# Making Sense of Data: *Genres* + *Tracks*

Dear Diary,
It is Saturday, September 21. I am sitting in the Bean🥫 .

The purpose of this notebook is to make sense of the data contained in the UCI FMA Music Analysis Dataset (https://archive.ics.uci.edu/ml/datasets/FMA:+A+Dataset+For+Music+Analysis): **genres, and tracks**.
For genres, we are interested in exploring the **colors** associated with each sub-genre and the **hierarchy structure** organizing the 164 genres. For tracks, we are interested in mapping tracks to genres to find the genres with the most songs to use for our initial model. We also want to explore associated track metadata, such as **year**.

---

# Genres

The `raw_genres.csv` file was small enough that it was easier to analyze the data in Google Sheets. Sorry to betray the CS community by using layman's tools.

The file had 164 rows with the following columns:

| genre_id | genre_color | genre_handle | genre_parent_id | genre_title |
|---|---|---|---|---|
| 46 | #CC3300 | Latin_America | 2 | Latin America |
| ... | ... | ... | ... | ... |

## Comments:

- Parent genres did not have `parent_ids`.
- The rows were in a haphazard order; they were not sorted numerically by `genre_id`/`genre_parent_id` nor alphabetically by `genre_handle`/`genre_title`.
- I did not consider `genre_color`, but if it was sorted by color, that's not useful to me.

## In Google Sheets, I did the following:

1. Sorted rows by `parent_id` to get a sense of which genres had the most breadth (the most sub-genres).
2. This moved all the parent rows to the bottom, and I pulled them out to the side.
3. I created two new columns for the parent sub-table, `num sub_genres`.
4. I counted all instances of each sub genre and added it to the parent table.

## Results:

| Top Genres (sub-genres) | Graph |
|---|---|

| Top Genres (sub-genres) | Graph |
|---|---|
| | |

## Tracks

The file containing track data is too big to assess in Google Sheets (wah). Let's do some pandas parsing activities. The goal here is to see if the top genres above (based on sub-genre) matches the quantity of tracks for each genre. I'll start by loading `raw_tracks.csv` into a pandas df:

In [67]:

```
import numpy as np
import pandas as pd

# change filepath if running on another machine, this is local to mine
tracks = pd.read_csv("/Users/mkarroqe/Desktop/github/dancing-screen/fma_metadata/raw_tracks.csv")
tracks
```

Out[67]:

| | track_id | album_id | album_title | albu |
|---|---|---|---|---|
| **0** | 2 | 1.0 | AWOL - A Way Of Life | http://freemusicarchive.org/music/AWOL/AWC _... |
| **1** | 3 | 1.0 | AWOL - A Way Of Life | http://freemusicarchive.org/music/AWOL/AWC _... |
| **2** | 5 | 1.0 | AWOL - A Way Of Life | http://freemusicarchive.org/music/AWOL/AWC _... |
| **3** | 10 | 6.0 | Constant Hitmaker | http://freemusicarchive.org/music/Kurt_Vile/C |
| **4** | 20 | 4.0 | Niris | http://freemusicarchive.org/music/Chris_and_ |
| **5** | 26 | 4.0 | Niris | http://freemusicarchive.org/music/Chris_and_ |
| **6** | 30 | 4.0 | Niris | http://freemusicarchive.org/music/Chris_and_ |
| **7** | 46 | 4.0 | Niris | http://freemusicarchive.org/music/Chris_and_ |
| **8** | 48 | 4.0 | Niris | http://freemusicarchive.org/music/Chris_and_ |
| **9** | 134 | 1.0 | AWOL - A Way Of Life | http://freemusicarchive.org/music/AWOL/AWC _... |
| **10** | 135 | 58.0 | mp3 | http://freemusicarchive.org/music/Abominog/r |
| **11** | 136 | 58.0 | mp3 | http://freemusicarchive.org/music/Abominog/r |
| **12** | 137 | 59.0 | Live at LACE | http://freemusicarchive.org/music/Airway/Live |
| **13** | 138 | 59.0 | Live at LACE | http://freemusicarchive.org/music/Airway/Live |
| **14** | 139 | 60.0 | Every Man For Himself | http://freemusicarchive.org/music/Alec_K_Rec |

|  | track_id | album_id | album_title | albu |
|---|---|---|---|---|
| **15** | 140 | 61.0 | The Blind Spot | http://freemusicarchive.org/music/Alec_K_Rec |
| **16** | 141 | 60.0 | Every Man For Himself | http://freemusicarchive.org/music/Alec_K_Rec |
| **17** | 142 | 62.0 | The Quiet Room | http://freemusicarchive.org/music/Alec_K_Rec |
| **18** | 144 | 64.0 | Amoebiasis | http://freemusicarchive.org/music/Amoebic_E |
| **19** | 145 | 64.0 | Amoebiasis | http://freemusicarchive.org/music/Amoebic_E |
| **20** | 146 | 65.0 | Limbic Rage | http://freemusicarchive.org/music/Amoebic_E |
| **21** | 147 | 65.0 | Limbic Rage | http://freemusicarchive.org/music/Amoebic_E |
| **22** | 148 | 66.0 | Contradiction | http://freemusicarchive.org/music/Contradictio |
| **23** | 149 | 67.0 | Two Gong/Wire Pieces | http://freemusicarchive.org/music/Andy_Hayle |
| **24** | 150 | 67.0 | Two Gong/Wire Pieces | http://freemusicarchive.org/music/Andy_Hayle |
| **25** | 151 | 68.0 | Make Them Suffer | http://freemusicarchive.org/music/Animal_Writ |
| **26** | 152 | 68.0 | Make Them Suffer | http://freemusicarchive.org/music/Animal_Writ |
| **27** | 153 | 69.0 | Arc and Sender | http://freemusicarchive.org/music/Arc_and_Se |
| **28** | 154 | 69.0 | Arc and Sender | http://freemusicarchive.org/music/Arc_and_Se |
| **29** | 155 | 70.0 | unreleased demo | http://freemusicarchive.org/music/Arc_and_Se |
| **...** | ... | ... | ... | ... |

|  | track_id | album_id | album_title | albu |
|---|---|---|---|---|
| **109697** | 155290 | 22935.0 | Return | http://freemusicarchive.org/music/Alex_Masor |
| **109698** | 155291 | 22935.0 | Return | http://freemusicarchive.org/music/Alex_Masor |
| **109699** | 155292 | 22935.0 | Return | http://freemusicarchive.org/music/Alex_Masor |
| **109700** | 155293 | 22935.0 | Return | http://freemusicarchive.org/music/Alex_Masor |
| **109701** | 155294 | 22935.0 | Return | http://freemusicarchive.org/music/Alex_Masor |
| **109702** | 155295 | 22935.0 | Return | http://freemusicarchive.org/music/Alex_Masor |
| **109703** | 155296 | 22935.0 | Return | http://freemusicarchive.org/music/Alex_Masor |
| **109704** | 155297 | 22935.0 | Return | http://freemusicarchive.org/music/Alex_Masor |
| **109705** | 155298 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |
| **109706** | 155299 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |
| **109707** | 155300 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |
| **109708** | 155301 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |
| **109709** | 155302 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |
| **109710** | 155303 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |
| **109711** | 155304 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |
| **109712** | 155305 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |
| **109713** | 155306 | 22936.0 | Scissors Paper Stone | http://freemusicarchive.org/music/Greg_Atkins |

|  | track_id | album_id | album_title | albu |
|---|---|---|---|---|
| **109714** | 155307 | 22937.0 | Live at WFMU with Scott Williams, 3/27/2017 | http://freemusicarchive.org/music/awott/Live_ |
| **109715** | 155308 | 22937.0 | Live at WFMU with Scott Williams, 3/27/2017 | http://freemusicarchive.org/music/awott/Live_ |
| **109716** | 155309 | 22937.0 | Live at WFMU with Scott Williams, 3/27/2017 | http://freemusicarchive.org/music/awott/Live_ |
| **109717** | 155310 | 22937.0 | Live at WFMU with Scott Williams, 3/27/2017 | http://freemusicarchive.org/music/awott/Live_ |
| **109718** | 155311 | 22937.0 | Live at WFMU with Scott Williams, 3/27/2017 | http://freemusicarchive.org/music/awott/Live_ |
| **109719** | 155312 | 22937.0 | Live at WFMU with Scott Williams, 3/27/2017 | http://freemusicarchive.org/music/awott/Live_ |
| **109720** | 155314 | 22940.0 | Live at Monty Hall, 2/17/2017 | http://freemusicarchive.org/music/Spowder/Li |
| **109721** | 155315 | 22940.0 | Live at Monty Hall, 2/17/2017 | http://freemusicarchive.org/music/Spowder/Li |
| **109722** | 155316 | 22940.0 | Live at Monty Hall, 2/17/2017 | http://freemusicarchive.org/music/Spowder/Li |

|  | track_id | album_id | album_title | albu |
|---|---|---|---|---|
| **109723** | 155317 | 22940.0 | Live at Monty Hall, 2/17/2017 | http://freemusicarchive.org/music/Spowder/Li |
| **109724** | 155318 | 22940.0 | Live at Monty Hall, 2/17/2017 | http://freemusicarchive.org/music/Spowder/Li |
| **109725** | 155319 | 22940.0 | Live at Monty Hall, 2/17/2017 | http://freemusicarchive.org/music/Spowder/Li |
| **109726** | 155320 | 22906.0 | What I Tell Myself Vol. 2 | http://freemusicarchive.org/music/Forget_the_ |

109727 rows × 39 columns

In [199]:

```
tracks['track_url'][0]
```

Out[199]:

```
'http://freemusicarchive.org/music/AWOL/AWOL_-_A_Way_Of_Life/Food'
```

Next, I want to examine the `track_genres` column:

In [138]:

```
genres = tracks['track_genres']
genres_df = pd.DataFrame(genres)
genres_df
```

```
Out[138]:
```

| | track_genres |
|---|---|
| **0** | [{'genre_id': '21', 'genre_title': 'Hip-Hop', ... |
| **1** | [{'genre_id': '21', 'genre_title': 'Hip-Hop', ... |
| **2** | [{'genre_id': '21', 'genre_title': 'Hip-Hop', ... |
| **3** | [{'genre_id': '10', 'genre_title': 'Pop', 'gen... |
| **4** | [{'genre_id': '76', 'genre_title': 'Experiment... |
| **5** | [{'genre_id': '76', 'genre_title': 'Experiment... |
| **6** | [{'genre_id': '76', 'genre_title': 'Experiment... |
| **7** | [{'genre_id': '76', 'genre_title': 'Experiment... |
| **8** | [{'genre_id': '76', 'genre_title': 'Experiment... |
| **9** | [{'genre_id': '21', 'genre_title': 'Hip-Hop', ... |
| **10** | [{'genre_id': '45', 'genre_title': 'Loud-Rock'... |
| **11** | [{'genre_id': '45', 'genre_title': 'Loud-Rock'... |
| **12** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **13** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **14** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **15** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **16** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **17** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **18** | [{'genre_id': '4', 'genre_title': 'Jazz', 'gen... |
| **19** | [{'genre_id': '4', 'genre_title': 'Jazz', 'gen... |
| **20** | [{'genre_id': '4', 'genre_title': 'Jazz', 'gen... |
| **21** | [{'genre_id': '4', 'genre_title': 'Jazz', 'gen... |
| **22** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **23** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **24** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **25** | [{'genre_id': '25', 'genre_title': 'Punk', 'ge... |
| **26** | [{'genre_id': '25', 'genre_title': 'Punk', 'ge... |
| **27** | [{'genre_id': '26', 'genre_title': 'Post-Rock'... |

| | track_genres |
|---|---|
| **28** | [{'genre_id': '26', 'genre_title': 'Post-Rock'... |
| **29** | [{'genre_id': '26', 'genre_title': 'Post-Rock'... |
| **...** | ... |
| **109697** | [{'genre_id': '18', 'genre_title': 'Soundtrack... |
| **109698** | [{'genre_id': '18', 'genre_title': 'Soundtrack... |
| **109699** | [{'genre_id': '18', 'genre_title': 'Soundtrack... |
| **109700** | [{'genre_id': '18', 'genre_title': 'Soundtrack... |
| **109701** | [{'genre_id': '18', 'genre_title': 'Soundtrack... |
| **109702** | [{'genre_id': '18', 'genre_title': 'Soundtrack... |
| **109703** | [{'genre_id': '18', 'genre_title': 'Soundtrack... |
| **109704** | [{'genre_id': '18', 'genre_title': 'Soundtrack... |
| **109705** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109706** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109707** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109708** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109709** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109710** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109711** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109712** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109713** | [{'genre_id': '17', 'genre_title': 'Folk', 'ge... |
| **109714** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **109715** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **109716** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **109717** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **109718** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **109719** | [{'genre_id': '1', 'genre_title': 'Avant-Garde... |
| **109720** | [{'genre_id': '25', 'genre_title': 'Punk', 'ge... |
| **109721** | [{'genre_id': '25', 'genre_title': 'Punk', 'ge... |
| **109722** | [{'genre_id': '25', 'genre_title': 'Punk', 'ge... |
| **109723** | [{'genre_id': '25', 'genre_title': 'Punk', 'ge... |

|  | track_genres |
|---|---|
| **109724** | [{'genre_id': '25', 'genre_title': 'Punk', 'ge... |
| **109725** | [{'genre_id': '25', 'genre_title': 'Punk', 'ge... |
| **109726** | [{'genre_id': '10', 'genre_title': 'Pop', 'gen... |

109727 rows × 1 columns

Next, I want to create a dictionary that maps genres to number of tracks with those genres.

In [188]:

```
lst = eval(genres_df.iloc[i][0])
lst
```

Out[188]:

```
[{'genre_id': '10',
  'genre_title': 'Pop',
  'genre_url': 'http://freemusicarchive.org/genre/Pop/'},
 {'genre_id': '12',
  'genre_title': 'Rock',
  'genre_url': 'http://freemusicarchive.org/genre/Rock/'},
 {'genre_id': '169',
  'genre_title': 'Rockabilly',
  'genre_url': 'http://freemusicarchive.org/genre/Rockabilly/'}]
```

In [174]:

```
rows = len(genres_df)
genre_count = {}
bad_rows = []

for i in range(0, rows):
    try:
        lst = eval(genres_df.iloc[i][0])
        for dic in lst:
            if dic['genre_id'] in genre_count:
                genre_count[dic['genre_id']] += 1
            else:
                genre_count[dic['genre_id']] = 1
    except:
        bad_rows.append(i)
```

Saving `genre_count` and `bad_rows` as a `.pkl` because the above cell took a while to run:

In [186]:

```
import pickle
with open("genre_count.pkl", "wb") as f:
    pickle.dump(genre_count, f)
with open("bad_rows.pkl", "wb") as f:
    pickle.dump(bad_rows, f)

print(round((len(bad_rows)/rows)*100, 2), 'percent of rows are bad.')
```

```
2.38 percent of rows are bad.
```

Now I want the top 5 genre_ids:

In [213]:

```
from collections import Counter
k = Counter(genre_count)
highest = k.most_common(5)

max_genre_ids = []
for tup in highest:
    max_genre_ids.append(tup[0])

highest
```

Out[213]:

```
[('38', 25493), ('15', 24530), ('1', 9183), ('12', 8406), ('76', 73
30)]
```

Now, I want to map the top ids to their names:

In [214]:

```
# change filepath if running on another machine, this is local to mine
raw_genres = pd.read_csv("/Users/mkarroqe/Desktop/github/dancing-screen/fma_metadata/raw_genres.csv")
raw_genres
```

```
Out[214]:
```

|  | genre_id | genre_color | genre_handle | genre_parent_id | genre_ |
|---|---|---|---|---|---|
| **0** | 1 | #006666 | Avant-Garde | 38.0 | Avant-Gar |
| **1** | 2 | #CC3300 | International | NaN | Internatior |
| **2** | 3 | #000099 | Blues | NaN | Blues |
| **3** | 4 | #990099 | Jazz | NaN | Jazz |
| **4** | 5 | #8A8A65 | Classical | NaN | Classical |
| **5** | 6 | #4D0000 | Novelty | 38.0 | Novelty |
| **6** | 7 | #009999 | Comedy | 20.0 | Comedy |
| **7** | 8 | #665666 | Old-Time__Historic | NaN | Old-Time / Historic |
| **8** | 9 | #663366 | Country | NaN | Country |
| **9** | 10 | #009900 | Pop | NaN | Pop |
| **10** | 11 | #E40089 | Disco | 14.0 | Disco |
| **11** | 12 | #840000 | Rock | NaN | Rock |
| **12** | 13 | #5B747C | Easy_Listening | 126.0 | Easy Listening |
| **13** | 14 | #330033 | Soul-RB | NaN | Soul-RnB |
| **14** | 15 | #FF6600 | Electronic | NaN | Electronic |
| **15** | 16 | #003366 | Sound_Effects | 6.0 | Sound Eff |
| **16** | 17 | #5E6D3F | Folk | NaN | Folk |
| **17** | 18 | #669933 | Soundtrack | 1235.0 | Soundtrac |
| **18** | 19 | #5E6D3F | Funk | 14.0 | Funk |
| **19** | 20 | #006699 | Spoken | NaN | Spoken |
| **20** | 21 | #CC0000 | Hip-Hop | NaN | Hip-Hop |
| **21** | 22 | #dddd00 | Audio_Collage | 38.0 | Audio Coll |
| **22** | 25 | #840000 | Punk | 12.0 | Punk |
| **23** | 26 | #840000 | Post-Rock | 12.0 | Post-Rock |
| **24** | 27 | #840000 | Lo-fi | 12.0 | Lo-Fi |
| **25** | 30 | #00eeff | Field_Recordings | 38.0 | Field Recording |

|  | genre_id | genre_color | genre_handle | genre_parent_id | genre_ |
|---|---|---|---|---|---|
| **26** | 31 | #777777 | Metal | 12.0 | Metal |
| **27** | 32 | #222222 | Noise | 38.0 | Noise |
| **28** | 33 | #5E6D3F | Psych-Folk | 17.0 | Psych-Fol |
| **29** | 36 | #840000 | Krautrock | 12.0 | Krautrock |
| **...** | ... | ... | ... | ... | ... |
| **134** | 491 | #FF6600 | Skweee | 468.0 | Skweee |
| **135** | 493 | #663366 | western_swing | 651.0 | Western Swing |
| **136** | 495 | #FF6600 | Downtempo | 15.0 | Downtemp |
| **137** | 502 | #CC3300 | Cumbia | 46.0 | Cumbia |
| **138** | 504 | #CC3300 | Latin | 2.0 | Latin |
| **139** | 514 | #dddd00 | Sound_Art | 38.0 | Sound Art |
| **140** | 524 | #CC3300 | Romany_Gypsy | 130.0 | Romany (Gypsy) |
| **141** | 538 | #E40089 | compilation | 18.0 | Compilatio |
| **142** | 539 | #CC0000 | rap | 21.0 | Rap |
| **143** | 542 | #CC0000 | breakbeat | 21.0 | Breakbeat |
| **144** | 567 | #000099 | Gospel | 3.0 | Gospel |
| **145** | 580 | #CC0000 | Abstract_Hip-Hop | 1172.0 | Abstract H Hop |
| **146** | 602 | #CC3300 | Reggae_-_Dancehall | 79.0 | Reggae - Dancehall |
| **147** | 619 | #CC3300 | Spanish | 130.0 | Spanish |
| **148** | 651 | #663366 | Country__Western | 9.0 | Country & Western |
| **149** | 659 | #8A8A65 | Contemporary_Classical_1147 | 5.0 | Contempo Classical |
| **150** | 693 | #CC0000 | Wonky | 21.0 | Wonky |
| **151** | 695 | #FF6600 | Jungle | 15.0 | Jungle |
| **152** | 741 | #CC3300 | Klezmer | 130.0 | Klezmer |
| **153** | 763 | #D4A017 | holiday | 763.0 | Holiday |
| **154** | 806 | #CC0000 | hiphop | 21.0 | hiphop |

|  | genre_id | genre_color | genre_handle | genre_parent_id | genre_ |
|---|---|---|---|---|---|
| **155** | 808 | #CC3300 | Salsa | 46.0 | Salsa |
| **156** | 810 | #5B747C | Nu-Jazz | 51.0 | Nu-Jazz |
| **157** | 811 | #CC0000 | Hip-Hop_Beats | 21.0 | Hip-Hop Beats |
| **158** | 906 | #990099 | Modern_Jazz | 4.0 | Modern Ja |
| **159** | 1032 | #CC3300 | Turkish | 102.0 | Turkish |
| **160** | 1060 | #CC3300 | tango | 46.0 | Tango |
| **161** | 1156 | #CC3300 | Fado | 130.0 | Fado |
| **162** | 1193 | #D4A017 | Christmas | 763.0 | Christmas |
| **163** | 1235 | #000000 | Instrumental | NaN | Instrumen |

164 rows × 5 columns

In [216]:

```
for i in range(0, len(raw_genres)):
    if str(raw_genres.iloc[i][0]) in max_genre_ids:
        print(raw_genres.iloc[i][4])

highest
```

```
Avant-Garde
Rock
Electronic
Experimental
Experimental Pop
```

Out[216]:

```
[('38', 25493), ('15', 24530), ('1', 9183), ('12', 8406), ('76', 73
30)]
```

# IN CONCLUSION,

Our top genres by *number of sub-genres* are:

- International: 15
- Rock: 15
- Electronic: 14
- Experimental: 14
- Spoken: 8

Our top genres by *number of tracks* are:

- Experimental: 25,493
- Electronic: 24,530
- Avant-Garde: 9,183
- Rock: 8,406
- Experimental Pop: 7,330

# THE PROBLEM IS,

The FMA dataset website is currently in the middle of a merger, and all the audio files are currently unavailable. So we don't know what any of the tracks under these genres sound like.

### sooo... to be discussed in the next notebook :-)