In [ ]:
```
# top artists
# top songs
# listen mins by month
# genres
```

In [103…
```
import pandas as pd
import numpy as np
import requests
```

In [227…
```
# read in my spotify data. I am grabbing both the 2019-2024 and 2024-2025 da
# because the latter only has dec 2024 onwards not all of 2024

df_19to24= pd.read_json("/Users/dipanjanadas/Downloads/Spotify Extended Stre
df_24to25= pd.read_json("/Users/dipanjanadas/Downloads/Spotify Extended Stre

#putting the two datasets together
df_spotify= pd.concat([df_19to24, df_24to25])
```

In [229…
```
df_spotify.tail(5)
```

Out[229…

	ts	platform	ms_played	conn_country	ip_addr	mast
1631	2025-04-07 22:46:20+00:00	ios	185450	US	207.212.33.43	
1632	2025-04-07 23:30:28+00:00	ios	230400	US	207.212.33.43	
1633	2025-04-09 18:21:07+00:00	ios	92190	US	98.182.29.165	
1634	2025-04-09 18:21:14+00:00	osx	202656	US	98.182.29.165	
1635	2025-04-09 18:21:47+00:00	ios	33471	US	98.182.29.165	

5 rows × 23 columns

In [231…
```
df_spotify.info(memory_usage="deep")
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 17506 entries, 0 to 1635
Data columns (total 23 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   ts                                  17506 non-null  datetime64[ns, UTC]
 1   platform                            17506 non-null  object
 2   ms_played                           17506 non-null  int64
 3   conn_country                        17506 non-null  object
 4   ip_addr                             17506 non-null  object
 5   master_metadata_track_name          17486 non-null  object
 6   master_metadata_album_artist_name   17486 non-null  object
 7   master_metadata_album_album_name    17486 non-null  object
 8   spotify_track_uri                   17486 non-null  object
 9   episode_name                        20 non-null     object
 10  episode_show_name                   20 non-null     object
 11  spotify_episode_uri                 20 non-null     object
 12  audiobook_title                     0 non-null      float64
 13  audiobook_uri                       0 non-null      float64
 14  audiobook_chapter_uri               0 non-null      float64
 15  audiobook_chapter_title             0 non-null      float64
 16  reason_start                        17506 non-null  object
 17  reason_end                          17506 non-null  object
 18  shuffle                             17506 non-null  bool
 19  skipped                             17506 non-null  bool
 20  offline                             17506 non-null  bool
 21  offline_timestamp                   3026 non-null   float64
 22  incognito_mode                      17506 non-null  bool
dtypes: bool(4), datetime64[ns, UTC](1), float64(5), int64(1), object(12)
memory usage: 11.9 MB
```

In [233…

```
# dropping some columns bc they are not useful and/or contain no data

df_spotify= (df_spotify
.drop(["episode_name", "episode_show_name", "spotify_episode_uri",
"audiobook_title","audiobook_uri",      "audiobook_chapter_uri", "audiobook_
            )
```

In [235…

```
df_spotify.info(memory_usage="deep")
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 17506 entries, 0 to 1635
Data columns (total 16 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   ts                                  17506 non-null  datetime64[ns, UTC]
 1   platform                            17506 non-null  object
 2   ms_played                           17506 non-null  int64
 3   conn_country                        17506 non-null  object
 4   ip_addr                             17506 non-null  object
 5   master_metadata_track_name          17486 non-null  object
 6   master_metadata_album_artist_name   17486 non-null  object
 7   master_metadata_album_album_name    17486 non-null  object
 8   spotify_track_uri                   17486 non-null  object
 9   reason_start                        17506 non-null  object
 10  reason_end                          17506 non-null  object
 11  shuffle                             17506 non-null  bool
 12  skipped                             17506 non-null  bool
 13  offline                             17506 non-null  bool
 14  offline_timestamp                   3026 non-null   float64
 15  incognito_mode                      17506 non-null  bool
dtypes: bool(4), datetime64[ns, UTC](1), float64(1), int64(1), object(9)
memory usage: 10.2 MB
```

In [255…
```
df_spotify['date'] = df_spotify['ts'].dt.date
```

In [263…
```
df_spotify["date"]=pd.to_datetime(df_spotify["date"])
```

In [265…
```
df_spotify.info(memory_usage="deep")
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 17506 entries, 0 to 1635
Data columns (total 17 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   ts                                  17506 non-null  datetime64[ns, UTC]
 1   platform                            17506 non-null  object
 2   ms_played                           17506 non-null  int64
 3   conn_country                        17506 non-null  object
 4   ip_addr                             17506 non-null  object
 5   master_metadata_track_name          17486 non-null  object
 6   master_metadata_album_artist_name   17486 non-null  object
 7   master_metadata_album_album_name    17486 non-null  object
 8   spotify_track_uri                   17486 non-null  object
 9   reason_start                        17506 non-null  object
 10  reason_end                          17506 non-null  object
 11  shuffle                             17506 non-null  bool
 12  skipped                             17506 non-null  bool
 13  offline                             17506 non-null  bool
 14  offline_timestamp                   3026 non-null   float64
 15  incognito_mode                      17506 non-null  bool
 16  date                                17506 non-null  datetime64[ns]
dtypes: bool(4), datetime64[ns, UTC](1), datetime64[ns](1), float64(1), int6
4(1), object(9)
memory usage: 10.3 MB
```

In [277…
```
#filtering data from march 1 2024 to feb 28 2025 only so I get one year's wo

df_spotify=df_spotify.query("date >= '2024-03-01' and date<= '2025-02-28'")
```

In [279…
```
df_spotify.info(memory_usage="deep")
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2503 entries, 14592 to 1224
Data columns (total 17 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   ts                                  2503 non-null   datetime64[ns, UTC]
 1   platform                            2503 non-null   object
 2   ms_played                           2503 non-null   int64
 3   conn_country                        2503 non-null   object
 4   ip_addr                             2503 non-null   object
 5   master_metadata_track_name          2502 non-null   object
 6   master_metadata_album_artist_name   2502 non-null   object
 7   master_metadata_album_album_name    2502 non-null   object
 8   spotify_track_uri                   2502 non-null   object
 9   reason_start                        2503 non-null   object
 10  reason_end                          2503 non-null   object
 11  shuffle                             2503 non-null   bool
 12  skipped                             2503 non-null   bool
 13  offline                             2503 non-null   bool
 14  offline_timestamp                   2503 non-null   float64
 15  incognito_mode                      2503 non-null   bool
 16  date                                2503 non-null   datetime64[ns]
dtypes: bool(4), datetime64[ns, UTC](1), datetime64[ns](1), float64(1), int6
4(1), object(9)
memory usage: 1.4 MB
```

In [115…
```
#filtering data from march 1 2024 to feb 28 2025 only so I get one year's wo

#df_spotify=df_spotify.set_index("ts").loc["2024-03-01":"2025-02-28"]

# didn't end up using this bc this created a multi-index df which was annoyi
```

In [285…
```
# changing timezone

df_spotify["ts"]= df_spotify["ts"].dt.tz_convert('America/Los_Angeles')
```

In [287…
```
df_spotify.info(memory_usage="deep")
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2503 entries, 14592 to 1224
Data columns (total 17 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   ts                                    2503 non-null   datetime64[ns, Ameri
ca/Los_Angeles]
 1   platform                              2503 non-null   object
 2   ms_played                             2503 non-null   int64
 3   conn_country                          2503 non-null   object
 4   ip_addr                               2503 non-null   object
 5   master_metadata_track_name            2502 non-null   object
 6   master_metadata_album_artist_name     2502 non-null   object
 7   master_metadata_album_album_name      2502 non-null   object
 8   spotify_track_uri                     2502 non-null   object
 9   reason_start                          2503 non-null   object
 10  reason_end                            2503 non-null   object
 11  shuffle                               2503 non-null   bool
 12  skipped                               2503 non-null   bool
 13  offline                               2503 non-null   bool
 14  offline_timestamp                     2503 non-null   float64
 15  incognito_mode                        2503 non-null   bool
 16  date                                  2503 non-null   datetime64[ns]
dtypes: bool(4), datetime64[ns, America/Los_Angeles](1), datetime64[ns](1),
float64(1), int64(1), object(9)
memory usage: 1.4 MB
```

In [289…

```python
# change ms_played to s_played
df_spotify["s_played"]=df_spotify["ms_played"]/1000
```

In [291…

```python
# create year and month columns

df_spotify["year"]= df_spotify["date"].dt.year
df_spotify["month"]= df_spotify["date"].dt.month
```

In [293…

```python
df_spotify.head()
```

Out[293…

	ts	platform	ms_played	conn_country	ip_addr	master_m
14592	2024-03-05 11:12:06-08:00	osx	200373	IN	106.212.95.233	
14593	2024-03-05 11:13:11-08:00	osx	5627	IN	106.212.95.233	He
14594	2024-03-05 11:16:30-08:00	osx	200373	IN	106.212.95.233	
14595	2024-03-05 11:16:53-08:00	osx	21024	IN	106.212.95.233	He
14596	2024-03-05 11:17:25-08:00	osx	32426	IN	106.212.95.233	

In [295…

```
# create a new track_uri column stripped of the spotify bit. this is going t
# handy later when we try to gather genre from spotify api

mask= df_spotify["spotify_track_uri"].str.split(":", expand=True)
df_spotify['track_uri']= mask[2]
```

In [299…

```
mask # splits by the : so creates 3 columns. I picked the third one (2 in co
```

Out[299…

	0	1	2
14592	spotify	track	4nc6XiUze2Yh7wFueGOPv7
14593	spotify	track	5PUXKVVVQ74C3gl5vKy9Li
14594	spotify	track	4nc6XiUze2Yh7wFueGOPv7
14595	spotify	track	5PUXKVVVQ74C3gl5vKy9Li
14596	spotify	track	4nc6XiUze2Yh7wFueGOPv7
...	...	...	...
1220	spotify	track	4lnAN2S1fcI0SjxEbksZVr
1221	spotify	track	44MuEHdlociG8KjhPhOVw5
1222	spotify	track	0uuQLn4o2ZCWiuzeyrAcAR
1223	spotify	track	4NJtlrapihMPiOlZ396uus
1224	spotify	track	5zCnGtCl5Ac5zlFHXaZmhy

2503 rows × 3 columns

In [536…
```
df_spotify= df_spotify.dropna()
```

In [538…
```
df_spotify.shape # the above code had dropped one row
#I did this bc the code below kept breaking on a null value in track_uri
```

Out[538… (2502, 21)

In [596…
```
# I didn't get genres from the entire list of track_uris bc spotipy kept bre
# intead bc I had many repeating track_uris I just took the unique values ar
# them. Later I join the genre to the list of 2502 track_uris so it all work


import spotipy
from spotipy.oauth2 import SpotifyClientCredentials
import pandas as pd

#  Spotify API credentials
CLIENT_ID = "ed105e6eed804c3abfd9a1ad4fbc3af8"
CLIENT_SECRET = "1e3ebf908ab8425fb28783b588023243"

# Initialize Spotify client with client credentials
client_credentials_manager = SpotifyClientCredentials(client_id=CLIENT_ID, c
sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)

# List of track IDs
#track_ids = ['TRACK_ID_1', 'TRACK_ID_2', 'TRACK_ID_3']
#track_ids= df_spotify["track_uri"].values
track_ids=df_spotify.track_uri.unique() # only the unique track uris( no rep

# Initialize lists to store data
track_ids_list = []
artist_names_list = []
first_genres_list = []
#genres_list=[]
for track_id in track_ids:
    #try:
        # Get track information
        track_info = sp.track(track_id)

       # if track_info:
        artist_name = track_info['artists'][0]['name']
        artist_id = track_info['artists'][0]['id']

            # Get artist information to retrieve genres
        artist_info = sp.artist(artist_id)
        genres = artist_info.get('genres', [])

            # Get the first genre if available
        first_genre = genres[1] if genres else None

            # Append data to lists
        track_ids_list.append(track_id)
        artist_names_list.append(artist_name)
        first_genres_list.append(first_genre)
        #genres_list.append(', '.join(genres) if genres else 'No genres avai
```

```
      # else:
       #    print(f"Could not retrieve information for track ID: {track_id}")
           # Append None values to maintain list lengths
         #  track_ids_list.append(track_id)
         #  artist_names_list.append(None)
         #  first_genres_list.append(None)

   # except spotipy.exceptions.SpotifyException as e:
     #   print(f"Error fetching data for track ID {track_id}: {e}")
        # Append None values in case of an error
      #  track_ids_list.append(track_id)
     #   artist_names_list.append(None)
      #  first_genres_list.append(None)

   # except Exception as e:
    #    print(f"An unexpected error occurred for track ID {track_id}: {e}")
        # Append None values in case of an error
    #    track_ids_list.append(track_id)
    #    artist_names_list.append(None)
     #   first_genres_list.append(None)

# Create the DataFrame
data = {
    'track_uri': track_ids_list,
    'Artist': artist_names_list,
    'Genre_list': genres_list
}
df = pd.DataFrame(data)

# Print the DataFrame
print(df)
```

```
                  track_uri               Artist  \
0    4nc6XiUze2Yh7wFueGOPv7  Anirudh Ravichander
1    5PUXKVVVQ74C3gl5vKy9Li        Jasleen Royal
2    6ZzYETKetIfNUsZUb23jgG               HONNE
3    1lNHWPDvKEbamKezpLq7HW               HONNE
4    0GPJSHYaXh8rZSSJoUMgyl               HONNE
..                      ...                 ...
665  67NdAOAn0EzvyrMYduzzZm         Anne Wilson
666  7cWnks0lsRtpAi87COOiXK      Salim–Sulaiman
667  3fPgIknlkDWXs1l2noKZbp             Badshah
668  3TAhWtQnpoL5Vl9VQPl9fU       Farhan Akhtar
669  2j2rmGPa2bNqvHijeyWLj2         Neha Kakkar

                                          Genre_list
0    tamil pop, kollywood, tamil dance, tollywood, ...
1    hindi pop, bollywood, gujarati pop, desi, hind...
2                                No genres available
3                                No genres available
4                                No genres available
..                                               ...
665        christian country, christian, worship, ccm
666         bollywood, sufi, desi, hindi pop, qawwali
667  bollywood, desi, hindi pop, desi hip hop, desi...
668                         bollywood, desi, hindi pop
669  bollywood, hindi pop, desi, desi pop, gujarati...

[670 rows x 3 columns]
Processing complete.
```

In [614…
```
df.head()
```

Out[614…

	track_uri	Artist	First Genre
0	4nc6XiUze2Yh7wFueGOPv7	Anirudh Ravichander	tamil pop
1	5PUXKVVVQ74C3gl5vKy9Li	Jasleen Royal	hindi pop
2	6ZzYETKetIfNUsZUb23jgG	HONNE	None
3	1lNHWPDvKEbamKezpLq7HW	HONNE	None
4	0GPJSHYaXh8rZSSJoUMgyl	HONNE	None

In [566…
```
#genre_data_unique_uris= df
```

In [610…
```
# renamed trackid column to match track_uri in the original df_spotify df, s

df.rename(columns={'Track ID': 'track_uri'}, inplace=True)
```

In [602…
```
#filtered the track_uris from df_spotify and turned in into a new data frame

genre_df= pd.DataFrame(df_spotify["track_uri"])
```

In [604…
```
genre_df.head() #check that the dataframe formed as intended, probably shoul
```

Out[604…

	track_uri
14592	4nc6XiUze2Yh7wFueGOPv7
14593	5PUXKVVVQ74C3gl5vKy9Li
14594	4nc6XiUze2Yh7wFueGOPv7
14595	5PUXKVVVQ74C3gl5vKy9Li
14596	4nc6XiUze2Yh7wFueGOPv7

In [608…
```
genre_df.shape # checking rows of genre_df to make sure it has all the data
```

Out[608… (2502, 1)

In [616…
```
# merge the genres from df to the track_uri list in genre_df
df_trackID_genre_merged= genre_df.merge(df,
                                        how="left",
                                        left_on="track_uri",
                                        right_on="track_uri")
```

In [624…
```
df_trackID_genre_merged.tail(25) # checking to see that the merge worked
```

Out[624…

	track_uri	Artist	First Genre
2477	4nc6XiUze2Yh7wFueGOPv7	Anirudh Ravichander	tamil pop
2478	0RBw4ODUQPO4cuAOZtBGga	Tory Lanez	None
2479	44MuEHdlociG8KjhPhOVw5	Kylie Minogue	dance pop
2480	4lnAN2S1fcI0SjxEbksZVr	Selena Gomez	pop
2481	4lnAN2S1fcI0SjxEbksZVr	Selena Gomez	pop
2482	1eZefeDb8uOsjvcbl1fJrG	Diljit Dosanjh	bhangra
2483	4lnAN2S1fcI0SjxEbksZVr	Selena Gomez	pop
2484	4lnAN2S1fcI0SjxEbksZVr	Selena Gomez	pop
2485	0RBw4ODUQPO4cuAOZtBGga	Tory Lanez	None
2486	7cWnks0lsRtpAi87COOiXK	Salim–Sulaiman	bollywood
2487	7cWnks0lsRtpAi87COOiXK	Salim–Sulaiman	bollywood
2488	7cWnks0lsRtpAi87COOiXK	Salim–Sulaiman	bollywood
2489	3fPgIknlkDWXs1l2noKZbp	Badshah	bollywood
2490	3fPgIknlkDWXs1l2noKZbp	Badshah	bollywood
2491	3TAhWtQnpoL5Vl9VQPl9fU	Farhan Akhtar	bollywood
2492	2j2rmGPa2bNqvHijeyWLj2	Neha Kakkar	bollywood
2493	3fPgIknlkDWXs1l2noKZbp	Badshah	bollywood
2494	5a2Hoi1wuhCA6Ob7pbOlpW	Divya Kumar	bollywood
2495	0RBw4ODUQPO4cuAOZtBGga	Tory Lanez	None
2496	1eZefeDb8uOsjvcbl1fJrG	Diljit Dosanjh	bhangra
2497	4lnAN2S1fcI0SjxEbksZVr	Selena Gomez	pop
2498	44MuEHdlociG8KjhPhOVw5	Kylie Minogue	dance pop
2499	0uuQLn4o2ZCWiuzeyrAcAR	Klangkarussell	None
2500	4NJtlrapihMPiOlZ396uus	MEYY	None
2501	5zCnGtCl5Ac5zlFHXaZmhy	Ram Sampath	bollywood

In [620…
```
df_trackID_genre_merged.shape # checking to see that the merge worked
```

Out[620…
```
(2502, 3)
```

In [626…
```
df_spotify.shape
```

Out[626…
```
(2502, 21)
```

In [628…
```
# saving df_spotify and df_trackID_genre_merged as csv files so I can load i
df_spotify.to_csv('mySpotifyData.csv')
df_trackID_genre_merged.to_csv('genres.csv')
```

In [642…
```
df_trackID_genre_merged.loc["First Genre"].unique()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[642], line 1
----> 1 df_trackID_genre_merged.loc["First Genre"].unique()

File /opt/anaconda3/lib/python3.12/site-packages/pandas/core/indexing.py:119
1, in _LocationIndexer.__getitem__(self, key)
   1189 maybe_callable = com.apply_if_callable(key, self.obj)
   1190 maybe_callable = self._check_deprecated_callable_usage(key, maybe_ca
llable)
-> 1191 return self._getitem_axis(maybe_callable, axis=axis)

File /opt/anaconda3/lib/python3.12/site-packages/pandas/core/indexing.py:143
1, in _LocIndexer._getitem_axis(self, key, axis)
   1429 # fall thru to straight lookup
   1430 self._validate_key(key, axis)
-> 1431 return self._get_label(key, axis=axis)

File /opt/anaconda3/lib/python3.12/site-packages/pandas/core/indexing.py:138
1, in _LocIndexer._get_label(self, label, axis)
   1379 def _get_label(self, label, axis: AxisInt):
   1380     # GH#5567 this will fail if the label is not present in the axi
s.
-> 1381     return self.obj.xs(label, axis=axis)

File /opt/anaconda3/lib/python3.12/site-packages/pandas/core/generic.py:430
1, in NDFrame.xs(self, key, axis, level, drop_level)
   4299             new_index = index[loc]
   4300 else:
-> 4301     loc = index.get_loc(key)
   4303     if isinstance(loc, np.ndarray):
   4304         if loc.dtype == np.bool_:

File /opt/anaconda3/lib/python3.12/site-packages/pandas/core/indexes/range.p
y:417, in RangeIndex.get_loc(self, key)
    415         raise KeyError(key) from err
    416 if isinstance(key, Hashable):
--> 417     raise KeyError(key)
    418 self._check_indexing_error(key)
    419 raise KeyError(key)

KeyError: 'First Genre'
```

In [650…
```
jupyter nbconvert --to pdf spotify.ipynb
```

```
  Cell In[650], line 1
    jupyter nbconvert --to pdf spotify.ipynb
            ^
SyntaxError: invalid syntax
```

In [ ]: