A Statistical Analysis of Music Taste

The Analysis Plaground

MAT 441 Applied Statistics | DePaul University

Yianni Mercer | Fall 2021

Below is a series of Python code relative to the statistical analysis portion of the project

Declaring the Correct Working Directory

In [3]:
```
## Changing current working directory to the base root of the project, so I ca
import os
cwd = os.getcwd() # get cwd
cwd_list = cwd.split('/')[:-1] # split the cwd on the '/' character into a lis
ch = '/' # declare the '/' character
os.chdir(ch.join(cwd_list)) #Rejoin the list on the '/' character and use the
os.getcwd() # show new cwd
```

Out[3]:
```
'/Users/yiannimercer/Library/Mobile Documents/iCloud~com~getrocketbook~Rocketb
ook/Documents/MAT441_Applied_Stats_I/Final/spotify_liked_songs_analysis'
```

Import the Data

In [41]:
```
import pandas as pd #practically tidyverse/dplyr R libraries equivalent, howev
df = pd.read_csv("data_collection/data_files/spotify_liked_final_df.csv",parse
df.head() #print first 10 rows of the df
```

Out[41]:

	Unnamed: 0	genre	artist_name	track_name	track_id	popularity	acousticr
0	0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0.0	(
1	1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	1.0	0

	Unnamed: 0	genre	artist_name	track_name	track_id	popularity	acousticr
2	2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3.0	0
3	3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tIvf	0.0	0
4	4	Movie	Fabien Nataf	Ouverture	0IuslXpMROHdEPvSl1fTQK	4.0	0

5 rows × 22 columns

Minor Clean Up

In [42]:
```
df = df.drop(['Unnamed: 0','date_liked'],axis=1) # Dropping the index and dupl
df.head()
```

Out[42]:

	genre	artist_name	track_name	track_id	popularity	acousticness	dance
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0.0	0.611	
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	1.0	0.246	
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3.0	0.952	
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tIvf	0.0	0.703	
4	Movie	Fabien Nataf	Ouverture	0IuslXpMROHdEPvSl1fTQK	4.0	0.950	

Initial Observations

In [43]:
```
print("The columns of our Data Frame: \n{}".format(df.columns))
```

```
The columns of our Data Frame:
Index(['genre', 'artist_name', 'track_name', 'track_id', 'popularity',
       'acousticness', 'danceability', 'duration_ms', 'energy',
       'instrumentalness', 'key', 'liveness', 'loudness', 'mode',
       'speechiness', 'tempo', 'time_signature', 'valence', 'liked',
       'liked_date'],
      dtype='object')
```

In [44]:
```
print("The shape of the Data Frame: {} \n\n\nSome basic information regarding
print(df.info())
```

```
The shape of the Data Frame: (176514, 20)


Some basic information regarding the column in our Data Frame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 176514 entries, 0 to 176513
Data columns (total 20 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   genre             176514 non-null  object
 1   artist_name       176514 non-null  object
 2   track_name        176514 non-null  object
 3   track_id          176514 non-null  object
 4   popularity        176514 non-null  float64
 5   acousticness      176514 non-null  float64
 6   danceability      176514 non-null  float64
 7   duration_ms       176514 non-null  float64
 8   energy            176514 non-null  float64
 9   instrumentalness  176514 non-null  float64
 10  key               176514 non-null  object
 11  liveness          176514 non-null  float64
 12  loudness          176514 non-null  float64
 13  mode              176514 non-null  object
 14  speechiness       176514 non-null  float64
 15  tempo             176514 non-null  float64
 16  time_signature    176514 non-null  object
 17  valence           176514 non-null  float64
 18  liked             176514 non-null  float64
 19  liked_date        2672 non-null    datetime64[ns, UTC]
dtypes: datetime64[ns, UTC](1), float64(12), object(7)
memory usage: 26.9+ MB
None
```

In [45]:
```
print("Basic Summary Descriptive Statistics of Data Frame")
df.describe()
```

```
Basic Summary Descriptive Statistics of Data Frame
```

Out[45]:

	popularity	acousticness	danceability	duration_ms	energy	instrument
count	176514.000000	176514.000000	176514.000000	1.765140e+05	176514.000000	176514.0
mean	36.257634	0.403876	0.541111	2.361540e+05	0.557203	0.
std	17.392089	0.366286	0.190441	1.305749e+05	0.275855	0.3
min	0.000000	0.000000	0.056900	1.538700e+04	0.000020	0.0
25%	25.000000	0.045500	0.415000	1.782800e+05	0.344000	0.0
50%	37.000000	0.288000	0.558000	2.194750e+05	0.592000	0.0
75%	49.000000	0.791000	0.683000	2.685730e+05	0.789000	0.0
max	100.000000	0.996000	0.989000	5.552917e+06	0.999000	0.9

In [46]:
```
print("Missing Values Per Each Column:")
df.isna().sum()
```

```
Missing Values Per Each Column:
genre                   0
artist_name             0
track_name              0
track_id                0
```

Out[46]:
```
popularity              0
```

```
acousticness             0
danceability             0
duration_ms              0
energy                   0
instrumentalness         0
key                      0
liveness                 0
loudness                 0
mode                     0
speechiness              0
tempo                    0
time_signature           0
valence                  0
liked                    0
liked_date          173842
dtype: int64
```

In [ ]: