musicanalysisreport

November 28, 2020

[2]:
```
# modules for research report
from datascience import *
import numpy as np
import random
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')

# module for YouTube video
from IPython.display import YouTubeVideo

# okpy config
from client.api.notebook import Notebook
ok = Notebook('music-final-project.ok')
_ = ok.auth(inline=True)
```

```
=====================================================================
Assignment: Final Project: Free Music Archive
OK, version v1.12.5
=====================================================================

Successfully logged in as vaibhavippili@berkeley.edu
```

1 Analysis Research Report Using the Free Music Archive

1.1 Introduction

For this report, we have done an in-depth analysis of the music dataset from the Free Music Archive. The information in this dataset, collected by Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, includes information about 106,574 songs from 16,341 artists. Information about the songs were collected through the Spotify API. The dataset has been split into three tables: tracks, genres, and features.

The tracks table includes information about individual songs, including the name, duration, and artist of the song. The track duration feature of this table is particularly interesting, as there may be a relationship between it and other features of the songs. The genre table includes information about all possible genres in the dataset, as well as the number of songs in each genre and the parent

genre of subgenres. The features table is the most detailed of the three tables, and includes features about individual songs calculated by Spotify. These features include the danceability (how likely an individual is to dance to a song,), tempo, and valence (how positive or cheerful a song is) of a song. The units for these variables are largely based on a scale from 0-1, with the exception of tempo, where lower values indicate a lower presence of the variable and a higher value represents a higher presence of the variable. These features are ideal for further study as studies can be conducted to find relationships between these features and each other, different genres, and different artists. In particular, we are interested in looking into how danceability is affected by and plays a role in the features of a song. For example, we can look into the danceability of different genres, or how variables like valence and liveness affect the danceability of a song.

1.2 Hypothesis Testing and Prediction Questions

Please bold your hypothesis testing and prediction questions.

After joining all the tables provided in the dataset, we will proceed in studying the theme of our report, the relationship between danceability and other variables such as different genres and other musical aspects such as valence.

Specifically, we are interested in whether the difference between the danceability of Hip-Hop genre songs and Folk genre songs are statistically significant. **Our null hypothesis for this hypothesis test is that in the sample, the distribution of danceability between Hip-Hop and Folk music is the same. Any difference in the sample is solely due to random chance. Our alternative hypothesis is that the danceability score of the Hip-Hop genre is higher than the Folk genre, on average.**

Also, we are curious to see whether we can use the valence score to determine the danceability of a certain song. We will conduct a linear regression and compute the correlation between these two variables to determine whether there is a linear relationship between valence and danceability.

1.3 Exploratory Data Analysis

You may change the order of the plots and tables.

Before conducting our Hypothesis Test and our Prediction, we will further examine the data provided to look for prominent trends.

Table Requiring a Join Operation:

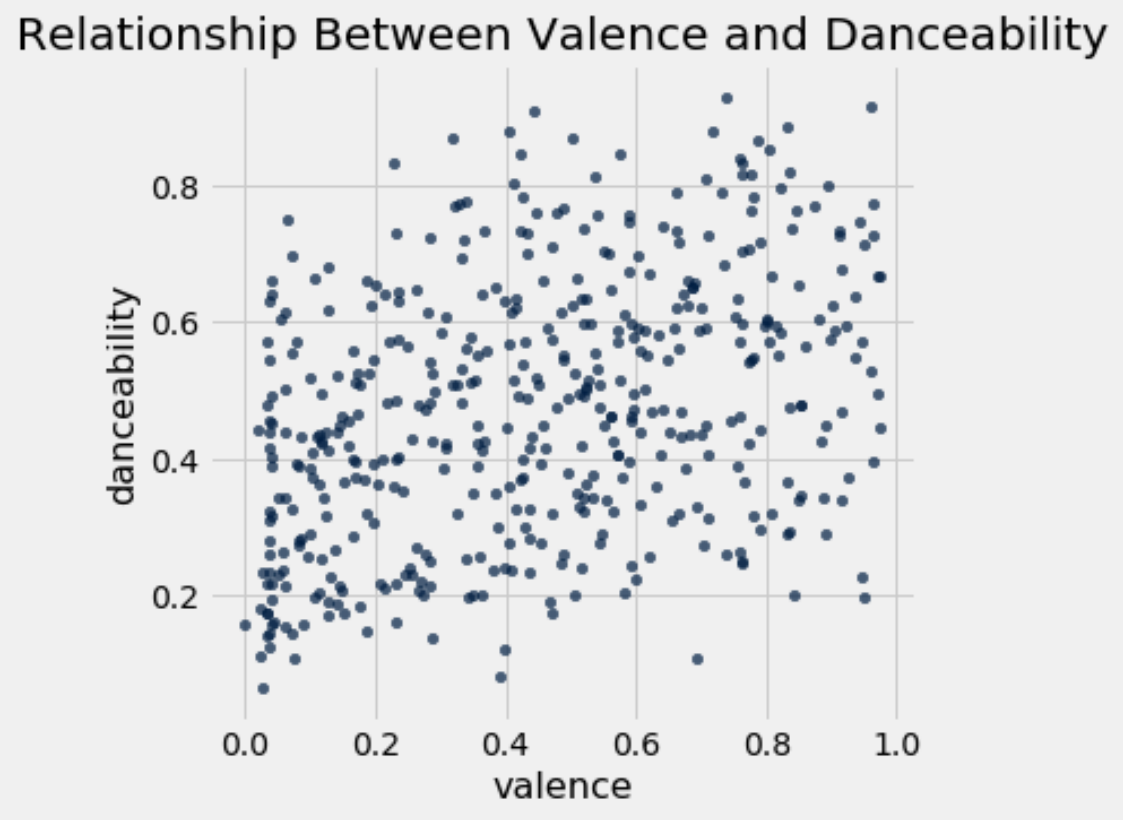
[6]:
```
# Use this cell to join two datasets
all_data = features.join('track_id', tracks).join('track_genre', genres,␣
 ↪'title')

all_genre_danceability = all_data.select('track_genre','valence').
 ↪group('track_genre', np.mean).sort(1)
all_genre_danceability.show()
```

```
<IPython.core.display.HTML object>
```

First, we look to see whether there is significant variance in the valence variable. In order for the valence variable to be an accurate predictor for danceability, it needs to have a lot of variability. In the table, it appears that there is indeed a significant amount of variability in the variance across different genres. This indicates that valence has the potential to be an effective predictor of danceability.

Quantitative Plot:

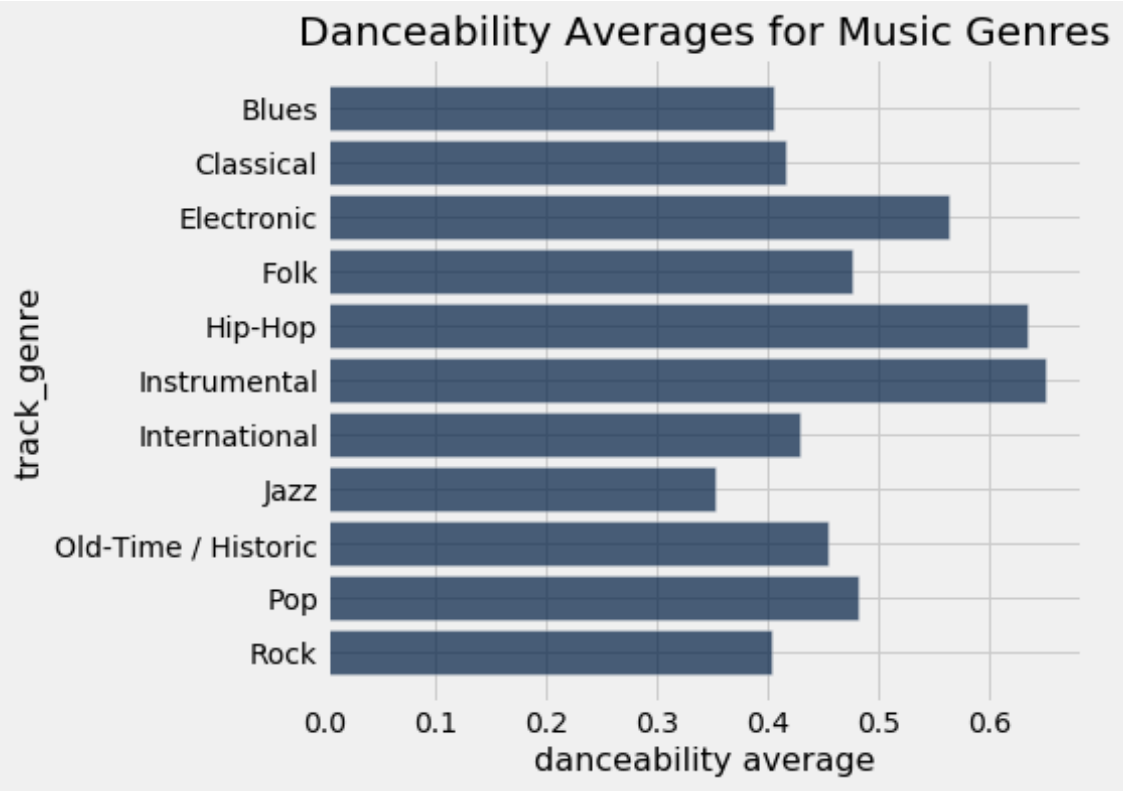
[7]:
```
# Use this cell to generate your quantitative plot
features.scatter('valence', 'danceability')
plots.title('Relationship Between Valence and Danceability');
```



For the quantitative plot, we are exploring the section of the data we intend to investigate thoroughly in our prediction section. We created a scatter plot between the variables valence, the positive or negative emotion of a song, and danceability, a track's suitability for dancing. Based on this scatter plot, we can see there is a weak positive linear association between the two variables. In the linear regression between these two variables that we will conduct in the prediction section of this exploratory data report, we will be able to determine the correlation between these two variables in more detail.

Qualitative Plot:

[8]:
```
# Use this cell to generate your qualitative plo# Use this cell to generate␣
 ↪your qualitative plot
genre_average = all_data.select('track_genre', 'danceability').group(0, np.
 ↪average).barh(0,1)
plots.title('Danceability Averages for Music Genres');
```



Before beginning our hypothesis test, we wanted to visualize the difference between danceability between different genres. By creating this bar graph, we were able to compare the average danceability between each of the genres. Using this graph we can visualize the observed difference in danceability between both genres before conducting a A/B Hypothesis test to see if this difference is statistically significant.

Aggregated Data Table:

[9]:
```
# Use this cell to generate your aggregated data table
all_data.select('track_genre').group('track_genre').sort('count').show()
```

```
<IPython.core.display.HTML object>
```

As we plan to conduct an A/B test between the difference of the danceability of Folk versus Hip-Hop songs, we need to first determine whether there is a similar sample size between the two variables, as if there is a very large difference in sample size between the two genres, our analysis might result in an inaccurate and unreliable conclusion.

To do this, we used the group function to find the number of songs in each genre of music. We were able to proceed with our hypothesis test as both folk and hip-hop songs had a similar sample size. While we were exploring the data, we also considered testing the difference in danceability between the Rock and Pop genre, but we had to remove this potential idea as the difference between sample sizes of those two categories was far too great.

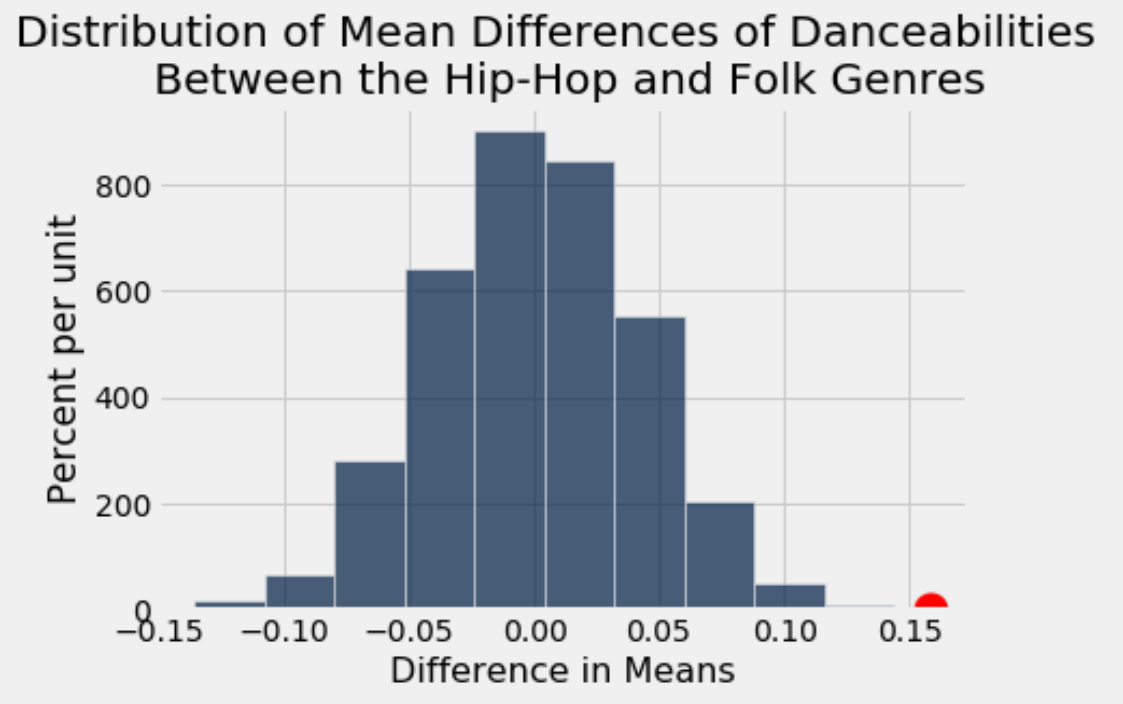
1.4 Hypothesis Testing

Do not copy code from demo notebooks or homeworks! You may split portions of your code into distinct cells. Also, be sure to set a random seed so that your results are reproducible.

Here we are performing a hypothesis test to determine whether the danceability of Hip-Hop songs are statistically more significant than Folk genre songs or this difference is due to chance. To do this, we will be conducting an A/B Hypothesis test. Our test statistic that we will be measuring as we conduct this experiment is the mean difference between danceability of Folk and Hip-Hop songs. To determine whether the result of this test is statistically significant, we are using a P-Value significance cutoff of 5%. If we observe that the observed difference is larger than the simulated distribution, we will favor the alternative hypothesis.

[10]:
```
# set the random seed so that results are reproducible
random.seed(1231)

#calculated the average danceability for both the Folk and Hip-Hop genres.
Folk_Hop_average = all_data.select('track_genre', 'danceability').group(0, np.
 ↪average).take(3,4)

#Found the observed difference in the mean between the averages of the␣
 ↪danceability of Folk and Hip-Hop.
observed_value = Folk_Hop_average.column(1).item(1) - Folk_Hop_average.
 ↪column(1).item(0)

#Created a table with the genres and danceability of each song to reference in␣
 ↪the future.
Folk_and_Hop = all_data.select('track_genre', 'danceability').
 ↪where('track_genre', are.contained_in(make_array('Folk', 'Hip-Hop')))

#Defined a function that shuffles the genres of the Pop and Rock table and␣
 ↪returns a simulated difference of means.
def new_shuffle():
    """Conducts one shuffle and returns one new difference of means between the␣
 ↪danceability of Folk and Hip-Hop. """

    #Creates an array of shuffled genre labels
    value_shuffled = Folk_and_Hop.sample(with_replacement = False).column(0)
```

```
    #New table with shuffled labels and danceability scores from original table.
    new_shuffled_table = Table().with_columns('Track Genre', value_shuffled,'Danceability', Folk_and_Hop.column(1))

    #Calculates difference of mean from the new created table.
    shuffled_mean_values = new_shuffled_table.group('Track Genre', np.mean).column(1)
    shuffled_mean = shuffled_mean_values.item(1) - shuffled_mean_values.item(0)
    return shuffled_mean

#Performing an A/B Hypothesis Test with 5000 trials, using the previously defined function.
shuffled_means = make_array()
repetitions = 5000

for i in np.arange(repetitions):
    new_mean = new_shuffle()
    shuffled_means = np.append(shuffled_means, new_mean)

#Create a Histogram to visualize our simulated values and compare it with the observered value.
results = Table().with_columns('Difference in Means', shuffled_means).hist()
plots.plot(observed_value, 0, 'ro', markersize = 15 )
plots.title('Distribution of Mean Differences of Danceabilities \n Between the Hip-Hop and Folk Genres');
```



[11]:
```
#Calculating the P-Value.
p_value = np.count_nonzero(observed_value <= shuffled_means)/ repetitions
print('As our P-Value of ' + str(p_value) + ' is less than the significane␣
 ↪level of 5%, we reject the null hypothesis.')
```

```
As our P-Value of 0.0 is less than the significane level of 5%, we reject the
null hypothesis.
```

As our observed value was deemed to be 0.1586, and we have a P-Value of 0, we have significant evidence that there is a difference of danceability between Hip-Hop and Folk genre songs, and that Hip-Hop songs have a higher danceability score, on average. As our P-Value is less than our predetermined significance level cutoff of 5%, we reject our null hypothesis that there is no statistically significant difference in danceability between the two genres.

1.5 Prediction

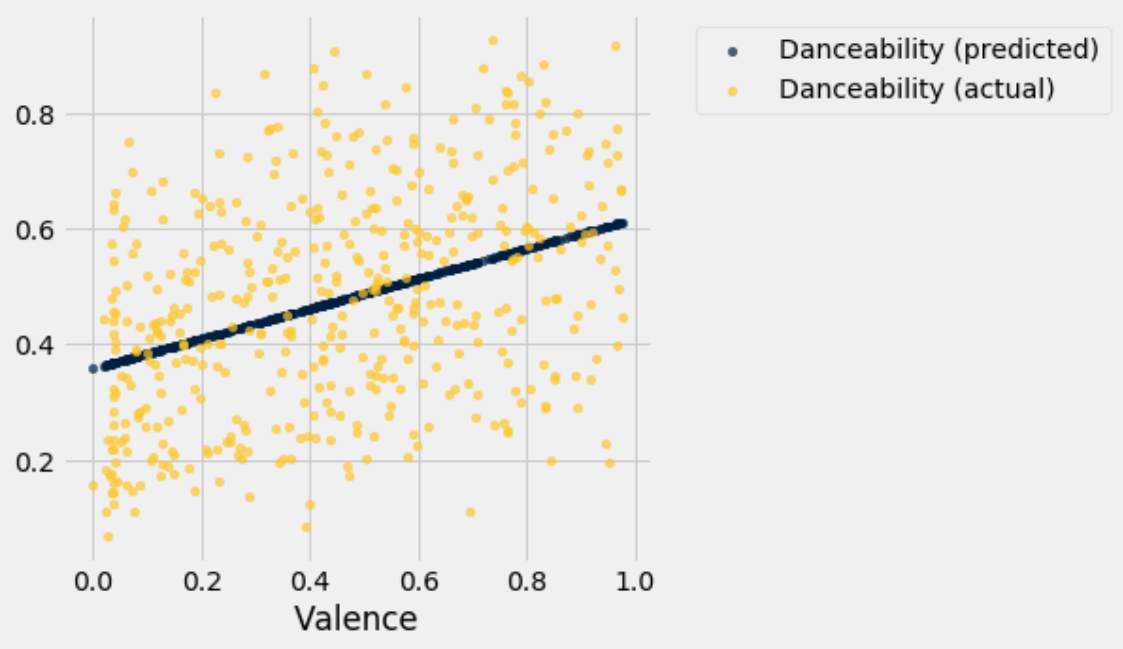
Be sure to set a random seed so that your results are reproducible.

For our prediction, we will be attempting to predict the danceability of a song based on the valence, or the positivity, of a song. In order to achieve this, we will be employing a linear regression model that will be fitted to the danceability and valence of songs in the dataset. We will then evaluate how well the linear regression can predict the danceability of songs. From a logical standpoint, it would be reasonable to believe that the positivity of a song may contribute to how likely one might dance. However, due to the trends in the exploratory data analysis, this relationship may not be as strong as we might expect.

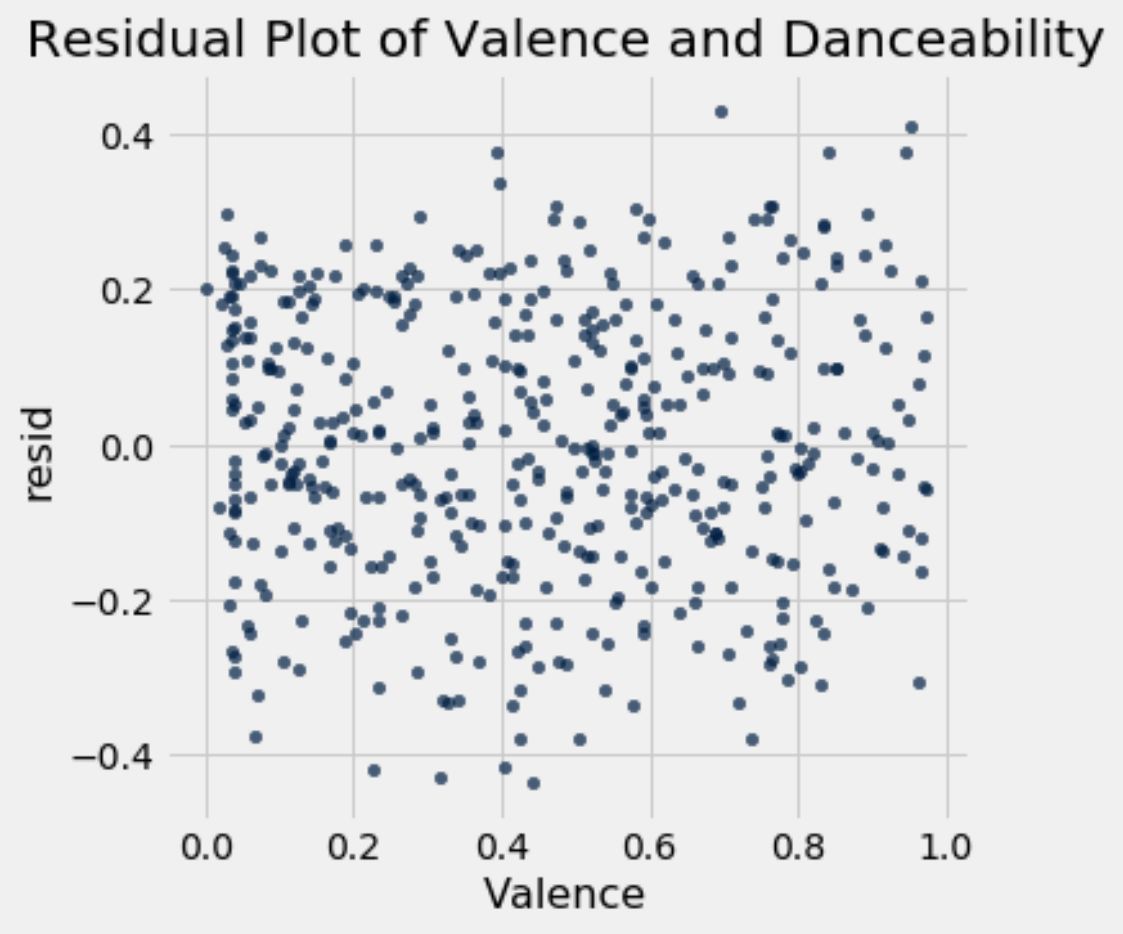
[12]:
```
# set the random seed so that results are reproducible
random.seed(1231)

# set x to the valence values and set y to the danceability values
x = features.column('valence')
y = features.column('danceability')

def standard_units(arr):
    '''return standard units of array'''
    return (arr-np.mean(arr))/np.std(arr)

def correlation(x,y):
    '''return correlation coefficient'''
    return np.mean(standard_units(x)*standard_units(y))

def slope_and_intercept(x,y):
    '''return slope and intercept'''
    slope = correlation(x,y)*(np.std(y)/np.std(x))
    intercept = np.mean(y) - slope*np.mean(x)
```

```
    return slope,intercept

#set line to the features of the line
line = slope_and_intercept(x, y)
preds =  line[0]*x + line[1]
print(f'Correlation: {correlation(x,y)}')

#display scatter plot with predicted and actual values
pred_table = Table().with_columns('Danceability (predicted)', preds,␣
 ↪'Danceability (actual)', y, 'Valence', x)
pred_table.scatter('Valence')
```

```
Correlation: 0.3762155481339832
```



[13]:
```
#Display residual plot.
preds_table_with_resid = pred_table.with_column('resid', preds - y)
preds_table_with_resid.scatter('Valence', 'resid')
plots.title('Residual Plot of Valence and Danceability');
```



With a correlation coefficient of 0.37, the relationship between the valence of a song and the danceability of a song appears to be a weak, positive linear one. Looking at the residuals, there appears to be no trends, which indicates that there are no issues with the linear model for valence and danceability. This reflects the prediction we made in the exploratory data analysis, but appears to be slightly weaker than we had assumed from a logical standpoint.

1.6 Conclusion

In our analysis of the music dataset, we attempted to investigate how danceability affects and is affected by features of a song. Specifically, we conducted a hypothesis test to see if Hip Hop songs have a higher danceability score, on average, than Folk songs, and attempted to use a linear regression to predict the danceability of a song based on the valence of the song. For our hypothesis test, we conducted an A/B test and found that Hip Hop did indeed have a higher danceability, on average, than Folk songs. In our linear regression, we found that valence had a weak, linear, positive relationship with danceability, and the residual plot did not show any significant issues with the linear model. Overall, the regression is a weakly effective predictor, but there may be better variables to predict danceability than valence. Further investigation may try to use a multiple linear regression to improve the accuracy of the predictor.

Also, there might potential sources of error within the data. Since the data that was given had been extensively cleaned and altered, some unforseen biases may arise from the fact that the sample is not entirely random. To fix this issue for future analyses, using the raw dataset might improve the accuracy of our results.