The Data Processing For the Visualization of The Popularity of Singers

By Team AZ : Khizer Ahmed Biyabani and Danyal Quazi

Importing necessary libraries

In [34]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Loading the data

In [35]:
```
df = pd.read_csv("data.csv")
```

In [36]:
```
df.head()
```

Out[36]:

	acousticness	artists	danceability	duration_ms	energy	explicit	
0	0.995	['Carl Woitschach']	0.708	158648	0.1950	0	6KbQ3uYMLKb5jDxLF7wYD
1	0.994	['Robert Schumann', 'Vladimir Horowitz']	0.379	282133	0.0135	0	6KuQTIu1KoTTkLXKrwlLF
2	0.604	['Seweryn Goszczyński']	0.749	104300	0.2200	0	6L63VW0PibdM1HDSBoqno
3	0.995	['Francisco Canaro']	0.781	180760	0.1300	0	6M94FkXd15sOAOQYRnWPN
4	0.990	['Frédéric Chopin', 'Vladimir Horowitz']	0.210	687733	0.2040	0	6N6tiFZ9vLTSOIxkj8qK

The Size of Data

In [37]:
```
df.shape
```

Out[37]:
```
(169909, 19)
```

Exploring the Data

In [38]:

```
df.describe()
```

Out[38]:

	acousticness	danceability	duration_ms	energy	explicit	instrumentalness
count	169909.000000	169909.000000	1.699090e+05	169909.000000	169909.000000	169909.000000
mean	0.493214	0.538150	2.314062e+05	0.488593	0.084863	0.161937
std	0.376627	0.175346	1.213219e+05	0.267390	0.278679	0.309329
min	0.000000	0.000000	5.108000e+03	0.000000	0.000000	0.000000
25%	0.094500	0.417000	1.710400e+05	0.263000	0.000000	0.000000
50%	0.492000	0.548000	2.086000e+05	0.481000	0.000000	0.000204
75%	0.888000	0.667000	2.629600e+05	0.710000	0.000000	0.086800
max	0.996000	0.988000	5.403500e+06	1.000000	1.000000	1.000000

In [39]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169909 entries, 0 to 169908
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   acousticness      169909 non-null  float64
 1   artists           169909 non-null  object 
 2   danceability      169909 non-null  float64
 3   duration_ms       169909 non-null  int64  
 4   energy            169909 non-null  float64
 5   explicit          169909 non-null  int64  
 6   id                169909 non-null  object 
 7   instrumentalness  169909 non-null  float64
 8   key               169909 non-null  int64  
 9   liveness          169909 non-null  float64
 10  loudness          169909 non-null  float64
 11  mode              169909 non-null  int64  
 12  name              169909 non-null  object 
 13  popularity        169909 non-null  int64  
 14  release_date      169909 non-null  object 
 15  speechiness       169909 non-null  float64
 16  tempo             169909 non-null  float64
 17  valence           169909 non-null  float64
 18  year              169909 non-null  int64  
dtypes: float64(9), int64(6), object(4)
memory usage: 24.6+ MB
```

In [40]:

```
df.isnull().sum()
```

Out[40]:

```
acousticness        0
artists             0
danceability        0
duration_ms         0
energy              0
explicit            0
id                  0
instrumentalness    0
key                 0
liveness            0
loudness            0
mode                0
name                0
```

```
popularity          0
release_date        0
speechiness         0
tempo               0
valence             0
year                0
dtype: int64
```

Identifying the attributes as

1. continous
2. categorical
3. discrete

In [41]:
```
continues = [features for features in df if df[features].dtypes==('float64')]
```

In [42]:
```
continues
```

Out[42]:
```
['acousticness',
 'danceability',
 'energy',
 'instrumentalness',
 'liveness',
 'loudness',
 'speechiness',
 'tempo',
 'valence']
```

In [ ]:
```

```

In [43]:
```
categorical = [features for features in df if df[features].dtypes=='O']
```

In [44]:
```
categorical
```

Out[44]:
```
['artists', 'id', 'name', 'release_date']
```

In [45]:
```
discrete = [features for features in df if df[features].dtypes==('int64')]
```

In [46]:
```
discrete
```

Out[46]:
```
['duration_ms', 'explicit', 'key', 'mode', 'popularity', 'year']
```

Correlation Matrix

In [47]:
```
corr = df.corr()
corr
```

Out[47]:

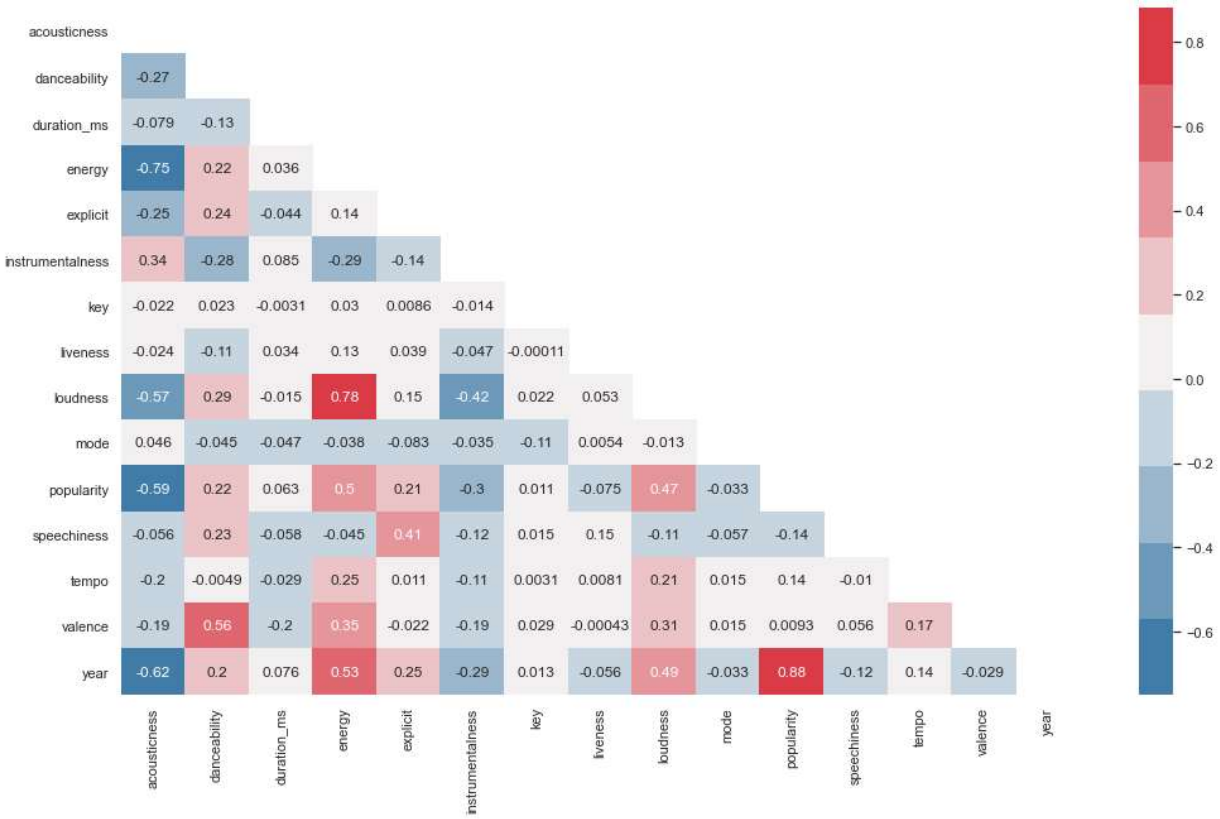
	acousticness	danceability	duration_ms	energy	explicit	instrumentalness	
acousticness	1.000000	-0.265950	-0.079311	-0.750283	-0.253690	0.335821	-0
danceability	-0.265950	1.000000	-0.134500	0.220569	0.241891	-0.281429	0

	acousticness	danceability	duration_ms	energy	explicit	instrumentalness	
duration_ms	-0.079311	-0.134500	1.000000	0.036396	-0.043811	0.084814	-0
energy	-0.750283	0.220569	0.036396	1.000000	0.142677	-0.287692	0
explicit	-0.253690	0.241891	-0.043811	0.142677	1.000000	-0.138292	0
instrumentalness	0.335821	-0.281429	0.084814	-0.287692	-0.138292	1.000000	-0
key	-0.021686	0.022599	-0.003116	0.029984	0.008578	-0.014268	1
liveness	-0.023871	-0.105532	0.034270	0.126293	0.039272	-0.047397	-0
loudness	-0.567072	0.294170	-0.014687	0.782982	0.152695	-0.417033	0
mode	0.046475	-0.045306	-0.046981	-0.038355	-0.083221	-0.035051	-0
popularity	-0.593345	0.221077	0.063292	0.497488	0.214044	-0.299829	0
speechiness	-0.056077	0.225305	-0.058449	-0.045226	0.413074	-0.115735	0
tempo	-0.204982	-0.004872	-0.028816	0.249936	0.011484	-0.107570	0
valence	-0.185540	0.560242	-0.198760	0.350086	-0.022327	-0.193929	0
year	-0.624550	0.203430	0.076293	0.532419	0.245227	-0.291571	0

In [48]:

```
sns.set(rc = {"figure.figsize":(17,10)})
sns.set_theme(style="white")
sns.plotting_context()
cmap = sns.diverging_palette(240, 10, n=9)
mask = np.triu(np.ones_like(corr, dtype=bool))
sns.heatmap(corr, annot = True, cmap=cmap, mask=mask)
```

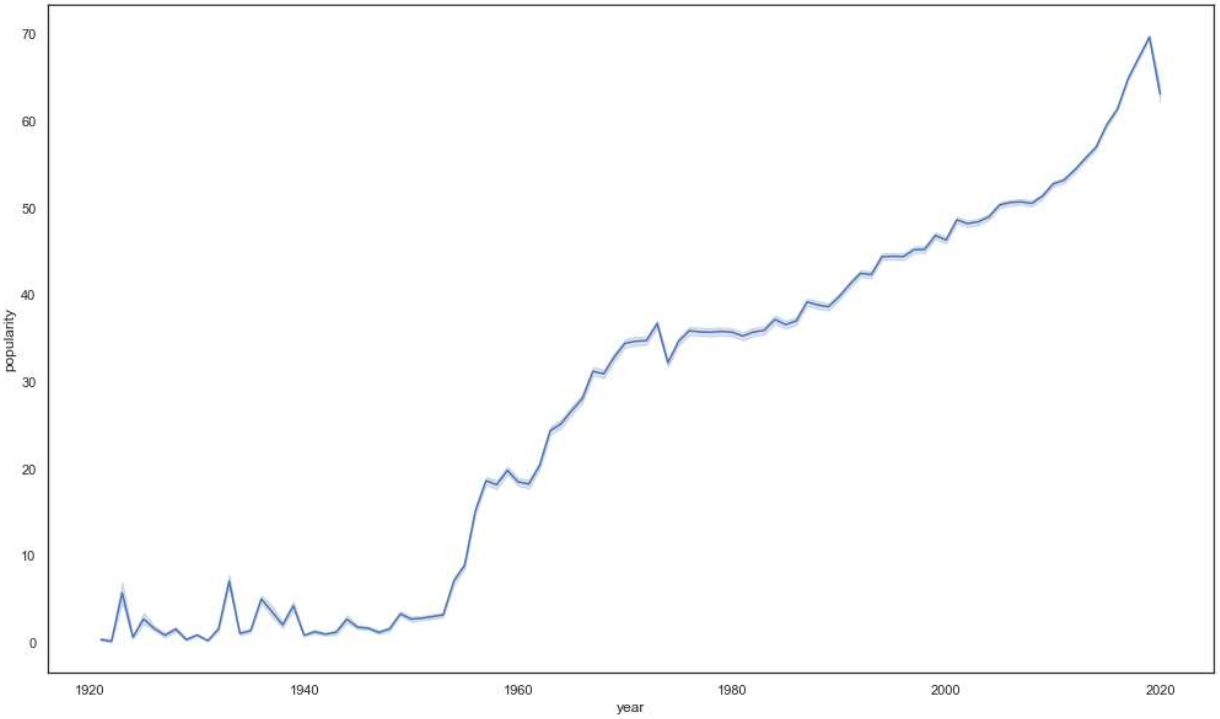
Out[48]: <AxesSubplot:>



Checking for popularity v/s year trend

In [49]:
```
sns.lineplot(x='year', y='popularity', data=df)
```

Out[49]:
```
<AxesSubplot:xlabel='year', ylabel='popularity'>
```



In [50]:
```
df['artists'].value_counts()
```

Out[50]:
```
['Эрнест Хемингуэй']                                                    1215
['Francisco Canaro']                                                     938
['Эрих Мария Ремарк']                                                    781
['Ignacio Corsini']                                                      620
['Frank Sinatra']                                                        592
                                                                        ...
['Shonazar Sakhebov']                                                      1
['Meek Mill', 'Lil Uzi Vert', 'Nicki Minaj']                               1
['Gary Burton', 'Chick Corea', 'Pat Metheny', 'Roy Haynes', 'Dave Holland']    1
['John Tartaglia', 'Shrek Ensemble']                                       1
['Breakout']                                                               1
Name: artists, Length: 33375, dtype: int64
```

In [51]:
```
#np.unique(df['artists'])
len(np.unique(df['name']))
```

Out[51]:
```
132940
```

In [52]:
```
df.head()
```

Out[52]:

	acousticness	artists	danceability	duration_ms	energy	explicit	
0	0.995	['Carl Woitschach']	0.708	158648	0.1950	0	6KbQ3uYMLKb5jDxLF7wYD
1	0.994	['Robert Schumann', 'Vladimir Horowitz']	0.379	282133	0.0135	0	6KuQTIu1KoTTkLXKrwlLF

	acousticness	artists	danceability	duration_ms	energy	explicit	
2	0.604	['Seweryn Goszczyński']	0.749	104300	0.2200	0	6L63VW0PibdM1HDSBoqno
3	0.995	['Francisco Canaro']	0.781	180760	0.1300	0	6M94FkXd15sOAOQYRnWPN
4	0.990	['Frédéric Chopin', 'Vladimir Horowitz']	0.210	687733	0.2040	0	6N6tiFZ9vLTSOIxkj8qK

In [53]:
```
data = df[['artists', 'popularity','name', 'year']]
```

In [54]:
```
data.head()
```

Out[54]:

	artists	popularity	name	year
0	['Carl Woitschach']	0	Singende Bataillone 1. Teil	1928
1	['Robert Schumann', 'Vladimir Horowitz']	0	Fantasiestücke, Op. 111: Più tosto lento	1928
2	['Seweryn Goszczyński']	0	Chapter 1.18 - Zamek kaniowski	1928
3	['Francisco Canaro']	0	Bebamos Juntos - Instrumental (Remasterizado)	1928
4	['Frédéric Chopin', 'Vladimir Horowitz']	1	Polonaise-Fantaisie in A-Flat Major, Op. 61	1928

In [55]:
```
data.shape
```

Out[55]:
```
(169909, 4)
```

In [56]:
```
data['artists'] = data['artists'].str[1:-1]
data.head()
```

```
<ipython-input-56-2df5c171d4b3>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['artists'] = data['artists'].str[1:-1]
```

Out[56]:

	artists	popularity	name	year
0	'Carl Woitschach'	0	Singende Bataillone 1. Teil	1928
1	'Robert Schumann', 'Vladimir Horowitz'	0	Fantasiestücke, Op. 111: Più tosto lento	1928
2	'Seweryn Goszczyński'	0	Chapter 1.18 - Zamek kaniowski	1928

	artists	popularity	name	year
3	'Francisco Canaro'	0	Bebamos Juntos - Instrumental (Remasterizado)	1928
4	'Frédéric Chopin', 'Vladimir Horowitz'	1	Polonaise-Fantaisie in A-Flat Major, Op. 61	1928

Getting Individual Artists

In [59]:
```
d=data.assign(artists=data.artists.str.split(",")).explode('artists')
```

In [60]:
```
d.head()
```

Out[60]:

	artists	popularity	name	year
0	'Carl Woitschach'	0	Singende Bataillone 1. Teil	1928
1	'Robert Schumann'	0	Fantasiestücke, Op. 111: Più tosto lento	1928
1	'Vladimir Horowitz'	0	Fantasiestücke, Op. 111: Più tosto lento	1928
2	'Seweryn Goszczyński'	0	Chapter 1.18 - Zamek kaniowski	1928
3	'Francisco Canaro'	0	Bebamos Juntos - Instrumental (Remasterizado)	1928

In [61]:
```
d.shape
```

Out[61]:
```
(226813, 4)
```

In [62]:
```
d.tail(10)
```

Out[62]:

	artists	popularity	name	year
169903	'Junior H'	68	Ojos De Maniaco	2020
169904	'DripReport'	75	Skechers (feat. Tyga) - Remix	2020
169904	'Tyga'	75	Skechers (feat. Tyga) - Remix	2020
169905	'Leon Bridges'	64	Sweeter (feat. Terrace Martin)	2020
169905	'Terrace Martin'	64	Sweeter (feat. Terrace Martin)	2020
169906	'Kygo'	70	How Would I Know	2020
169906	'Oh Wonder'	70	How Would I Know	2020
169907	'Cash Cash'	70	I Found You	2020
169907	'Andy Grammer'	70	I Found You	2020
169908	'Ingrid Andress'	65	More Hearts Than Mine	2020

In [63]:
```
#d['artists'].value_counts()
len(np.unique(d['artists']))
```

Out[63]:
```
31527
```

In [64]:
```
d.isnull().sum()
```

Out[64]:
```
artists       0
popularity    0
name          0
year          0
dtype: int64
```

In [65]:
```
d.duplicated().sum()
```

Out[65]:
```
2117
```

In [66]:
```
d = d.reset_index(drop=True)
```

In [67]:
```
d.head()
```

Out[67]:

	artists	popularity	name	year
0	'Carl Woitschach'	0	Singende Bataillone 1. Teil	1928
1	'Robert Schumann'	0	Fantasiestücke, Op. 111: Più tosto lento	1928
2	'Vladimir Horowitz'	0	Fantasiestücke, Op. 111: Più tosto lento	1928
3	'Seweryn Goszczyński'	0	Chapter 1.18 - Zamek kaniowski	1928
4	'Francisco Canaro'	0	Bebamos Juntos - Instrumental (Remasterizado)	1928

In [68]:
```
len(np.unique(d['artists']))
```

Out[68]:
```
31527
```

Determing the Average Popularity of Each Artist in Each year

In [69]:
```
d1 = d.groupby(['artists','year']).agg({'popularity':[np.mean]}).reset_index()
```

In [70]:
```
d1
```

Out[70]:

	artists	year	popularity
			mean
0	"'Be More Chill' Ensemble"	2015	52.000000
1	"'In The Heights' Original Broadway Company"	2008	46.000000
2	"'Legally Blonde' Ensemble"	2007	48.571429
3	"'Legally Blonde' Greek Chorus"	2007	49.000000
4	"'Little Women' Original Broadway Cast"	2005	42.000000
...	...	...	...
66306	Coda	2015	61.000000
66307	Giuseppe di Stefano	1953	1.000000

	artists	year	popularity
			mean
66308	JIN∽'	2015	61.000000
66309	nowhere.'	2016	59.000000
66310	nowhere.'	2018	66.000000

66311 rows × 3 columns

Saving The Final dataset

In [71]:
```
d1.to_csv('d1.csv')
```

The End