COMU3120

Digital Analytics

Individual Project Report

From The Perspective Of Loudness, Dancingability, Energy to Analyze The Homogeneity Of Pop Music

Name: YULIN WU

Student ID: 45659016

WORD COUNT: 1502

SUBMITTED DATE: 6.6.2022

TABLE OF CONTENT

Introduction

Pop music is one of the popular music genres. As social and cultural procedures continue to evolve. More and more people think that pop music is becoming more and more homogenous. Contemporary pop music all sounds the same, and pop music has remained largely the same pattern for a long time without noticeable change (Serra et al., 2012). Songs sound more and more similar, relying on the same simple structures and tones, and the diversity of widespread music declines (Trigger, 2013). Popular music production has industrialized based on more direct imitations of top producers (Daniels & Thompson, 2018).Nevertheless, this opinion has been opposed by some. Some people think that pop music is not homogenous and if similar contemporary songs are too close, the song is unlikely to succeed. Whether or not there is homogeneity in popular music is still up for debate (Askin & Mauskapf, 2017).

The report aims to test the hypothesis that popular music became homogeneous from 1970 to 2020. Loudness and associated energy are the "greatest common divisors" of music (Weninger et al., 2013). When analyzing audio characteristics, loudness and capability are also critical criteria. The characteristics of successful songs are also closely related to danceability (Interiano et al., 2018). With improving people's cultural level, popular music is constantly improving and innovating. The nature of popular music has undergone many changes. This report will focus on loudness characteristics, danceability, and energy to examine whether popular music is homogenized. Extracting and analyzing the data is used to verify the hypothesis that popular music has become homogeneous.

Method

Sample and procedure

This report uses Billboard's website to search the top 20 charts for the first week of April, August, and December each year from 1970 to 2019. Music information for these lists includes the year, month, rank, artist, and title. All music information was collected and saved in the first data file (N = 3000).

The second step of the research is to obtain the Trackid of each music. Because Trackid is equivalent to the role of the key. At the same time, Trackid is also a necessary condition for the subsequent steps. Trackid is through Spotify's Application Programming Interface (API) interface. It is queried and searched based on the artist name and title information, and it acts as a bridge between the link data access service and the service server. Some music works Spotify does not have their copyright or the artist and title information is not enough to help the Spotify API to find the trackid search results are the main reasons for the lack of 431 trackid data. The second data file is about trackid data collection.

The third stage in the research is to use the Spotify API again to search for the audio feature information of the music based on the Track id of the music. The searched audio feature information in the study includes ten main audio features, including danceability, energy, key, loudness, mode, speech, acoustics, instrumentality, vitality, valence and rhythm (Spotify, 2022). But in this study, we will mainly analyze three audio characteristics of danceability, energy, and loudness. The contents of the third data file are the ten main searched audio features.

The fourth stage in the study was to clean up missing data. The data-cleaned code was written to clean up 431 missing data. The cleaned data leaves only 2569 complete samples. It can be used for continuous measurement. Samples with complete information are collected in the fourth data file.

The final stage of the study is to aggregate each audio feature data, and time, explore the standard deviation (SD) of the audio feature versus year, and the trend of each audio feature from 1970 to 2020. The study will explore the relationship between the standard deviation of the characteristics and the year to analyze whether popular music is homogenized. The changing trend of each audio feature can analyze the development of popular music. The appendix contains the detailed collection code for all steps.

Measure

Based on Spotify's definition of music characteristics, this study examines three features of music samples (Spotify, 2022).

On a scale of 0.0 to 1.0, danceability "is the combination of musical characteristics, including rhythm, rhythmic stability, beat intensity, and overall regularity" (Spotify, 2022). The average dance ability score was 0.64 (SD = 0.15).

On a scale of 0.0 to 1.0, energy measures perceived intensity and activity. High-energy music has a quick, loud, and chaotic sound, while low-energy music has the opposite effect (Spotify, 2022). The average energy score was 0.63 (SD = 0.19).

The total decibel level (dB) of the entire audio track is referred to as loudness (Spotify, 2022). The average loudness score was -8.38 (SD = 3.49).

Results

The study's hypothesis is that popular music is becoming more homogenized. For each musical feature, time series plots are used to examine the link between SD and year, as well as how each musical feature has changed through time.

Hypothesis - Music characteristic loudness

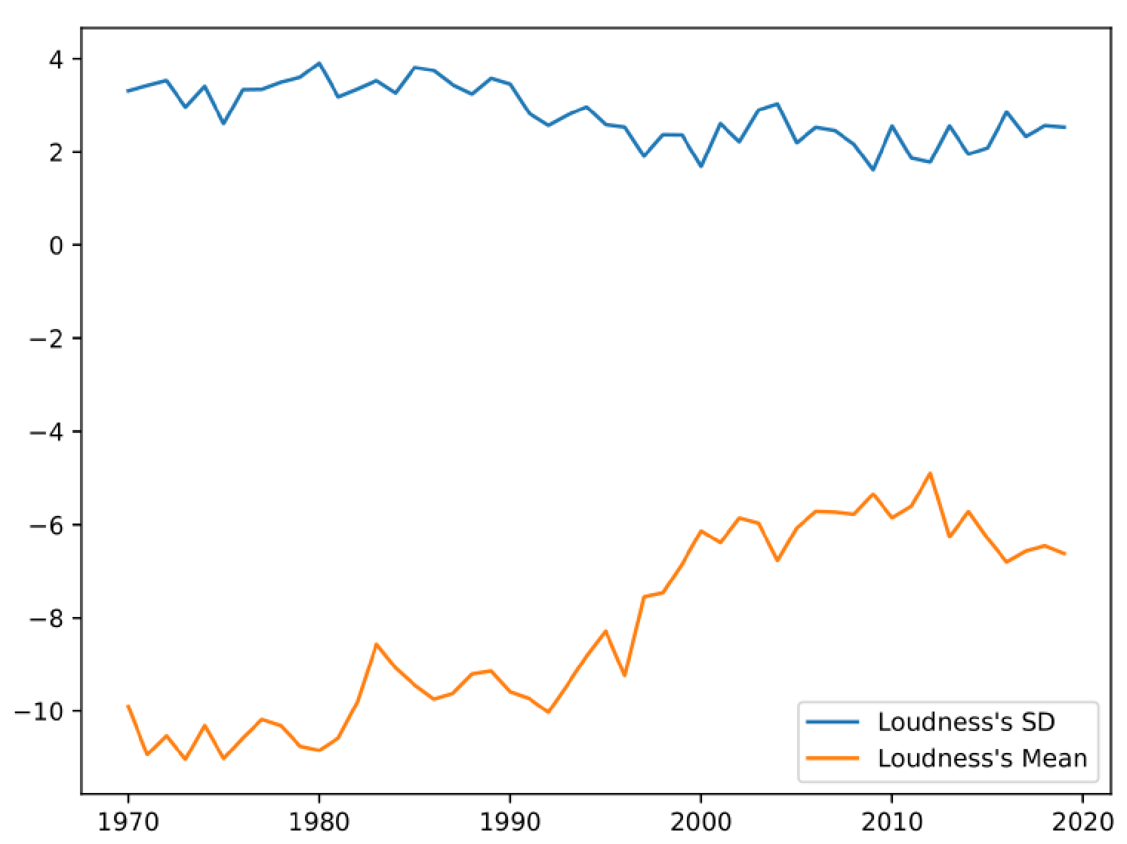

Figure 1 Time series plot of SD/Mean for Loudness

There is a clear link between music characteristic loudness and year ($r(48) = -.74$, $p < .05$) as shown in Figure 1.

The loudness feature exhibits a homogenous trend, and its SD diminishes with time, as seen in Figure 1.

Figure 1 further shows that the presented loudness of the average loudness does not have a consistent trend. For the past 50 years, there has been an increase in noise levels. This also demonstrates that the decibel level in 2020 is higher than in 1970. The loudness characteristics may be deduced from the SD and mean values in the picture, proving the theory.

Hypothesis - Music Feature Danceability

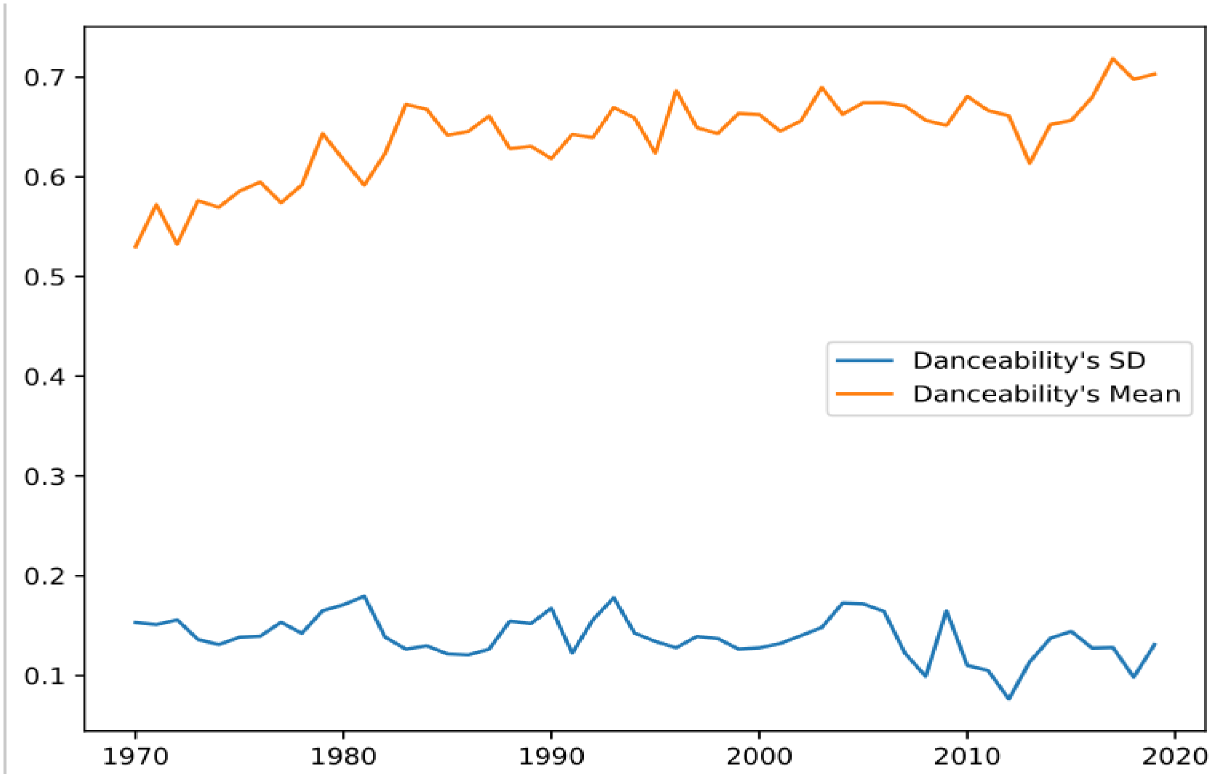

Figure 2 Time series plot of SD/Mean for Danceability

Figure 2 shows that there is no evident link between music characteristic loudness and year (r (48) = - .38, p < .05).

The SD dropped over time, according to the danceability parameters in Figure 2. Danceability is getting increasingly uniform.

Between 1970 and 2020, Figure 2 demonstrates an increased trend, with the average value of danceability going from moderate to high.

Hypothesis - Music Feature Energy

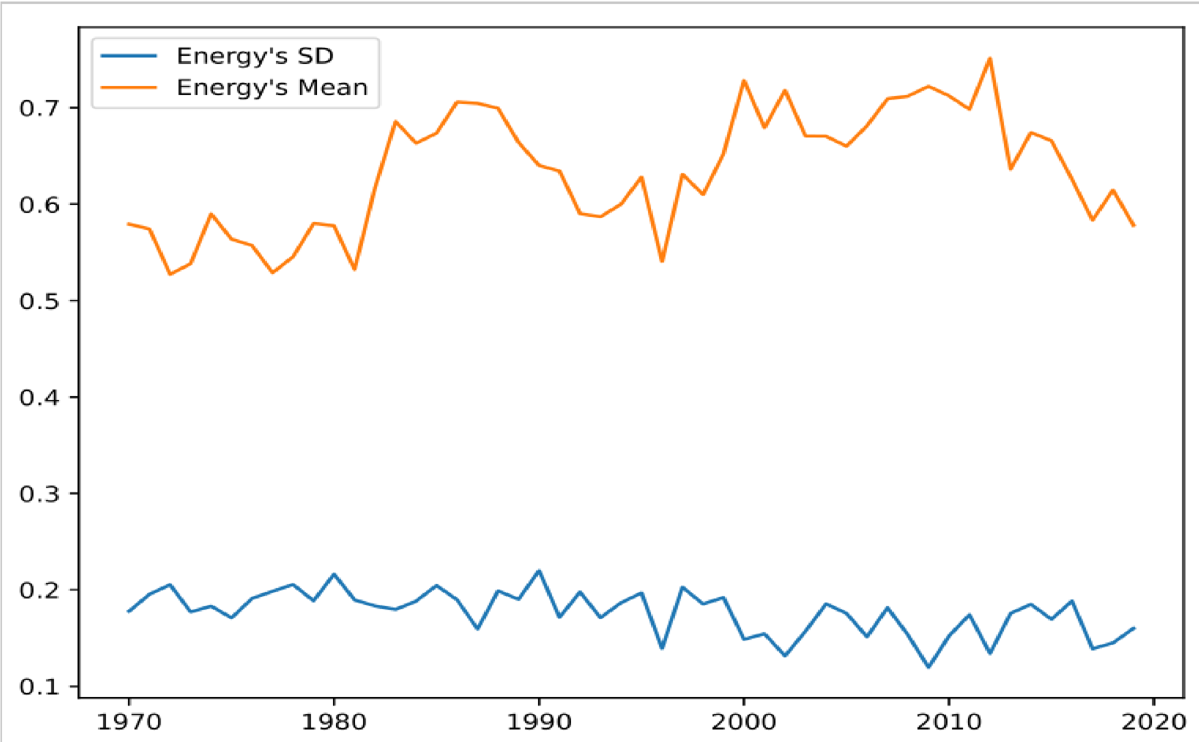

Figure 3 Time series plot of SD/Mean for Energy

According to Figure 3, there is a negative relationship between music feature energy and year ($r (48) = -.55$, $p < .05$).

Figure 3 shows that the music feature energy SD decreases with time, indicating an energy homogenization trend.

Figure 3 also shows that, despite the fact that the energy of the musical feature peaked in 2010, the musical feature suddenly dropped sharply afterwards. Musical characteristics' energy has generally increased over the last 50 years, which means becoming more excited. Energy signatures can also be used to test hypotheses.

Discussion

By examining three qualities (danceability, energy, and loudness) of the top 20 popular music from 1970 to 2020, this study verifies whether the idea that popular music is homogenized is correct. This study determined that there is a trend of homogenization or homogenization in popular music after extensive data analysis and research. As a result, this study's hypothesis has been proven.

The SD of the three most essential musical aspects of danceability, energy, and loudness are all adversely connected with the year, according to the study's findings. The statistics also suggest that the diversity of popular music is declining year after year, indicating a downward trend. We can deduce from the research that the diversity of popular music is dwindling as the industry of popular music creation becomes more industrialized.

Furthermore, previous research indicates that all three musical characteristics are changing in a positive way. Because of the constant increase in energy, pop music becomes more dynamic and exciting. It appears to be louder and noisier. Based on the preceding analysis, we can conclude that popular music in 2020 is highly danceable and thus more appropriate for dancing. Dance is directly linked to changes in rhythm. More rhythm, intensity, beat, and regularity are seen in music with improved danceability. The previous study also found that as time passes, the volume of popular music becomes louder. Although popular music is getting increasingly homogenized, it is also becoming more passionate and vibrant as the times go.

There are many flaws in this study. In today's society, there are many cover versions of popular songs that people like. The audio characteristics of these covers differ greatly from the original. These differences may affect the science and uniformity of the survey. The lack of investigation into the causes of homogenization in this study is also an important point for improvement. Future research should look at broader musical features and characteristics. Due to the lack of understanding of music in the research, it is impossible to observe the development trend of musical characteristics from a more professional perspective. This should be avoided and noted in future studies. In order to obtain explanations and factors, future research needs to communicate and interview with music-related experts to obtain influencing factors and reasons.

Reference

Askin, N., & Mauskapf, M. (2017). What makes popular culture popular? Product features and optimal differentiation in music. *American Sociological Review*, *82*(5), 910–944. https://doi.org/10.1177/0003122417728662

Daniels, M., & Thompson, A. (2018, May). *Are Hit Songs Becoming Less Musically Diverse?* The Pudding. https://pudding.cool/2018/05/similarity/

Interiano, M., Kazemi, K., Wang, L., Yang, J., Yu, Z., & Komarova, N. L. (2018). Musical trends and predictability of success in contemporary songs in and out of the top charts. *Royal Society Open Science*, *5*(5), 171274. https://doi.org/10.1098/rsos.171274

Serrà, J., Corral, Á., Boguñá, M., Haro, M., & Arcos, J. Ll. (2012). Measuring the evolution of contemporary western popular music. *Scientific Reports*, *2*(1). https://doi.org/10.1038/srep00521

Spotify. (2022). *Web API reference | spotify for developers*. Developer.spotify.com. https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features

Trigger. (2013, August 12). *The science behind why pop music sucks*. Saving Country Music. https://www.savingcountrymusic.com/the-science-behind-why-pop-music-sucks/

Weninger, F., Eyben, F., Schuller, B. W., Mortillaro, M., & Scherer, K. R. (2013). On the Acoustics of Emotion in Audio: What Speech, Music, and Sound have in Common. *Frontiers in Psychology*, *4*. https://doi.org/10.3389/fpsyg.2013.00292

Appendix

```
def scraper(url):

    headers = {'User-Agent': 'Mozilla/5.0 (Macinstosh; Intel Mac OS X 10_10_1)
AppleWebKit/537.36(KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36'}
    req = Request(url, headers=headers)
    context = ssl._create_unverified_context()

    uClient = urlopen(req, context=context)
    html = uClient.read()
```

```
    uClient.close()


    return BeautifulSoup(html, 'html.parser')



# Nested for loop:
# First for loop loops trough all numbers from 1970 to (not including) 2021 - We need this to generate years
# Second loop loops through the strings 04, 08, 12 - We sample 1 April, 1 August, and 1 December
for year in range(1970, 2020):
    for month in ['04', '08', '12']:

        # We build a custom url for the charts of every month of every year 1970-2020. Per iteration of the nested for loops, we get a different year-month combination
        url = 'https://www.billboard.com/charts/hot-100/' + \
            str(year) + '-' + str(month) + '-01/'
        print(url)

        soup = scraper(url)  # We get the html

        # Narrow down to the chart info
        chartdata = soup.find(
            'div', class_='u-max-width-960 lrv-u-margin-lr-auto')

        # loop through each chart item
```

```
    for i in chartdata.find_all('ul', class_='lrv-a-unstyle-list lrv-u-flex lrv-u-height-100p lrv-u-flex-direction-column@mobile-max'):

        rank = i.find('span', class_='c-label  a-font-primary-bold-l u-font-size-32@tablet u-letter-spacing-0080@tablet')  # Get rank

        title = i.find('h3').getText().strip()  # Get title text

        artist = i.find('span').getText().strip()  # Get artist text

        print(year, month, rank, '-', title, 'by', artist)



        if rank <= 20:  # Only keep if it's in the top 20


            entry = {

                'year': year,

                'month': month,

                'rank': rank,

                'artist': artist,

                'title': title,

            }  # Define a dictionary with all the required info


            data.append(entry)  # Add dictionary to the list data


    savedata = pd.DataFrame(data)  # Convert list data to a dataframe

    # Save dataframe to file

    savedata.to_csv('billboard.csv', sep=',', index=False)
```

```
    pause.seconds(2)  # Pause for two seconds in between every request


import json

import requests

import pandas as pd

import pause


# Read the data produced by the previous script

chartdata = pd.read_csv('billboard.csv',sep=',')

chartdata = chartdata.T.to_dict().values()


for entry in chartdata:


        # Build query (consists of artist and title)

        query = entry['artist'] + ' ' + entry['title']

        print('Searching',query)


        token =
'BQC-OqSMeATPn1jboU3AIfkB8H15eQZFO7-4pjdSLMoAHdtGKEyn-r70ST6ivcqnyoRlU991squ2_fgI
H5E'


        headers = {

                'Accept': 'application/json',

                'Content-Type': 'application/json',

                'Authorization': 'Bearer ' + token,
```

```
}

# Set parameters
params = (
                    ('q', query),
                    ('type', 'track'),
                    ('limit', '1'),
          )

response = requests.get('https://api.spotify.com/v1/search', headers=headers, params=params) #
Make request

print(response) # We want code [200] here

data = json.loads(response.text) # Convert response to text/dict

try: # Not every query returns a positive result. If we try to get information from a response of a
failed search, it crashes the script because that information is not there, hence the try/except logic
     trackid = data['tracks']['items'][0]['id'] # Isolate track id
     print(trackid)
except:
     trackid = ''
     print('Track id not found')

entry['trackid'] = trackid # Add track id to the active dictionary
```

```
        savedata = pd.DataFrame(chartdata) # Convert list data to a dataframe
        savedata.to_csv('spotify_trackid.csv',sep=',',index=False) # Save dataframe to file

        pause.seconds(.15) # Pause in between requests

import json
import requests
import pandas as pd
import time

# Read the data produced by the previous script
spotifydata = pd.read_csv('spotify_trackid.csv',sep=',')
spotifydata = spotifydata.T.to_dict().values()

for entry in spotifydata:

        token =
'BQAisSCJXcU6GWz31_lYmE_m8LuhImOVKB30nrBQxYPVNASl3Terex_rqb6J56o0AeX5SIDpCgP
WyqR_qKs'
        headers = {
                'Accept': 'application/json',
                'Content-Type': 'application/json',
                'Authorization': 'Bearer ' + token,
        }
```

trackid = entry['trackid'] # Add in the trackid

try: # Not every song has a valid trackid, if not available, the request would fail and crash the script, hence the try-except.

print('Querying',trackid)

response = requests.get('https://api.spotify.com/v1/audio-features/'+trackid, headers=headers) # Make request

print(response) # [200] in case of success

data = json.loads(response.text) # Convert response to text/dict

Get every audio feature from the response and add it as a key in the active dictionary

entry['danceability'] = data['danceability']

entry['energy'] = data['energy']

entry['key'] = data['key']

entry['loudness'] = data['loudness']

entry['mode'] = data['mode']

entry['speechiness'] = data['speechiness']

entry['acousticness'] = data['acousticness']

entry['instrumentalness'] = data['instrumentalness']

entry['liveness'] = data['liveness']

entry['valence'] = data['valence']

```
            entry['tempo'] = data['tempo']

            savedata = pd.DataFrame(spotifydata) # Convert list data to a dataframe
            savedata.to_csv('spotify_audiofeatures.csv',sep=',',index=False) # Save dataframe to file

            #time.sleep(.15) # Pause in between requests

        except:
            print('No audio features returned',trackid)
# Step Read
import pandas as pd

# Step Read the data
dataclean = pd.read_csv('spotify_audiofeatures.csv',sep=',')

# Step Get an overview of the variables
print(dataclean.info())

print(dataclean.shape)
print()

# StepPrint and clean missing data
```

```
print('\n # Missing data:\n', dataclean.isnull().sum())

isolatemissing = pd.isnull(dataclean['trackid'])

print('\n Rows with missing data:\n', dataclean[isolatemissing])

dataclean = dataclean.dropna()

print('\nDF after:\n', dataclean)


savedata = pd.DataFrame(dataclean)


savedata.to_csv('cleaningdata.csv',sep=',',index=False)


import pandas as pd



alldata = pd.read_csv('datacleaning.csv',sep=',')


print(alldata['danceability'].describe())

print()

print(alldata['energy'].describe())

print()

print(alldata['loudness'].describe())


import pandas as pd

import researchpy as rp
```

```
alldata = pd.read_csv('datacleaning.csv',sep=',')

part= alldata.groupby(['year'], as_index=False)[['danceability', 'energy','loudness']].std()

print(rp.correlation.corr_pair(part[['year', 'danceability']]))
print()
print(rp.correlation.corr_pair(part[['year', 'energy']]))
print()
print(rp.correlation.corr_pair(part[['year', 'loudness']]))
print()

import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('datacleaning.csv',sep=',')

group= data.groupby(['year'], as_index=False)[['danceability', 'energy','loudness']].std()

group_mean = data.groupby(['year'], as_index=False)[['danceability','energy','loudness']].mean()
```

```
plt.plot(group['year'], group['danceability'], label="Danceability's SD")


plt.plot(group_mean['year'], group_mean['danceability'], label="Danceability's Mean")

plt.legend()

plt.tight_layout()

plt.savefig('danceability-M.pdf')


import pandas as pd

import matplotlib.pyplot as plt


data = pd.read_csv('datacleaning.csv',sep=',')


group= data.groupby(['year'], as_index=False)[['danceability', 'energy','loudness']].std()



group_mean = data.groupby(['year'], as_index=False)[['danceability','energy','loudness']].mean()


plt.plot(group['year'], group['energy'], label="Energy's SD")

plt.plot(group_mean['year'], group_mean['energy'], label="Energy's Mean")

plt.legend()

plt.tight_layout()

plt.savefig('energy-M.pdf')


import pandas as pd
```

```
import matplotlib.pyplot as plt


data = pd.read_csv('datacleaning.csv',sep=',')


group= data.groupby(['year'], as_index=False)[['danceability', 'energy','loudness']].std()



group_mean = data.groupby(['year'], as_index=False)[['danceability','energy','loudness']].mean()


plt.plot(group['year'], group['loudness'], label="Loudness's SD")

plt.plot(group_mean['year'], group_mean['loudness'], label="Loudness's Mean")

plt.legend()

plt.tight_layout()

plt.savefig('loudness-M.pdf')
```