Appendix 9: Hypothesis 1 Test

In [75]:

```
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy.stats import chi2_contingency
from scipy.stats import chi2
from sklearn.preprocessing import MinMaxScaler
```

In [76]:

```
df=pd.read_csv('data_final.csv')
df.head()
```

Out[76]:

	Unnamed: 0	track	artist	uri	danceability	energy
0	0	Wild Things	Alessia Cara	spotify:track:2ZyuwVvV6Z3XJaXIFbspeE	0.741	0.62
1	1	Love Someone	Lukas Graham	spotify:track:2JqnpexIO9dmvjUMCaLCLJ	0.550	0.41
2	2	Here's To Never Growing Up	Avril Lavigne	spotify:track:0qwcGscxUHGZTgq0zcaqk1	0.482	0.87
3	3	Crawling Back To You	Daughtry	spotify:track:6BDtTzjbJ5kKKSWcJT8MIX	0.438	0.91
4	4	Faster	Matt Nathanson	spotify:track:6pIKFdrBnKF0y3CRuceTDh	0.742	0.85

5 rows × 32 columns

In [77]:

```
df=df.iloc[:,1:]
df.head()
```

Out[77]:

	track	artist	uri	danceability	energy	key	louc
0	Wild Things	Alessia Cara	spotify:track:2ZyuwVvV6Z3XJaXIFbspeE	0.741	0.626	1	-
1	Love Someone	Lukas Graham	spotify:track:2JqnpexIO9dmvjUMCaLCLJ	0.550	0.415	9	-
2	Here's To Never Growing Up	Avril Lavigne	spotify:track:0qwcGscxUHGZTgq0zcaqk1	0.482	0.873	0	-
3	Crawling Back To You	Daughtry	spotify:track:6BDtTzjbJ5kKKSWcJT8MIX	0.438	0.919	0	-
4	Faster	Matt Nathanson	spotify:track:6pIKFdrBnKF0y3CRuceTDh	0.742	0.853	9	-

5 rows × 31 columns

In [78]:

```
genres=df['genres'].unique()
decades=df['Decade'].unique()
print(genres)
print(decades)
```

```
['Pop' 'Easy listening' 'Hip hop' 'Metal' 'Country' 'Electronic' 'Rock'
 'Folk' 'Latin' 'Classical' 'Jazz' 'R&B' 'Caribbean' 'Blues']
['10s' '00s' '90s' '80s' '70s' '60s']
```

In [79]:

```
tb = [[0 for x in range(len(decades)+1)] for y in range(len(genres))]
tb
```

Out[79]:

```
[[0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0]]
```

In [80]:

```
tb[0][0]
```

Out[80]:

```
0
```

In [81]:

```
for i in range(len(decades)+1):
    if(i!=0):
        df_1=df[df['Decade']==decades[i-1]]
        df_1=df_1[df_1['target']==1]
        df_1=df_1.reset_index(drop=True)
        hits=len(df_1)
    for j in range(len(genres)):
        if(i==0):
            tb[j][i]=genres[j]
        else:
            df_2=df_1[df_1['genres']==genres[j]]
            df_2=df_2.reset_index(drop=True)
            tb[j][i]=len(df_2)
```

In [82]:

```
tb
```

Out[82]:

```
[['Pop', 1170, 994, 583, 476, 590, 1204],
 ['Easy listening', 4, 0, 2, 4, 42, 265],
 ['Hip hop', 781, 452, 322, 25, 0, 0],
 ['Metal', 26, 145, 18, 9, 1, 1],
 ['Country', 567, 561, 163, 81, 136, 151],
 ['Electronic', 62, 12, 147, 55, 7, 1],
 ['Rock', 83, 216, 554, 1541, 1099, 365],
 ['Folk', 3, 10, 21, 31, 113, 73],
 ['Latin', 20, 18, 3, 1, 1, 0],
 ['Classical', 0, 0, 0, 0, 7, 18],
 ['Jazz', 1, 3, 13, 47, 28, 65],
 ['R&B', 27, 96, 271, 471, 999, 1044],
 ['Caribbean', 20, 14, 35, 8, 3, 5],
 ['Blues', 1, 6, 4, 8, 24, 62]]
```

In [83]:

```
x=['Genre']
for i in range(len(decades)):
    x.append(decades[i])
x
```

Out[83]:

```
['Genre', '10s', '00s', '90s', '80s', '70s', '60s']
```

In [84]:

```
df_tb=pd.DataFrame(data=tb,columns=x)
df_tb
```

Out[84]:

	Genre	10s	00s	90s	80s	70s	60s
0	Pop	1170	994	583	476	590	1204
1	Easy listening	4	0	2	4	42	265
2	Hip hop	781	452	322	25	0	0
3	Metal	26	145	18	9	1	1
4	Country	567	561	163	81	136	151
5	Electronic	62	12	147	55	7	1
6	Rock	83	216	554	1541	1099	365
7	Folk	3	10	21	31	113	73
8	Latin	20	18	3	1	1	0
9	Classical	0	0	0	0	7	18
10	Jazz	1	3	13	47	28	65
11	R&B	27	96	271	471	999	1044
12	Caribbean	20	14	35	8	3	5
13	Blues	1	6	4	8	24	62

In [85]:

```
y=df_tb.iloc[:,1:]
y
```

Out[85]:

	10s	00s	90s	80s	70s	60s
0	1170	994	583	476	590	1204
1	4	0	2	4	42	265
2	781	452	322	25	0	0
3	26	145	18	9	1	1
4	567	561	163	81	136	151
5	62	12	147	55	7	1
6	83	216	554	1541	1099	365
7	3	10	21	31	113	73
8	20	18	3	1	1	0
9	0	0	0	0	7	18
10	1	3	13	47	28	65
11	27	96	271	471	999	1044
12	20	14	35	8	3	5
13	1	6	4	8	24	62

In [86]:

```
norm = MinMaxScaler().fit(y)
y_norm = norm.transform(y)
y_norm
```

Out[86]:

```
array([[1.00000000e+00, 1.00000000e+00, 1.00000000e+00, 3.08890331e-01,
        5.36851683e-01, 1.00000000e+00],
       [3.41880342e-03, 0.00000000e+00, 3.43053173e-03, 2.59571707e-03,
        3.82165605e-02, 2.20099668e-01],
       [6.67521368e-01, 4.54728370e-01, 5.52315609e-01, 1.62232317e-02,
        0.00000000e+00, 0.00000000e+00],
       [2.22222222e-02, 1.45875252e-01, 3.08747856e-02, 5.84036340e-03,
        9.09918107e-04, 8.30564784e-04],
       [4.84615385e-01, 5.64386318e-01, 2.79588336e-01, 5.25632706e-02,
        1.23748863e-01, 1.25415282e-01],
       [5.29914530e-02, 1.20724346e-02, 2.52144082e-01, 3.56911097e-02,
        6.36942675e-03, 8.30564784e-04],
       [7.09401709e-02, 2.17303823e-01, 9.50257290e-01, 1.00000000e+00,
        1.00000000e+00, 3.03156146e-01],
       [2.56410256e-03, 1.00603622e-02, 3.60205832e-02, 2.01168073e-02,
        1.02820746e-01, 6.06312292e-02],
       [1.70940171e-02, 1.81086519e-02, 5.14579760e-03, 6.48929267e-04,
        9.09918107e-04, 0.00000000e+00],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        6.36942675e-03, 1.49501661e-02],
       [8.54700855e-04, 3.01810865e-03, 2.22984563e-02, 3.04996755e-02,
        2.54777070e-02, 5.39867110e-02],
       [2.30769231e-02, 9.65794769e-02, 4.64837050e-01, 3.05645685e-01,
        9.09008189e-01, 8.67109635e-01],
       [1.70940171e-02, 1.40845070e-02, 6.00343053e-02, 5.19143413e-03,
        2.72975432e-03, 4.15282392e-03],
       [8.54700855e-04, 6.03621730e-03, 6.86106346e-03, 5.19143413e-03,
        2.18380346e-02, 5.14950166e-02]])
```

In [87]:

```
stat, p, dof, expected = chi2_contingency(y)
print('dof=%d' % dof)
print(expected)
```

```
dof=65
[[841.28843471 768.873734   649.9067257  838.85432713 928.0035175
  990.07326096]
 [ 53.15695312  48.58141791  41.06446722  53.00315362  58.6360604
   62.55794772]
 [264.94632785 242.14082115 204.67463157 264.1797562  292.25544302
  311.8030202 ]
 [ 33.53750986  30.65073685  25.90818121  33.44047547  36.99435988
   39.46873673]
 [278.19364425 254.24786221 214.90836315 277.38874401 306.86821517
  327.39317121]
 [ 47.62326399  43.52404633  36.78961732  47.48547517  52.53199102
   56.04560616]
 [646.9385651  591.25271393 499.76881557 645.06677179 713.62120201
  761.35193159]
 [ 42.08957487  38.46667475  32.51476742  41.96779671  46.42792164
   49.5332646 ]
 [  7.21056462   6.58990842   5.57025896   7.18970223   7.95378737
    8.4857784 ]
 [  4.19218873   3.83134211   3.23852265   4.18005943   4.62429498
    4.93359209]
 [ 26.32694524  24.06082843  20.33792225  26.25077324  29.0405725
   30.98295834]
 [487.63539329 445.66171387 376.70495482 486.22451331 537.8979926
  573.87543211]
 [ 14.25344169  13.02656316  11.01097701  14.21220207  15.72260295
   16.77421311]
 [ 17.60719267  16.09163685  13.60179514  17.55624962  19.42203894
   20.72108679]]
```

In [88]:

```
prob = 0.999
critical = chi2.ppf(prob, dof)
```

In [89]:

```
critical
```

Out[89]:

```
105.98814308961282
```

In [90]:

```
print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))
```

```
probability=0.999, critical=105.988, stat=9916.167
```

In [91]:

```
alpha = 1.0 - prob
print('significance=%.3f, p=%.5f' % (alpha, p))
```

```
significance=0.001, p=0.00000
```

In [ ]:

```

```