In [1]:
```
import pandas as pd
```

In [2]:
```
# Elimina les columnes del dataFrame passades per param

def drop_specific_cols(df, cols):
    df = df.drop(cols,axis=1)
    return df
```

In [3]:
```
# csv to df del msd
msd_df = pd.read_csv('./datasets/msd_reduced.csv', index_col=0)
```

In [4]:
```
# csv to df dels grammy, billboard i spotify
grammy_df = pd.read_csv('./datasets/grammy_reduced.csv', index_col=0)
billboard_df = pd.read_csv('./datasets/billboard_reduced.csv', index_col=0)
spotify_df = pd.read_csv('./datasets/spotify_combined_artists.csv', index_col=0)
```

In [5]:
```
# inner join between spotify and msd per obtenir param fisics de les cançons d'spotify
new_df = pd.merge(msd_df, spotify_df, left_on=['title','artist_name'], right_on = ['title_spoty','all_artists'])
print ("Number of rows matched : %d"%new_df['popularity'].count())
new_df.describe()
```

```
Number of rows matched : 1211
```

Out[5]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time
count	1211.0	1211.000000	1211.000000	1211.0	1211.000000	1211.000000	1211.000000	1211.000000	1211.000000	1
mean	0.0	234.941919	1.148684	0.0	5.224608	-9.704753	0.710157	225.665208	122.233003	
std	0.0	92.735189	2.318206	0.0	3.610923	5.513450	0.453877	91.567689	32.333065	
min	0.0	10.605260	0.000000	0.0	0.000000	-34.198000	0.000000	10.605000	0.000000	
25%	0.0	180.897505	0.089000	0.0	2.000000	-12.037500	0.000000	173.212000	100.083500	
50%	0.0	229.067300	0.229000	0.0	5.000000	-8.238000	1.000000	218.860000	119.945000	
75%	0.0	273.122810	0.877000	0.0	8.000000	-5.935000	1.000000	262.313000	139.783000	
max	0.0	1400.267300	30.674000	0.0	11.000000	-0.558000	1.000000	1389.592000	243.049000	

In [6]:
```
# left joins for matches between msd and grammy per saber quants grammies ha guanyat cada cançó
new_df = pd.merge(new_df, grammy_df, how='left', left_on=['title','artist_name'], right_on = ['nominee','artist'])
# display matches
print ("Number of rows matched : %d"%new_df['nominee'].count())
new_df.describe()
```

```
Number of rows matched : 27
```

Out[6]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time
count	1211.0	1211.000000	1211.000000	1211.0	1211.000000	1211.000000	1211.000000	1211.000000	1211.000000	1
mean	0.0	234.941919	1.148684	0.0	5.224608	-9.704753	0.710157	225.665208	122.233003	
std	0.0	92.735189	2.318206	0.0	3.610923	5.513450	0.453877	91.567689	32.333065	
min	0.0	10.605260	0.000000	0.0	0.000000	-34.198000	0.000000	10.605000	0.000000	
25%	0.0	180.897505	0.089000	0.0	2.000000	-12.037500	0.000000	173.212000	100.083500	
50%	0.0	229.067300	0.229000	0.0	5.000000	-8.238000	1.000000	218.860000	119.945000	
75%	0.0	273.122810	0.877000	0.0	8.000000	-5.935000	1.000000	262.313000	139.783000	
max	0.0	1400.267300	30.674000	0.0	11.000000	-0.558000	1.000000	1389.592000	243.049000	

In [7]:
```
# left joins for matches between msd and billboard per saber quantes setmanes ha estat a la billboard cada cançó
new_df = pd.merge(new_df, billboard_df, how='left', left_on=['title','artist_name'], right_on = ['song','artist'])
print ("Number of rows matched : %d"%new_df['song'].count())
new_df.describe()
```

```
Number of rows matched : 144
```

Out[7]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time
count	1211.0	1211.000000	1211.000000	1211.0	1211.000000	1211.000000	1211.000000	1211.000000	1211.000000	1
mean	0.0	234.941919	1.148684	0.0	5.224608	-9.704753	0.710157	225.665208	122.233003	
std	0.0	92.735189	2.318206	0.0	3.610923	5.513450	0.453877	91.567689	32.333065	
min	0.0	10.605260	0.000000	0.0	0.000000	-34.198000	0.000000	10.605000	0.000000	
25%	0.0	180.897505	0.089000	0.0	2.000000	-12.037500	0.000000	173.212000	100.083500	
50%	0.0	229.067300	0.229000	0.0	5.000000	-8.238000	1.000000	218.860000	119.945000	
75%	0.0	273.122810	0.877000	0.0	8.000000	-5.935000	1.000000	262.313000	139.783000	
max	0.0	1400.267300	30.674000	0.0	11.000000	-0.558000	1.000000	1389.592000	243.049000	

8 rows × 22 columns

In [8]:
```
display(new_df)
```

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time_signature	...	popular
0	0.0	223.94730	0.316	0.0	7	-7.417	1	216.160	119.301	4	...	
1	0.0	317.90975	2.862	0.0	1	-7.714	1	303.386	90.024	4	...	
2	0.0	193.93261	0.160	0.0	7	-8.397	0	182.967	143.997	4	...	
3	0.0	207.77751	0.160	0.0	9	-7.642	1	197.143	167.888	1	...	
4	0.0	302.81098	0.000	0.0	11	-3.575	1	293.634	170.026	4	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
1206	0.0	259.52608	4.952	0.0	0	-12.990	1	246.143	133.894	4	...	
1207	0.0	243.12118	0.074	0.0	4	-5.323	0	234.818	88.044	4	...	
1208	0.0	217.44281	0.183	0.0	7	-7.515	0	214.332	87.010	4	...	
1209	0.0	675.91791	0.000	0.0	10	-8.901	0	671.057	92.820	3	...	
1210	0.0	246.49098	0.334	0.0	0	-14.073	1	234.672	70.055	3	...	

1211 rows × 31 columns

In [9]:
```
# Creem una nova columna, e_grammy, que pren valors enters segons sel nombre de grammies que ha guanyat una can
# Ojo perquè una cançó pot guanyar més d'un Grammy.
new_df['e_grammy'] = new_df['prizes'].fillna(0)
```

In [10]:
```
new_df.describe()
```

Out[10]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time
count	1211.0	1211.000000	1211.000000	1211.0	1211.000000	1211.000000	1211.000000	1211.000000	1211.000000	1
mean	0.0	234.941919	1.148684	0.0	5.224608	-9.704753	0.710157	225.665208	122.233003	
std	0.0	92.735189	2.318206	0.0	3.610923	5.513450	0.453877	91.567689	32.333065	
min	0.0	10.605260	0.000000	0.0	0.000000	-34.198000	0.000000	10.605000	0.000000	
25%	0.0	180.897505	0.089000	0.0	2.000000	-12.037500	0.000000	173.212000	100.083500	
50%	0.0	229.067300	0.229000	0.0	5.000000	-8.238000	1.000000	218.860000	119.945000	
75%	0.0	273.122810	0.877000	0.0	8.000000	-5.935000	1.000000	262.313000	139.783000	
max	0.0	1400.267300	30.674000	0.0	11.000000	-0.558000	1.000000	1389.592000	243.049000	

8 rows × 23 columns

In [11]:
```
count = (new_df['e_grammy'] != 0).sum()
print(count)
```

```
27
```

In [12]:
```
new_df['is_billboard'] = new_df['peak-rank'].notnull().astype('int')
```

In [13]:
```
def billboard_success(m, s, is_billboard):
    if is_billboard:
        return (101 - m + s)
    else:
        return 0
```

In [14]:
```
# Per cada fila de la taula new_df, cridem la funció billboard_success amb els atributs seg. i guardem el valor
new_df['e_billboard'] = new_df.apply(lambda x: billboard_success(x['peak-rank'], x['weeks-on-board'],x['is_billb
```

In [15]:
```
display(new_df)
```

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time_signature	...	artist_l
0	0.0	223.94730	0.316	0.0	7	-7.417	1	216.160	119.301	4	...	NaN
1	0.0	317.90975	2.862	0.0	1	-7.714	1	303.386	90.024	4	...	NaN
2	0.0	193.93261	0.160	0.0	7	-8.397	0	182.967	143.997	4	...	NaN
3	0.0	207.77751	0.160	0.0	9	-7.642	1	197.143	167.888	1	...	NaN
4	0.0	302.81098	0.000	0.0	11	-3.575	1	293.634	170.026	4	...	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
1206	0.0	259.52608	4.952	0.0	0	-12.990	1	246.143	133.894	4	...	NaN
1207	0.0	243.12118	0.074	0.0	4	-5.323	0	234.818	88.044	4	...	NaN
1208	0.0	217.44281	0.183	0.0	7	-7.515	0	214.332	87.010	4	...	NaN
1209	0.0	675.91791	0.000	0.0	10	-8.901	0	671.057	92.820	3	...	NaN
1210	0.0	246.49098	0.334	0.0	0	-14.073	1	234.672	70.055	3	...	NaN

1211 rows × 34 columns

In [16]:
```
count_b = (new_df['e_billboard'] != 0).sum()
print(count_b)
```

```
144
```

In [17]:
```
new_df['e_spotify'] = new_df['popularity'].fillna(0)
```

In [18]:
```
count_s = (new_df['e_spotify'] != 0).sum()
print(count_s)
```

```
1163
```

In [19]:
```
# Eliminem columnes innecessàries per a estudiar les dades
cols = [ 'nominee', 'artist_x', 'rank', 'song', 'artist_y', 'peak-rank', 'weeks-on-board', 'is_billboard', 'all
new_df = drop_specific_cols(new_df, cols)
```

In [20]:
```
display(new_df)
```

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time_signature	...	artist_l
0	0.0	223.94730	0.316	0.0	7	-7.417	1	216.160	119.301	4	...	Kingsl
1	0.0	317.90975	2.862	0.0	1	-7.714	1	303.386	90.024	4	...	Madri
2	0.0	193.93261	0.160	0.0	7	-8.397	0	182.967	143.997	4	...	Charle t
3	0.0	207.77751	0.160	0.0	9	-7.642	1	197.143	167.888	1	...	
4	0.0	302.81098	0.000	0.0	11	-3.575	1	293.634	170.026	4	...	St. Jos
...	...	...	...	...	...	...	...	...	...	...	...	...
1206	0.0	259.52608	4.952	0.0	0	-12.990	1	246.143	133.894	4	...	Charle t
1207	0.0	243.12118	0.074	0.0	4	-5.323	0	234.818	88.044	4	...	New Yo
1208	0.0	217.44281	0.183	0.0	7	-7.515	0	214.332	87.010	4	...	New Yo
1209	0.0	675.91791	0.000	0.0	10	-8.901	0	671.057	92.820	3	...	Barn
1210	0.0	246.49098	0.334	0.0	0	-14.073	1	234.672	70.055	3	...	Om

1211 rows × 23 columns

In [21]:
```
new_df.describe()
```

Out[21]:

	danceability	duration	end_of_fade_in	energy	key	loudness	mode	start_of_fade_out	tempo	time
count	1211.0	1211.000000	1211.000000	1211.0	1211.000000	1211.000000	1211.000000	1211.000000	1211.000000	1
mean	0.0	234.941919	1.148684	0.0	5.224608	-9.704753	0.710157	225.665208	122.233003	
std	0.0	92.735189	2.318206	0.0	3.610923	5.513450	0.453877	91.567689	32.333065	
min	0.0	10.605260	0.000000	0.0	0.000000	-34.198000	0.000000	10.605000	0.000000	
25%	0.0	180.897505	0.089000	0.0	2.000000	-12.037500	0.000000	173.212000	100.083500	
50%	0.0	229.067300	0.229000	0.0	5.000000	-8.238000	1.000000	218.860000	119.945000	
75%	0.0	273.122810	0.877000	0.0	8.000000	-5.935000	1.000000	262.313000	139.783000	
max	0.0	1400.267300	30.674000	0.0	11.000000	-0.558000	1.000000	1389.592000	243.049000	

In [22]:
```
# export to csv file
new_df.to_csv("successes_with_artists.csv")
```