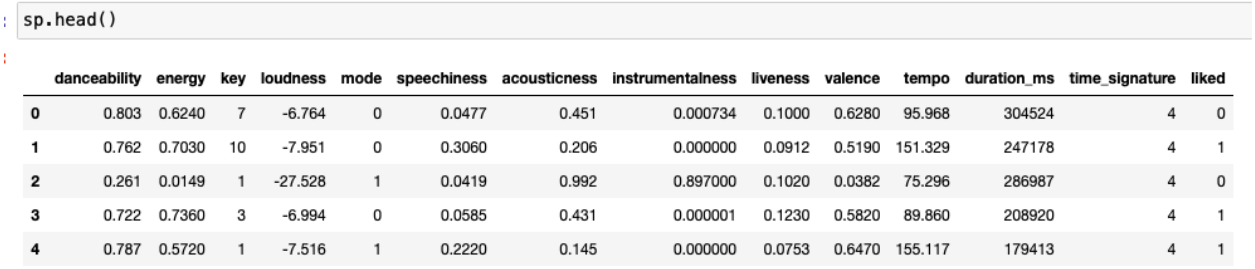
key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	liked
5.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000
5.497436	-9.481631	0.538462	0.148957	0.319093	0.192337	0.148455	0.493632	121.086174	213408.933333	3.912821	0.512821
3.415209	6.525086	0.499802	0.120414	0.320782	0.346226	0.105975	0.267695	28.084829	72152.392864	0.451332	0.501122
0.000000	-42.261000	0.000000	0.027800	0.000003	0.000000	0.033100	0.035300	60.171000	77203.000000	1.000000	0.000000
2.000000	-9.962000	0.000000	0.056800	0.042200	0.000000	0.084000	0.269000	100.242000	178300.500000	4.000000	0.000000
6.000000	-7.766000	1.000000	0.096200	0.213000	0.000008	0.105000	0.525000	124.896000	204000.000000	4.000000	1.000000
8.000000	-5.829000	1.000000	0.230500	0.504000	0.097500	0.177000	0.717500	142.460500	242373.500000	4.000000	1.000000
1.000000	-2.336000	1.000000	0.540000	0.995000	0.969000	0.633000	0.980000	180.036000	655213.000000	5.000000	1.000000

This is a countplot showing the number of samples in the dataset. This is an imbalanced dataset because both classes 1 and 0 are not equal as the class 1 has the most samples in this dataset.



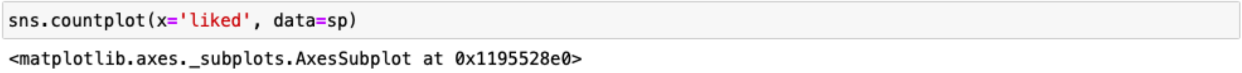
```
sns.countplot(x='liked', data=sp, palette='RdBu_r')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x119416280>
```



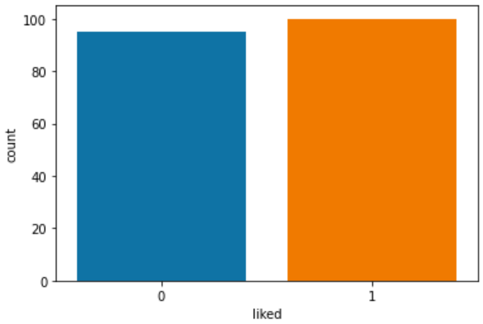
```
sp.head()
```

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	liked
0	0.803	0.6240	7	-6.764	0	0.0477	0.451	0.000734	0.1000	0.6280	95.968	304524	4	0
1	0.762	0.7030	10	-7.951	0	0.3060	0.206	0.000000	0.0912	0.5190	151.329	247178	4	1
2	0.261	0.0149	1	-27.528	1	0.0419	0.992	0.897000	0.1020	0.0382	75.296	286987	4	0
3	0.722	0.7360	3	-6.994	0	0.0585	0.431	0.000001	0.1230	0.5820	89.860	208920	4	1
4	0.787	0.5720	1	-7.516	1	0.2220	0.145	0.000000	0.0753	0.6470	155.117	179413	4	1

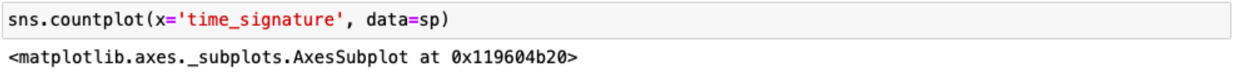


```
sns.countplot(x='liked', data=sp)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1195528e0>
```

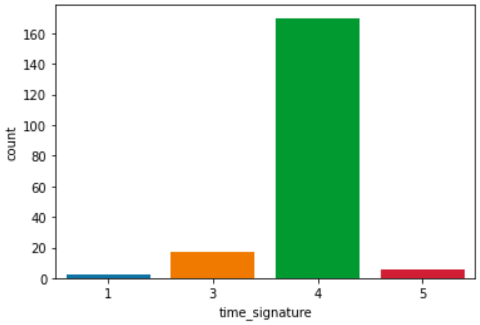


This is a countplot of time signature column where we can see the majority of the songs have a time signature of 4 while only few songs have time signature of 1 or 5. Some also have time signature of 3.

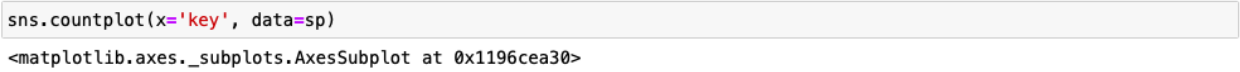


```
sns.countplot(x='time_signature', data=sp)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x119604b20>
```

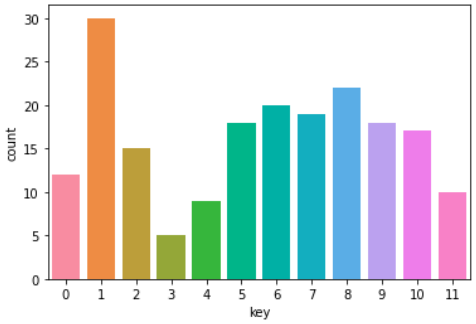


Here we can see the countplot for different keys. The majority key is 30 over here.

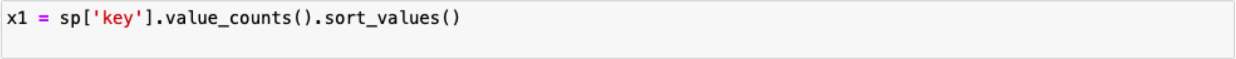


```
sns.countplot(x='key', data=sp)
```

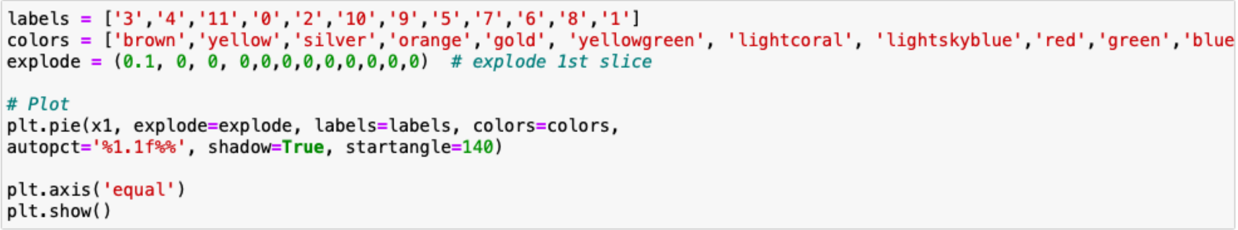
```
<matplotlib.axes._subplots.AxesSubplot at 0x1196cea30>
```



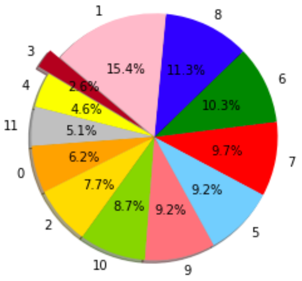
This pie chart shows the percentage of keys in the songs of the datasets. Majority of the songs is in key 1 and then in 8 and then in 6.



```
x1 = sp['key'].value_counts().sort_values()
```



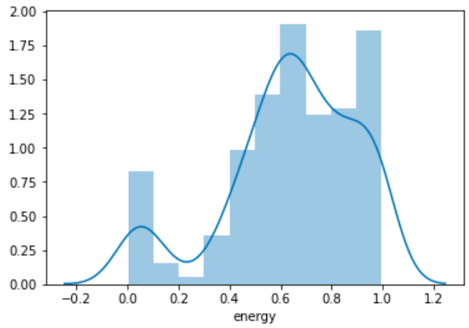
```
labels = ['3','4','11','0','2','10','9','5','7','6','8','1']
colors = ['brown','yellow','silver','orange','gold', 'yellowgreen', 'lightcoral', 'lightskyblue','red','green','blue
explode = (0.1, 0, 0, 0,0,0,0,0,0,0,0,0)  # explode 1st slice

# Plot
plt.pie(x1, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=140)

plt.axis('equal')
plt.show()
```



Here I have plotted the distribution plot for some of the columns such as energy, loudness, acousticness, tempo, liveness, danceability and valence.

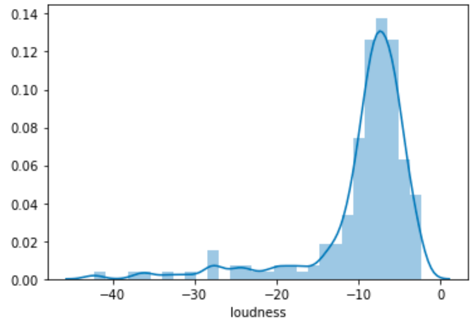
```
sns.distplot(sp.energy)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x119416b80>
```



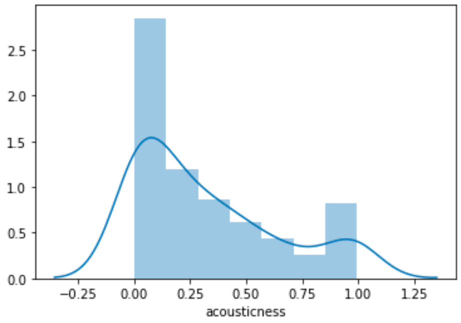
```
sns.distplot(sp.loudness)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1199bbd90>
```



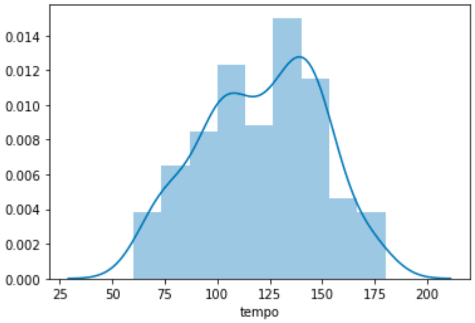
```
sns.distplot(sp.acousticness)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x119acc340>
```



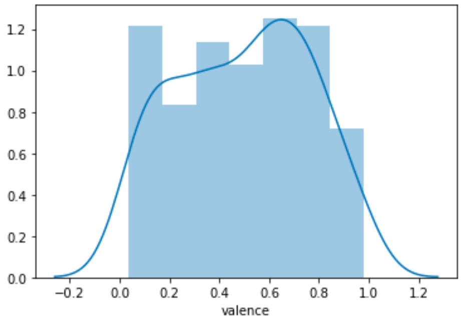
```
sns.distplot(sp.tempo)
```

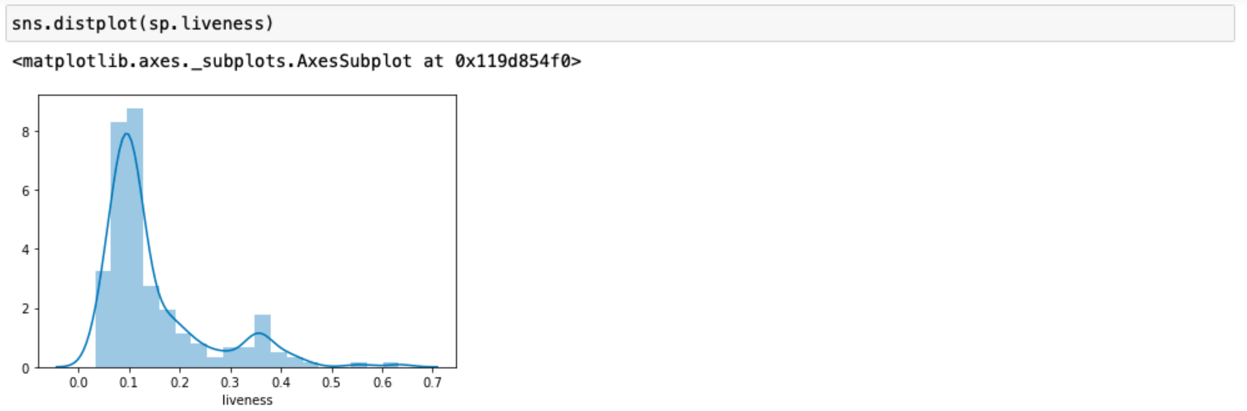
```
<matplotlib.axes._subplots.AxesSubplot at 0x119baf130>
```



```
sns.distplot(sp.valence)
```

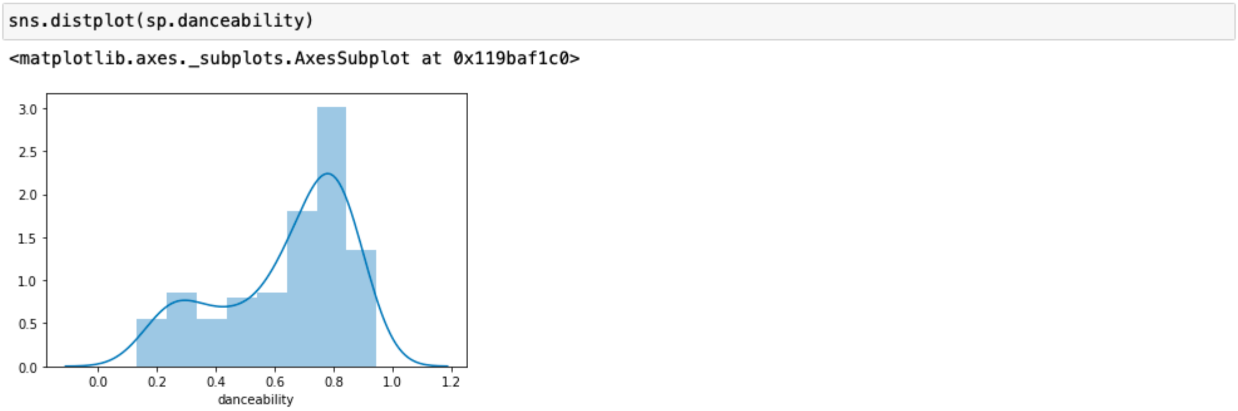
```
<matplotlib.axes._subplots.AxesSubplot at 0x119cacac0>
```





```
sns.distplot(sp.liveness)
```

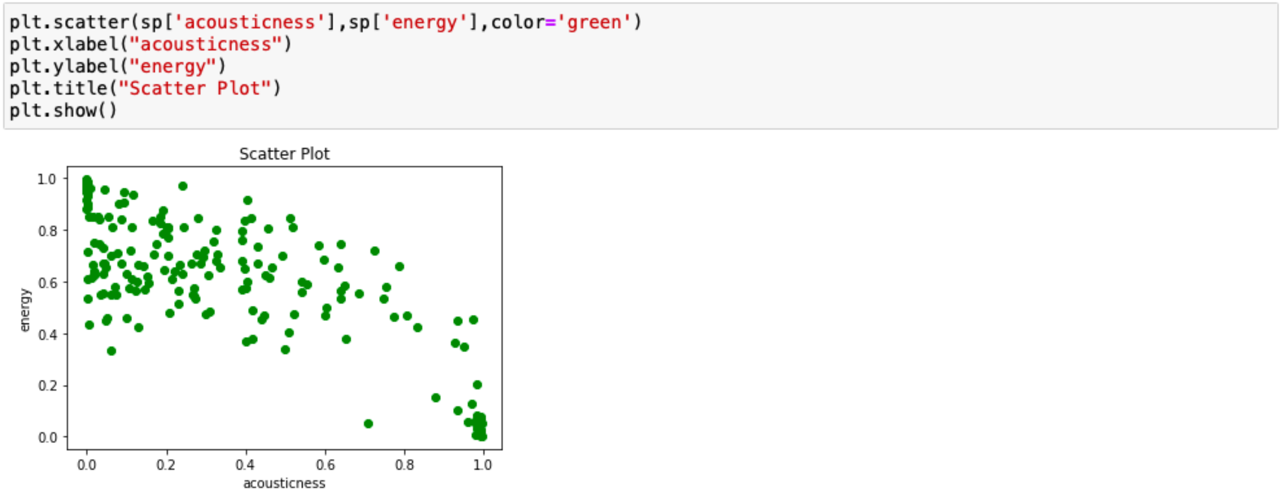
```
<matplotlib.axes._subplots.AxesSubplot at 0x119d854f0>
```



```
sns.distplot(sp.danceability)
```

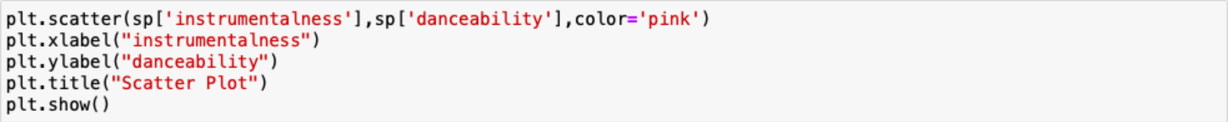
```
<matplotlib.axes._subplots.AxesSubplot at 0x119baf1c0>
```

Here you can see a scatterplot that depicts how the acoustic ness affects the energy of the songs. The plot shows that when the acoustic ness is low the energy of the song is high but in some cases when the acoustic ness is high the energy of the song is low. Therefore, there is a negative correlation between acoustic ness and energy. If we imagine that a line is drawn across the points on this plot the line would be declining linearly thus, it is negatively correlated.

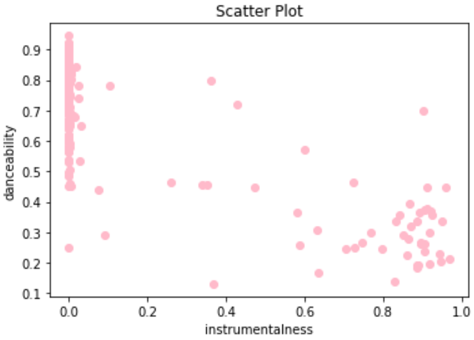


```
plt.scatter(sp['acousticness'],sp['energy'],color='green')
plt.xlabel("acousticness")
plt.ylabel("energy")
plt.title("Scatter Plot")
plt.show()
```

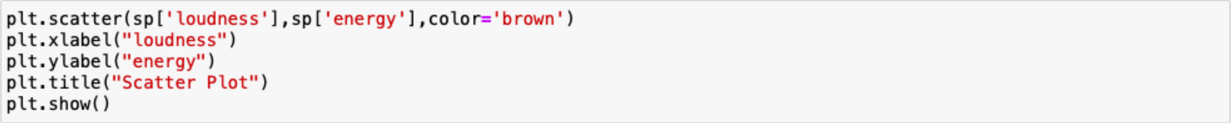
This scatterplot shows that if the instrumentalness is high then the danceability is low but if the instrumentalness is low then some songs are having high danceability.



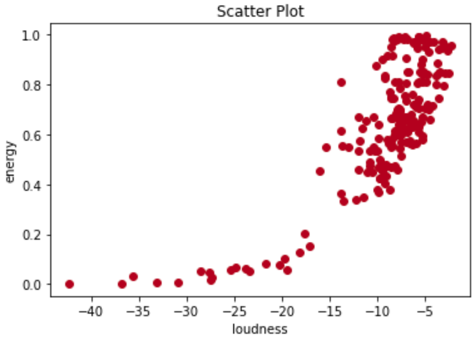
```
plt.scatter(sp['instrumentalness'],sp['danceability'],color='pink')
plt.xlabel("instrumentalness")
plt.ylabel("danceability")
plt.title("Scatter Plot")
plt.show()
```

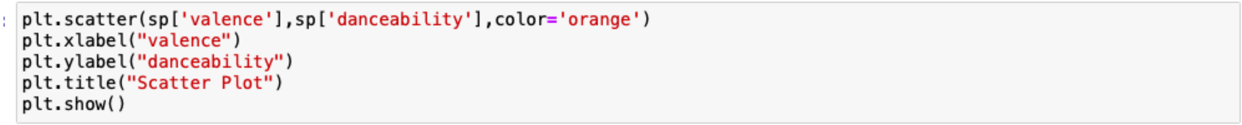


Here you can see a scatterplot that depicts how the loudness affects the energy of the songs. The plot shows that when the loudness is high the energy of the song is high but in some cases when the loudness is low the energy of the song is low. Therefore, there is a positive correlation between loudness and energy. If we imagine that a line is drawn across the points on this plot the line would be increasing in a parabolic line or linear line thus, it is positively correlated.

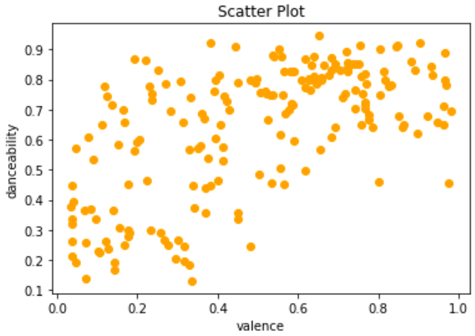


```
plt.scatter(sp['loudness'],sp['energy'],color='brown')
plt.xlabel("loudness")
plt.ylabel("energy")
plt.title("Scatter Plot")
plt.show()
```

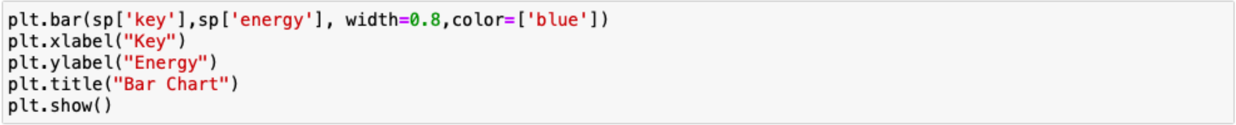




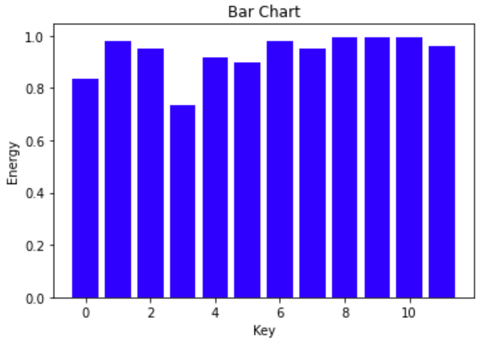
```
plt.scatter(sp['valence'],sp['danceability'],color='orange')
plt.xlabel("valence")
plt.ylabel("danceability")
plt.title("Scatter Plot")
plt.show()
```



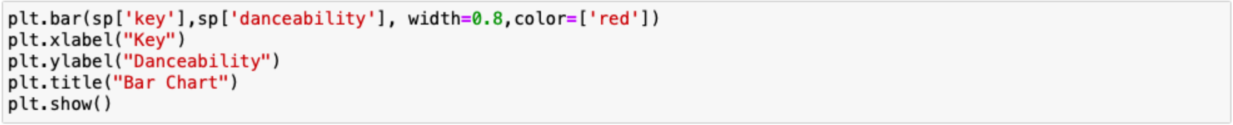
This bar chart shows how different keys have the most energy of the song



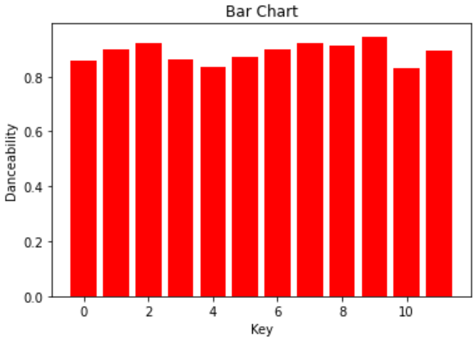
```
plt.bar(sp['key'],sp['energy'], width=0.8,color=['blue'])
plt.xlabel("Key")
plt.ylabel("Energy")
plt.title("Bar Chart")
plt.show()
```



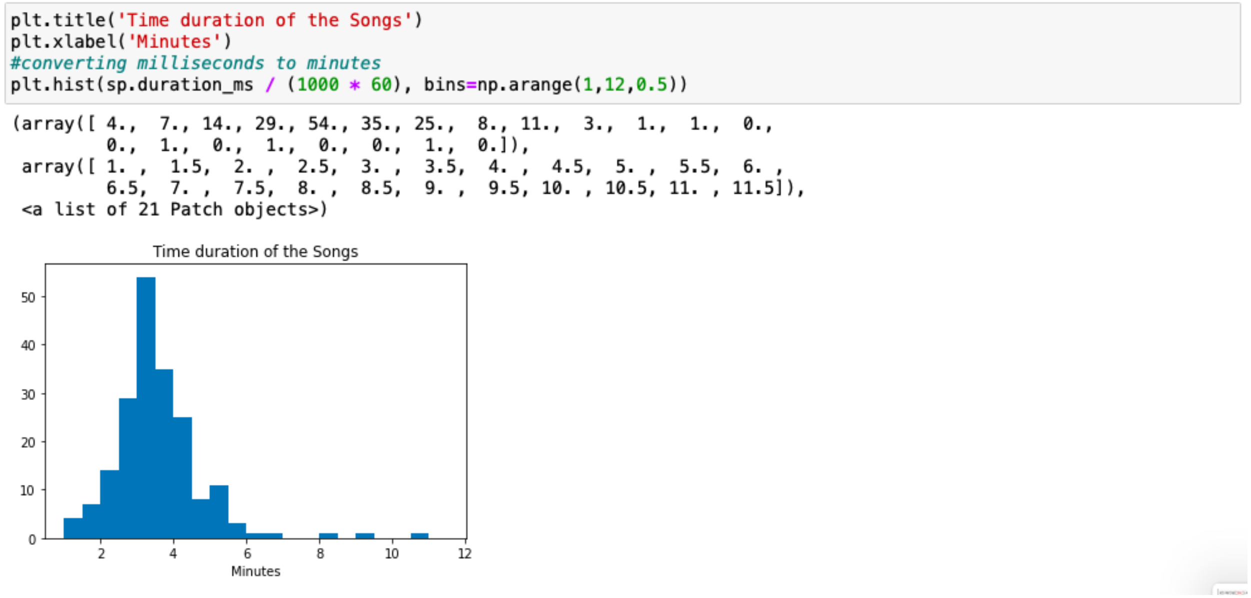
This bar chart shows how different keys have the most danceability of the song.



```
plt.bar(sp['key'],sp['danceability'], width=0.8,color=['red'])
plt.xlabel("Key")
plt.ylabel("Danceability")
plt.title("Bar Chart")
plt.show()
```



This is a histogram showcasing the time duration of the songs in minutes. Most songs have a time duration of 3 to 4 minutes.

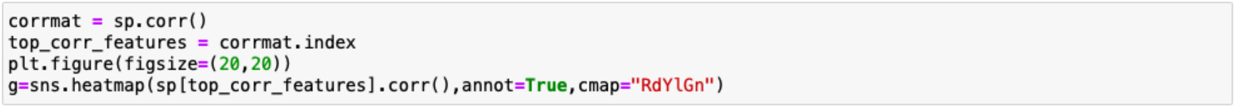


```
plt.title('Time duration of the Songs')
plt.xlabel('Minutes')
#converting milliseconds to minutes
plt.hist(sp.duration_ms / (1000 * 60), bins=np.arange(1,12,0.5))
```

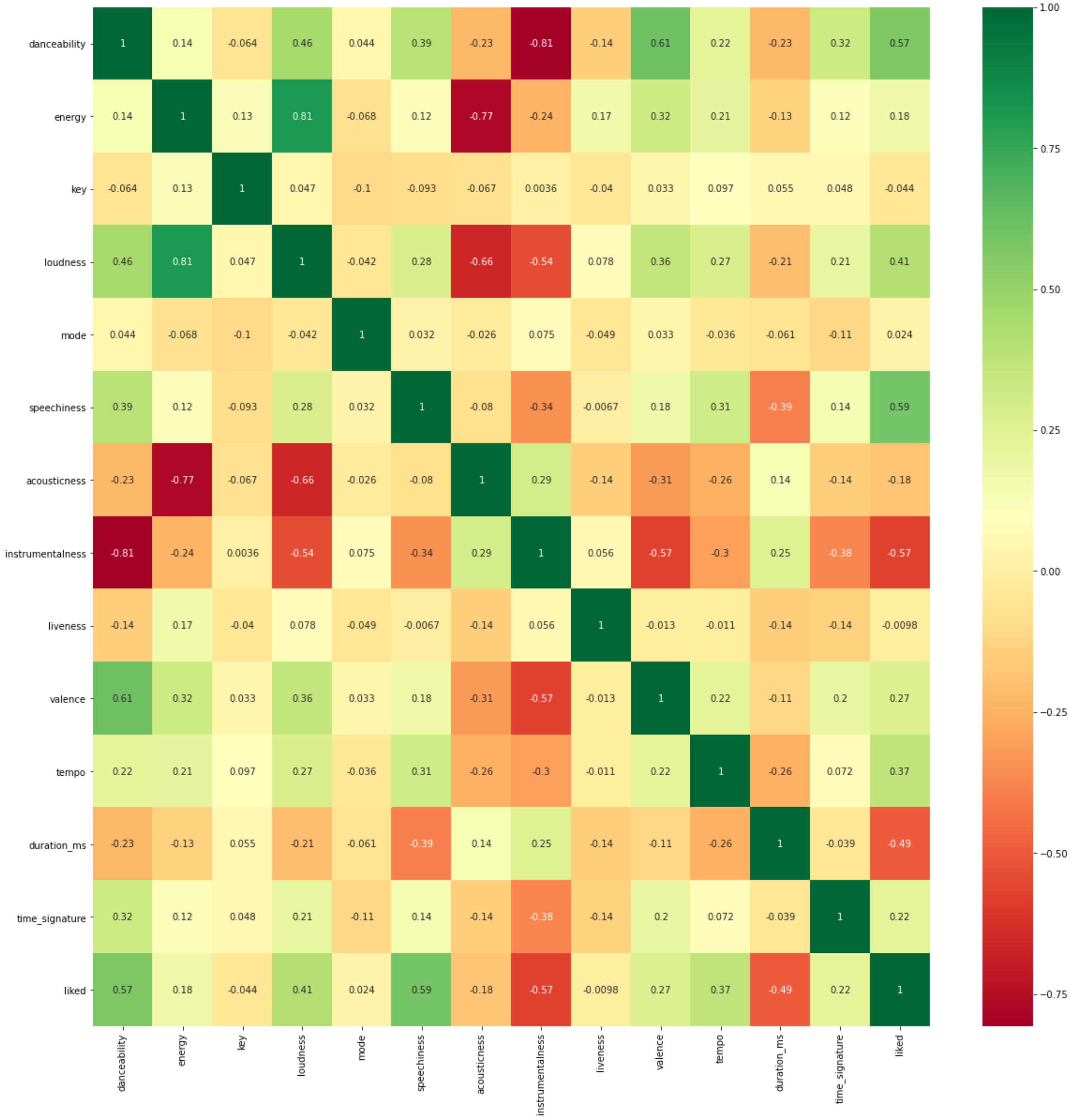
```
(array([ 4.,  7., 14., 29., 54., 35., 25.,  8., 11.,  3.,  1.,  1.,  0.,
         0.,  1.,  0.,  1.,  0.,  0.,  1.,  0.]),
 array([ 1. ,  1.5,  2. ,  2.5,  3. ,  3.5,  4. ,  4.5,  5. ,  5.5,  6. ,
         6.5,  7. ,  7.5,  8. ,  8.5,  9. ,  9.5, 10. , 10.5, 11. , 11.5]),
 <a list of 21 Patch objects>)
```

Below is a correlation matrix that shows the correlation of all columns with all other columns. From this matrix we can see that energy and loudness have a positive correlation of 0.81 and danceability and valence have a positive correlation of 0.61. Plus, danceability and liked have a positive correlation of 0.57. Speechiness and liked have a positive correlation of 0.59.

Instrumentalness and danceability have a negative correlation of -0.81. Acousticness and energy have a negative correlation of -0.77. Plus, acoustic ness and loudness have a negative correlation of -0.66.



```
cormat = sp.corr()
top_corr_features = cormat.index
plt.figure(figsize=(20,20))
g=sns.heatmap(sp[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```


danceability
energy
key
loudness
mode
speechiness
acousticness
instrumentalness
liveness
valence
tempo
duration_ms
time_signature
liked
1 0.14 -0.064 0.46 0.044 0.39 -0.23 -0.81 -0.14 0.61 0.22 -0.23 0.32 0.57
0.14 1 0.13 0.81 -0.068 0.12 -0.77 -0.24 0.17 0.32 0.21 -0.13 0.12 0.18
-0.064 0.13 1 0.047 -0.1 -0.093 -0.067 0.0036 -0.04 0.033 0.097 0.055 0.048 -0.044
0.46 0.81 0.047 1 -0.042 0.28 -0.66 -0.54 0.078 0.36 0.27 -0.21 0.21 0.41
0.044 -0.068 -0.1 -0.042 1 0.032 -0.026 0.075 -0.049 0.033 -0.036 -0.061 -0.11 0.024
0.39 0.12 -0.093 0.28 0.032 1 -0.08 -0.34 -0.0067 0.18 0.31 -0.39 0.14 0.59
-0.23 -0.77 -0.067 -0.66 -0.026 -0.08 1 0.29 -0.14 -0.31 -0.26 0.14 -0.14 -0.18
-0.81 -0.24 0.0036 -0.54 0.075 -0.34 0.29 1 0.056 -0.57 -0.3 0.25 -0.38 -0.57
-0.14 0.17 -0.04 0.078 -0.049 -0.0067 -0.14 0.056 1 -0.013 -0.011 -0.14 -0.14 -0.0098
0.61 0.32 0.033 0.36 0.033 0.18 -0.31 -0.57 -0.013 1 0.22 -0.11 0.2 0.27
0.22 0.21 0.097 0.27 -0.036 0.31 -0.26 -0.3 -0.011 0.22 1 -0.26 0.072 0.37
-0.23 -0.13 0.055 -0.21 -0.061 -0.39 0.14 0.25 -0.14 -0.11 -0.26 1 -0.039 -0.49
0.32 0.12 0.048 0.21 -0.11 0.14 -0.14 -0.38 -0.14 0.2 0.072 -0.039 1 0.22
0.57 0.18 -0.044 0.41 0.024 0.59 -0.18 -0.57 -0.0098 0.27 0.37 -0.49 0.22 1
danceability
energy
key
loudness
mode
speechiness
acousticness
instrumentalness
liveness
valence
tempo
duration_ms
time_signature
liked
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75