PROJET EN DATASET

14.12.2020 - 18.01.2020

Hajar LACHHEB 1AC-A

Vue d'ensemble

Mon projet a pour but de faire une analyse des résultats obtenues et liées au TOP 50 des morceaux musicaux au niveau de la plateforme SPOTIFY pour l'année 2019.

Le choix des morceaux et leur classement se fait en prenant en compte plusieurs critères et c'est ce qu'on prendrait en compte aussi pour effectuer une analyse plus ou moins complète de ce dataset.

L'ensemble du dataset se trouve au niveau du site officiel de SPOTIFY ou au niveau de KAGGLE.

Mon Dataset rassemble alors 50 morceaux avec 13 variables chacun.

Objectifs

- Chercher la dataset qui se trouve sous format EXCEL.
- Trouver l'environnement adéquat afin de réaliser l'analyse voulue.
- Se familiariser avec les différentes bibliothèques.
- Réaliser l'analyse.

OUTILS INFORMATIQUES UTILISÉS :

❖ Le langage de programmation **python** ;

❖ Les librairies disponibles sur **Anaconda Navigateur** ;

· **Numpy** ; *Algèbre linéaire*

· **Pandas** ; Traitement *de la Data (Data Sets Fichier CSV)*

· **Seaborn ;** *visualisation statique de la Data*

· **Matplotlib. Pyplot ;** *Représentation graphique da la data*

❖ Environnement de développement ; **SPIDER , JUPITER NOTEBOOK ;**

ÉTAPES DE MON ANALYSE :

- Chercher et trouver la dataset sur laquelle il faut travailler.

  J'ai finit par choisir la DATASET lié au TOP 50 des morceaux musicaux au sein de SPOTIFY. Et ceci en prenant en compte 14 critères différents.

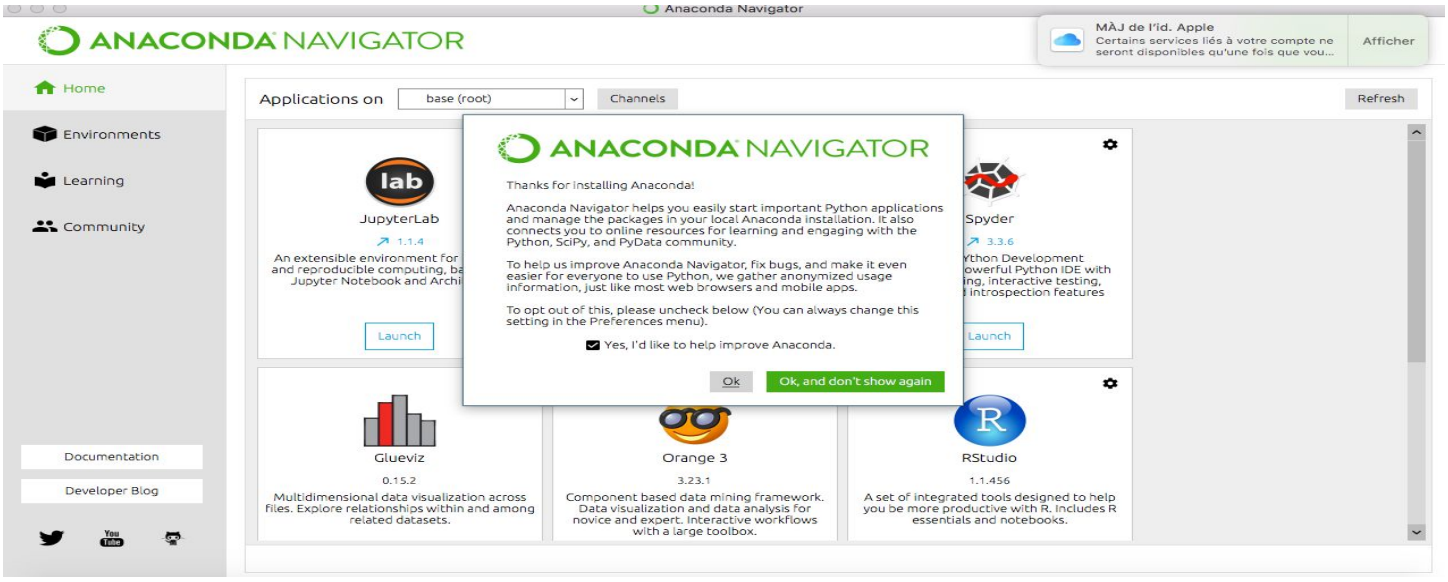
  https://www.kaggle.com/leonardopena/top50spotify2019#top50.csv

	Unnamed: 0	track_name	artist_name	Genre	beats_per_minute	Energy	Danceability
0	1	Señorita	Shawn Mendes	canadian pop	117	55	76
1	2	China	Anuel AA	reggaeton flow	105	81	79
2	3	boyfriend (with Social House)	Ariana Grande	dance pop	190	80	40
3	4	Beautiful People (feat. Khalid)	Ed Sheeran	pop	93	65	64
4	5	Goodbyes (Feat. Young Thug)	Post Malone	dfw rap	150	65	58
5	6	I Don't Care (with Justin Bieber)	Ed Sheeran	pop	102	68	80
6	7	Ransom	Lil Tecca	trap music	180	64	75
7	8	How Do You Sleep?	Sam Smith	pop	111	68	48
8	9	Old Town Road - Remix	Lil Nas X	country rap	136	62	88
9	10	bad guy	Billie Eilish	electropop	135	43	70
10	11	Callaita	Bad Bunny	reggaeton	176	62	61
11	12	Loco Contigo (feat. J. Balvin & Tyga)	DJ Snake	dance pop	96	71	82
12	13	Someone You Loved	Lewis Capaldi	pop	110	41	50
13	14	Otro Trago - Remix	Sech	panamanian pop	176	79	73
14	15	Money In The Grave (Drake ft. Rick Ross)	Drake	canadian hip hop	101	50	83

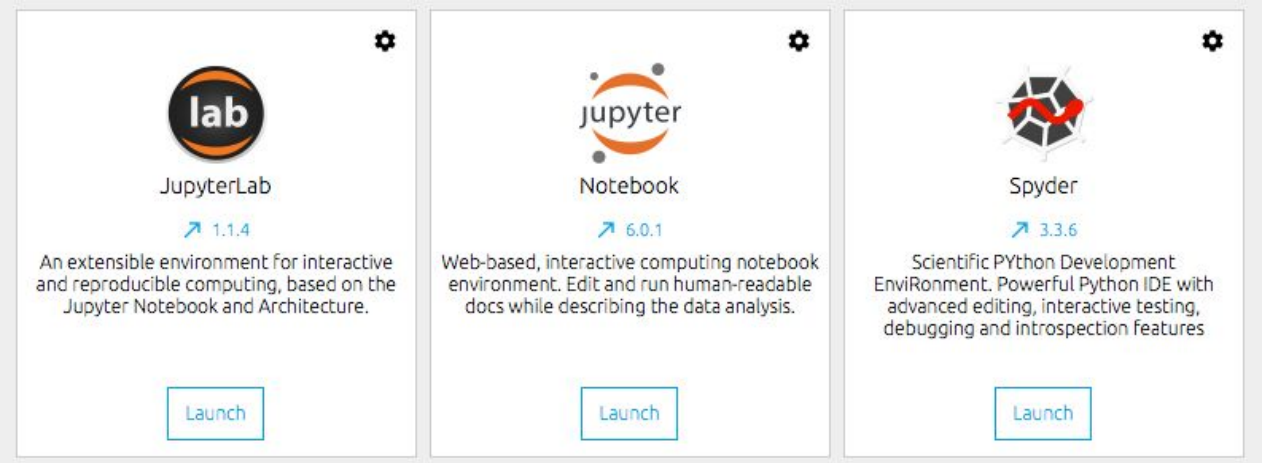
#C'est un extrait , elle contient en fait , 50 titres musicaux et 14 critères.

■ Chercher l'environnement adéquat pour effectuer l'analyse.

1. Telecharger ANACONDA - L'environnement d'ANACONDA.

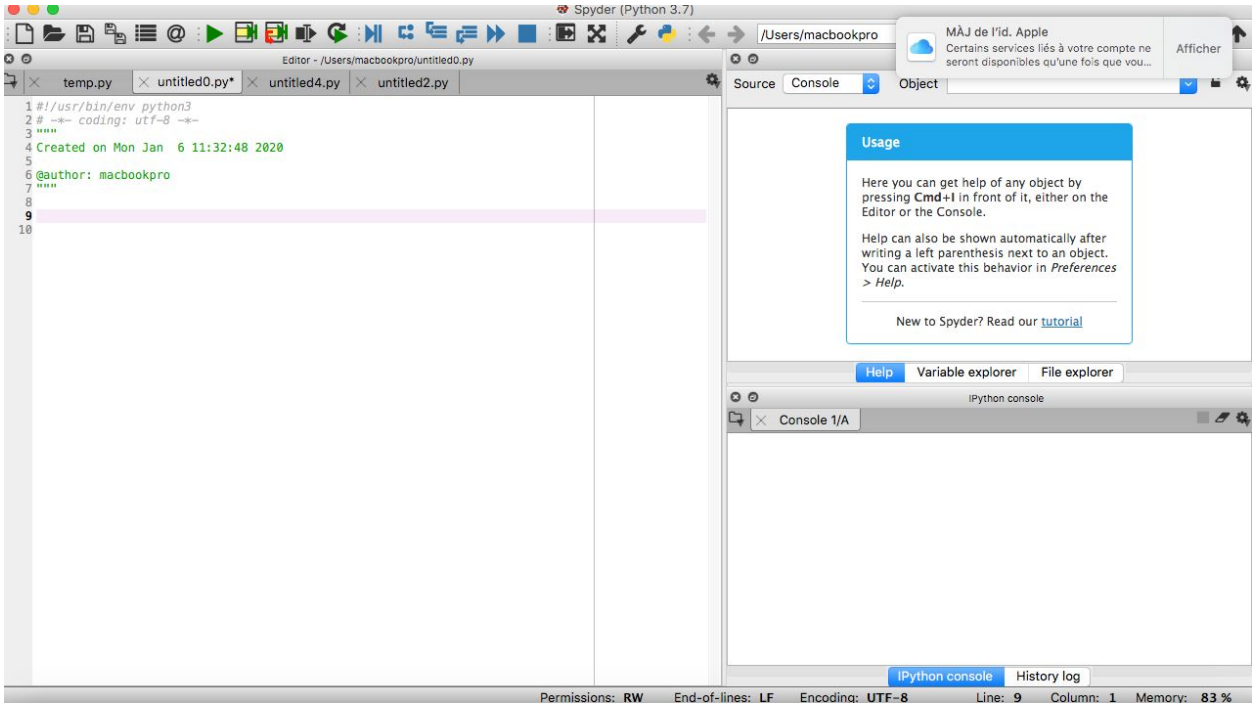


2. Trouver le bon environnement de développement.



3. Environment choisi , il faut télécharger et mettre a jour ce dernier.

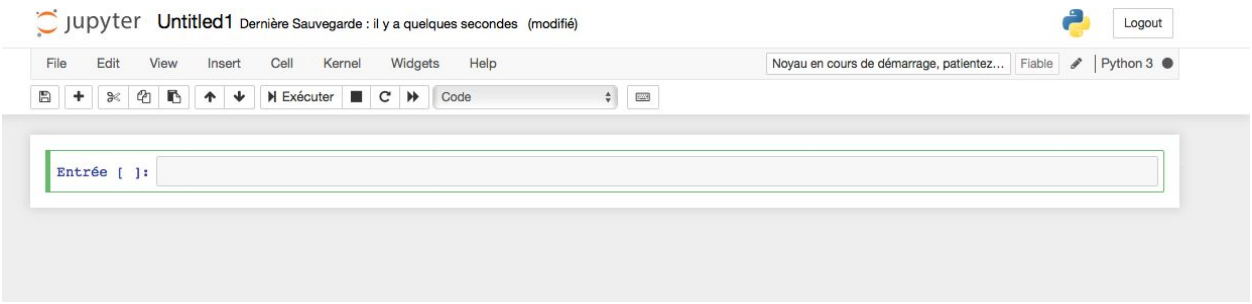
4. Se familiariser avec l'environnement surtout que le langage de programmation utilisé est PYTHON.



5. J'ai eu l'obligation de travailler aussi avec de nouvelle bibliothèque tel que panda , sns et plotly (là ou j'ai eu le problème avec SPYDER). #pip install NOM DU MODUL

```
Successfully built retrying
Installing collected packages: retrying, plotly
conda install -c plotly plotly==4.1.0
Successfully installed plotly-4.1.0 retrying-1.3.3
(base) mbp-de-macbook:~ macbookpro$ conda install -c plotly plotly==4.1.0

Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/macbookpro/opt/anaconda3

  added / updated specs:
    - plotly==4.1.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    conda-4.8.1                |           py37_0         2.8 MB
    plotly-4.1.0               |             py_0         4.0 MB  plotly
    retrying-1.3.3             |           py37_2          16 KB
```

6. J'ai pris le temps après de télécharger JUPYTER NOTEBOOK , afin de l'utiliser et ceci à travers ANACONDA.



■ Commencer l'analyse du dataset choisi.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 25 11:19:48 2019

@author: macbookpro
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pylab import rcParams
from pandas.plotting import scatter_matrix
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
data = pd.read_csv("/Users/macbookpro/Desktop/top50spotify2019/top50.csv",encoding = "ISO-8859-1")
data.head()
print(data.head())
data=data.rename(columns={"Loudness..dB..": "Loudness", "Acousticness..": "Acousticness", "Speechiness.":"Speechiness","Valence.":"Valence"
 "Length.":"Length"})
data.rename(columns={'Artist.Name':'artist_name','Track.Name':'Track-name'},inplace=True)
data = data.drop('Unnamed: 0' , axis=1)
print(data)
```

#Dans le programme ci dessus , on intègre la dataset qu'on a telecharger sous format CSV , et on utilisant surtout la bibliothèque PANDA as pd mais aussi sans oublier la norme ISO pour nettoyer plus ou moins le texte (Caractères).

#Data head va me permettre aussi de lire les informations qu'on trouve au niveau du DATASET. A cette étape aussi , je vais inclure les IMPORT de toute les bibliothèques que je vais utiliser. J'ai pris aussi le temps de corriger des erreurs que j'ai trouvé de mon DATASET , c'est pour cette raison que j'ai changé les noms des colonnes et supprimer une colonne que je considère comme INUTILE.

#Le print (data.head()) m'a permis de retourner en output la dataset tel qu'elle es. Et ça me rassure.

```
[5 rows x 14 columns]
                                        Track.Name  ... Popularity
0                                         Señorita  ...         79
1                                            China  ...         92
2                      boyfriend (with Social House)  ...         85
3                    Beautiful People (feat. Khalid)  ...         86
4                        Goodbyes (Feat. Young Thug)  ...         94
5                  I Don't Care (with Justin Bieber)  ...         84
6                                           Ransom  ...         92
7                                  How Do You Sleep?  ...         90
8                           Old Town Road - Remix  ...         87
9                                          bad guy  ...         95
10                                        Callaita  ...         93
11            Loco Contigo (feat. J. Balvin & Tyga)  ...         86
12                               Someone You Loved  ...         88
13                             Otro Trago - Remix  ...         87
14          Money In The Grave (Drake ft. Rick Ross)  ...         92
15                        No Guidance (feat. Drake)  ...         82
16                                       LA CANCIÓN  ...         90
17      Sunflower - Spider-Man: Into the Spider-Verse  ...         91
18                                          Lalala  ...         88
19                                     Truth Hurts  ...         91
20                               Piece Of Your Heart  ...         91
21                                          Panini  ...         91
22                               No Me Conoce - Remix  ...         83
23                                    Soltera - Remix  ...         91
24                     bad guy (with Justin Bieber)  ...         89
25                               If I Can't Have You  ...         70
26                                     Dance Monkey  ...         83
27                                         It's You  ...         89
28                                       Con Calma  ...         91
29                                     QUE PRETENDES  ...         89
30                                         Takeaway  ...         84
31                                          7 rings  ...         89
32                                 0.958333333333333  ...         89
33          The London (feat. J. Cole & Travis Scott)  ...         89
34                               Never Really Over  ...         89
```

Ce que j'ai obtenue comme comme output.

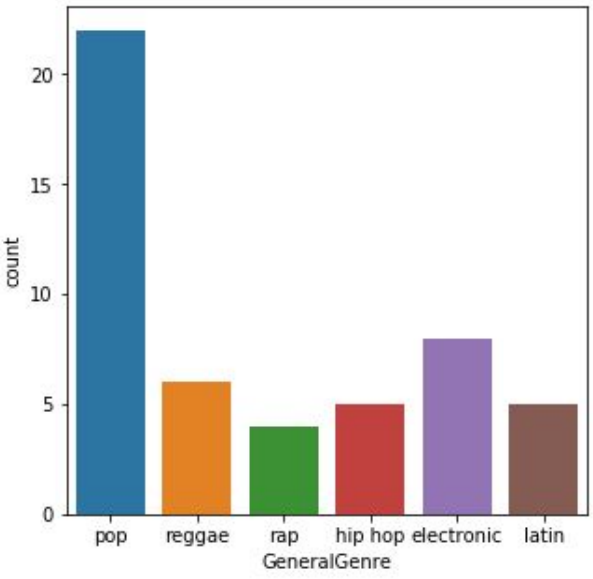
```
data['GeneralGenre']=['hip hop' if each =='atl hip hop'
                      else 'hip hop' if each =='canadian hip hop'
                      else 'hip hop' if each == 'trap music'
                      else 'pop' if each == 'australian pop'
                      else 'pop' if each == 'boy band'
                      else 'pop' if each == 'canadian pop'
                      else 'pop' if each == 'dance pop'
                      else 'pop' if each == 'panamanian pop'
                      else 'pop' if each == 'pop'
                      else 'pop' if each == 'pop house'
                      else 'electronic' if each == 'big room'
                      else 'electronic' if each == 'brostep'
                      else 'electronic' if each == 'edm'
                      else 'electronic' if each == 'electropop'
                      else 'rap' if each == 'country rap'
                      else 'rap' if each == 'dfw rap'
                      else 'escape room' if each == 'hip hop'
                      else 'latin' if each == 'latin'
                      else 'r&b' if each == 'r&n en espanol'
                      else 'reggae' for each in data['Genre']]
print(data.groupby('GeneralGenre').size())
print(data.groupby('artist_name').size())
plt.figure(figsize=(5,5))
sns.countplot(x="GeneralGenre", data=data)
plt.figure(figsize=(5,5))


sns.distplot(data["Popularity"])
plt.figure(figsize=(5,5))

sns.distplot(data["Danceability"])
plt.figure(figsize=(5,5))

sns.scatterplot(x=data.Danceability,y=data.Popularity,data=data)
plt.figure(figsize=(5,5))
```

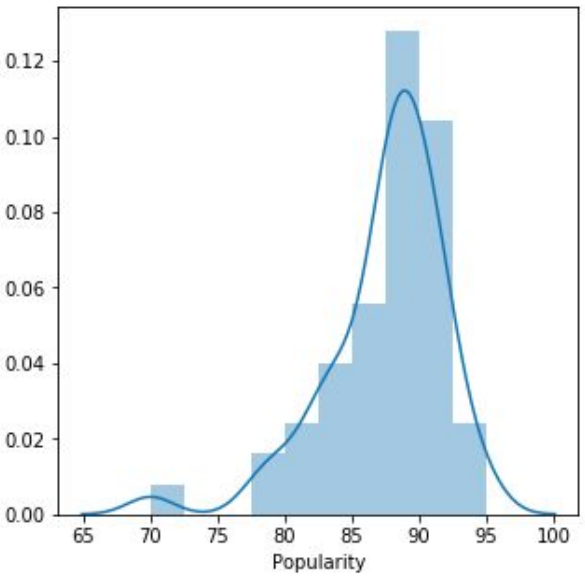
#Dans cette étape , j'ai décidé de trier les genres par genre principales surtout.

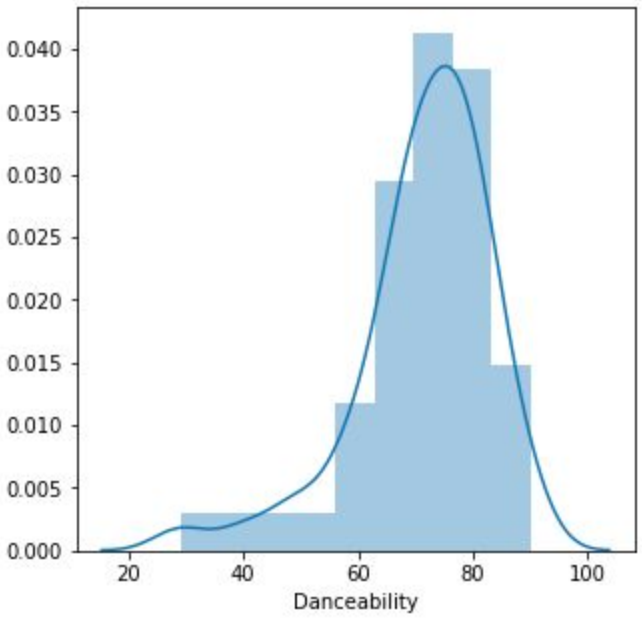
#J'utilise la bibliothèque Seaborn , parceque je veux integrer la densité dans ma visualisation. DISTPLOT

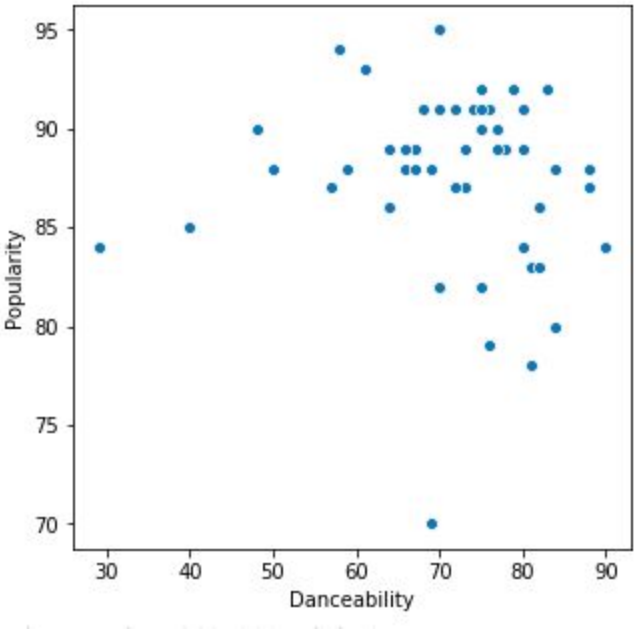
#J'utilise la COUNTPLOT pour avoir un compte exacte du nombre de titres liés a chaque GENERALGENRE.

#La SCATTERPLOT , me permet d'avoir une visualisation avec des points plus ou moins de la danceability en fonction de la popularity pour voir si NÉCESSAIREMENT , les chansons populaires sont exactement les chansons dont la valeur de la danceability est grande , et c'est pour cette raison , ils sont populaires.









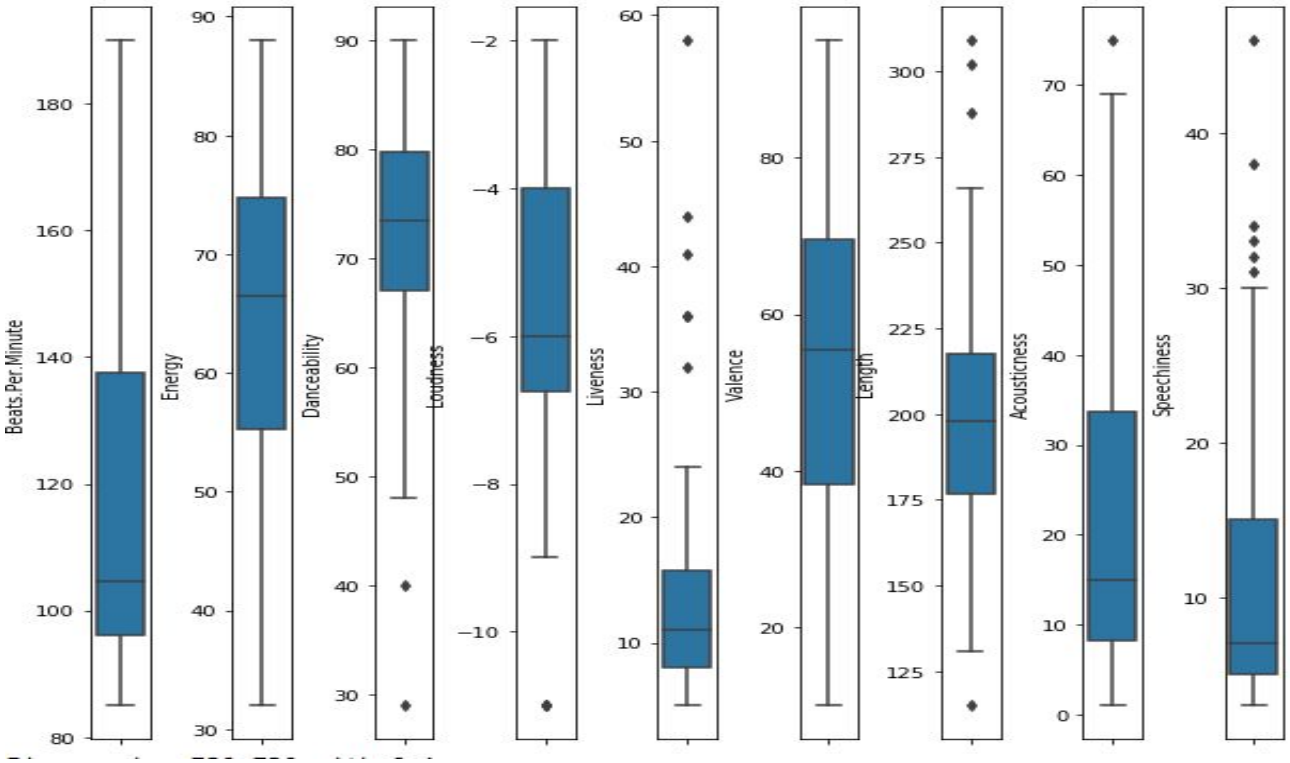
#A noter que les valeurs au niveau de l'axe des y dans les DISPLOTS , correspondent a la densité ou plutôt la densité de la probabilité. Ca nous donne ainsi une idée plus claire sur ces deux critères : popularité et la danceability.

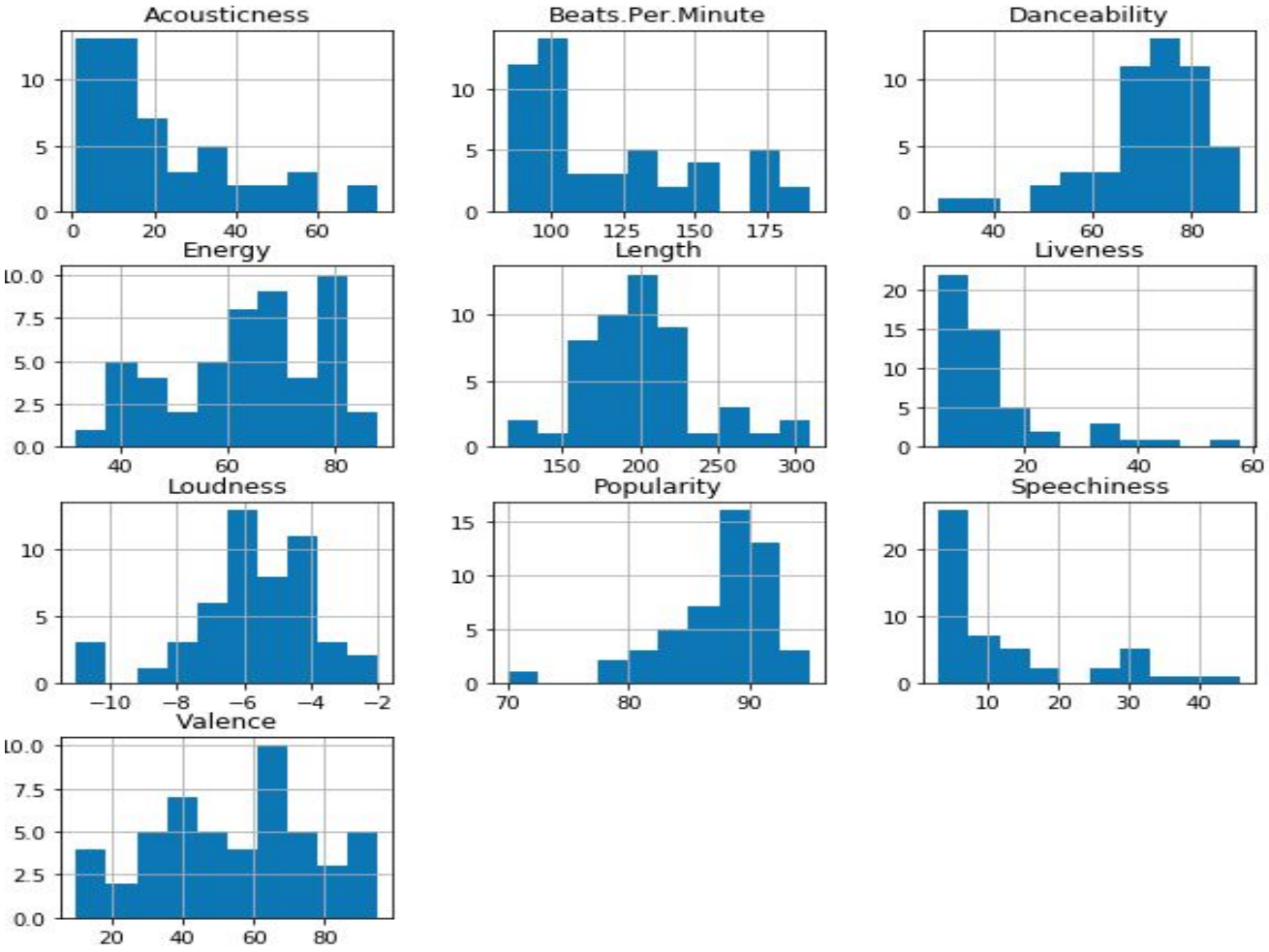
```
fig, ax = plt.subplots(1,9)
fig.subplots_adjust(hspace=0, wspace=1.4)
sns.boxplot( y = data["Beats.Per.Minute"], ax=ax[0])
sns.boxplot( y = data["Energy"], ax=ax[1])
sns.boxplot( y = data["Danceability"], ax=ax[2])
sns.boxplot( y = data["Loudness"], ax=ax[3])
sns.boxplot( y = data["Liveness"], ax=ax[4])
sns.boxplot( y = data["Valence"], ax=ax[5])
sns.boxplot( y = data["Length"], ax=ax[6])
sns.boxplot( y = data["Acousticness"], ax=ax[7])
sns.boxplot( y = data["Speechiness"], ax=ax[8])
plt.figure(figsize=(10,10))

categorie3 = ['Track-name','artist_name']
artist_list=data['artist_name'].values.tolist()
rcParams['figure.figsize'] = 10, 10
data.drop(categorie3,axis=1).hist();
```

#Je vais utiliser maintenant la BOXPLOT afin d'avoir des informations sur le max , le min et la valeur moyenne des différentes données.

#Pour le deuxième programme , j'ai découvert que c'était long et qu'il valait mieux utiliser un autre. " data.hist(figsize=(10,10))"

Ce programme va me permettre non seulement d'avoir un histogramme liées à un élément , mais va me donner tout les histogrammes liées aux différents critères et en prenant en compte un BIN = Nombre de morceaux qui ont une valeur X par exemple d'un critère Y.

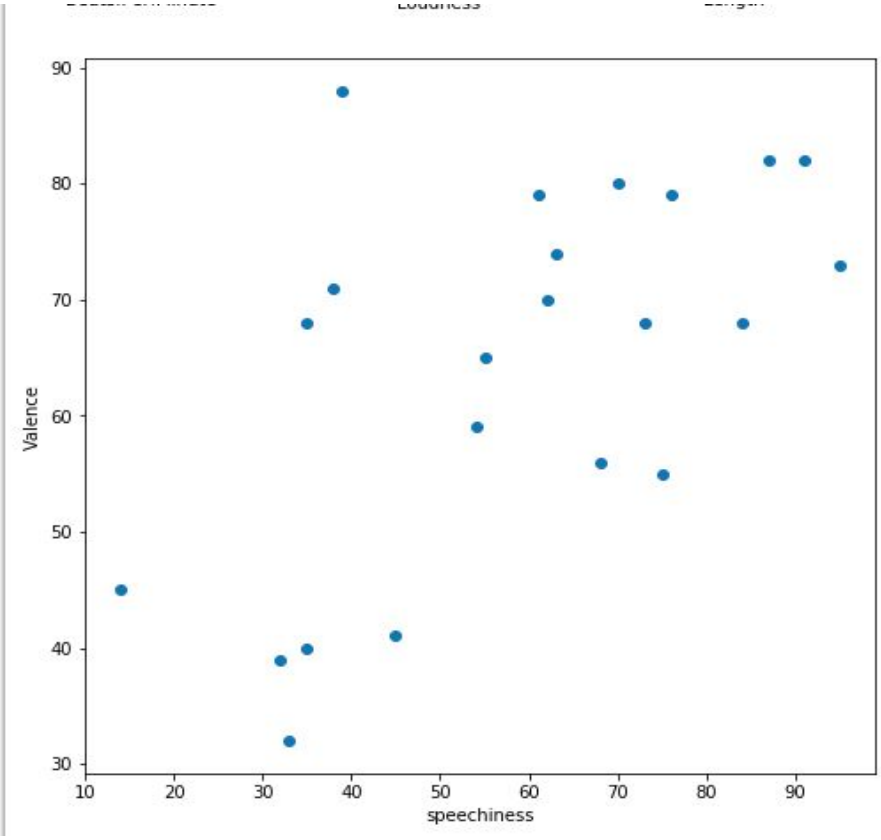




#L'étape suivante consiste alors a choisir un autre SCATTERPLOT mais cette fois , on va utiliser le critère de SPEECHINESS et celui de la VALENCE , à noter que la valeur de la valence nous prouve si oui ou non le morceau est positive ou pas , c'est a dire que si le sentiments qu'on aura après c'est que de la positivité et non la dépression et le malheur. Alors que la speechiness va me montrer si les morceaux sont remplis de paroles (LYCRIS) ou pas , et comme ca j'aurais une idée sur les morceaux dont la valence est très grande , est ce qu'ils transmettent des messages a travers leur morceaux ou c'est juste de la musique instrumentale.

#Mais cette fois , on va utiliser PLT au lieu de SNS , une bibliothèque au lieu d'une autre , et cela m'a permis de m'habituer et a connaître l'utilite de plusieurs fonctions au seins des bibliotheques.

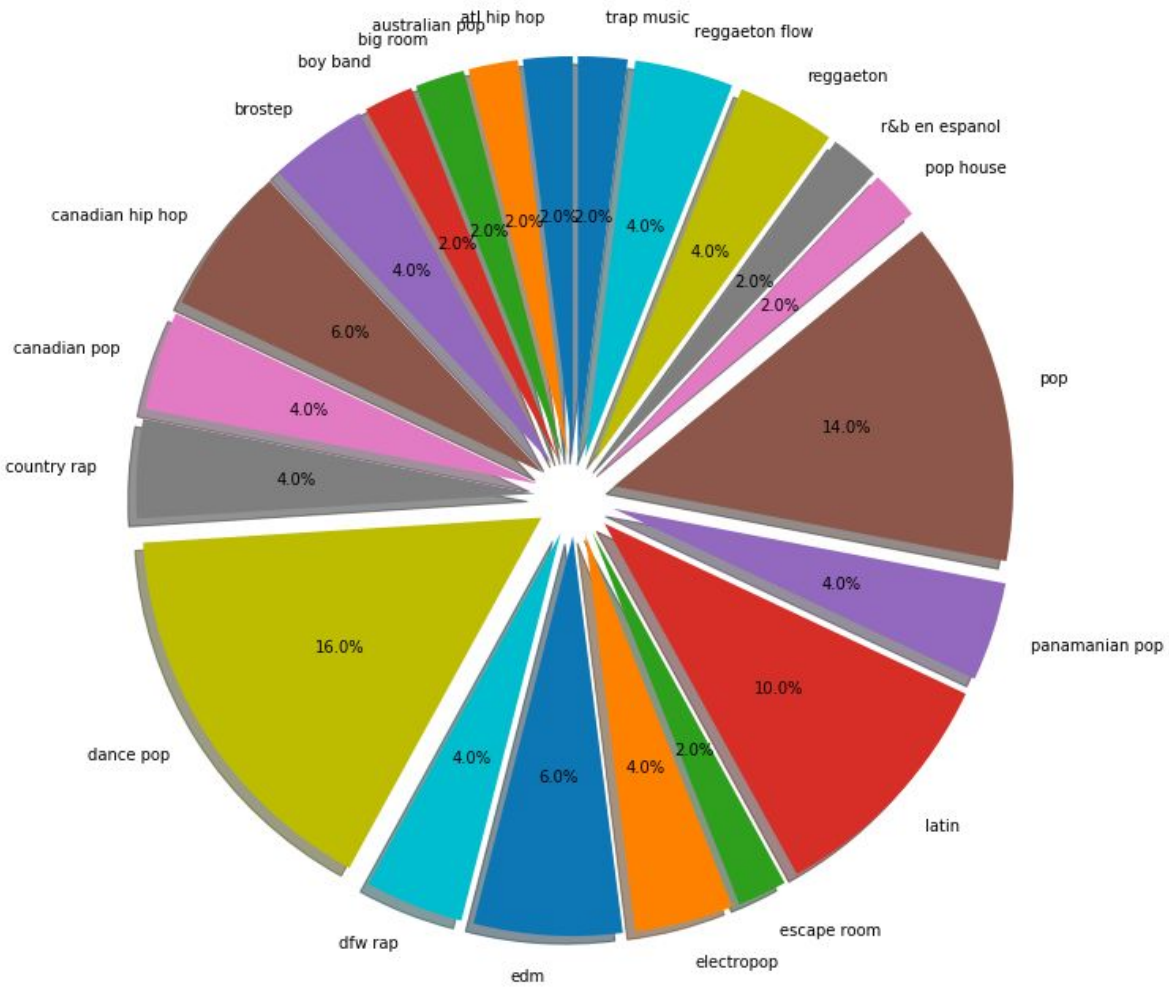
```
plt.figure(figsize=[8,8])
plt.scatter(x=x1,y=y1)
plt.xlabel('speechiness')
plt.ylabel('Valence')
plt.show()
print(data.dtypes)
skew=data.skew()
print(skew)
plt.show()
```



#J'ai eu l'idée de faire un pourcentage de chaque genre et sa dominance au sein du top 50 , question de voir si le genre POP est toujours premier. Et rien n'es meilleure qu'un présentation au niveau d'un cercle.

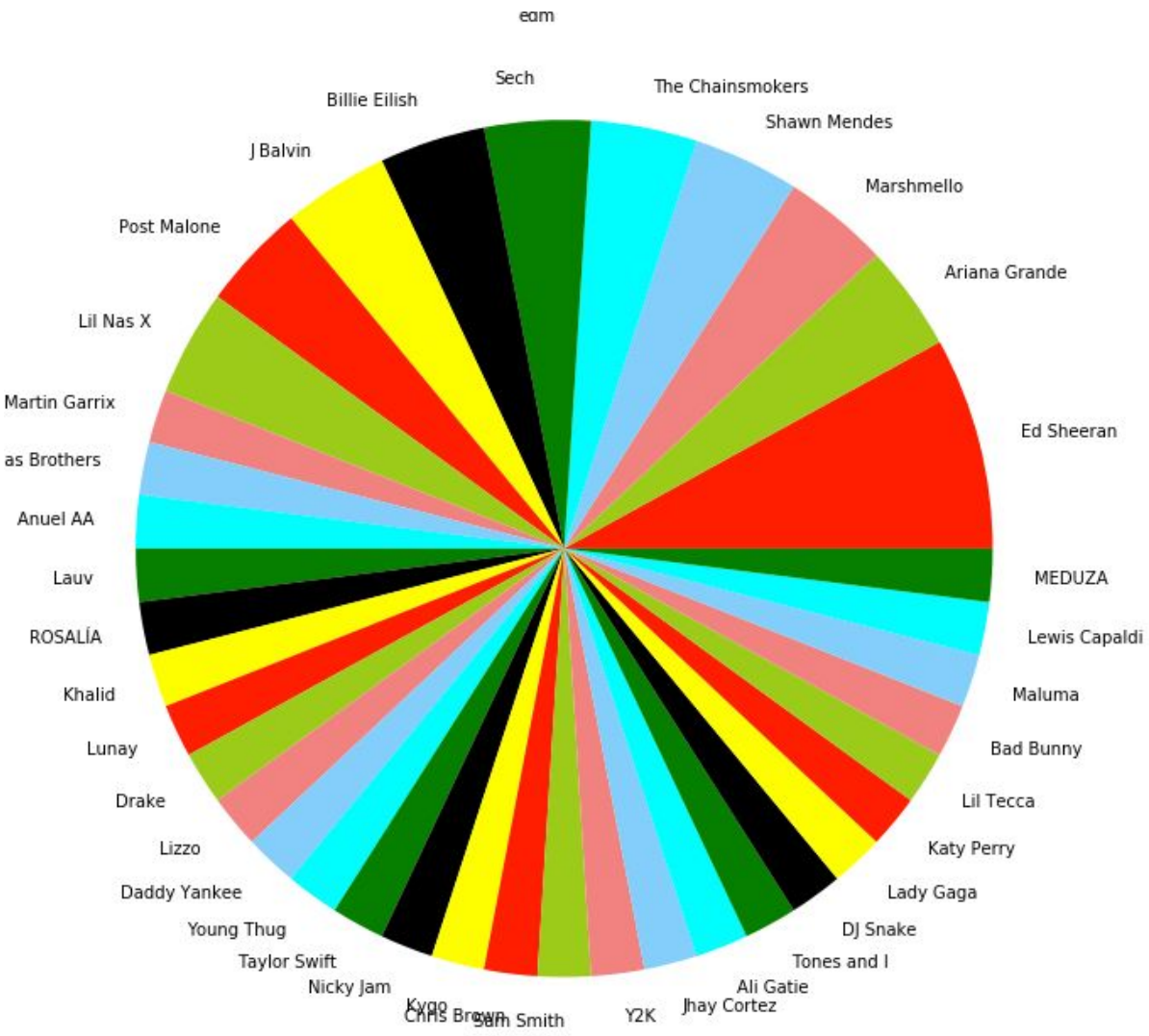
```
size=[]
genre_list=[1,1,1,1,2,3,2,2,8,2,3,2,1,5,2,7,1,1,2,2,1]
pourc=0
for i in(genre_list):
     pourc= round (100*i/45,3)

     size.append(pourc)
print(size)
print(len(size))
genre_list=[1,1,1,1,2,3,2,2,8,2,3,2,1,5,2,7,1,1,2,2,1]
labels=['atl hip hop','australian pop','big room','boy band','brostep','canadian hip hop',
        'canadian pop','country rap','dance pop','dfw rap','edm','electropop','escape room','latin','panamanian pop',
        'pop','pop house','r&b en espanol','reggaeton','reggaeton flow','trap music']
explode = (0.1, 0.1, 0.1, 0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1)
fig1, ax1 = plt.subplots(figsize=(11,11))
ax1.pie(size, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```

#J'ai créé une liste vide , transformer le nombre de titres qu'on trouve dans chaque genre en pourcentage et remplis la liste vide avec le nom de chaque genre et ce qui va constituer le LABEL. J'ai aussi utilisé la fonction EXPLODE afin de représenter plus les pourcentages.



#Avant de passer a la prochaine et dernière partie de mon analyse , j'ai aussi décidé de faire une autre PIE mais cette fois ci avec les artistes et leur nombres de titres sélectionnés dans le top 50 2019 , cela va nous montrer et nous donner une idée sur le travail acharné qu'un artiste a fait durant une année et la qualité de ces titres.

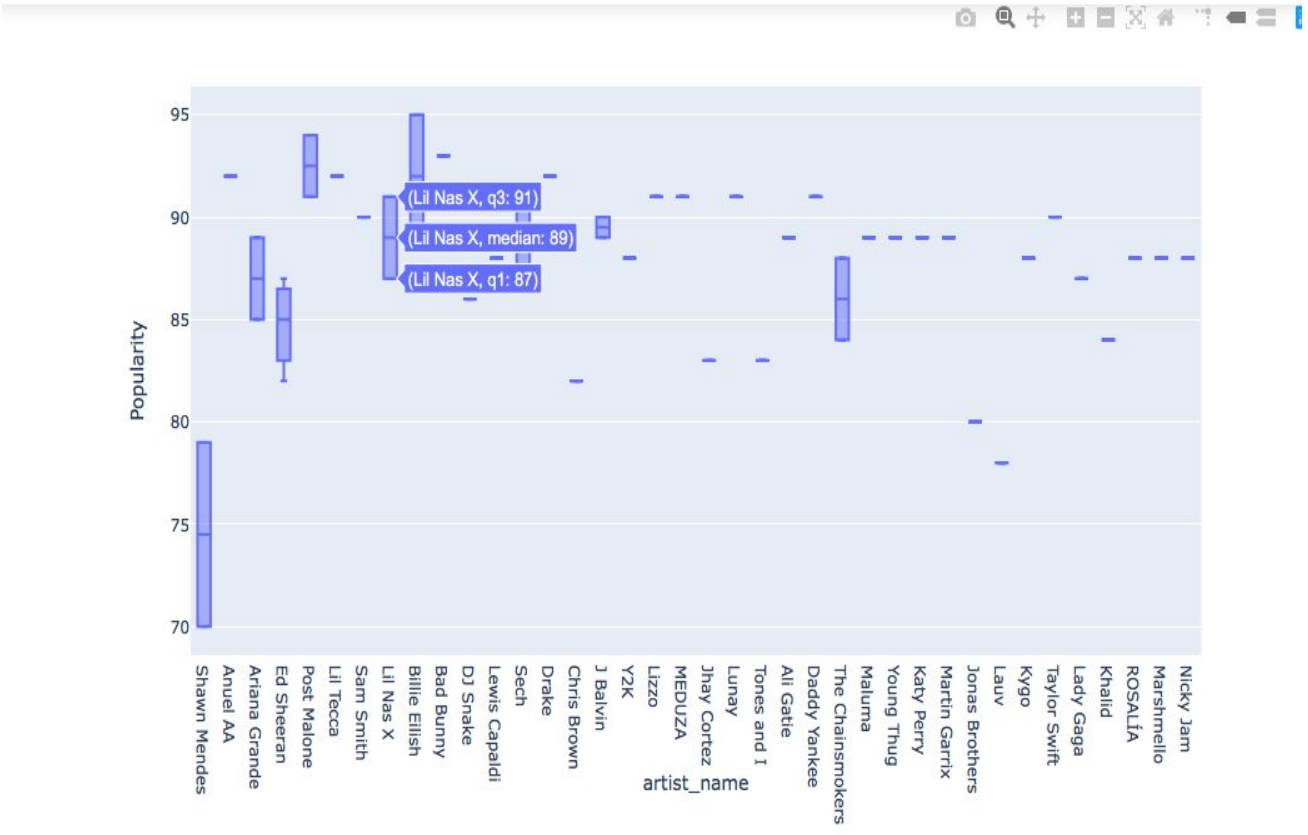
```
133 labels = data.artist_name.value_counts().index
134 sizes = data.artist_name.value_counts().values
135 plt.figure(figsize = (10,10))
136 plt.pie(sizes, labels=labels, colors=colors)
137 autopct=('%1.1f%%')
138 plt.axis('equal')
139
```



#On passe a l'étape ou je vais utiliser JUPITER. En cherchant , j'ai trouvé qu'on pouvait faire des graphes qui sont interactives et cela en utilisant une bibliothèque dont le nom est PLOTLY , en faisant l'installation sur mon TERMINATOR et voulant l'utiliser sur SPYDER , je me rends compte que ca ne marche pas et ca ne me donne pas de résultats , après maintes reprises , j'ai décidé d'utiliser JUPYTER NOTEBOOK.

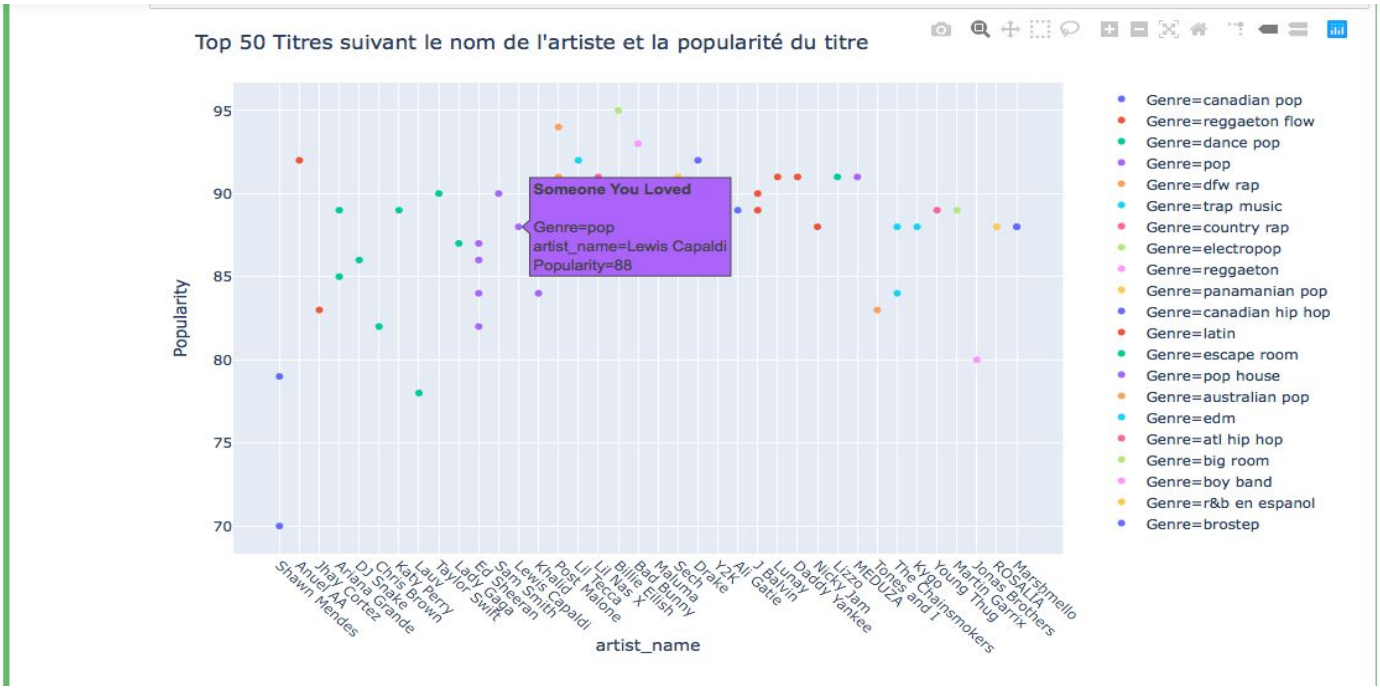
Entrée [6]:

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

from pandas.plotting import scatter_matrix
import plotly_express as px

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
data = pd.read_csv("/Users/macbookpro/Desktop/top50spotify2019/top50.csv",encoding = "ISO-8859-1")
data.head()
data=data.rename(columns={"Loudness..dB..": "Loudness", "Acousticness..": "Acousticness", "Speechiness.":"Speechiness"

data.rename(columns={'Artist.Name':'artist_name'},inplace=True)
data.head()

print(data.head())
artist_contribution = data[['artist_name','Genre', 'Popularity']]


fig = px.box(data, x="artist_name", y="Popularity")

fig.show()
```



#En cliquant comme on le voit sur chaque boxplot , on a les valeurs du max et min et milieu et ceci en fonction de la popularité surtout pour les artistes qui ont deux morceaux ou plus.

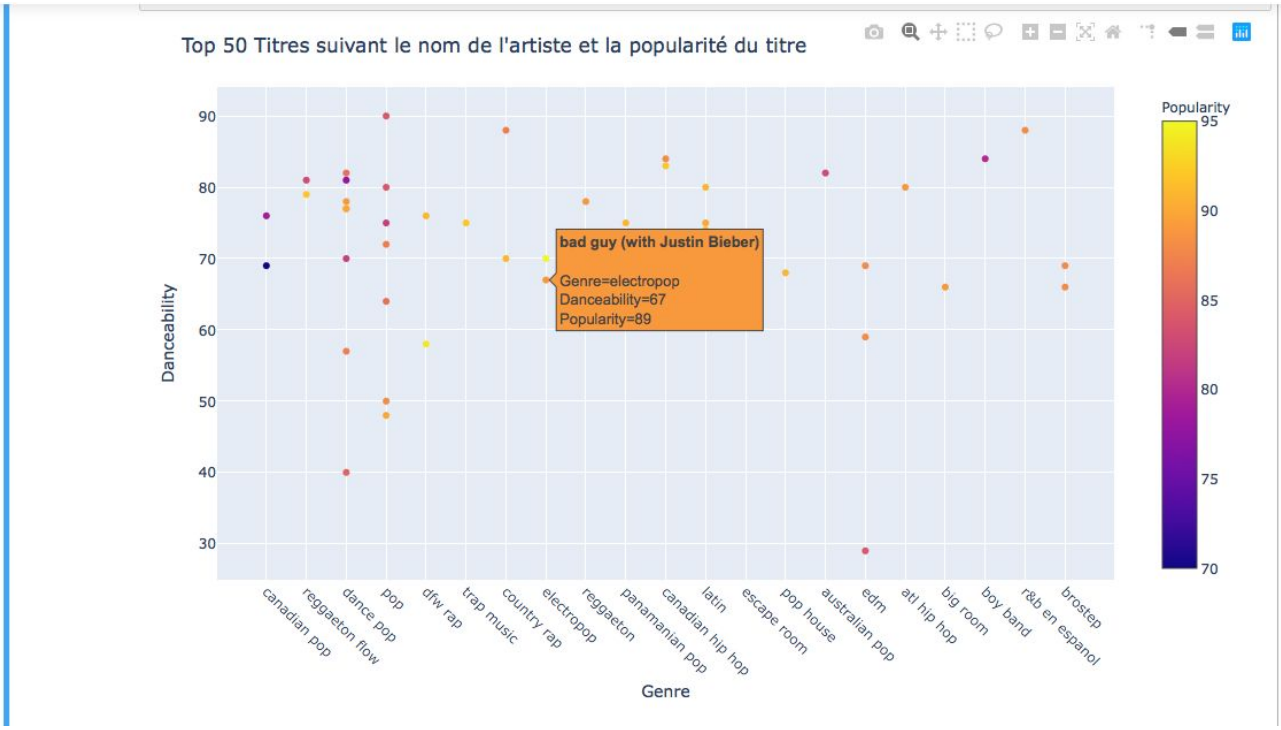
```
Entrée [19]: fig = px.scatter(data, x="artist_name", y="Popularity", color="Genre", hover_name="Track.Name")
fig.update_layout(
    title_text = "Top 50 Titres suivant le nom de l'artiste et la popularité du titre",

    xaxis=dict(tickangle=45)
)
fig.show()
print('Il faut cliquer pour trouver ou connaitre le titre du morceau')
```



#J'ai utilisé et decouvert en meme temps comment utiliser ce scatter interactif. Et j'ai utilisé la popularité avec le genre mais aussi en mettant en consideration le nom des artistes et le plus parfait dans tout ça c'est l'interface interactif qui me permet de connaître le titre de chaque titre. Un titre entendu dans le TOP 50 , d'un artiste X et du genre Y avec un taux de popularité Z.

#le HOVER_NAME c'est là ou je peux préciser qu'est ce que j'ai envie de voir en cliquant sur les différents points présents au seins du graphe. Je peux meme mettre un autre critère comme "Danceability".

```
Entrée [42]: fig = px.scatter(data, x="Genre", y="Danceability", color="Popularity", hover_name="Track.Name")
fig.update_layout(
    title_text = "Top 50 Titres suivant le nom de l'artiste et la popularité du titre",

    xaxis=dict(tickangle=45)
)
fig.show()
print('Il faut cliquer pour trouver ou connaitre le titre du morceau')
```



#J'ai re-utilisé le même graphe vu son utilité surtout a être interactif. En cliquant sur les petits points , je remarque que ca me donne le genre , le taux de danceability et la popularity et en final le nom du morceau musical et ceci va me donner beaucoup d'infos sur les chansons populaires , est ce qu'elles sont nécessairement dansante.

#Le tickAngle me permet juste d'ajuster l'angle de rotation du label qu'on trouve au niveau des axes des Xsticks , si on avais fait 90 , ca aurait ete verticale.

Le rapport de stage prend fin , et ce que je peux dire c'est que ce projet m'a appris énormément de choses. Partant de comment installer un autre environnement , installer de nouvelles bibliothèques et surtout se familiariser avec ces derniers.

Python est un monde très vaste. Le projet m'a permis aussi de m'identifier et de connaître plus le monde de l'analyse de données, ce que j'ignorais et j'ignorais la démarche a suivre pour faire une analyse de données.

Il faut suivre beaucoup d'étapes pour faire une analyse. Mais le plus important c'est de comprendre le pourquoi de cette analyse.