Has Popular Music Become More Homogeneous in Recent Years in Favour of Synthetic Dance Music?

Danyu Wan

45215337

COMU3120

Contents

Introduction

Is Coldplay a Britpop rock band or an electric band? I never had doubt until the album *A Head Full of Dreams* was released. Nowadays, there is a heated debate about the increasing homogenisation of popular music. Popular radio hits tend to sound more similar, whereas the genres of songs made by so-called "rock" bands like Coldplay are more ambiguous. This begs the question of whether popular music has become more homogeneous in more recent years in favour of synthetic electric music.

Serrà et al. (2012) undertook an analysis of 464,411 songs between 1955 and 2010 based on timbre (colour, texture, or tune quality of sound), pitch (chords, melody), and loudness. Results indicate that while timbre and pitch remained largely unchanged over the course of more than fifty years, music loudness increased significantly. Furthermore, Percino et al. (2014) conducted research on more than 500,000 albums and discovered that the more popular the music genres are, the more similar they sound by quantifying music features and genres. Moreover, as music became more formulaic instrumentally, sales numbers increased, implying the popularity of music with little variety. Barnes (2014) believes that the current mainstream consumption market is uniform, commercial, and safe, music that lacks authentic characteristics and is formulaic is extremely marketable.

Past research has revealed important findings denoting that music has not changed significantly over time. The methodology, however, may have skewed the results. The database used for analysis in the Serrà et al. (2012) study is biased because it contains 2,650 songs from 1955 to 1959 but 177,808 songs from 2005 to 2009. That is, the research is primarily based on tracks released in recent years. Therefore, the current research aims to improve the methodology and gain a more comprehensive understanding of how popular music has evolved over the last 50 years. To represent the most popular music over the last 50 years, I will first collect songs from Billboard's end-of-year top 100 charts spanning 1970 to 2019, then use the Spotify API endpoint to acquire the audio features, finally analyse the data. Python will be used for all data collection and analysis (see Appendix).

Method

Data Collection

To find the most representative popular music each year, I chose to collect the most popular 100 songs from 1970 to 2019 using Billboard's year-end charts (https://www.billboard.com/charts/year-end/hot-100-songs). This chart, which is based on radio airplay, sales, and streaming data in the United States, can be considered a fair and objective representation of popular songs. Also, because the research seeks to explore how music evolves over time, a time span of fifty years is appropriate for observing the variation trend. Therefore, I first used Python to collect all the artist and track information for all of the songs on the charts from 1970 to 2019, yielding 4653 songs as a result (see Appendix).

As one of the largest music streaming platforms, Spotify has a massive music library. Using the Spotify API endpoint, users can obtain audio features for nearly every song. Spotify's proprietary analysis algorithms extract common audio features such as tempo and key, as well as more specialised features such as liveness and instrumentalness (Skidén, 2016). It is essential to obtain a track ID prior to getting the audio features. As a result, I began by using Spotify’s Search API (https://developer.spotify.com/console/get-search-item/) to obtain all the track IDs. The search query is activated by entering the concatenated artist and track name. However, I noticed that some of the previously obtained tracks have difficult names that prevent the results from being found. When there are multiple artists or a single artist has multiple names, the extra artists/names must be removed. To make the search query work, I removed all the occurrences of ‘featuring’ ‘duet’ ‘x’ ‘&’ ‘and’ ‘/’ ‘with’ ‘(’ ‘or’ (in both capital and lower case) and the words that followed. Same works for track names. In addition to removing ‘/’ ‘(’ and the words that followed, I removed the term ‘theme from’ because the track *Theme from Star Wars – Meco* could not be found with it. Also, the track *Ran$om – Lil Tecca* could not be found. I noticed that the song is also called *Ransom,* thus by replacing ‘$’ with ‘s’, the result appeared. The audio features for the tracks were then obtained through Spotify Audio Features endpoint (https://developer.spotify.com/console/get-audio-features-track/) using the track IDs. Spotify Audio Features (Spotify, n.d.) Energy (a scale from 0 to 1, representing a perceptual measure of activity and intensity), loudness (the decibel level of a track's overall volume), danceability (a measure ranging from 0-1, describing how ideal a track is for dancing) and acousticness (a scale from 0 to 1 indicating whether the track is acoustic) were selected as the features to focus on. Because of the missing information from 14 tracks, the outcome was unavoidably skewed. Nonetheless, the impact is minor given the small amount of missing data. After removing the missing data, the final result contains 4639 pieces of data, including the year, artist, track, track ID, and relevant audio features.

Data Processing

Following the completion of data collection, the next step is data processing. To learn about the level of music homogenisation and the evolution of central tendency over time, I aggregated the data to one observation of the mean/standard deviation per relevant audio feature per year, resulting in two sets of data.

Results

Data Visualisation & Analysis

The time series figures were generated using the previously processed aggregated data. Figure 1 depicts how each audio feature is dispersed chronologically. In general, all four indicators point to a downward trend. All features have two noticeable low points, which are around the years 2000 and 2010. All features have their highest points around the year 1980. All the data is least dispersed at the lowest point around 2010, then gradually increases until now. Most recently, acousticness, danceability and loudness are all on the rise, while energy has been on the decline since around 2015.

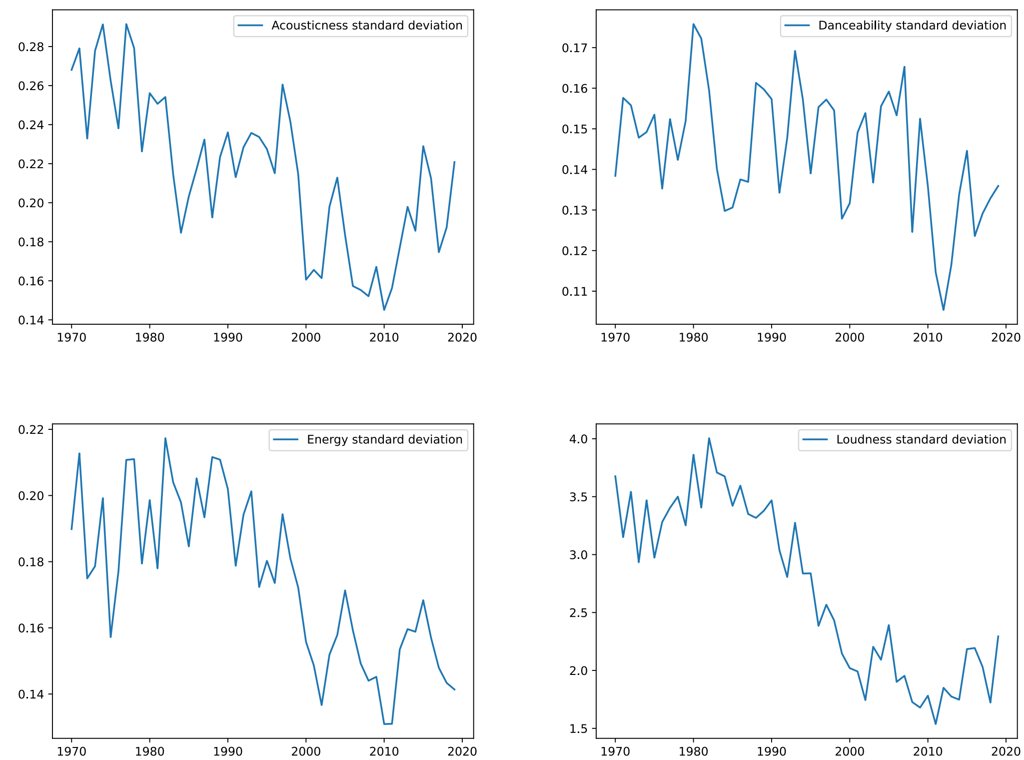

Figure 1. Time series of audio features' standard deviation

Figure 2 displays the four features' central tendency over time. Overall, danceability, energy, and loudness are on the rise, while acousticness is on the decline. Similar to standard deviation, 2010 appears to be a turning point for all data. The lowest points of danceability, energy and loudness are at around the year 1970; the lowest of acousticness is at around 2010. The peak of energy and loudness is at around 2010, the peak of danceability is at around 2019, and the peak of acousticness is at around 1972.

Figure 2. Time series of audio features' mean

Finally, I calculated the magnitude of the correlation between the year and the four features. The relevant correlation coefficient is shown in Table 1: the year has a positive correlation with danceability, loudness, and energy, but a negative correlation with acousticness. The correlation between the year and loudness is particularly strong, with r =.56.

Table 1. The magnitude of the correlations between year and audio features

	Year & Acousticness	Year & Danceability	Year & Loudness	Year & Energy
R-value	-0.2889	0.1827	0.5603	0.2182

Discussion

The main aim of the research was to investigate whether popular music has become more homogeneous in favour of synthetic electric music over time. The results are consistent with previous studies that over the last 50 years, popular music has become more homogeneous due to the changes of acousticness, danceability, loudness and energy. It's worth noting that popular music sounded the most similar in around 2010. Milano (2019) believes that emerging pop stars such as Beyoncé, Taylor Swift, and Justin Bieber, as well as the rise of streaming services, revolutionized pop music in the 2010s. The study by Serrà et al. (2012) only covers music up to 2010, indicating very little change of popular music. However, it is a delightful discovery that music has started to sound more 'different' since 2010. Furthermore, with a more complete database, the current study overcomes the limitation of the Serrà et al. (2012) study, which was that their data were biased. The observations also supported the findings of Percino et al. (2014) that instrumental music has become increasingly formulaic.

It is not surprising to find that music acousticness has declined dramatically over the last fifty years. The correlation coefficient also demonstrates a negative relationship between the year and the acousticness. It illustrates how, as music has become more synthesised, the proportion of real instruments in music has decreased. Similarly, music danceability has increased substantially, which means that music from recent years is more suitable for dancing in terms of tempo, rhythm, beat, and so on. Music has become more energetic as it has become faster, louder, and noisier. Music loudness has undoubtedly increased noticeably. It is worth noting that the year has a strong positive correlation with loudness. It draws to a conclusion that music has become more homogeneous to synthetic dance music. However, there is no need to be pessimistic about the future of popular music; some results show that the situation is changing. At the very least, music is less homogeneous now than it was ten years ago.

Because of the large amount of data used, the research is overall comprehensive. It shows a clear trend of how music has evolved over the last 50 years. However, a number of limitations needs to be addressed. First, the results returned by the Spotify API endpoint are not always consistent. I noticed that the track IDs I collected differed slightly between attempts. Therefore, bias may exist when using this method. Second, the audio features of my selection are not the most representative ones. There is some overlap between the feature loudness and the feature energy. Besides that, the specific measurements are ambiguous because they are based on Spotify's standard. Finally, Billboard's year-end charts only include the most popular songs in the United States. Despite the fact that the United States has one of the world's largest streaming markets (Statista, 2021), future research could be expanded to include the global market.

Ultimately, the current research reveals significant findings about the evolution of popular music from 2010 to 2019, filling in the gaps left by previous research. Future research could be conducted to explore why music sounded almost "the same" in 2010 and why the situation has improved.

References

Barnes, T. (2014, December 18). This Is Why Hip-Hop Is Going to Lose Its Power. *Mic.* https://www.mic.com/articles/106724/hip-hop-could-become-completely-white

Milano, B. (2019, December 6). What Did 2010s Music Do for Us? Behind A Transformative Decade. *U Discover Music.* https://www.udiscovermusic.com/stories/2010s-music-history/

Percino, G., Klimek, P., & Thurner, S. (2014). Instrumentational complexity of music genres and why simplicity sells. *PloS One, 9*(12), e115255. https://doi.org/10.1371/journal.pone.0115255

Serrà, J., Corral, A., Boguñá, M., Haro, M., & Arcos, J. (2012). Measuring the evolution of contemporary western popular music. *Scientific Reports, 2*(1), 521–521. https://doi.org/10.1038/srep00521

Skidén, P. (2016, March 29). New Endpoints: Audio Features, Recommendations and User Taste. *Spotify for Developers.* https://developer.spotify.com/community/news/2016/03/29/audio-features-recommendations-user-taste/

Statista. (2021). *Music Streaming.* Statista. https://www.statista.com/outlook/dmo/digital-media/digital-music/music-streaming/united-states

Appendix

Python Codes

```
#Step 1:
from urllib.request import Request, urlopen
import ssl
from bs4 import BeautifulSoup
import pause
import pandas as pd

x = []

yearnumber = 1970
while yearnumber <= 2020:
    url = 'https://www.billboard.com/charts/year-end/'+str(yearnumber)+'/hot-100-songs'

    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()

    soup = BeautifulSoup(html, 'html.parser')
    divofinterest = soup.find('div',class_='chart-details')

    for i in divofinterest.find_all('div',class_='ye-chart-item__primary-row'):
        artist = i.find('div',class_='ye-chart-item__artist').getText().strip()
        track = i.find('div',class_='ye-chart-item__title').getText().strip()

        print(artist,'-',track)
        print()

        pause.seconds(0.05)

        x.append({
            'year':yearnumber,
            'artist':artist,
            'track':track
        })

    yearnumber += 1

x = pd.DataFrame(x)
x.to_csv('billboard.csv',sep=';',index=False)

#Step 2:
import pandas as pd
```

```
import json
import requests
import pause
import spotifyaudiofeatures

data = pd.read_csv('http://www.digitalanalytics.id.au/static/files/chartdata.csv',sep=';')
data = data.T.to_dict().values()

x = []


for i in data:
    artist = i['artist']
    track = i['track']

    headers = {
                'Accept': 'application/json',
                  'Content-Type': 'application/json',
            'Authorization': 'Bearer
BQDhOzfCl3PRa5cUHYfYDkkeMnC1HePXheTAYikGsP98DxZAM77fOsGTW34OUj7LRbQc
zEmrz4NImcBgSR1rxCNmci9RB4WnWZimgCWjPOchDYda8W4euB_KrkVMPEzffN3eQICF-
iSqjh9mVf-Ki3710NQ0IBI',
        }

    if 'Featuring' in artist:
        artist = artist.split('Featuring')[0]
    if 'featuring' in artist:
        artist = artist.split('featuring')[0]
    if 'Duet' in artist:
        artist = artist.split('Duet')[0]
    if 'x' in artist:
        artist = artist.split('x')[0]
    if 'X' in artist:
        artist = artist.split('X')[0]
    if '&' in artist:
        artist = artist.split('&')[0]
    if 'and' in artist:
        artist = artist.lower().split('and')[0]
    if '/' in artist:
        artist = artist.split('/')[0]
    if 'with' in artist:
        artist = artist.split('with')[0]
    if 'With' in artist:
        artist = artist.split('With')[0]
    if '(' in artist:
        artist = artist.split('(')[0]
    if 'Or' in artist:
        artist = artist.split('Or')[0]

    if '/' in track:
        track = track.split('/')[0]
```

```
    if '(' in track:
        track = track.split('(')[0]
    if 'Theme From' in track:
        track = track.remove('Theme From')
    if '$' in track:
        track = track.replace('$','s')


    query = artist + ' ' + track


    params = (
                                ('q', query),
                ('type', 'artist,track'),
                  )

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

    print(response)

    json_data = json.loads(response.text)

    try:
        trackid = json_data['tracks']['items'][0]['id']
        print('ok',query)
    except:
        trackid = ''
        print('ERROR',query)

    year = i['year']
    print(str(year) + ' - ' + i['artist'] + ' - ' + i['track'] + trackid)

    pause.seconds(0.05)

    x.append({
        'year':i['year'],
        'artist':i['artist'],
        'track':i['track'],
        'track id':trackid,
        })

x = pd.DataFrame(x)
x.to_csv('spotifyid.csv',sep=';',index=False)

#Step 3:
import pandas as pd
import json
import requests
import pause

data = pd.read_csv('spotifyid.csv',sep=';')
```

```
data = data.T.to_dict().values()

y = []

for i in data:
    query = i['track id']

    headers = {
        'Accept': 'application/json',
        'Content-Type': 'application/json',
        'Authorization': 'Bearer BQCKofAdcjd6Pcbz_OMJKq-oBPqhCt_GONv7MBBU95-
Q65M8wI42atkUXFBFCW2KLb3-6bcdYYpW9D-
H1iKCs3noMpqKGAdOdDfMmb7ZgGb9LpGiOaq6xtQDl_oiw7NMzv9ugB49Bv6DbYKBRzG
XYJexCW3_5Og',
    }

    url = 'https://api.spotify.com/v1/audio-features/' + str(query)

    response = requests.get(url, headers=headers)

    print(response)

    json_data = json.loads(response.text)

    year = i['year']
    print(str(year) + ' - ' + i['artist'] + ' - ' + i['track'])

    try:
        danceability = json_data['danceability']
    except:
        danceability = ''

    print(danceability)

    try:
        energy = json_data['energy']
    except:
        energy = ''

    print(energy)

    try:
        loudness = json_data['loudness']
    except:
        loudness = ''

    print(loudness)

    try:
        acousticness = json_data['acousticness']
    except:
```

```
        acousticness = ''

    print(acousticness)

    pause.seconds(0.1)

    y.append({
        'year':i['year'],
        'artist':i['artist'],
        'track':i['track'],
        'track id':i['track id'],
        'danceability':danceability,
        'energy':energy,
        'loudness':loudness,
        'acousticness':acousticness,
    })

y = pd.DataFrame(y)
y.to_csv('spotify-audio-features.csv',sep=';',index=False)

#Step 4:
import pandas as pd

df = pd.read_csv('spotify-audio-features.csv',delimiter = ';')

print(df.info())

print(df.shape)

print(df.isnull().sum())

isolatemissing = pd.isnull(df['track id'])
print(df[isolatemissing])

isolatemissing1 = pd.isnull(df['danceability'])
print(df[isolatemissing1])

df = df.dropna()

print(df.shape)

print(df.duplicated().sum())

print(df[df.duplicated()])

df = df.drop_duplicates()

print(df.shape)

df.to_csv('spotify-clean.csv',sep=';',index=False)
```

```
#Step 5:
import pandas as pd
import researchpy as rp
import matplotlib.pyplot as plt

df = pd.read_csv('spotify-clean.csv',delimiter = ';')

dataset = []

#time series
group1 = df.groupby(['year'],
as_index=False)[['danceability','loudness','acousticness','energy']].mean()
print(group1)

plt.plot(group1['year'],group1['acousticness'],label='Acousticness mean')

plt.legend()
plt.savefig('timeseries_acousticness_mean.pdf')
plt.clf()

plt.plot(group1['year'],group1['danceability'],label='Danceability mean')

plt.legend()
plt.savefig('timeseries_danceability_mean.pdf')
plt.clf()

plt.plot(group1['year'],group1['energy'],label='Energy mean')

plt.legend()
plt.savefig('timeseries_energy_mean.pdf')
plt.clf()

plt.plot(group1['year'],group1['loudness'],label='Loudness mean')

plt.legend()
plt.savefig('timeseries_loudness_mean.pdf')
plt.clf()

group2 = df.groupby(['year'],
as_index=False)[['danceability','loudness','acousticness','energy']].std()
print(group2)

plt.plot(group2['year'],group2['acousticness'],label='Acousticness standard deviation')

plt.legend()
plt.savefig('timeseries_acousticness_std.pdf')
plt.clf()

plt.plot(group2['year'],group2['danceability'],label='Danceability standard deviation')

plt.legend()
```

```
plt.savefig('timeseries_danceability_std.pdf')
plt.clf()

plt.plot(group2['year'],group2['energy'],label='Energy standard deviation')

plt.legend()
plt.savefig('timeseries_energy_std.pdf')
plt.clf()

plt.plot(group2['year'],group2['loudness'],label='Loudness standard deviation')

plt.legend()
plt.savefig('timeseries_loudness_std.pdf')
plt.clf()


plt.plot(group2['year'],group2['acousticness'],label='Acousticness standard deviation')
plt.plot(group2['year'],group2['danceability'],label='Danceability standard deviation')
plt.plot(group2['year'],group2['energy'],label='Energy standard deviation')
plt.plot(group2['year'],group2['loudness'],label='Loudness standard deviation')

plt.legend()
plt.savefig('timeseries_all_std.pdf')
plt.clf()

#correlation
print(rp.correlation.corr_pair(df[['year','acousticness']]))
print(rp.correlation.corr_pair(df[['year','danceability']]))
print(rp.correlation.corr_pair(df[['year','loudness']]))
print(rp.correlation.corr_pair(df[['year','energy']]))

data = pd.DataFrame(group1)
data.to_csv('aggregated_mean.csv',sep=';',index=False)

data = pd.DataFrame(group2)
data.to_csv('aggregated_std.csv',sep=';',index=False)
```