D A T A S C I E N C E
and
B U S I N E S S I N T E L L I G E N T

Author: Egi Safitri

Meeting 8

SIMULASI EDA with Python

Menggunakan Bahasa python

Task 1 : Define Library

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

Task 2 : Read Data (CSV)

```
df = pd.read_csv('data_by_year.csv')
```

Beberapa library yang biasa digunakan untuk mempermudah dalam melakukan explorasi data dengan python adalah pandas, matplotlib, seaborn dan numpy. Pandas dan numpy mempermudah dalam pemrosesan data numeric dan analisis data sedangkan matplotlib dan seaborn berfungsi dalam menciptakan visualisasi

Dataset diperoleh dari : https://www.kaggle.com/datasets/yamaerenay/spotify-dataset-19212020-160k-tracks

Head and Tail

Task 3 : Head and Tail Data

```
df.head()
```

	year	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo
0	1921	0.895823	0.425661	229911.914062	0.236784	0.322330	0.215814	-17.095437	0.077258	100.397758
1	1922	0.939236	0.480000	167904.541667	0.237026	0.440470	0.238647	-19.179958	0.115419	101.376139
2	1923	0.976329	0.568462	178356.301775	0.246936	0.401932	0.236656	-14.373882	0.098619	112.456598
3	1924	0.935575	0.548654	188461.649789	0.347033	0.583955	0.237875	-14.202304	0.090210	120.653359
4	1925	0.965422	0.571890	184130.699620	0.264373	0.408893	0.243094	-14.516707	0.115457	115.671715

```
df.tail()
```

	year	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo
95	2016	0.280290	0.599976	219400.763840	0.592877	0.074646	0.180198	-7.949913	0.107298	119.070344
96	2017	0.289916	0.612286	209343.613000	0.586739	0.098209	0.194218	-8.422697	0.111752	116.840278
97	2018	0.271941	0.664930	200919.119000	0.590591	0.035948	0.171781	-7.253666	0.128140	122.004325
98	2019	0.289298	0.644215	197733.133000	0.578796	0.076518	0.167161	-8.041738	0.124799	118.868163
99	2020	0.247374	0.673077	197114.662301	0.611914	0.039052	0.177048	-7.204024	0.143505	121.228704

Function **head()** dan **tail()** mengijinkan pengguna untuk melihat sampel data. Secara default elemen yang akan ditampilkan adalah 5 data awal dan 5 data akhir, tentunya fungsi ini sangat berguna saat anda berkerja dengan jumlah data yang banyak, sebab akan memberikan gambaran awal kepada anda seperti apa isi data yang sedang anda hadapi.

Seperti pada tabel disamping, sekilas data spotify yang kita memiliki seluruhnya bertipe data numerical, dimana kebanyakan variabel memuat nilai desimal.

Number of Row and Col

Task 4 : Total Number of Rows and Columns

```
df.shape
```

```
(100, 14)
```

Task 5 : Columns Name and Index

```
df.columns
```

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

```
df.index
```

```
RangeIndex(start=0, stop=100, step=1)
```

Dataset yang ada terdiri dari 100 baris dan 14 kolom, dimana nama variabel untuk ke 14 kolom tersebut dapat diketahui dengan menggunakan fungsi **df.columns** sedangkan index pada data berada pada range 0 sampai 100 dengan kenaikan atau step = 1.

Missing Value

Task 6 : Missing Value

```
df.isna().sum()
```

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

Jumlah Missing Value pada tiap variabel bernilai 0 atau tidak ada nilai yang hilang disetiap kolom, yang perlu diperhatikan adalah **isna()** digunakan untuk mendeteksi missing value pada data, apabila pada data terdapat nilai 0 maka tidak akan terdeteksi sebagai missing value, sebab missing value tidak sama dengan 0.

Data Info

Task 7 : Data Info

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   year              100 non-null    int64  
 1   acousticness      100 non-null    float64
 2   danceability      100 non-null    float64
 3   duration_ms       100 non-null    float64
 4   energy            100 non-null    float64
 5   instrumentalness  100 non-null    float64
 6   liveness          100 non-null    float64
 7   loudness          100 non-null    float64
 8   speechiness       100 non-null    float64
 9   tempo             100 non-null    float64
 10  valence           100 non-null    float64
 11  popularity        100 non-null    float64
 12  key               100 non-null    int64  
 13  mode              100 non-null    int64  
dtypes: float64(11), int64(3)
memory usage: 11.1 KB
```

Sebenarnya kita dapat menggunakan fungsi info() yang disediakan oleh pandas untuk merangkum beberapa fungsi yang telah dijelaskan sebelumnya, beberapa informasi mengenai data yang dapat kita peroleh dengan menerapkan fungsi **info()** adalah sebagai berikut :

1. Jumlah Baris dan Kolom data
2. Nama variabel kolom
3. Tipe data tiap kolom
4. Memori yang dipakai
5. Jumlah Missing Value

Statistika Deskriptif

Selanjutnya untuk memahami lebih dalam data yang kita miliki, kita akan melakukan eksplorasi data dengan menggunakan statistika deskriptif. Statistika deskriptif adalah metode-metode yang berkaitan dengan pengumpulan dan penyajian suatu gugus data sehingga memberikan informasi yang berguna, ada 3 cara yang umum digunakan untuk mendeskripsikan data, yaitu:

A. Measures of Central Tendency
   1. Mean
   2. Median
   3. Mode

B. Measures of Spread
   1. Rage
   2. Quartile dan Interquartile Range
   3. Variance
   4. Standar deviasi

C. Measures to Describe Shape of Distribution
   1. Skewness
   2. Kurtosis

Central Tendency

1. Measures of Central Tendency

Yaitu cara untuk mendeskripsikan posisi titik tengah dari distribusi frekuensi suatu kelompok.

a. **Mean**

Mean adalah jumlah dari seluruh data continous (numerical) dibagi dengan jumlah data yang ada. Mean adalah measure of central yang paling sering digunakan untuk data numerical

```
df.mean()
```

```
year                1970.500000
acousticness           0.552625
danceability           0.537485
duration_ms       226930.361382
energy                 0.455864
instrumentalness       0.192160
liveness               0.209335
loudness             -11.916648
speechiness            0.101247
tempo                116.026461
valence                0.536384
popularity            27.223173
key                    4.190000
mode                   1.000000
dtype: float64
```

Fungsi **mean()** yang disediakan oleh pandas akan mengembalikan nilai rata-rata.

Tabel di samping merupakan nilai rata-rata dari setiap kolom pada data, variabel duration_ms memiliki nilai rata-rata yang sangat besar dibandingkan variabel lain, hal ini terjadi karena variabel tersebut memuat data waktu musik dalam milisecond.

Median

Median adalah nilai tengah dari suatu data numerical yang diurutkan. jika jumlah data ganjil, maka nilai median tepat berada tengah dari data, sedangkan apabila genap maka nilai median berada diantara kedua nilai yang berada ditengah. Median lebih sering digunakan jika mean tidak mampu menjelaskan data kita dengan baik, sebagai contoh akan dibahas dalam bab skewness

```
df.median()
```

```
year                1970.500000
acousticness           0.452951
danceability           0.544746
duration_ms       234673.799962
energy                 0.498728
instrumentalness       0.124916
liveness               0.207946
loudness             -11.761679
speechiness            0.086237
tempo                117.597018
valence                0.548000
popularity            33.655500
key                    5.000000
mode                   1.000000
dtype: float64
```

Fungsi **median()** yang disediakan oleh pandas akan mengembalikan nilai tengah dari data

Modus/Mode

c. Mode

Mode adalah suatu data categorical atau data continous yang dapat dihitung dimana frekuensi dari data tersebut paling besar atau data yang paling sering muncul.

Fungsi **mode()** yang disediakan oleh pandas akan mengembalikan nilai yang sering muncul pada data

Mode digunakan jika data merupakan data categorical atau data numerical yang dianggap sebagai data categorical. Pada data spotify hampir seluruhnya merupakan data numerical, hanya variabel ‘year’ yang merupakan data numerical yang dapat dianggap sebagai data categorical namun variabel ‘year’ seluruhnya bersifat unique, jadi menggunakan mode pada data spotify tidak terlalu diperlukan.

Measures of Spread

2. Measures of Spread

Digunanakan untuk mendeskripsikan seberapa menyebar data yang ada.

a. **Range**

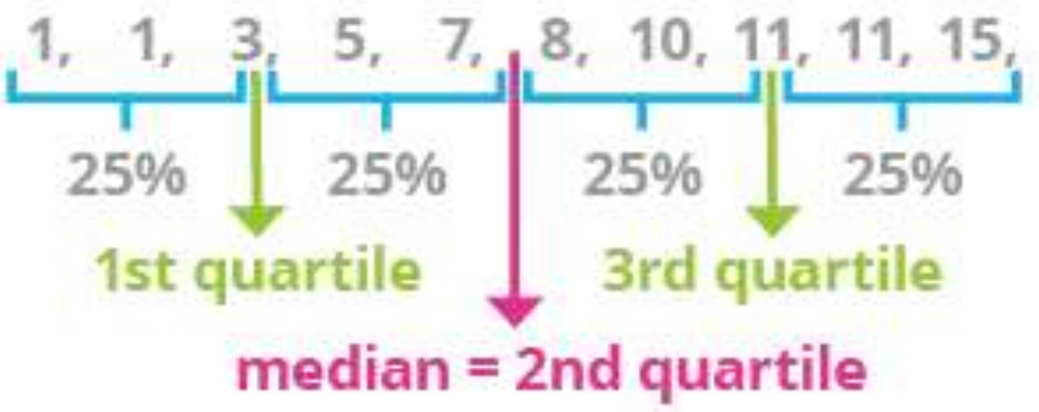
Range adalah perbedaan antara nilai terbesar dengan nilai terkecil pada data. range menjelaskan seberapa jauh data kita tersebar.

```
Range = df.max()-df.min()
Range
```

```
year                99.000000
acousticness         0.747617
danceability         0.269563
duration_ms      98483.191333
energy               0.489444
instrumentalness     0.548007
liveness             0.097187
loudness            12.485687
speechiness          0.434228
tempo               23.259060
valence              0.294951
popularity          69.516611
key                 10.000000
mode                 0.000000
dtype: float64
```

Quartile dan IQR

Quartile adalah nilai yang membagi data menjadi 4 bagian (25%). Terdapat 3 jenis Quartile yaitu Q1 yang merupakan nilai antara median dengan data terkecil, Q2 yang merupakan Median data dan Q3 adalah nilai antara median dengan data terbesar



Quartile dan IQR

Quartile sangat berguna karena berhubungan dengan konsep statistik lain yaitu Interquartile Range (IQR), IQR didapat dengan mengurangi Q3 dengan Q1. Dengan memanfaatkan nilai IQR anda dapat menemukan batas atas dan batas bawah data dengan perhitungan sebagai berikut

```
Batas Bawah = Q1 - 1.5 * IQR
Batas Atas = Q3 + 1.5 * IQR
```

Apabila ada data anda berada diluar rentang batas atas dan bawah maka nilai tersebut dikatakan nilai outlier

Variance

Variance digunakan untuk mengukur seberapa menyebar data yang ada dari mean-nya. Jika nilai variance dari data kita kecil, maka data tersebar dekat nilai mean-nya sedangkan jika nilai variance besar maka menunjukan data tersebar jauh dari mean-nya

```
df.var()
```

```
year                8.416667e+02
acousticness        7.958764e-02
danceability        2.707409e-03
duration_ms         6.386024e+08
energy              2.709956e-02
instrumentalness    1.680197e-02
liveness            3.326025e-04
loudness            9.808809e+00
speechiness         4.356401e-03
tempo               3.311003e+01
valence             3.409859e-03
popularity          4.411583e+02
key                 1.286253e+01
mode                0.000000e+00
dtype: float64
```

Standart Deviation

Standart deviasi menunjukan seberapa berbeda nilai pada data terhadap rata-ratanya, sama seperti variance, semakin besar nilai standart deviasi semakin jauh nilai dari mean-nya, yang membedaakan dengan variane adalah penggunaan standart deviasi dinilai lebih jelas dan intuitif, sebagai gambaran dalam bahasa sehari-hari, standart deviasi adalah nilai plus minus dari mean.

Untuk mendapatkan nilai standart deviasi kita hanya perlu melakukan akar kuadrat terhadap variance

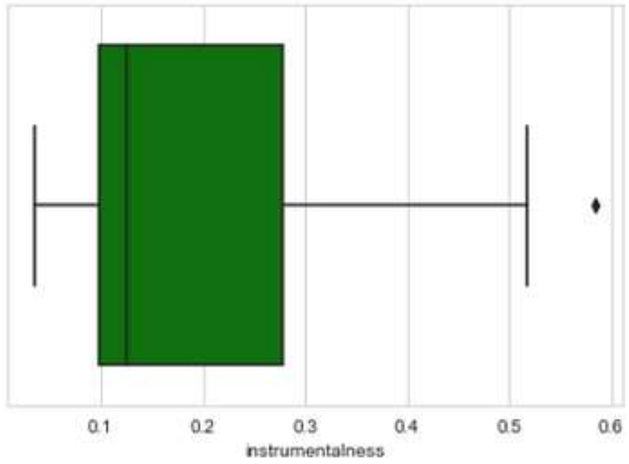
```
df.std()
```

```
year                29.011492
acousticness         0.282113
danceability         0.052033
duration_ms      25270.584548
energy               0.164619
instrumentalness     0.129622
liveness             0.018237
loudness             3.131902
speechiness          0.066003
tempo                5.754132
valence              0.058394
popularity          21.003768
key                  3.586436
mode                 0.000000
dtype: float64
```

Visualisasi EDA

Untuk mempermudah menganalisis persebaran data, kita dapat menggunakan visualisasi boxplot sebagai berikut

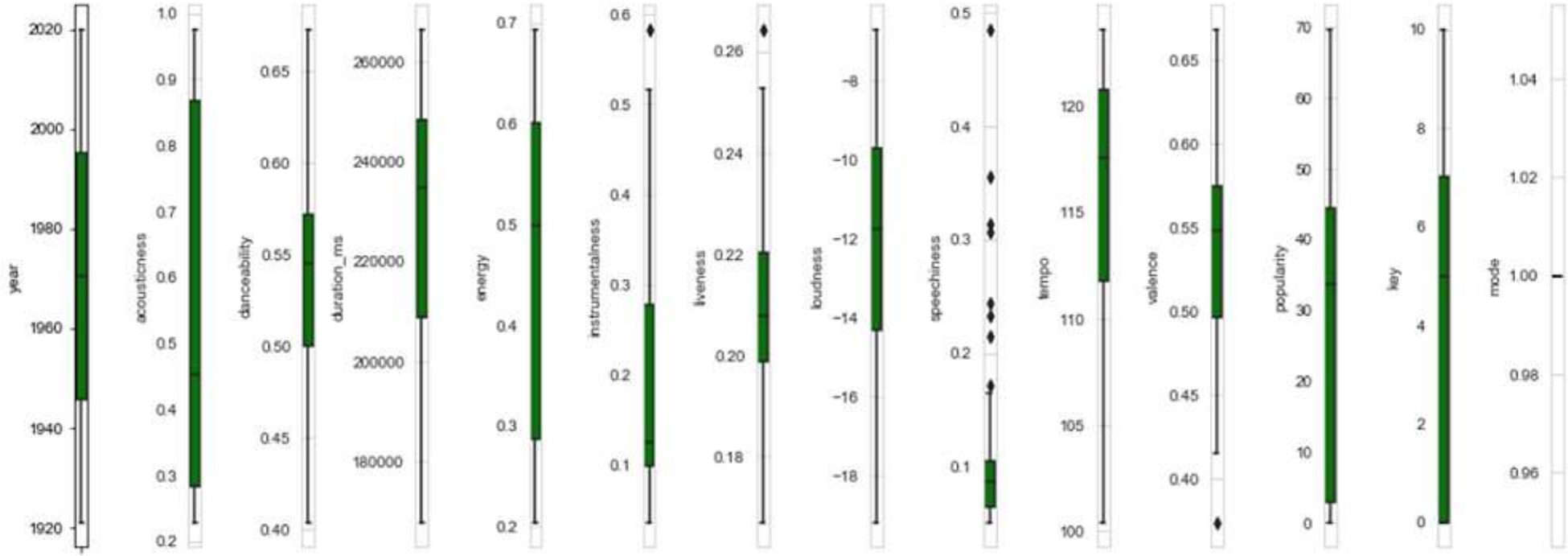
```
sns.boxplot(df['instrumentalness'], color = 'green')
plt.show()
```



Gambar disamping adalah hasil visualisasi variabel instrumentalness dalam bentuk boxplot, kelebihan dari menggunakan visualisasi ini adalah secara bersamaan dapat memberikan informasi mengenai, nilai maksimum, nilai minimum, range (max-min), Q1, Median(Q2), Q3 dan juga outlier.

Visualisasi EDA

Dengan bantuan library seaborn kita dapat memvisualisasikan seluruh variabel kedalam boxplot seperti gambar dibawah



Visualisasi EDA

Beberapa informasi yang dapat kita peroleh pada visualisasi sebelumnya adalah :

Variabel instrumentalness, liveness, speechiness dan valence masing-masing memiliki nilai yang outlier, untuk menghandle nilai outlier ini kembali kepada anda untuk apa data akan diproses selanjutnya.

Variabel Mode tidak memiliki range sebab hanya tidak memiliki nilai lain selain 1.

Visualisasi EDA

Selain menggunakan boxplot untuk melihat persebaran data, anda juga dapat menggunakan histogram untuk melihat persebaran data berdasarkan frequensi, sebagai berikut:

Visualisasi EDA

Untuk menjelaskan gambar sebelumnya, sumbu x adalah nilai pada variabel sedangkan sumbu y adalah frekuensi data, sebagai contoh pada variabel accousticness, sebagian besar data bernilai antara 0.2~0.3, hal ini ditandai dengan diantara nilai tersebut menghasilkan bar paling tinggi.

Sebagai informasi tambahan selain menggunakan visualisasi data untuk merangkum persebaran data, anda juga dapat menggunakan fungsi **describes()** yang akan memberikan informasi mengenai measures of tendency dan measures of spread data dalam sebuah table.

Visualisasi EDA

```
df.describe()
```

	year	acousticness	danceability	duration_ms	energy	instrumentalness
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	1970.500000	0.552625	0.537485	226930.361382	0.455864	0.192160
std	29.011492	0.282113	0.052033	25270.584548	0.164619	0.129622
min	1921.000000	0.228712	0.403515	167904.541667	0.204252	0.035948
25%	1945.750000	0.282738	0.500161	208882.913176	0.286154	0.098798
50%	1970.500000	0.452951	0.544746	234673.799962	0.498728	0.124916
75%	1995.250000	0.867022	0.571744	248234.122455	0.600955	0.277763
max	2020.000000	0.976329	0.673077	266387.733000	0.693696	0.583955

Seperti gambar diatas, dengan munggunakan **describes()** beberapa informasi yang dapat diperoleh yaitu

- Jumlah data pada tiap variabel
- Mean data pada tiap variabel
- std / standart deviasi pada tiap variabel
- min dan max tiap variabel
- Q1, Q2 dan Q3

Measures to Describe Shape of Distribution

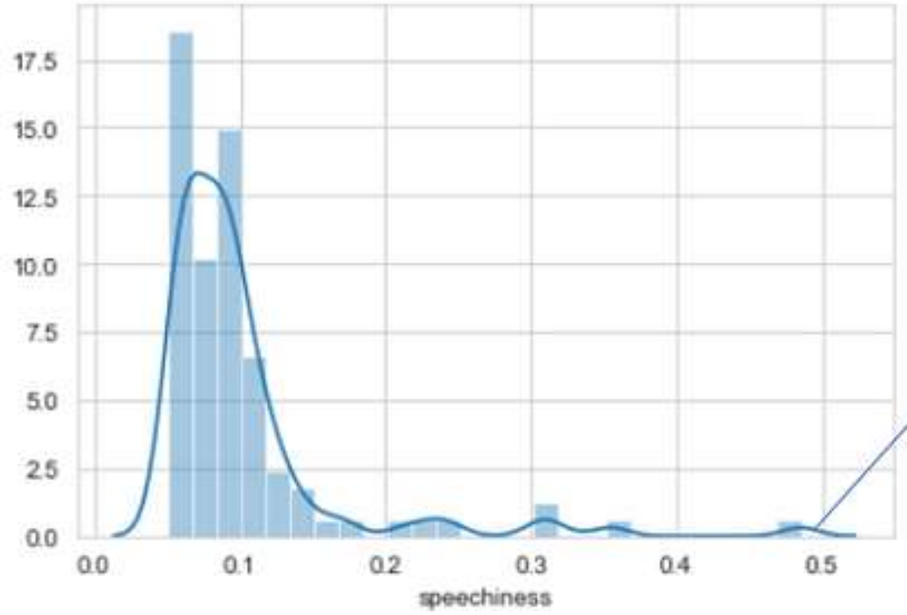
a. Skewness

adalah ukuran ketidaksimetrisan dalam distribusi nilai. skewness dapat bernilai positif, negatif dan nol. Skewness yang bernilai positif berarti ekor distribusi berada di sebeah kanan nilai terbanyak. skewness yang bernilai negatif berarti ekor distribusi berada disebelah kiri, ini menunjukan bahwa sebagian besar nilai berada disisi kiri kurva. sementara skewness bernilai nol berarti nilai terdistribusi secara simetris

Skewness dapat terjadi sebab terdapat nilai outlier pada data, sebagai contoh saya akan menggunakan variabel speechiness untuk melihat apakah terdapat nilai outlier atau tidak. anda dapat menggunakan seaborn untuk menghasilkan grafik skewness sebagai berikut :

Skewness

```
sns.distplot(df['speechiness'])
plt.show()
```

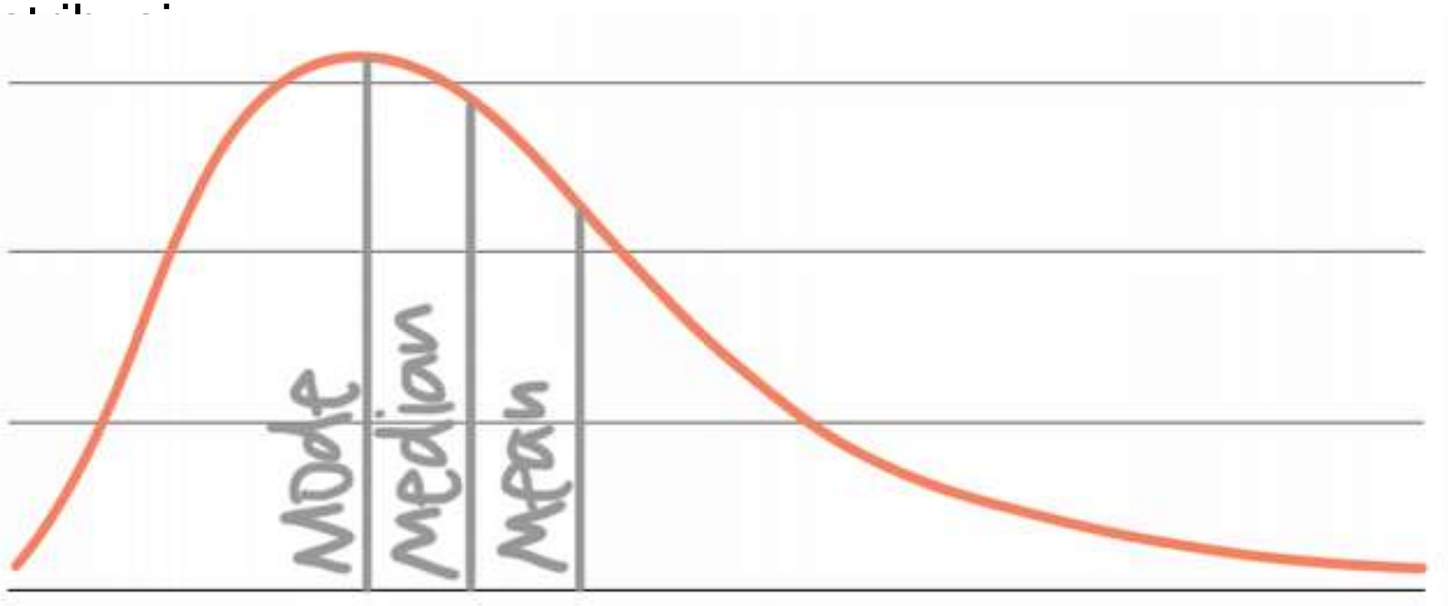


Ekor Distribusi Positif

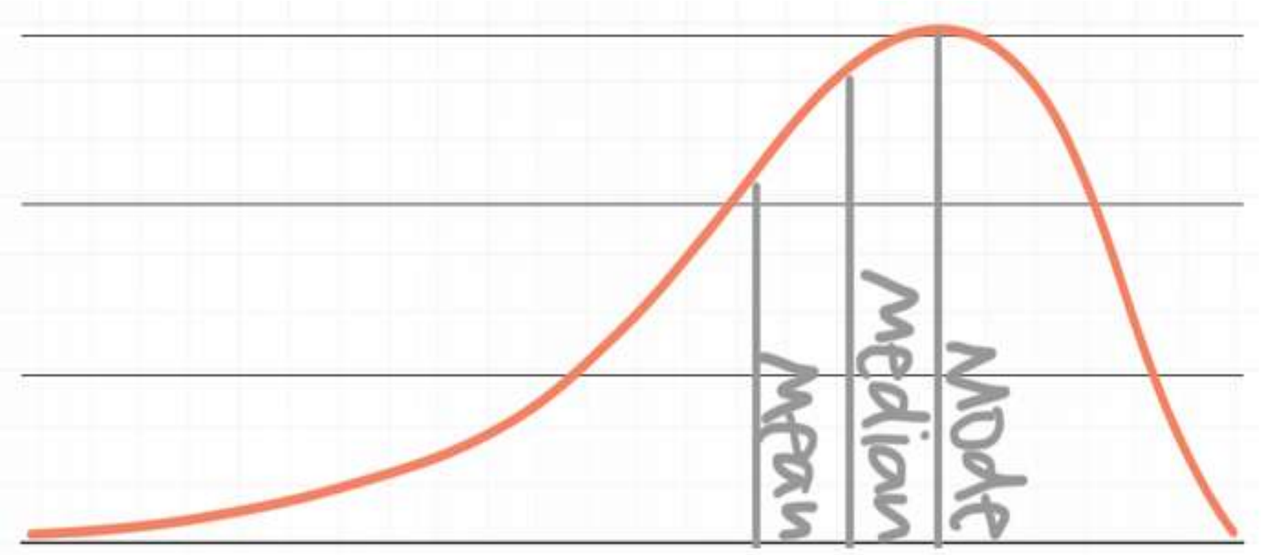
Variabel speechiness memiliki nilai outlier yang nilainya lebih besar dari mean-nya sehingga ekor distribusi terbentuk kekanan atau positif

Skewness

Sebagai informasi tambahan untuk mengukur central sebuah data yang skewness, lebih baik menggunakan median dari pada mean, sebab saat sebuah data skewness maka nilai mean-nya akan tertarik kearah ekor distribusi. Contoh : data Skewness Positif, Mean mendekati Ekor



Skewness



data Skewness Negatif, Mean mendekati Ekor Distribusi

Sedangkan untuk mengukur persebaran data yang skewness anda dapat menggunakan IQR (Interquartile Range) dibandingkan dengan Standart Deviasi

Kurtosis

Kurtosis adalah indikator untuk menunjukan derajat keruncingan, semakin besar nilai kurtosis maka kurva semakin runcing. Nilai referensi kurtosis adalah 3.

Skewness dan kurtois dapat menunjukan distribusi data. Kondisi ideal adalah saat data terdistribusi normal, yaitu skewness bernilai 0 dan kurtosis bernilai 3. Semakin jauh dari kondisi ideal berarti data tersebar semakin tidak merata.

Berikut adalah nilai skewness dan kurtosis tiap variabel yang ada dalam data spotify

	skewness	kurtosis
year	0.000000	-1.200000
acousticness	0.228083	-1.745978
danceability	0.032675	0.136101
duration_ms	-0.567899	-0.772094
energy	-0.125503	-1.691329
instrumentalness	1.028267	0.050488
liveness	0.229717	0.211713
loudness	0.066280	-0.889842
speechiness	3.446855	14.272313
tempo	-0.674085	-0.487244
valence	-0.399485	-0.080470
popularity	0.043466	-1.326863
key	-0.017320	-1.656646
mode	0.000000	0.000000

Dari nilai hasil perhitungan skewnees dan kurtosis diatas dapat disimpulkan bahwa data spotify belum terdistribusi secara normal

Korelasi

Selanjutnya adalah analisis korelasi yang menjelaskan ada atau tidaknya hubungan antar dua variabel. Nilai korelasi antara 2 variabel berada pada rentang -1 ~ +1 dimana semakin mendekati +1 maka korelasi antara kedua data semakin kuat, artinya jika terjadi kenaikan nilai pada variabel 1, maka variabel ke-2 juga akan bertambah. Berbeda apabila nilai korelasi dari 2 variabel semakin kecil dan mendekati -1, artinya saat variabel 1 mengalami kenaikan, variabel ke-2 justru mengalami penurunan.

Korelasi

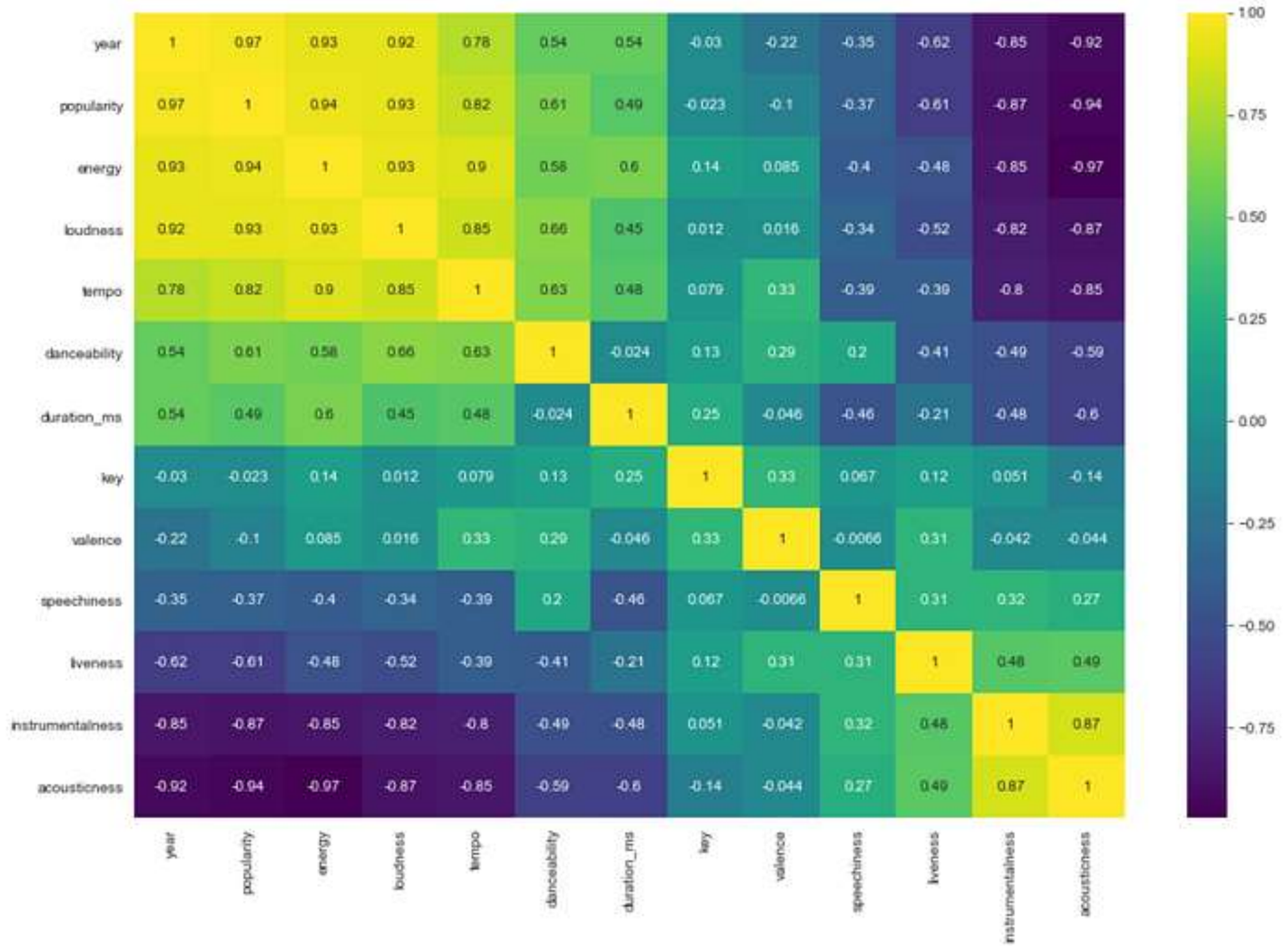
Dengan memanfaatkan fungsi **corr()** yang disediakan pandas maka kita dapat mengetahui korelasi antar variabel sebagai berikut :

```
df.corr()
```

	year	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
year	1.000000	-0.922472	0.536385	0.536314	0.929389	-0.853655	-0.616820	0.923169	-0.353955	0.778410	-0.217423
acousticness	-0.922472	1.000000	-0.592065	-0.598229	-0.971622	0.869827	0.494254	-0.874271	0.274185	-0.854827	-0.043512
danceability	0.536385	-0.592065	1.000000	-0.024408	0.577360	-0.490333	-0.410532	0.656247	0.204205	0.633426	0.289993
duration_ms	0.536314	-0.598229	-0.024408	1.000000	0.600383	-0.484800	-0.213847	0.448097	-0.459290	0.478158	-0.046297
energy	0.929389	-0.971622	0.577360	0.600383	1.000000	-0.851709	-0.477962	0.933734	-0.403722	0.897127	0.085447
instrumentalness	-0.853655	0.869827	-0.490333	-0.484800	-0.851709	1.000000	0.480734	-0.819140	0.316847	-0.796840	-0.041861
liveness	-0.616820	0.494254	-0.410532	-0.213847	-0.477962	0.480734	1.000000	-0.515343	0.307722	-0.388700	0.308525
loudness	0.923169	-0.874271	0.656247	0.448097	0.933734	-0.819140	-0.515343	1.000000	-0.340797	0.850734	0.015680
speechiness	-0.353955	0.274185	0.204205	-0.459290	-0.403722	0.316847	0.307722	-0.340797	1.000000	-0.390921	-0.006619
tempo	0.778410	-0.854827	0.633426	0.478158	0.897127	-0.796840	-0.388700	0.850734	-0.390921	1.000000	0.325831
valence	-0.217423	-0.043512	0.289993	-0.046297	0.085447	-0.041861	0.308525	0.015680	-0.006619	0.325831	1.000000
popularity	0.973089	-0.940926	0.610879	0.487069	0.944727	-0.869664	-0.612549	0.929296	-0.369584	0.823671	-0.100054
key	-0.029561	-0.136749	0.129861	0.254648	0.137373	0.050664	0.124081	0.012451	0.067035	0.078629	0.328044
mode	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Korelasi

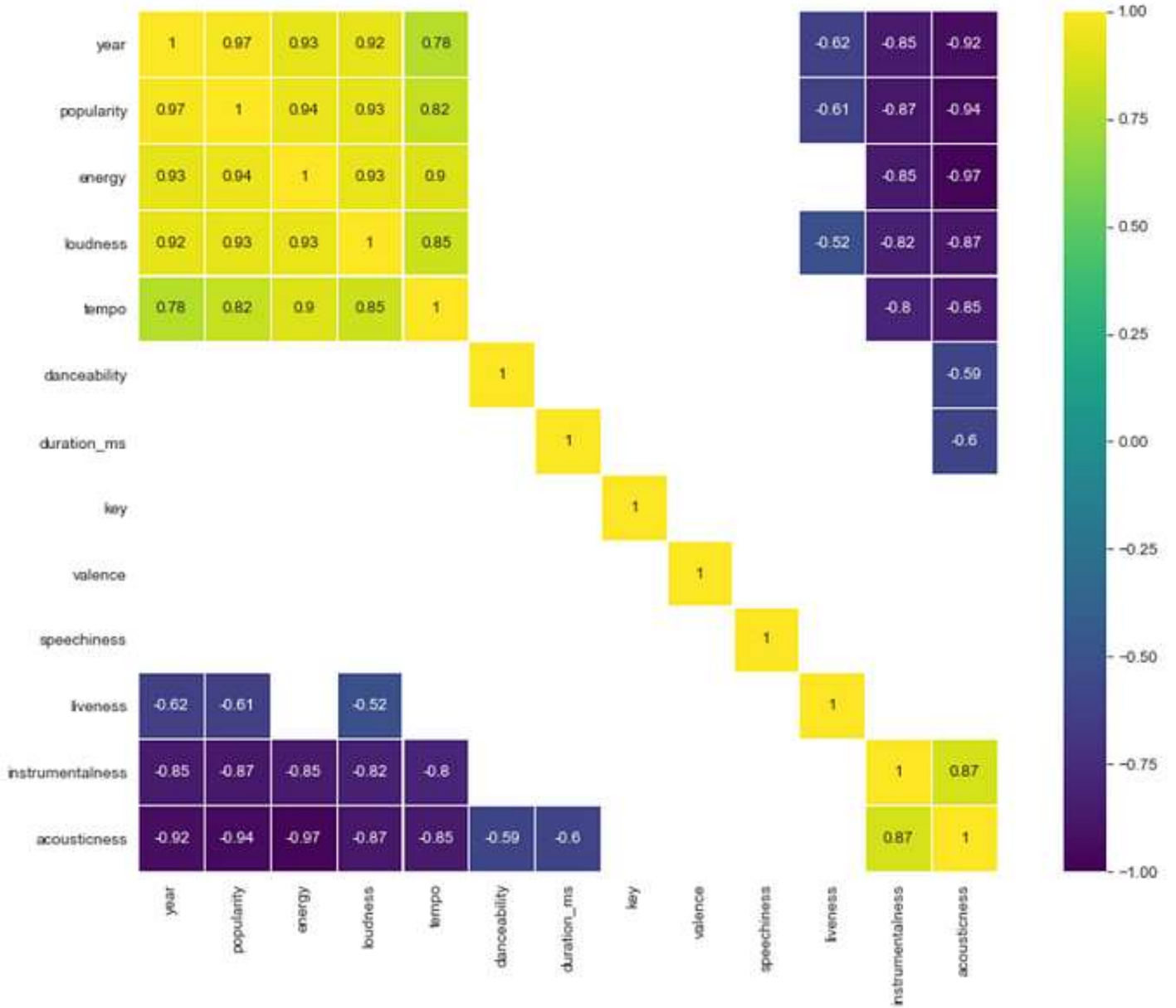
Untuk mempermudah dalam analisis, kita juga dapat menggunakan visualisasi heatmap yang disediakan oleh seaborn untuk menampilkan data korelasi antar variabel.



Dari heatmap diatas, semakin cerah warna pada box menunjukan semakin tinggi nilai korelasi antara 2 variabel.

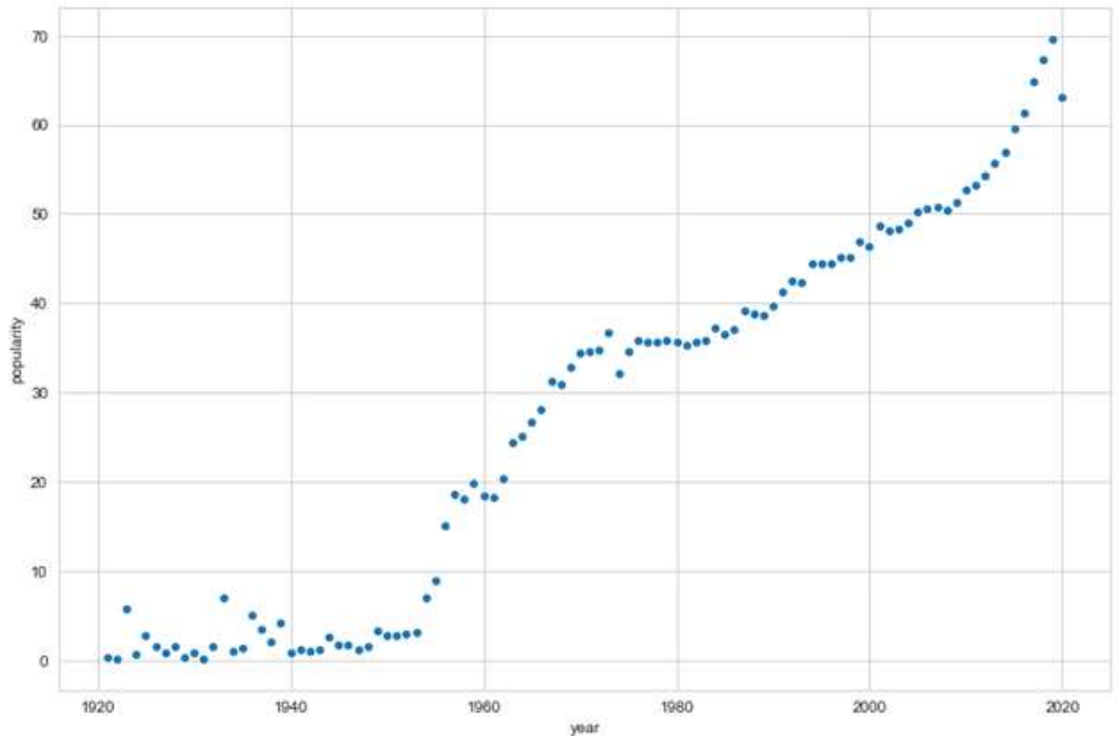
Korelasi

variabel yang memiliki korelasi paling tinggi adalah variabel year dengan popularity, hal ini berarti semakin bertambahnya tahun (year) maka nilai popularity juga akan bertambah. Untuk membuktikan hal tersebut kita dapat menggunakan scatter plot untuk melihat hubungan antar variabel year dan popularity.



Scatter Plot

```
plt.figure(figsize = (12,8))
sns.scatterplot(y = df['popularity'], x = df['year'])
plt.show()
```

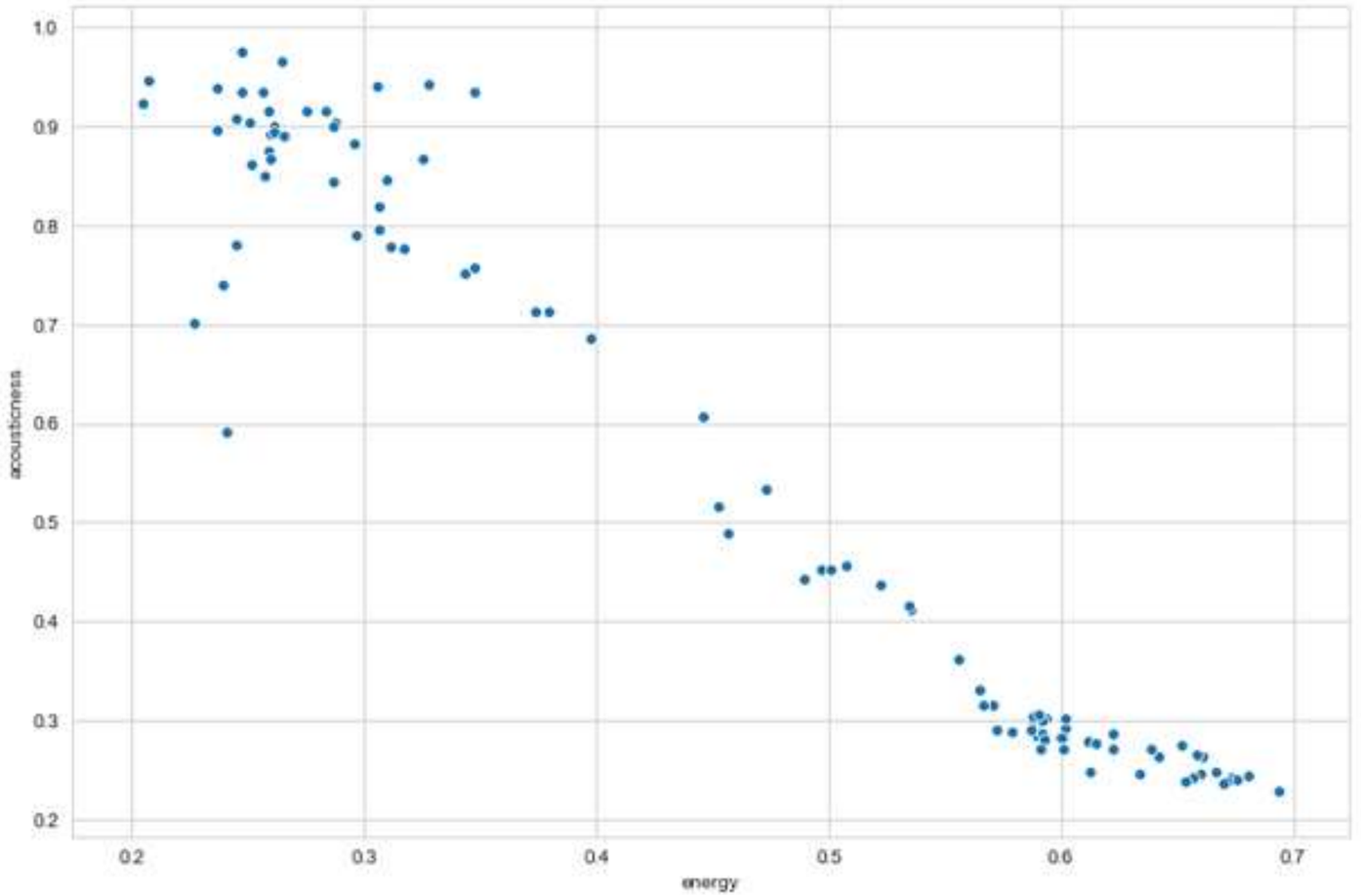


Seperti yang ditunjukan diagram scatter diatas, semakin bertambah nilai variabel year (sumbu x) maka diikuti oleh penambahan nilai variabel popularity (sumbu y).

Scatter Plot

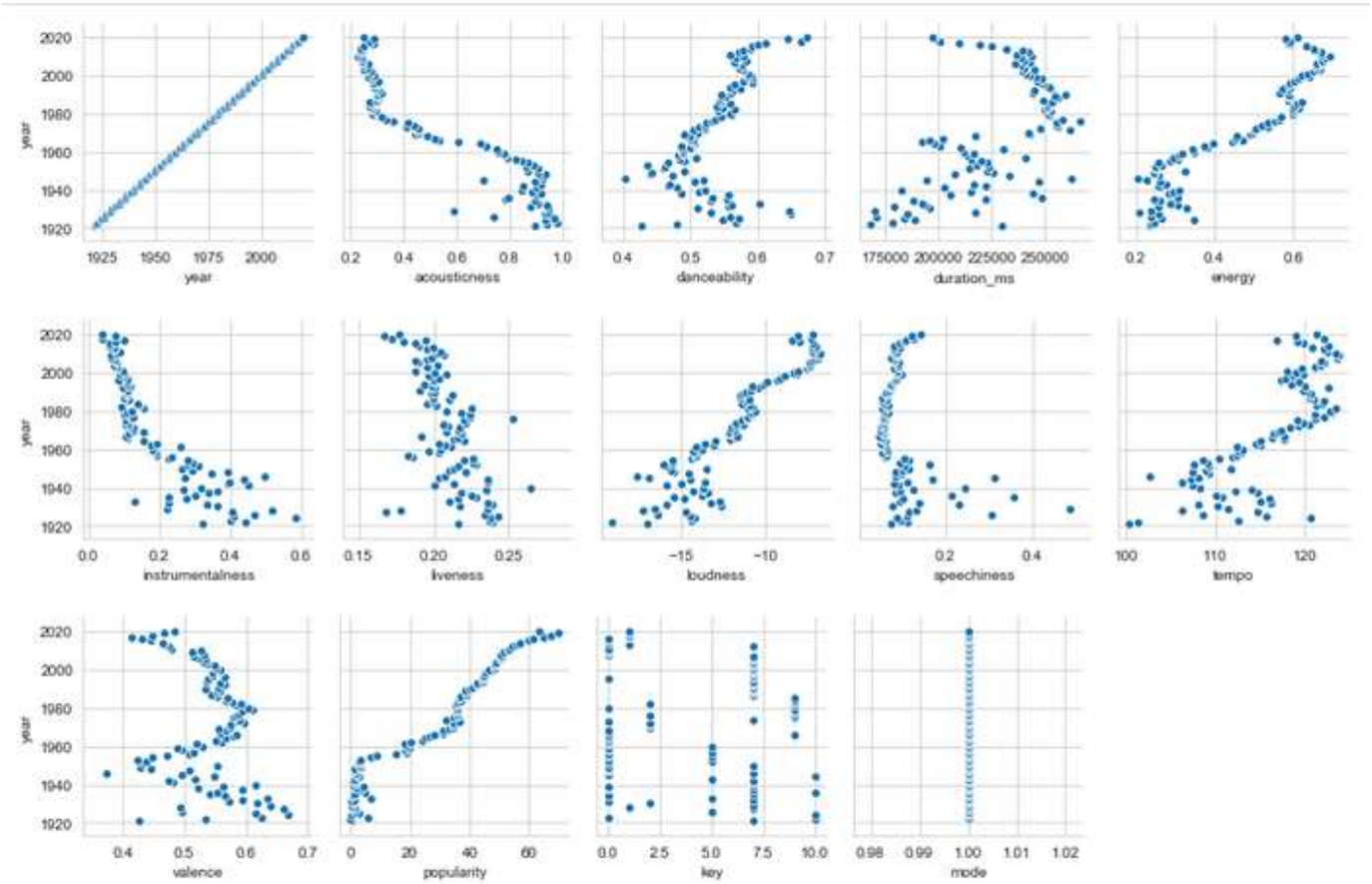
Berbeda dengan kedua variabel diatas, variabel energy dengan acousticness menunjukan nilai korelasi yang sangat kecil -0.97, hal ini menunjukan nilai kedua variabel tersebut berbanding terbalik seperti yang ditunjukan diagram scatter di samping ini

```
plt.figure(figsize = (12,8))
sns.scatterplot(y = df['acousticness'], x = df['energy'])
plt.show()
```



Scatter Plot

Kita juga dapat melihat korelasi antara seluruh variabel dengan 1 variabel, sebagai contoh, kita ingin melihat bagaimana tren keseluruhan variabel dari tahun ketahun, untuk menjawab pertanyaan tersebut anda dapat menggunakan scatter plot seperti gambar di samping



Berdasarkan diagram diatas terlihat bahwa sebagian besar variabel akan bertambah nilainya seiring dengan bertambahnya tahun, dan variabel valence serta acousticness yang menunjukan pengurangan nilai seiring dengan bertambahnya tahun.

Kesimpulan

Dalam melakukan Exploratory Data Analysis banyak hal mendasar terkait data yang dapat kita ketahui, dengan mengetahui isi dari data secara detail akan membantu tahapan selanjutnya dalam data mining.


THANK
YOU