Semester Project - Spotify

Paul Larison

University of Arkansas Little Rock

STAT 7340: Advanced Statistical Methods

December 8, 2022

Table of Contents

Introduction

This project utilizes a preexisting Spotify data set to examine the relationship, if any, of explicit content, and danceability to popularity in specific musical genres. Tempo was also considered, but discarded due to lack of interesting results.

To this end, the following hypotheses will be tested on each of the selected genres:

Hypothesis One:

- H_0 : The popularity of explicit songs is approximately equal to that of non-explicit songs.
- H_1 : Explicit songs are more (or less?) popular than non-explicit songs.

Hypothesis Two:

- H_0 : Danceability has no effect on popularity
- H_1 : High danceability correlates with high popularity

Tools and Methods

This dataset is relatively well organized, but still needs manipulation in order to be properly analyzed. The primary tool I used was Python, specifically the Spyder IDE. I also used Excel for a few images of Tables.

- Python
  - Modules
    - Pandas
    - Scipy
    - MatPlotLib
    - Seaborn
    - Math
  - Load, sort and prune datasets
  - Create summary statistics
  - Plot visualizations
  - Perform statistical analysis
  - Validate statistical analysis
  - Export tables to csv
- Excel
  - CSV to PNG
  - Row level data view

Data

The data from this project is originally from the Spotify API. Spotify is an audio streaming service based in Sweden with approximately 456 million monthly active users. The API is intended to allow developers to play audio and display related data in 3rd party apps, but it can also allow for metadata collection of a large number of tracks, as has been done here. The dataset creator appears to have downloaded all metadata for the top 1000 songs in each genre

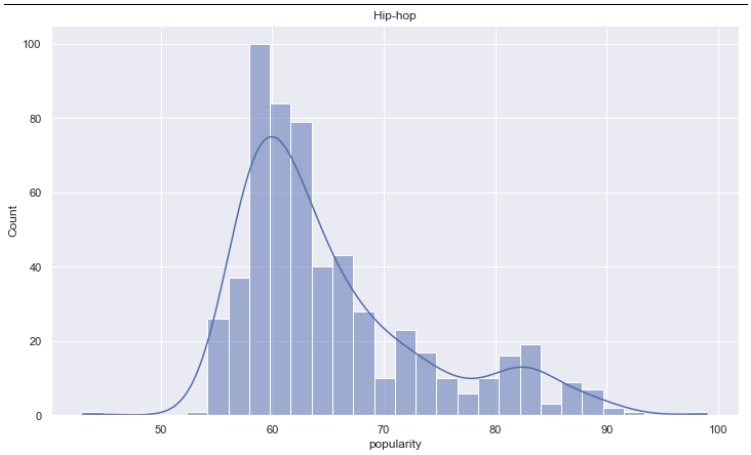
at a moment in time approximately two months ago (October, 2022). It was then uploaded to Kaggle, a machine learning community, for use in projects such as this one. It can be downloaded from https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset.

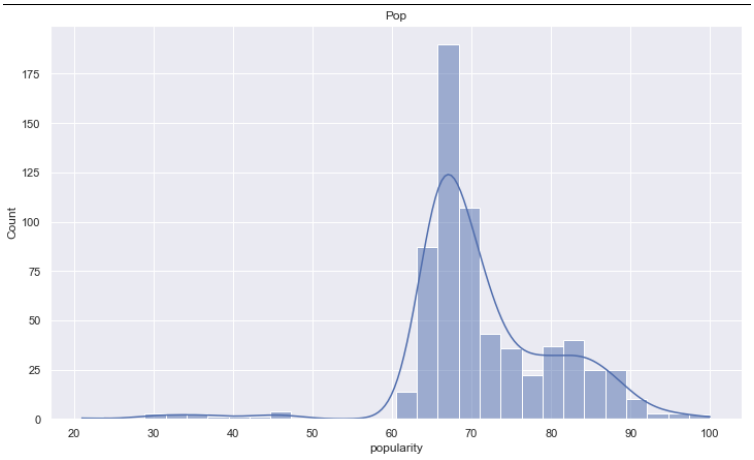
Data Structure

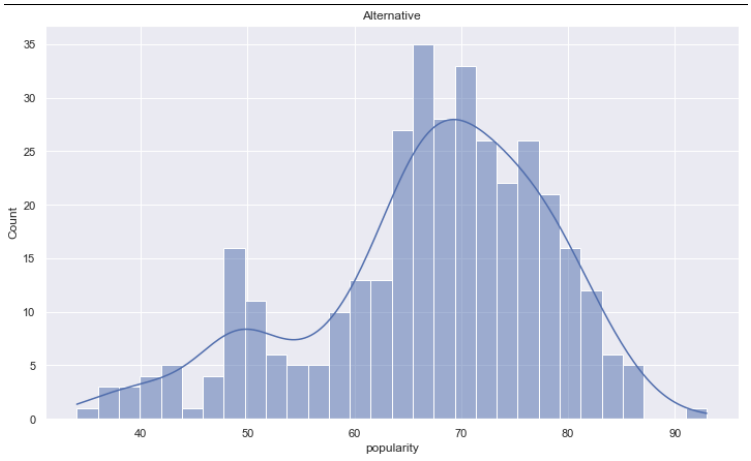
- dataset.csv
  - Description: A list containing 1000 song in each of 114 genres
  - Columns:
    - track_id - The Spotify ID for the track
    - artists - The artists' names who performed the track. If there is more than one artist, they are separated by a ;
    - album_name -The album name in which the track appears
    - track_name - Name of the track
    - popularity - The popularity of a track is a value between 0 and 100, with 100 being the most popular
    - duration_ms - The track length in milliseconds
    - explicit - Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown)
    - danceability - Danceability describes how suitable a track is for dancing
    - energy - Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity
    - key - The key the track is in (0=C)
    - loudness - The overall loudness of a track in decibels (dB)
    - mode - Mode indicates the modality (major or minor) of a track. Major is represented by 1 and minor is 0
    - speechiness - Speechiness detects the presence of spoken words in a track
    - acousticness - A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic
    - instrumentalness - Predicts whether a track contains no vocals
    - liveness - Detects the presence of an audience in the recording
    - valence - A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track
    - tempo - The overall estimated tempo of a track in beats per minute (BPM)
    - time_signature - An estimated time signature
    - track_genre - The genre in which the track belongs
  - Contains header
  - Record count: 114,000

Genre Histograms

Three genres were selected for analysis: Hip-hop, Alternative and Pop. These were chosen due to being popular genres that have a sufficient number of explicit songs, as well as having distributions that are at least close to unimodal. (Most genres have popularity distributions that are bimodal, and unpacking the reason for that is beyond the scope of this paper.)







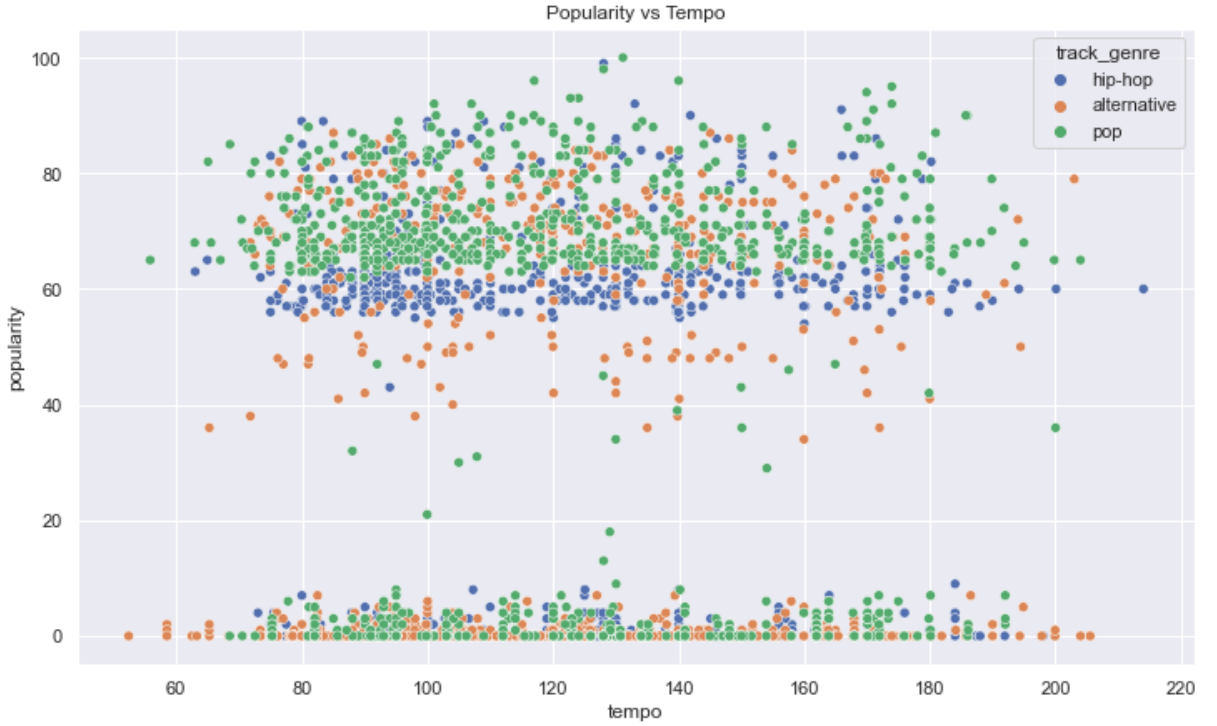
Data Preparation

This data contains many more columns that are used in this project, so the extra data is dropped, leaving artists, track_name, popularity, explicit, danceability, tempo and track_genre. Unused genres are then dropped, resulting in a much smaller dataframe to process.

The resulting dataframe has 3000 rows in 3 genres.

	artists	track_name	popularity	explicit	tempo	track_genre
count	3000	3000	3000	3000	3000	3000
unique	1185	2042		2		3
top	Justin Bieber	Mistletoe		FALSE		hip-hop
freq	58	31		2443		1000
mean			36.55733333		119.9763497	
std			34.42406694		30.41641369	
min			0		52.401	
25%			0		95.004	
50%			56		116.147	
75%			68		141.0005	
max			100		214.016	

A closer look reveals an issue with popularity scores, in that there are a number of songs with popularity 0 complicating analysis, as shown in the chart below:



Sorting by ascending popularity and then looking at the track-level data, we find that the zero popularity songs are well-known songs that are considered to be separate tracks by Spotify due to appearing on compilation albums of varying types. We can safely prune these entries.

	artists	album_name	track_name	popularity	explicit	tempo	track_genre
1499	Rob Zombie	Halloween Metal Nights	Demon Speeding	0	FALSE	133.984	alternative
2285	Justin Bieber	All I Want For Christmas Is You	Mistletoe	0	FALSE	161.948	pop
1149	Weezer	Alternative Christmas 2022	We Wish You A Merry Christmas	0	FALSE	149.806	alternative
1148	Weezer	Christmas Rock 2022	We Wish You A Merry Christmas	0	FALSE	149.806	alternative
1147	Weezer	Alternative Christmas 2022	We Wish You A Merry Christmas	0	FALSE	149.806	alternative
1146	Weezer	Rock n Roll Christmas	We Wish You A Merry Christmas	0	FALSE	149.806	alternative
1145	Skillet	Metal Lives!	Feel Invincible	0	FALSE	126.027	alternative
1143	Mon Laferte;Enrique Bunbury	noches de lluvia	Mi Buen Amor	0	FALSE	98.741	alternative
1142	Mon Laferte;Enrique Bunbury	Rock un sentimiento vol. I	Mi Buen Amor	0	FALSE	98.741	alternative
1139	Weezer	Christmas Rock 2022	O Come All Ye Faithful	0	FALSE	121.907	alternative
1138	The Smashing Pumpkins	Alternative Christmas 2022	Christmastime	0	FALSE	77.983	alternative
1137	Disturbed	Just Best Covers	If I Ever Lose My Faith In You	0	FALSE	173.861	alternative
1136	Skillet	Rock Brandneu	Surviving the Game	0	FALSE	103.996	alternative
1135	Weezer	Rock n Roll Christmas	O Come All Ye Faithful	0	FALSE	121.907	alternative
1133	Skillet	Top of the Rock	Surviving the Game	0	FALSE	103.996	alternative
1132	Disturbed	Metal Lives!	Hey You	0	FALSE	103.033	alternative
1131	Weezer;AJR	All Out Alternative	All My Favorite Songs	0	FALSE	84.962	alternative
1130	Mon Laferte	noches de lluvia	Amor Completo	0	FALSE	175.567	alternative
2284	Justin Bieber	Christmas Faves 2022	Mistletoe	0	FALSE	161.948	pop
1129	The Smashing Pumpkins	Alternative Christmas 2022	Christmastime	0	FALSE	77.983	alternative

Scatterplot after trim:

Description of the trimmed dataframe:

	artists	album_name	track_name	popularity	explicit	tempo	track_genre
count	1591	1591	1591	1591	1591	1591	1591
unique	899	1164	1437		2		3
top	Sidhu Moose Wala	Moosetape	STAY (with Justin Bieber)		FALSE		pop
freq	46	27	4		1390		660
mean				68.20427404		118.8393765	
std				9.950336138		30.20224438	
min				21		55.832	
25%				62		93.9985	
50%				67		114.089	
75%				74		140.0475	
max				100		214.016	

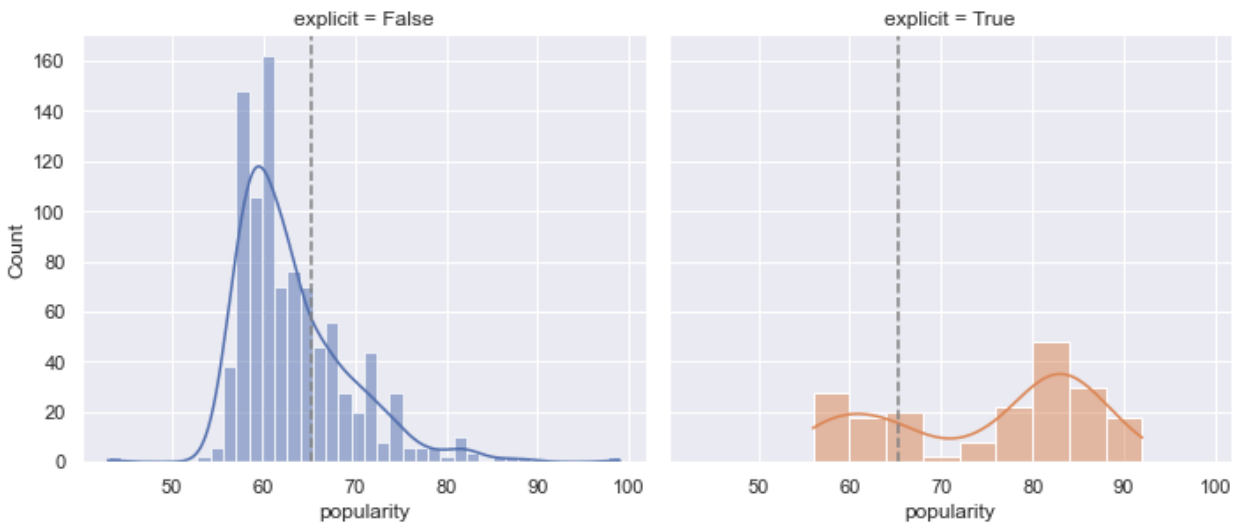
This is the data analyzed in the next section. For the first hypothesis, one more split is performed, forming explicit and non-explicit datasets for comparison.

Statistical Analysis

Hypothesis One:

H_0 : The popularity of explicit songs is approximately equal to that of non-explicit songs.
vs
H_1 : Explicit songs are more (or less?) popular than non-explicit songs.

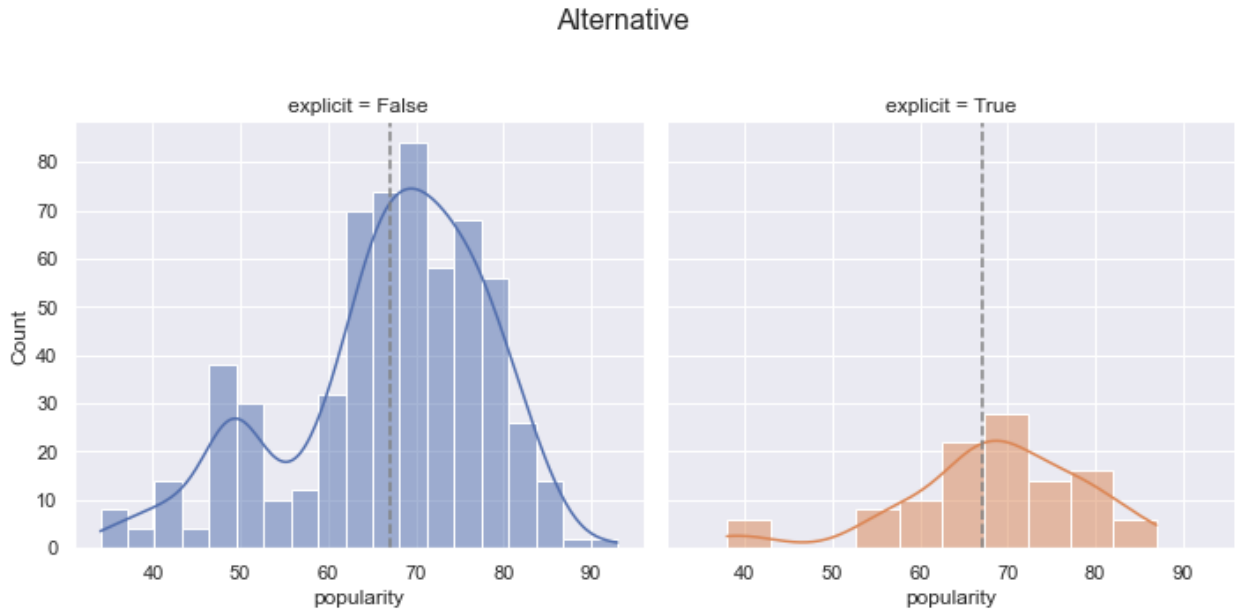
I used Welch's t-test to account for different variances. The popularity histogram for most genres is bimodal, so I selected genres that minimize this. I first tested the hypothesis using SciPy's ttest_ind function, then confirmed the calculation using manually coded formulae for standard error, degrees of freedom, and the t and p values. This code is available in Appendix section 2.



Welch's Two Sample T-test
Genre: Hip-hop
Data: Explicit=True and Explicit=False for the selected genre
Explicit=True Info:
Track Count (N): 194 Mean Popularity: 75.1340206185567 StDev: 11.074527432090829 SEM: 0.7951047322705559
Explicit=False Info:
Track Count (N): 952 Mean Popularity 63.260504201680675 StDev: 6.244862869483796 SEM: 0.20239716825360254

Calculate using SciPy:
t = 14.471763228102079 p-value = 9.700078932659345e-34
Reject (Popularity IS affected by explicit lyrics
Mean of Explicit: 75.13 , Mean of Non-explicit: 63.26
Explicit lyrics make hip-hop songs more popular

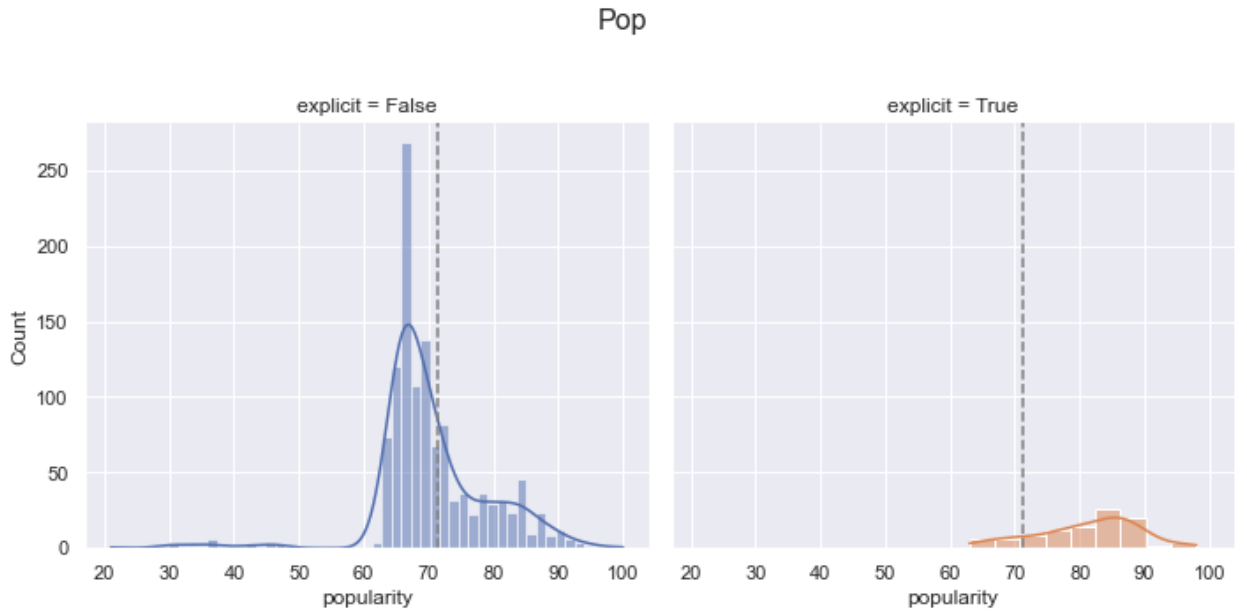
Check results using "manual" calculations:
t-statistic: 14.471763228102079 df: 218.63600340545466 p-statistic: 9.700078932659347e-34
Calculations match to at least 5 decimal places.




```
------------------------------------------------------------------
Welch's Two Sample T-test
Genre: Alternative
Data: Explicit=True and Explicit=False for the selected genre
Explicit=True Info:
Track Count (N): 110 Mean Popularity: 68.43636363636364 StDev:
10.620671615045296 SEM: 1.0126413057608254
Explicit=False Info:
Track Count (N): 606 Mean Popularity 66.85148514851485 StDev:
11.35313213879483 SEM: 0.4611894675922753

Calculate using SciPy:
t =  1.4243326173262696 p-value =  0.15632621325272786
Accept (Popularity IS NOT affected by explicit lyrics)
Mean of Explicit:  68.44 , Mean of Non-explicit:  66.85

Check results using "manual" calculations:
t-statistic:  1.4243326173262696 df:  157.6844420917497 p-statistic:
0.15632621325272786
Calculations match to at least 5 decimal places.
------------------------------------------------------------------
```



Welch's Two Sample T-test
Genre: Pop
Data: Explicit=True and Explicit=False for the selected genre
Explicit=True Info:
Track Count (N): 98 Mean Popularity: 81.18367346938776 StDev:
8.006887420235143 SEM: 0.8088177701493616
Explicit=False Info:
Track Count (N): 1222 Mean Popularity 70.56464811783961 StDev:
9.07609079271201 SEM: 0.25963499507775833

Calculate using SciPy:
t = 12.500789546238018 p-value = 2.3869956923136562e-23
Reject (Popularity IS affected by explicit lyrics
Mean of Explicit: 81.18 , Mean of Non-explicit: 70.56
Explicit lyrics make pop songs more popular

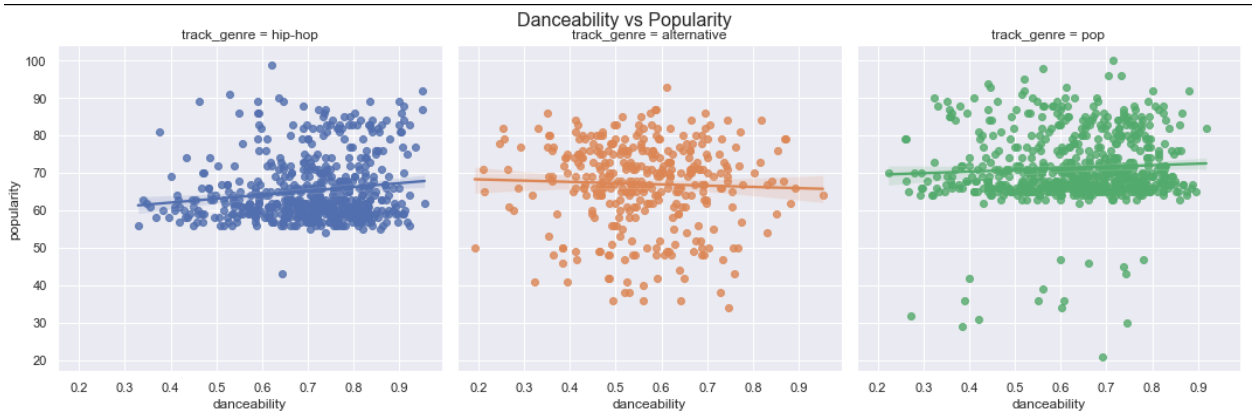
Check results using "manual" calculations:
t-statistic: 12.50078954623802 df: 117.92113790038928 p-statistic:
2.3869956923136057e-23
Calculations match to at least 5 decimal places.

Hypothesis Two:

H_0 : Danceability has no effect on popularity

H_1 : High danceability correlates with high popularity

Pearson's Correlation Coefficient is used to test for correlation between two independent variables, in this case danceability and popularity. The correlation value, r, has a value between 0 and 1. Values close to 1 show that the variables are highly correlated. The p-value is used to assess the accuracy of the test




```
---------------------------------------------------------------------
Pearson's Correlation Coefficient
Genre: Hip-hop
Data: Danceability and Popularity for the selected genre
Danceability Info:
Track Count (N): 573 Mean Danceability: 0.7050628272251307 StDev:
0.1226813894548242 SEM: 0.00512508856920297
Popularity Info:
Track Count (N): 573 Mean Popularity 65.27050610820244 StDev: 8.54227164244598
SEM: 0.35685851736988966

Calculate using SciPy:
Correlation Coefficient =  0.14918219196041682 P-value =
0.0003392684324033826
Reject (Danceability IS correlated with Popularity)

Check results using "manual" calculations:
r-statistic: 0.1491821919604167 t-statistic: 3.605141446412455 DF: 571
p-statistic: 0.0003392684324034552
Calculations match to at least 5 decimal places.
```

```
----------------------------------------------------------------
Pearson's Correlation Coefficient
Genre: Alternative
Data: Danceability and Popularity for the selected genre
Danceability Info:
Track Count (N): 358 Mean Danceability: 0.5596312849162014 StDev:
0.12988233556316103 SEM: 0.006864494734839647
Popularity Info:
Track Count (N): 358 Mean Popularity 67.09497206703911 StDev:
11.258954848694076 SEM: 0.5950542538640565

Calculate using SciPy:
Correlation Coefficient =  -0.03873434395269703 P-value =  0.4650239307657357
Accept (Danceability IS NOT correlated with Popularity)

Check results using "manual" calculations:
r-statistic: -0.03873434395269703 t-statistic: -0.7313870140119872 DF: 356
p-statistic: 0.465023930765622
Calculations match to at least 5 decimal places.

----------------------------------------------------------------
Pearson's  Correlation Coefficient
Genre: Pop
Data: Danceability and Popularity for the selected genre
Danceability Info:
Track Count (N): 660 Mean Danceability: 0.6301787878787877 StDev:
0.13927482303836428 SEM: 0.005421265134182477
Popularity Info:
Track Count (N): 660 Mean Popularity 71.35303030303031 StDev:
9.423016403879943 SEM: 0.36679041606186236

Calculate using SciPy:
Correlation Coefficient =  0.06264721716232217 P-value =  0.10784284228976711
Accept (Danceability IS NOT correlated with Popularity)

Check results using "manual" calculations:
r-statistic: 0.06264721716232215 t-statistic: 1.6101585397111977 DF: 658
p-statistic: 0.10784284228978491
Calculations match to at least 5 decimal places.

----------------------------------------------------------------
```

Conclusion

The results of the analysis are as follows, with 95% confidence:

	Hip-hop	Alternative	Pop
Explicit = Popular?	Yes	No	Yes
Danceable = Popular?	Yes	No	No

Explicit lyrics strongly correlated to more popular songs in both the pop and hip-hop genres, which saw 11 and 12 point increases, respectively, for the expected popularity of explicit songs over non-explicit songs. This pattern did not hold for alternative songs, which, with popularity means only a point and apart, did not have enough separation to disprove the null hypothesis.

Danceability is a useless measure for predicting Alternative and Pop songs. Not only does the correlation coefficient show poor or negative correlation, but the P-values for these genres indicate that r is not to be trusted. There does seem to be a slight correlation between danceability and popularity for hip-hop songs; the r value is small, at only 0.149, but with a p-value of 0.0003, the formula is confident in this prediction. Comparing coefficient calculations with mean danceability suggests that genres that are already more danceable might gain more benefit from high danceability.

Appendix: Python Code

Section 1 - Data Formatting and Summary Statistics

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path


#Suppress scientific notation
pd.set_option('display.float_format', lambda x: '%.5f' % x)
#Show all columns
pd.options.display.max_columns = None

###################load data#############################
workfolder = Path('E:\p.larisonFolders\Documents\AschoolFolder')
df0 = pd.read_csv(workfolder / 'spotifydataset.csv')
dfx = df0

#
df0=df0[['artists','album_name','track_name','popularity','explicit','tempo','
track_genre']]
df=pd.DataFrame()
dftemp=pd.DataFrame()

alpha = 0.95
genrelist = ['hip-hop','alternative','pop']

sns.set(rc={'figure.figsize':(12,7)})

for i in genrelist:
      dftemp = df0.loc[df0.track_genre == i]
      df = pd.concat([df,dftemp])
df.reset_index(drop=True,inplace=True)

sns.scatterplot(data=df, y=df.popularity, x=df.tempo, hue=df.track_genre)
plt.title('Popularity vs Tempo')
plt.show()

dfx_desc=dfx.describe(include='all')
df_desc=df.describe(include='all')
```

```
df.sort_values(by=['popularity'], inplace=True)
df_head=df.head(20)

df = df[df.popularity >= 20]
df.sort_index()

sns.scatterplot(data=df, y=df.popularity, x=df.tempo, hue=df.track_genre)
plt.title('Popularity vs Tempo, After Trim')
plt.show()

dftrim_desc=df.describe(include='all')
#Export to office software
dfx_desc.to_csv(workfolder / 'dfx_desc.csv')
df_desc.to_csv(workfolder / 'df_desc.csv')
df_head.to_csv(workfolder / 'df_head.csv')
dftrim_desc.to_csv(workfolder / 'dftrim_head.csv')
```

Section 2 - Hypothesis 1 Calculations and Visualizations

```
import pandas as pd
import math
import matplotlib.pyplot as plt
from scipy import stats as st
import seaborn as sns
from pathlib import Path


#Suppress scientific notation
pd.set_option('display.float_format', lambda x: '%.5f' % x)

###################load data#############################
workfolder = Path('E:\p.larisonFolders\Documents\AschoolFolder')
df = pd.read_csv(workfolder / 'spotifydataset.csv')


df = df[df.popularity >= 20]
df=df[['artists','track_name','popularity','explicit','danceability','tempo','
track_genre']]
dftemp=pd.DataFrame()

alpha = 0.95
genrelist = ['hip-hop','alternative','pop']
sns.set(rc={'figure.figsize':(12,7)})
```

```
for i in genrelist:
      dftemp = df.loc[df.track_genre == i]
      df = pd.concat([df,dftemp])
      sns.histplot(data=dftemp, x=dftemp.popularity, kde=True, bins=30)
      plt.title(i)
      plt.show()
df.reset_index(drop=True,inplace=True)


for i in genrelist:
      dftemp = df.loc[df.track_genre == i]
      sea = sns.FacetGrid(dftemp, col='explicit',hue='explicit', height=5)
      sea.map(sns.histplot,'popularity',kde=True)
      sea.refline(x=dftemp.popularity.mean())
      sea.fig.subplots_adjust(top=0.8)
      sea.fig.suptitle(i.capitalize(), fontsize=16)
      plt.show

      yesX = dftemp.loc[dftemp.explicit==True]
      nonX = dftemp.loc[dftemp.explicit==False]

      track_count_y = len(yesX)
      pop_mean_y = yesX.popularity.mean()
      pop_sdev_y = st.tstd(yesX.popularity)
      pop_sem_y = st.sem(yesX.popularity)

      track_count_n = len(nonX)
      pop_mean_n = nonX.popularity.mean()
      pop_sdev_n = st.tstd(nonX.popularity)
      pop_sem_n = st.sem(nonX.popularity)

      t,p = st.ttest_ind(yesX.popularity, nonX.popularity, equal_var=False)

      SE =
math.sqrt(((pop_sdev_y**2)/track_count_y)+((pop_sdev_n**2)/track_count_n))
      DF = ((((pop_sdev_y**2)/track_count_y) +
((pop_sdev_n**2)/track_count_n))**2) /
(((((pop_sdev_y**2)/track_count_y)**2)/(track_count_y-1)) +
((((pop_sdev_n**2)/track_count_n)**2)/(track_count_n-1)))

      t1 = ((pop_mean_y-pop_mean_n)/SE)
```

```
      p1 = st.t.sf(abs(t1),df=DF)*2


print('------------------------------------------------------------------')

      print('Welch\'s Two Sample T-test')

      print('Genre: ' + i.capitalize())
      print('Data: Explicit=True and Explicit=False for the selected genre' )
      print('Explicit=True Info: ')
      print('Track Count (N):',track_count_y,'Mean
Popularity:',pop_mean_y,'StDev:',pop_sdev_y,'SEM:',pop_sem_y)
      print('Explicit=False Info: ')
      print('Track Count (N):',track_count_n,'Mean
Popularity',pop_mean_n,'StDev:',pop_sdev_n,'SEM:',pop_sem_n)

      print()
      print('Calculate using SciPy:')
      print('t = ',t,'p-value = ',p)
      if p <= ((1-alpha)/2):
      print('Reject (Popularity IS affected by explicit lyrics')
      print('Mean of Explicit: ', round(pop_mean_y,2), ', Mean of
Non-explicit: ', round(pop_mean_n,2))
      if pop_mean_y > pop_mean_n:
            print('Explicit lyrics make',i,'songs more popular')
      else:
            print('Explicit lyrics make',i,'songs less popular')

      else:
      print('Accept (Popularity IS NOT affected by explicit lyrics)')
      print('Mean of Explicit: ', round(pop_mean_y,2), ', Mean of
Non-explicit: ', round(pop_mean_n,2))
      print()
      print('Check results using "manual" calculations:')
      print('t-statistic: ',t1,'df: ',DF,'p-statistic: ',p1)
      if round(t1,5) == round(t,5) and round(p1,5) == round(p,5):
      print('Calculations match to at least 5 decimal places.')
      else:
      print('ERROR: Calculations do not match.')

      print()
```

Section 3 - Hypothesis 2 Calculations and Visualizations

```
import pandas as pd
import math
import seaborn as sns
from scipy import stats as st
from pathlib import Path

#Suppress scientific notation
pd.set_option('display.float_format', lambda x: '%.5f' % x)

###################load data#############################
workfolder = Path('E:\p.larisonFolders\Documents\AschoolFolder')
df0 = pd.read_csv(workfolder / 'spotifydataset.csv')
df0 = df0[df0.popularity >= 20]
df0=df0[['artists','album_name','track_name','popularity','explicit','danceabi
lity','tempo','track_genre']]
df=pd.DataFrame()
dftemp=pd.DataFrame()

alpha = 0.95
genrelist = ['hip-hop','alternative','pop']

sns.set(rc={'figure.figsize':(6,6)})

for i in genrelist:
      dftemp = df0.loc[df0.track_genre == i]
      df = pd.concat([df,dftemp])
df.reset_index(drop=True,inplace=True)

for i in genrelist:
      dftemp = df.loc[df.track_genre == i]

      track_count = len(dftemp)
      pop_mean = dftemp.popularity.mean()
      pop_sdev = st.tstd(dftemp.popularity)
      pop_sem = st.sem(dftemp.popularity)
      pop_CIlo, pop_CIhi = st.norm.interval(alpha, loc=pop_mean)

      dance_mean = dftemp.danceability.mean()
      dance_sdev = st.tstd(dftemp.danceability)
      dance_sem = st.sem(dftemp.danceability)
      dance_CIlo, dance_CIhi = st.norm.interval(alpha, loc=dance_mean)
```

```
      #Pearson's SciPy
      r, p = st.pearsonr(dftemp.danceability,dftemp.popularity)

      #Pearson's manually
      x = list(dftemp.danceability)
      y = list(dftemp.popularity)

      x_mean = st.tmean(x)
      y_mean = st.tmean(y)
      dFree = len(x)-2

      top = float()
      bx = float()
      by = float()

      for j in range(len(x)):
      top = top + (x[j]-x_mean)*(y[j]-y_mean)
      bx = bx + (x[j]-x_mean)**2
      by = by + (y[j]-y_mean)**2

      bx = math.sqrt(bx)
      by = math.sqrt(by)
      bt = bx * by
      r1 = top/bt
      t1 = (r1 * math.sqrt(dFree)) / math.sqrt(1-(r1**2))
      p1 = st.t.sf(abs(t1),df=dFree)*2



print('-----------------------------------------------------------------')

      print('Pearson\'s  Correlation Coefficient')
      print('Genre: ' + i.capitalize())
      print('Data: Danceability and Popularity for the selected genre' )
      print('Danceability Info: ')
      print('Track Count (N):',track_count,'Mean
Danceability:',dance_mean,'StDev:',dance_sdev,'SEM:',dance_sem)
      print('Popularity Info: ')
      print('Track Count (N):',track_count,'Mean
Popularity',pop_mean,'StDev:',pop_sdev,'SEM:',pop_sem)
```

```
      print()
      print('Calculate using SciPy:')
      print('Correlation Coefficient = ',r,'P-value = ',p)
      if p <= ((1-alpha)):
      print('Reject (Danceability IS correlated with Popularity')
      else:
      print('Accept (Danceability IS NOT correlated with Popularity)')
      print()

      print('Check results using "manual" calculations:')
      print('r-statistic:',r1,'t-statistic:',t1,'DF:',dFree,'p-statistic:',p1)
      if round(r1,5) == round(r,5) and round(p1,5) == round(p,5):
      print('Calculations match to at least 5 decimal places.')
      else:
      print('ERROR: Calculations do not match.')

      print()


print('---------------------------------------------------------------------')
      print()


sea = sns.FacetGrid(df, col='track_genre',hue='track_genre', height=5)
sea.map(sns.regplot,'danceability','popularity')
sea.fig.subplots_adjust(top=0.9)
sea.fig.suptitle('Danceability vs Popularity', fontsize=16)
```