

Spotify Data Analysis

Emre Ozan Oral

Agenda

Topics Covered

- ▲ Gathering Data
- ▲ Data Analysis
- ▲ Visualization
- ▲ Hypothesis Testing

Data Gathering

Spotify collects your data for you.

I used extended streaming history for my project.


Download your data

By using our Download your data tool, you can request a copy of your personal data. You may download three different packages of data, either separately or all at once. Please see below what the packages include (if applicable to you) and choose what you want to download.

For more details about the data categories in the packages, please see [Understanding My Data](#) or [contact us](#).

Account data


- Playlists
- Streaming history for the past year
- A list of items saved in your library
- Search queries
- No. of followers, accounts you follow, and blocked accounts
- Payment and subscription data
- User data
- Customer Service History
- Family Plan data
- Inferences
- Voice input
- Podcast interactivity
- Spotify for Artists data

 Preparation time 5 days

☐ Select Account data

Extended streaming history


Extended streaming history for the lifetime of your account, including track information, and when and how you streamed content.

 Preparation time 30 days

☐ Select Extended streaming history

Technical log information

Technical log information that we have collected about your account to provide and troubleshoot the Spotify service.

 Preparation time 30 days

☐ Select Technical log information

Request data

DataFrame

There were 10 files for my entire Spotify history and i collected each dataframe and put them in a final_dataframe

```
import pandas as pd
import json

# Function to read and process a Spotify data file
def read_spotify_data(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        data = json.load(file)

    # Create a DataFrame from the list of song entries
    df = pd.DataFrame(data)

    # Convert 'ts' column to datetime format
    df['ts'] = pd.to_datetime(df['ts'])

    return df

# List of file paths for files 1.json to 10.json
file_paths = ['1.json', '2.json', '3.json', '4.json', '5.json', '6.json', '7.json', '8.json', '9.json', '10.json']

# List to store DataFrames for each file
dataframes = []

# Loop through each file and create a DataFrame
for file_path in file_paths:
    df = read_spotify_data(file_path)
    dataframes.append(df)

# Concatenate DataFrames into a single DataFrame
final_dataframe = pd.concat(dataframes, ignore_index=True)

# Display the final DataFrame

final_dataframe
```

What is the data format?

```
{
  "ts": "YYY-MM-DD 13:30:30",
  "username": "_____",
  "platform": "_____",
  "ms_played": _____,
  "conn_country": "_____",
  "ip_addr_decrypted": "____.____.____.____",
  "user_agent_decrypted": "_____",
  "master_metadata_track_name": "_____",
  "master_metadata_album_artist_name": "_____",
  "master_metadata_album_album_name": "_____",
  "spotify_track_uri": "_____",
  "episode_name": "_____",
  "episode_show_name": "_____",
  "spotify_episode_uri": "_____",
  "reason_start": "_____",
  "reason_end": "_____",
  "shuffle": null/true/false,
  "skipped": null/true/false,
  "offline": null/true/false,
  "offline_timestamp": "_____",
  "incognito_mode": null/true/false,
}
```

Some data i gathered from .JSON files were too technical for my research so narrowed my data for my research

Using Spotify API

In order to use Spotify API I need to have an application on the Spotify for Developers

Dashboard



for_210
scraping

```
import spotipy
from spotipy.oauth2 import SpotifyOAuth
import pandas as pd
import time

# Set up Spotify API credentials
client_id = '733e1d9701fe43f9922d208202bf57ed'
client_secret = '3ed569a922274d26b327e3954a5f973c'
redirect_uri = 'http://localhost:8888/callback'

# Set up Spotify API authentication
sp = spotipy.Spotify(auth_manager=SpotifyOAuth(client_id=client_id, client_secret=client_secret, redirect_uri=redirect_uri, scope='playlist-modify-private'))
```

Experimenting

- ▶ Collecting each year's data
- ▶ Making playlists from dataframes
- ▶ Making assumptions about data
- ▶ Training AI to make playlists for my taste

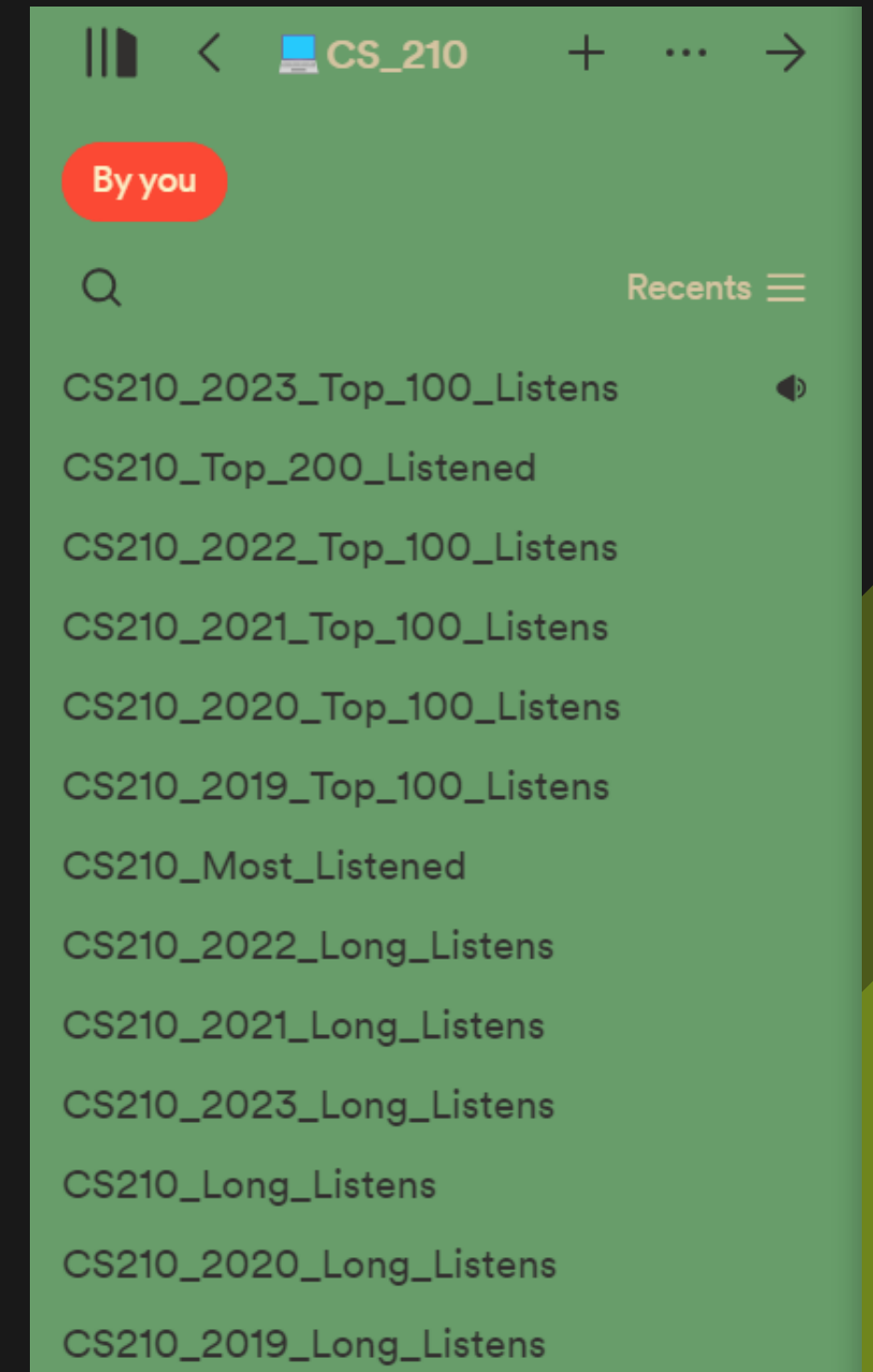
Experimenting with ideas

I created playlists with the application using Spotify API

We tell the API to include songs by their track_uri (in the data, this information is included for each song)

After experimenting, I created 2 code pieces for creating 2 types of playlists.

- 1) Creates top{given_value} for each year with your data
- 2) Creates all time top{given_value}



2 Algorithms

My code provided top100 for 2023

Which one is better?

Spotify algorithm provided top100 for 2023

The image displays two Spotify playlist interfaces side-by-side for comparison. The top interface, titled 'CS210_2023_Top_100_Listens', is a public playlist by Emre Ozan Oral containing 100 songs with a total duration of 5 hours and 41 minutes. The bottom interface, titled 'Your Top Songs 2023', is a Spotify Wrapped playlist for the same user, containing 100 songs with a total duration of 5 hours and 21 minutes. Both playlists are sorted by 'Custom order'.

CS210_2023_Top_100_Listens

#	Title	Album	Date added	Duration
1	New Person, Same Old Mistakes Tame Impala	Currents	15 hours ago	6:03
2	Borderline Tame Impala	The Slow Rush	15 hours ago	3:58
3	Chamber Of Reflection Mac DeMarco	Salad Days	15 hours ago	3:52
4	telepatía Kali Uchis	Sin Miedo (del Amor y Otros Demonios...	15 hours ago	2:40
5	Little Dark Age MGMT	Little Dark Age	15 hours ago	5:00
6	Doldum Adamlar	Dünya Günlükleri	15 hours ago	7:22
7	YKWIM? Yot Club	Bipolar	15 hours ago	3:33
8	Swing Lynn Harmless	I'm Sure	15 hours ago	5:21
9	İçimizdeki Canavarlar Adamlar	Harekete kimse mâni olamaz.	15 hours ago	4:39
10	Riders On The Storm - Fredwreck Remix Snoop Dogg, The Doors	Riders On The Storm (Fredwreck Remix)	15 hours ago	6:22

Your Top Songs 2023

#	Title	Album	Date added	Duration
1	telepatía Kali Uchis	Sin Miedo (del Amor y Otros Demonios...		2:40
2	New Person, Same Old Mistakes Tame Impala	Currents		6:03
3	Borderline Tame Impala	The Slow Rush		3:58
4	Chamber Of Reflection Mac DeMarco	Salad Days		3:52
5	Omae Wa Mou deadman 死人	Omae Wa Mou		1:54
6	Moonlight Kali Uchis	Red Moon In Venus		3:08
7	chipotle bbno\$	bag or die		2:29
8	YKWIM? Yot Club	Bipolar		3:33
9	Little Dark Age MGMT	Little Dark Age		5:00
10	METAMORPHOSIS INTERWORLD	METAMORPHOSIS		2:23

Beginning Research

I used all-time top1000 for my research

I used a Spotify playlist analyzer #Chosic

I got musical data and a popularity data.

Some songs are duplicated with different album names and their data is corrupted (Popularity = 0)

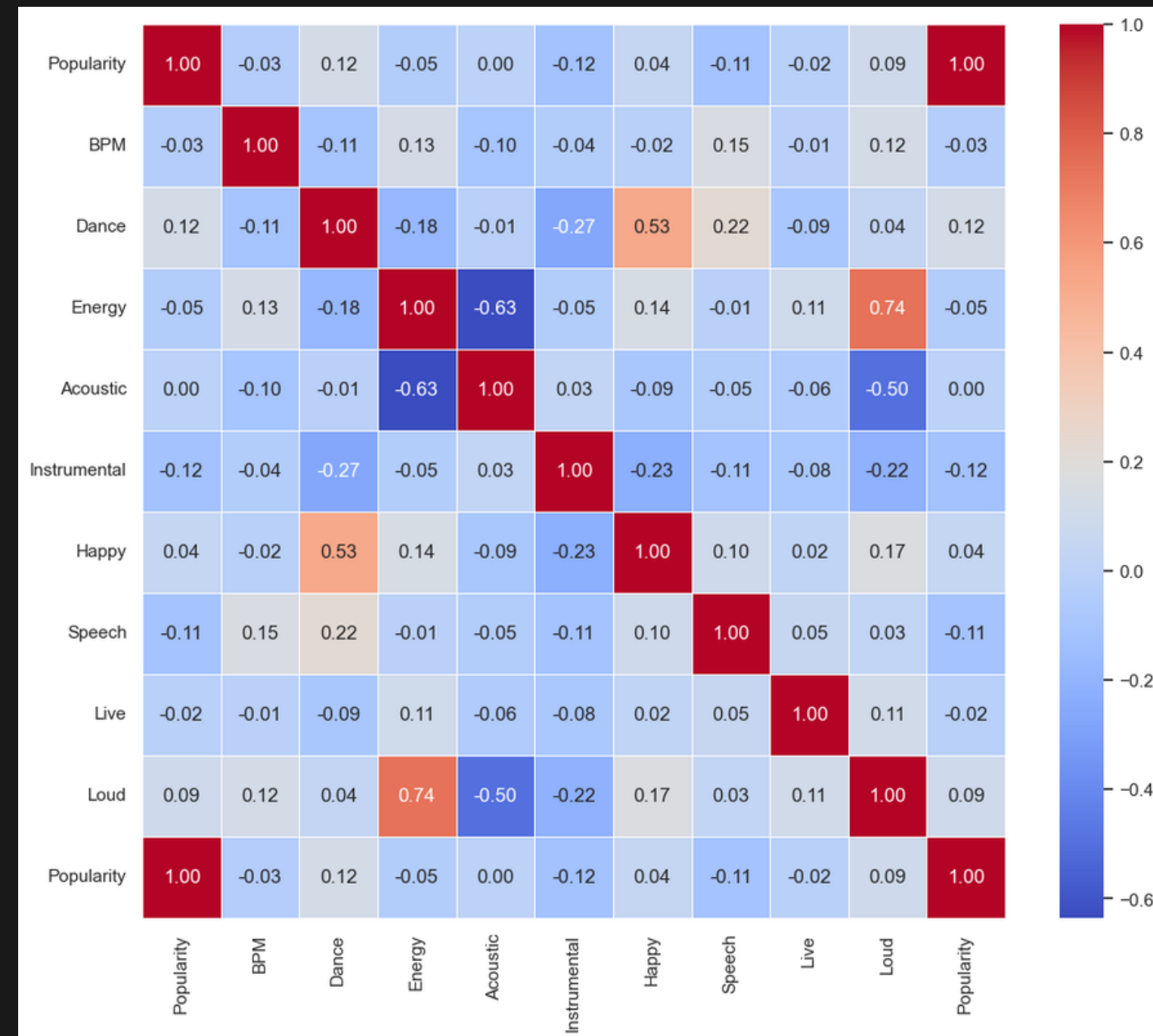
#	Column	Non-Null Count	Dtype
0	#	1000 non-null	int64
1	Song	1000 non-null	object
2	Artist	1000 non-null	object
3	Popularity	1000 non-null	int64
4	BPM	1000 non-null	int64
5	Genres	969 non-null	object
6	Parent Genres	962 non-null	object
7	Album	1000 non-null	object
8	Album Date	1000 non-null	object
9	Time	1000 non-null	object
10	Dance	1000 non-null	int64
11	Energy	1000 non-null	int64
12	Acoustic	1000 non-null	int64
13	Instrumental	1000 non-null	int64
14	Happy	1000 non-null	int64
15	Speech	1000 non-null	int64
16	Live	1000 non-null	int64
17	Loud	1000 non-null	int64
18	Key	1000 non-null	object
19	Time Signature	1000 non-null	int64
20	Added At	1000 non-null	object
21	Spotify Track Id	1000 non-null	object
22	Album Label	999 non-null	object
23	Camelot	1000 non-null	object
24	Spotify Track Img	630 non-null	object
25	Song Preview	1000 non-null	object

Visualization

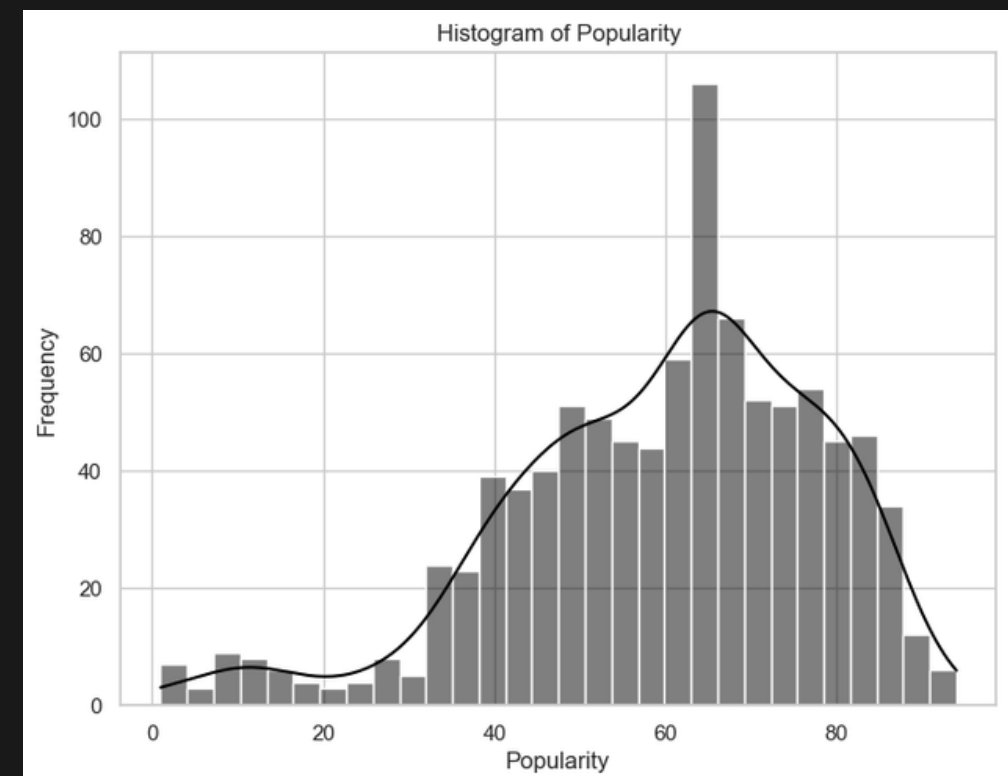
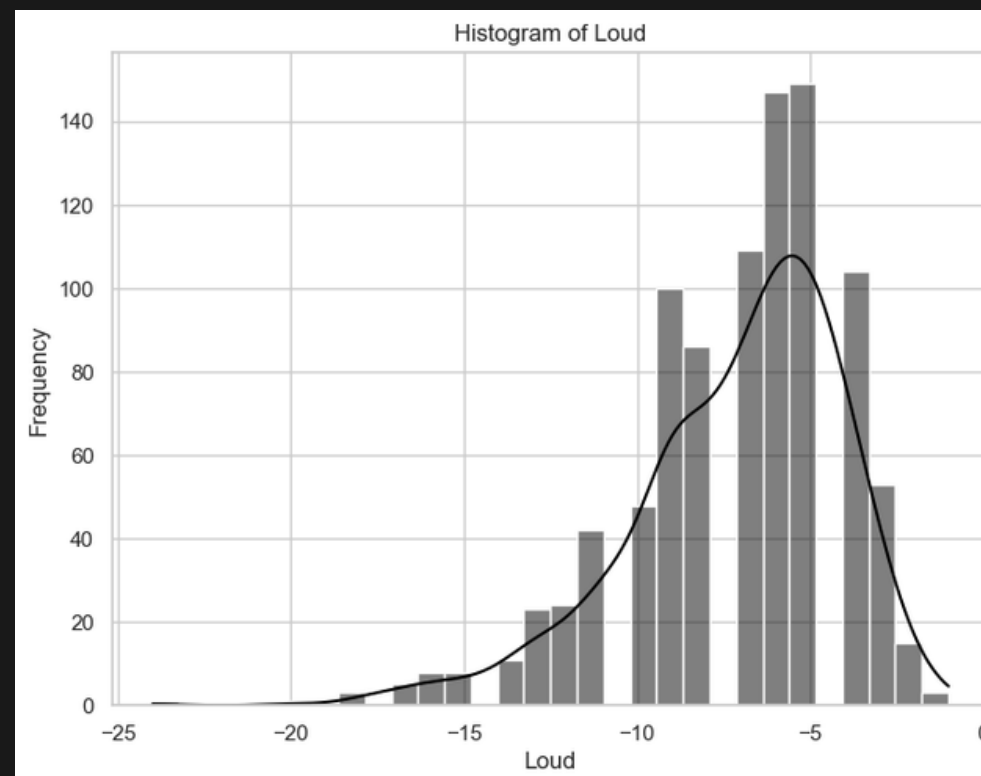
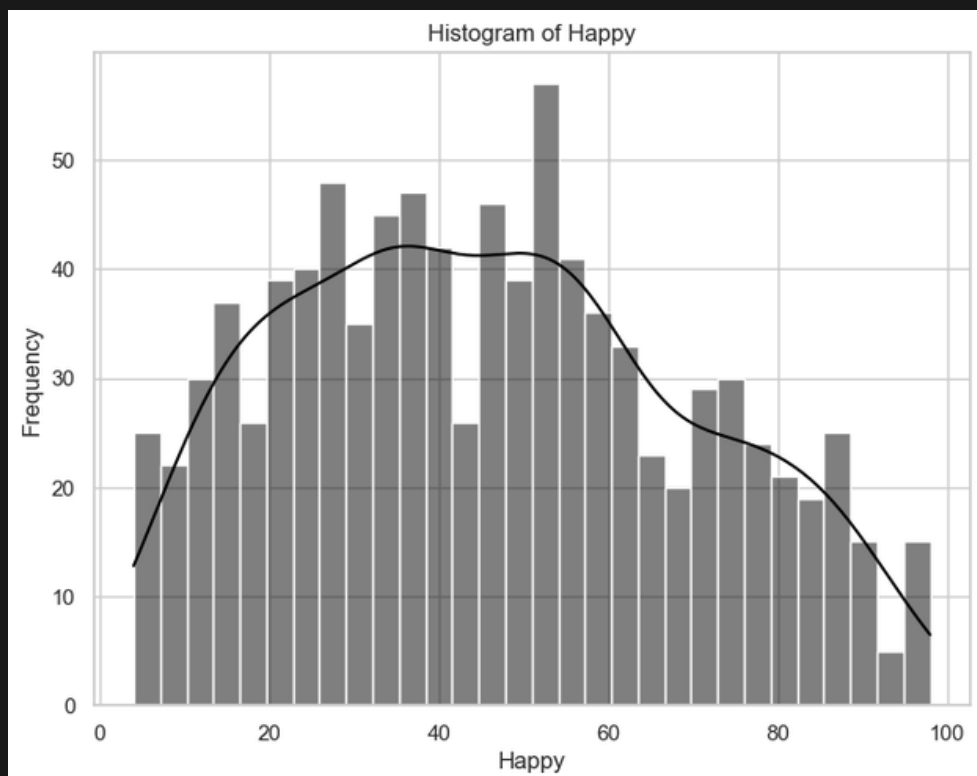
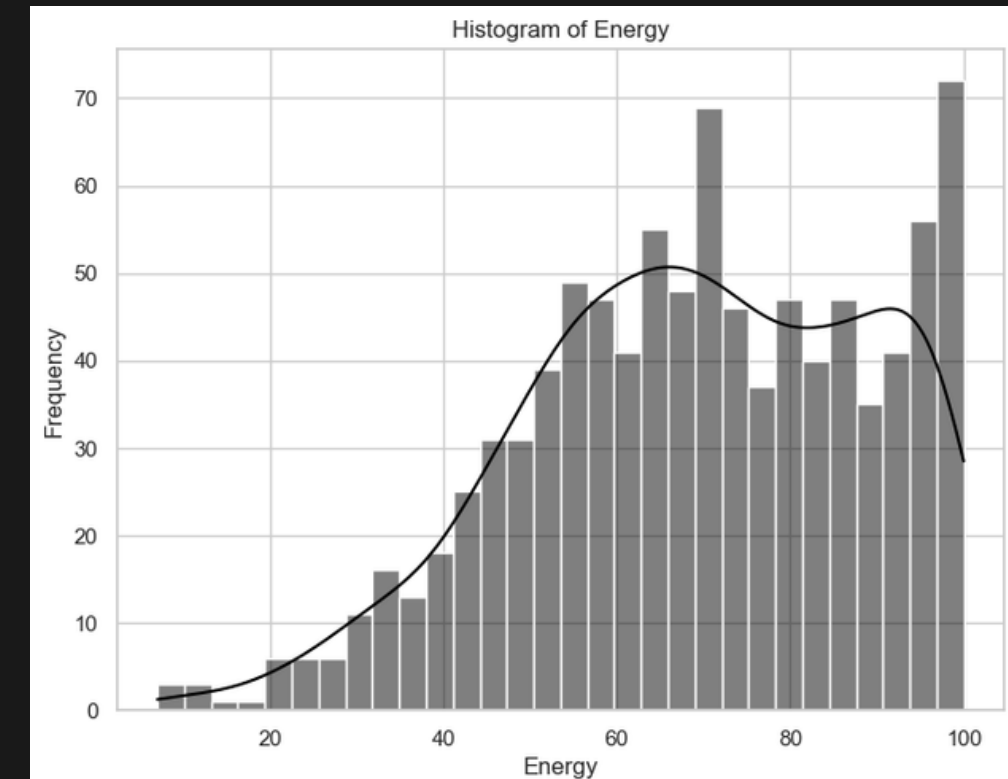
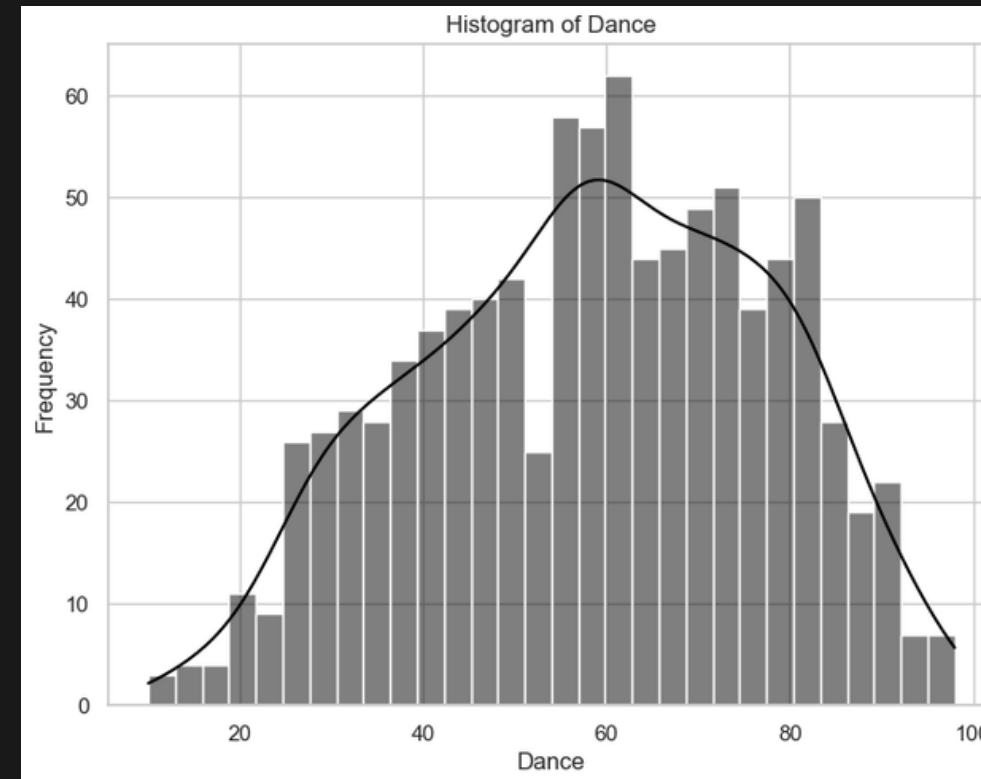
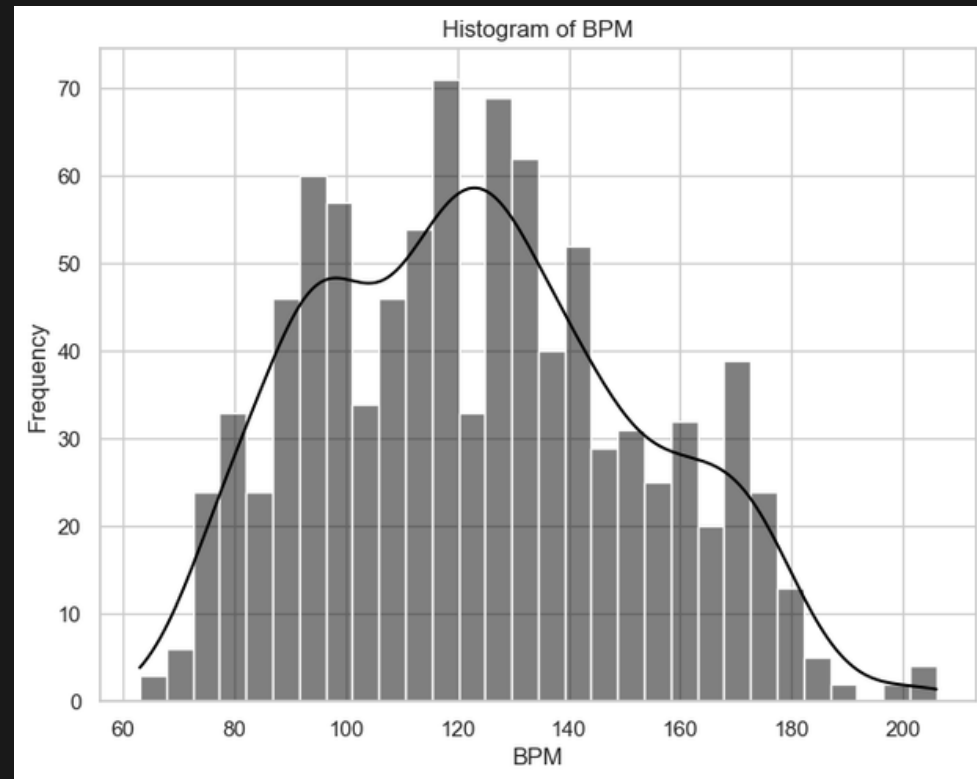
Starting with correlation data

All values seems like doesn't correlate with popularity at all

I wanted to see how I choose my songs and I targeted Popularity column



Musical values' histograms



Null Hypothesis

Popular songs have the same distribution of musical values (BPM, Dance, Energy, Acoustic, Instrumental, Happy, Speech, Live, Loud) as less popular songs. The musical values do not have a significant impact on the popularity of songs, and any observed differences are due to random chance.

Alternative Hypothesis

There is a significant difference in the distribution of musical values between popular and less popular songs. The musical values, such as BPM, Dance, Energy, etc., have a measurable influence on the popularity of songs. Popularity is not solely determined by random chance, and there are identifiable patterns in the musical characteristics of popular songs.

Hypothesis Testing

Picking the right test

We are checking relationships between values and we don't have normal distributions -> Spearman Correlation Test

```
from scipy.stats import spearmanr

musical_value_columns = ['Loud', 'Happy', 'Energy', 'Dance', 'BPM']

for column in musical_value_columns:
    threshold = most_df[column].median()

    # Check if there is more than one unique value
    if most_df[column].nunique() > 1:
        correlation, p_value = spearmanr(most_df[column], most_df['Popularity'])

        # Print results
        print(f'Correlation for {column}: {correlation:.4f}, p-value = {p_value:.4f}')

        # Check for significance at a 0.05 level (adjust as needed)
        if p_value < 0.05:
            print(f"The correlation between Popularity and {column} is statistically significant.")
        else:
            print(f"The correlation between Popularity and {column} is not statistically significant.")
```

✓ 0.0s

```
Correlation for Loud: 0.0878, p-value = 0.0070
The correlation between Popularity and Loud is statistically significant.
Correlation for Happy: 0.0378, p-value = 0.2473
The correlation between Popularity and Happy is not statistically significant.
Correlation for Energy: -0.0477, p-value = 0.1442
The correlation between Popularity and Energy is not statistically significant.
Correlation for Dance: 0.1281, p-value = 0.0001
The correlation between Popularity and Dance is statistically significant.
Correlation for BPM: -0.0299, p-value = 0.3592
The correlation between Popularity and BPM is not statistically significant.
```


What does this mean?

This shows that when I pick popular songs I tend to pick ones with high values of Dance and Loud.

There isn't enough data for other musical values to reject the null hypothesis.





Thank you!