Final Project Part 1, Social Science Inquiry II (SOSC13200-W22-3)

Tiffanie Huang

Monday 2/14/22 at 5pm

```
ts <- read.csv("../data/taylor_swift_spotify.csv")
```

Data set name: Taylor Swift Spotify Dataset *[Link]*(https://www.kaggle.com/datasets/jarredpriester/taylor-swift-spotify-dataset?resource=download&select=taylor_swift_spotify.csv)

1. Describe where the data comes from, and how it was generated.

The data was downloaded from Kaggle (jarredpriester) and found by Karen. It was originally from Spotify's web API for Taylor Swift, which developers can use to download metadata on artist albums/tracks, playlists, or user-related information. The Kaggle user used the Spotipy library to extract the data into a Jupyter notebook before reformatting for the public. Spotify, an audio streaming subscription service, collects data on user listening habits for purposes ranging from generating more personalized playlists to real-time context targeting for ads. Open APIs allow third-party companies or developers to interact more with the platform and could possibly benefit Spotify by keeping the streaming service relevant to users when it's easily "integratable" on other platforms. Spotify is known for their use of CNNs on audio data, which is how they are able to generate data on track characteristics that might not seem quantifiable at first, such as "acousticness" or "speechiness."

2. What does each observation describe in the data set?

Each observation describes a unique track by Taylor Swift. This could be a song or song commentary. The data was measured all at once rather than over different periods in time, since the Kaggle data set updates monthly; for Taylor Swift, the Spotify API has a temporal coverage start date of October 23, 2006, but this data set just includes the most recent version. The geospatial coverage is worldwide.

My data is wide, since each row is a new observation row per unit (unique track). We can see that the measurements for different variables like danceability, popularity, and tempo are all stored on the same row for each unique track. There is also an extra column for indexing (X) that I might drop later. Something I thought was interesting was how the same song on a normal vs deluxe (ex. "cardigan" from folklore vs "cardigan" from folklore (deluxe version)) are assigned different track IDs rather than consolidating into one observation.

3. What is the population represented in the data set? If you were to analyze the data set, would you expect the results to be relevant for other units or groups?

The population represented would be Taylor Swift's released recorded discography until now, including her commentary tracks, since it draws from Spotify's data limited to Taylor Swift's tracks/albums. The data set itself includes all relevant units from the population, as all her released songs/commentary are now on Spotify. If she releases any new tracks and my target population is her updated discography, then the data set wouldn't include all relevant units. I expect the results of my analysis to be relevant for other units/new tracks she releases since the "sample" for this data set is representative of the target population. I can't apply the results of my analysis to any other artist data pulled from the same source, so the population is restricted to Taylor Swift's music. However, it would be interesting to analyze data from other artists in a similar fashion and compare results.

4. Provide some descriptive summary of the dataset using code.

There are 1265 rows and 18 columns. The track ID will be what I use to uniquely identify tracks, since many song titles are similar/identical.

I was surprised that the genre of each track was not included as a categorical variable, as that is usually included as track information with the track album name and track number. I noticed that a few variables that the Spotify API typically provides are missing, such as mentioned artists, mode, key, time signature, and whether or not the track is explicit. For the variables I do have, there are no missing values.

The majority of variables are numeric and have to do with audio classification. My categorical variables include song title and album name. Because tracks were released over time and audio characteristics haven't been re-measured and replaced since their release, I can use release date as a way to analyze her track audio characteristics over time even if it does not mark intervals at which measurements were taken since. However, for measures of track popularity, I would need a measurement of track popularity at every interval in order to look at its relationship with time (here it just gives the most recent popularity scores for February 14th), since it is based on recent streaming patterns.

```
sum(ts$popularity==0) #52 values where song popularity = 0
```

```
## [1] 52
```

```
#str(ts) #9 numeric class variables, 4 integer class variables, 5 character class variables
dim(ts) #1265 rows/observations + #18 columns
```

```
## [1] 1265   18
```

```
sum(is.na(ts)) #0 missing values
```

```
## [1] 0
```

```
#numerical variables:
range(ts$acousticness) #2.47e-05, 9.83e-01
```

```
## [1] 2.47e-05 9.83e-01
```

```
range(ts$danceability) #0.175, 0.897
```

```
## [1] 0.175 0.897
```

I thought it was interesting how Spotify could classify different aspects of track audios to measure seemingly qualitative attributes. With acousticness and danceability I can imagine it has to do with BPM or timbre of the instrumentals, but valence surprised me because I didn't realize they could also classify how happy/sad a track's audio was and quantify that. Most of the numerical variables related to audio classification are measured on a scale of 0 to 1 (0 being none of that quality, 1 being completely of that quality), with the exception of loudness, which is typically measured between -60 to 0 in dB (dB are logged values). Loudness is measured in relative sound pressure units (not on a scale like danceability) to the reference level of -14 dB. We can use this when looking at loudness of some tracks relative to others.

```
range(ts$loudness) #-17.932  -1.953 (max is kind of loud)
```

```
## [1] -17.932  -1.953
```

For duration (ms) and popularity, we get integer values. The duration can be any integer above 0, while popularity is between 0 and 100.

```
range(ts$duration_ms) #41769, 613026 (All Too Well - Taylor's Version)
```

```
## [1]  41769 613026
```

```
range(ts$popularity) #0, 92
```

```
## [1]  0 92
```

```
#future modifications to make:
#drop extra index column, possibly url
#separate commentary and songs
#release date -> release month (?) for time
```

References

1. "Welcome to Spotipy!" Welcome to Spotipy! - spotipy 2.0 documentation. Accessed February 17, 2023. https://spotipy.readthedocs.io/en/2.22.1/. #'
2. "Web Api." Spotify for Developers. Accessed February 19, 2023. https://developer.spotify.com/documentation/web-api/.