Project 2: Phân tích dữ liệu

Trang Soundcloud là một trang web cho phép người dùng upload và chia sẻ các bài hát. Chúng ta sẽ phân tích dữ liệu về các nghệ sĩ, ban nhạc, podcast và người sáng tác âm nhạc trên trang Soundcloud thông qua đồ án này

Bảng phân công công việc:

MSSV	Họ và tên	Công việc
19120057	Lê Quốc Cường	Phân tích dữ liệu của users
19120124	Võ Thị Cẩm Quỳnh	Phân tích dữ liệu của playlists - Tiền xử lý, phân tích câu hỏi 2
19120142	Nguyễn Thị Phương Trang	Phân tích dữ liệu của playlists - Phân tích câu hỏi 1.3
18120429	Phạm Trung Kiên	Phân tích dữ liệu của tracks - Tiền xử lý, trả lời câu hỏi số 1
19120145	Lê Đào Duy Trọng	Phân tích dữ liệu của tracks - Tiền xử lý, trả lời câu hỏi số 2

Cài đặt môi trường

Import các thư viện cần thiết

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import datetime as dt
import warnings
import copy
```

Phần 1: Phân tích dữ liệu của user:

Thông tin cơ bản về dữ liệu

- **id**: Mã định danh của người dùng
- **avartar_url**: Đường dẫn đến avatar của người dùng
- **followers_count**: Số lượng người theo dõi người dùng
- **full_name**: Họ tên đầy đủ của người dùng
- **last_modified**: Lần chỉnh sửa cuối cùng về profile của người dùng
- **permalink_url**: Đường dẫn trực tiếp đến người dùng trên soundcloud
- **username**: Tên người dùng trên soundcloud
- **verified**: Trạng thái xác nhận của soundcloud đối với người dùng
- **city**: Thành phố của người dùng
- **country_code**: Mã quốc gia của người dùng
- **badges**: Danh hiệu người dùng đã đạt được

Nội dung thực hiện

1. Tiền xử lí dữ liệu

- Thay thế "None" bằng Nan
- Xóa column "first_name", "last_name"
- Xóa các bộ dữ liệu có tỉ lệ missing lớn hơn 25%

2. Khám phá dữ liệu

- Dữ liệu có kiểu là số
- Dữ liệu có kiểu không phải là số

3. Câu hỏi

- Phân bố user ở các quốc gia ?
- Quốc gia nào được yêu thích nhất ?
- Tương quan giữa số lượng followers và trạng thái xác nhận tài khoản

1.1 Tiền xử lí dữ liệu

In [2]:
```
data = pd.read_csv("user.csv")
data.head()
```

Out[2]:

	id	avatar_url	first_name	followers_count	full_name	last_modified	last_nar
0	8	https://i1.sndcdn.com/avatars-000208970259-vng...	Alexander	180223	Alexander Ljung	2020-10-20T14:38:52Z	Lju
1	183	https://i1.sndcdn.com/avatars-000012778523-59q...	Eric	133715	Eric Wahlforss	2020-12-16T21:40:33Z	Wahlfor
2	1691	https://i1.sndcdn.com/avatars-000000018772-c5d...	Ark	265	Ark Kozuki	2020-08-17T12:33:59Z	Kozu
3	2157	https://i1.sndcdn.com/avatars-N3HNyXVCJnJ8YiLQ...	Kim	1501	Kim	2021-08-17T10:55:32Z	No
4	1719	https://i1.sndcdn.com/avatars-000000012820-990...	None	877	None	2017-09-07T21:15:46Z	No

In [3]:
```
#Thay thế giá trị "None" bằng NaN
df = data.replace(to_replace = 'None', value = np.nan)
df.head()
```

Out[3]:

	id	avatar_url	first_name	followers_count	full_name	last_modified	last_nar
0	8	https://i1.sndcdn.com/avatars-000208970259-vng...	Alexander	180223	Alexander Ljung	2020-10-20T14:38:52Z	Lju
1	183	https://i1.sndcdn.com/avatars-000012778523-59q...	Eric	133715	Eric Wahlforss	2020-12-16T21:40:33Z	Wahlfor
2	1691	https://i1.sndcdn.com/avatars-000000018772-c5d...	Ark	265	Ark Kozuki	2020-08-17T12:33:59Z	Kozu
3	2157	https://i1.sndcdn.com/avatars-N3HNyXVCJnJ8YiLQ...	Kim	1501	Kim	2021-08-17T10:55:32Z	Na
4	1719	https://i1.sndcdn.com/avatars-000000012820-990...	NaN	877	NaN	2017-09-07T21:15:46Z	Na

In [4]:
```
# Dữ liệu của User có trường full_name, first_name, last_name vì vậy ta chỉ giữ l
filter = ['first_name' , 'last_name']
df = df.drop(filter, axis = 1)
print("Kich thuoc cu: ",data.shape)
print("Kich thuoc moi: ",df.shape)
```

```
Kich thuoc cu:  (1001, 13)
Kich thuoc moi:  (1001, 11)
```

In [5]:
```
#Kế tiếp ta sẽ xóa các bộ dữ liệu missing data quá ngưỡng limit(ở đây limit = 25%
limit = 25
def isNaN(num):
      return num!=num

def delMissingRow(Data, limit):
    data_temp=copy.deepcopy(Data)
    percent=0
    count=0
    end = len(data_temp['id'])
    i=0
    while i<end:
        count=0
        for key in data_temp:
            if isNaN(data_temp[key][i]):
                count+=1
        if(count/len(data_temp.iloc[i]))>limit/100:
            data_temp.drop(i,inplace = True)
            end-=1
        i+=1
    return data_temp

print("Kich thuoc cu: ",df.shape)
df = delMissingRow(df,limit)
print("Kich thuoc moi: ",df.shape)
```

```
Kich thuoc cu:  (1001, 11)
Kich thuoc moi:  (865, 11)
```

1.2. Khám phá dữ liệu

1.2.1. Phân bố các giá trị ở các cột có kiểu dữ liệu là số

- Ở cột "id" thì không có nhiều ý nghĩa nên ta bỏ qua
- Ở cột "followers_count" ta thấy độ lệch chuẩn quá lớn so với trung bình, nên khi ta phân tích sẽ không mang lại được nhiều ý nghĩa, nên ta không phân tích sâu về cột này mà sẽ sử dụng ở 1 vấn đề khác

In [6]:
```
df.describe()
```

Out[6]:

	id	followers_count
count	865.000000	865.000000
mean	31713.197688	2701.028902
std	22835.064535	16277.508384
min	8.000000	0.000000
25%	10891.000000	25.000000
50%	26627.000000	109.000000
75%	54666.000000	649.000000
max	70740.000000	331030.000000

1.2.2. Phân bố các giá trị ở các cột có kiểu dữ liệu không phải là số 1.

- Ta thấy được ở các cột "verified", "city", "country_code","badges" số lượng các giá trị unique không quá lớn so với số lượng giá trị, nên ta sẽ khám phá sâu vào các trường dữ liệu này
- Các cột còn lại thì hầu như số lượng các giá trị unique bằng số lượng giá trị, nên sẽ không mang lại nhiều ý nghĩa khi phân tích

In [7]:
```
df.describe(include = "all")
```

Out[7]:

	id	avatar_url	followers_count	full_name	last
count	865.000000	865	865.000000	691	
unique	NaN	784	NaN	689	
top	NaN	https://a1.sndcdn.com/images/default_avatar_la...	NaN	Nightology Records	07T
freq	NaN	82	NaN	2	
mean	31713.197688	NaN	2701.028902	NaN	
std	22835.064535	NaN	16277.508384	NaN	
min	8.000000	NaN	0.000000	NaN	
25%	10891.000000	NaN	25.000000	NaN	
50%	26627.000000	NaN	109.000000	NaN	
75%	54666.000000	NaN	649.000000	NaN	
max	70740.000000	NaN	331030.000000	NaN	

1.3. Câu hỏi

1.3.1. Số lượng user ở các quốc gia phân bố như thế nào?

In [8]:
```
userInCountry = df.country_code.value_counts()
print(userInCountry.head())
print(userInCountry.tail())
```

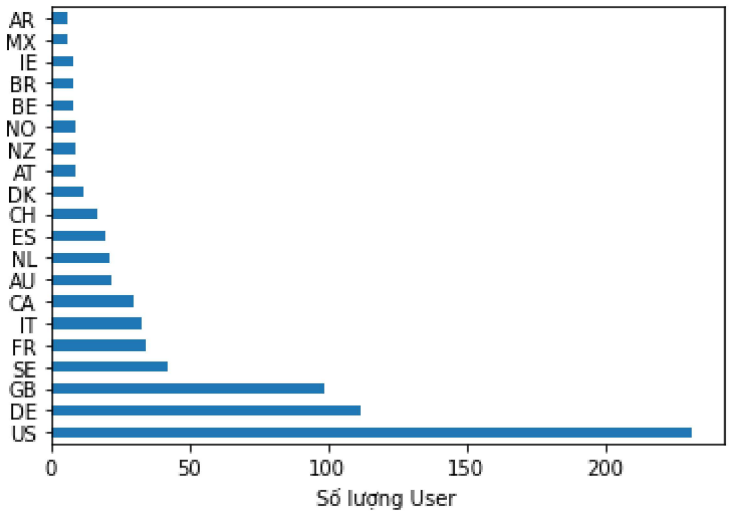
```
US    231
DE    112
GB     99
SE     42
FR     34
Name: country_code, dtype: int64
CO    1
LT    1
KG    1
EC    1
MY    1
Name: country_code, dtype: int64
```

Trực quan hóa dữ liệu

- Có quốc gia chỉ có 1 vài user nên ta loại bỏ các quốc gia này ra khỏi biểu đồ

In [9]:
```
userInCountry.head(20).plot.barh().set_xlabel('Số lượng User')
```

Out[9]:
```
Text(0.5, 0, 'Số lượng User')
```



1.3.2 Quốc gia nào được yêu thích nhất?

- Quốc gia được yêu thích được tính bằng tổng số lượng các flower của các user ở quốc gia đó

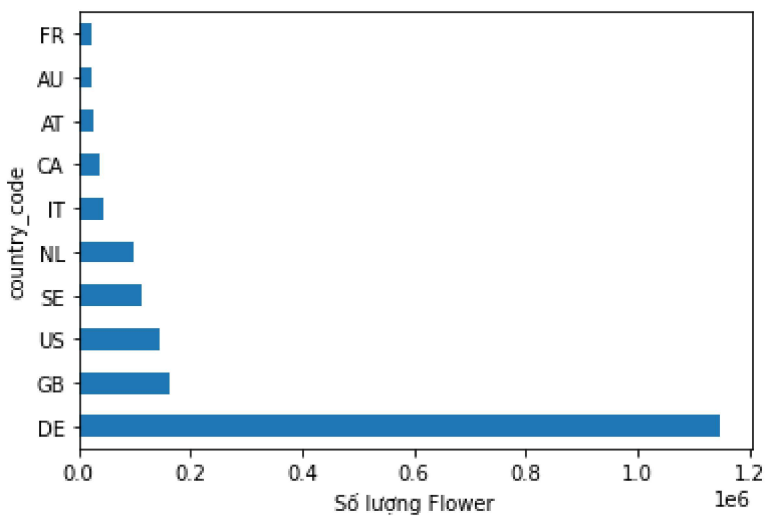
In [10]:
```
lovely_country = df.groupby("country_code")["followers_count"].sum()
lovely_country = lovely_country.sort_values(ascending = False)
lovely_country.head()
```

Out[10]:
```
country_code
DE    1145362
GB     163388
US     144214
SE     113721
NL      98518
Name: followers_count, dtype: int64
```

Trực quan hóa dữ liệu

In [11]:
```
lovely_country.head(10).plot.barh().set_xlabel('Số lượng Flower')
```

Out[11]:
```
Text(0.5, 0, 'Số lượng Flower')
```



1.3.3 Xem xét sự tương quan của một user được xác nhận với số lượng follower của user đó?

In [12]:
```
row = ["Non_Verified","Verified"]
col = ["Users_count","Followers_count"]
follower_data = []
follower_data.append(df.groupby("verified")["followers_count"].count().values)
follower_data.append(df.groupby("verified")["followers_count"].sum().values)
```

In [13]:
```
follower_df = pd.DataFrame(follower_data, columns = row, index = col)
follower_df
```

Out[13]:

	Non_Verified	Verified
Users_count	831	34
Followers_count	858656	1477734

Nhận xét

- Ta nhận thấy được số lượng user được xác nhận rất ít so với số lượng user chưa được xác nhận nhưng tổng số lượng follower của user đã được xác nhận cao hơn hẳn so với tổng số lượng follower của user chưa được xác nhận.

Phần 2: Phân tích dữ liệu của Playlist

2.1. Danh sách mong đợi:

- Sự phân bố của các biến chính, ở đây khi nhìn vào các biến có kiểu dữ liệu là số, bao gồm `duration` , `likes_count` , `reposts_count` , `track_count` tụi em mong muốn rằng gần như giá trị của các biến này đều sẽ trên mức trung bình, có thể một số vị trí ít hơn hoặc thấp hơn giá trị trung bình (có hình tương tự như hình tháp).
- Mối quan hệ/ tương quan giữa các cặp biến, ở đây tụi em đặt câu hỏi liệu có sự tương quan nào giữa các cặp biến `reposts_count - likes_count` , `track_count - duration` hay không?
  - `reposts_count - likes_count` : có phải những playlist có lượt repost (đăng tải lại) càng nhiều thì lượt like (thích) của chúng càng nhiều không?
  - `track_count - duration` : số lượng các track có trong 1 playlist có ảnh hưởng gì đến thời lượng của playlist này hay không?

2.2. Danh sách kiểm tra phân tích dữ liệu

Trả lời những câu hỏi về bộ dữ liệu:

- Xác định số liệu trước khi bắt đầu:
  - Thực hiện tổng quan về dữ liệu: thực hiện tìm hiểu ý nghĩa các cột dữ liệu, ở đây tụi em dựa trên nội dung hiểu và tài liệu 'SoundCloud Public API Specification' để đưa nội dung của các trường dữ liệu.
- Thực hiện tổng quan về dữ liệu.

In [14]:
```
df=pd.read_csv("playlist.csv")
df.head()
```

Out[14]:

	id	artwork_url	created_at	description	duration	genre	label_name	la
0	1	https://i1.sndcdn.com/artworks-000000009550-e6...	2008-04-29T22:14:25Z	A collection of old sounds	826086	None	None	2
1	18	https://i1.sndcdn.com/artworks-000067273270-ui...	2008-04-30T01:21:15Z	My 2003 debut album made on the road with a la...	3247119	Electronic	sonarkollektiv	2
2	32	None	2008-05-01T04:12:43Z	artist / noe\r\n	354742	ambient pop	Floor Limit	1
3	33	None	2008-05-01T04:13:55Z	None	254066	abstract hip-hop	Floor Limit	1
4	34	None	2008-05-01T17:31:06Z	None	50620378	None	None	1

5 rows × 25 columns

Tổng quan về dữ liệu

Type *Markdown* and LaTeX: α^2

Ý nghĩa các cột dữ liệu

- `id` :Playlist identifier
- `artwork_url` : URL to a JPEG image
- `created_at` :Created timestamp
- `description` :Playlist description
- `duration` :Playlist duration in milliseconds
- `genre` :Playlist genre
- `label_name` :Label name
- `last_modified` :Last modified timestamp
- `likes_count` :Count of playlist likes
- `managed_by_feeds` :Whether or not the playlist is managed by an RSS feed
- `permalink_url` :Playlist permalink URL
- `public` :Is the playlist public?
- `purchase_title` :Purchase title

- `purchase_url` :Purchase URL
- `release_date` :Release timestamp
- `reposts_count` :number of reposts from user
- `secret_token` :secret_token
- `tag_list` :Tags
- `title` :Playlist title
- `user_id` :User identifier
- `is_album` :True if the playlist is an album, false if it is just a playlist
- `display_date` :Display timestamp
- `user` `SoundCloud` :User object who uploaded the track
- `tracks` :List of tracks
- `track_count` :Count of tracks

Xác định số liệu, kích thước của bộ dữ liệu

In [15]:
```
num_rows=len(df)
num_cols=len(df.columns)
df=df.replace('None',np.NaN)
num_rows,num_cols
```

Out[15]:
```
(1508, 25)
```

Xác định kiểu dữ liệu có trong bộ dữ liệu

In [16]:
```python
df = df.astype({"id":object,"user_id":object,"user":object})
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1508 entries, 0 to 1507
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   id                1508 non-null   object
 1   artwork_url       485 non-null    object
 2   created_at        1508 non-null   object
 3   description       852 non-null    object
 4   duration          1508 non-null   int64
 5   genre             614 non-null    object
 6   label_name        330 non-null    object
 7   last_modified     1508 non-null   object
 8   likes_count       1508 non-null   int64
 9   managed_by_feeds  1508 non-null   bool
 10  permalink_url     1508 non-null   object
 11  public            1508 non-null   bool
 12  purchase_title    16 non-null     object
 13  purchase_url      185 non-null    object
 14  release_date      242 non-null    object
 15  reposts_count     1508 non-null   int64
 16  secret_token      0 non-null      float64
 17  tag_list          357 non-null    object
 18  title             1508 non-null   object
 19  user_id           1508 non-null   object
 20  is_album          1508 non-null   bool
 21  display_date      1508 non-null   object
 22  user              1508 non-null   object
 23  tracks            1137 non-null   object
 24  track_count       1508 non-null   int64
dtypes: bool(3), float64(1), int64(4), object(17)
memory usage: 263.7+ KB
```

In [17]:
```
df.describe(include='all')
```

Out[17]:

	id	artwork_url	created_at	description	duration	genre
count	1508.0	485	1508	852	1.508000e+03	614
unique	1508.0	485	1508	843	NaN	291
top	1.0	https://i1.sndcdn.com/artworks-000000009550-e6...	2008-04-29T22:14:25Z	Notice! These are hi MP3 files to download for...	NaN	Bluegrass
freq	1.0	1	1	4	NaN	115
mean	NaN	NaN	NaN	NaN	5.459063e+06	NaN
std	NaN	NaN	NaN	NaN	1.656793e+07	NaN
min	NaN	NaN	NaN	NaN	0.000000e+00	NaN
25%	NaN	NaN	NaN	NaN	5.346672e+05	NaN
50%	NaN	NaN	NaN	NaN	1.343620e+06	NaN

2.3. Tiền xử lý dữ liệu

Xóa dữ liệu:

- Dữ liệu ở cột `managed_by_feeds` , `public` , `secret_token` chỉ mang 1 giá trị, không có ảnh hưởng đến các giá trị khác.
- Các dữ liệu đều phù hợp, không có giá trị ngoại lệ

In [18]:
```
def del_columns(dataframe):
    dele=[]
    for col in dataframe.columns:
        count=dataframe[col].isnull().sum()
        if count == len(dataframe) or len(dataframe[col].unique())<=1:
            dele.append(col)
    return dele

dele=del_columns(df)
print(dele)
df=df.drop(columns=dele)
df.head()
```

```
['managed_by_feeds', 'public', 'secret_token']
```

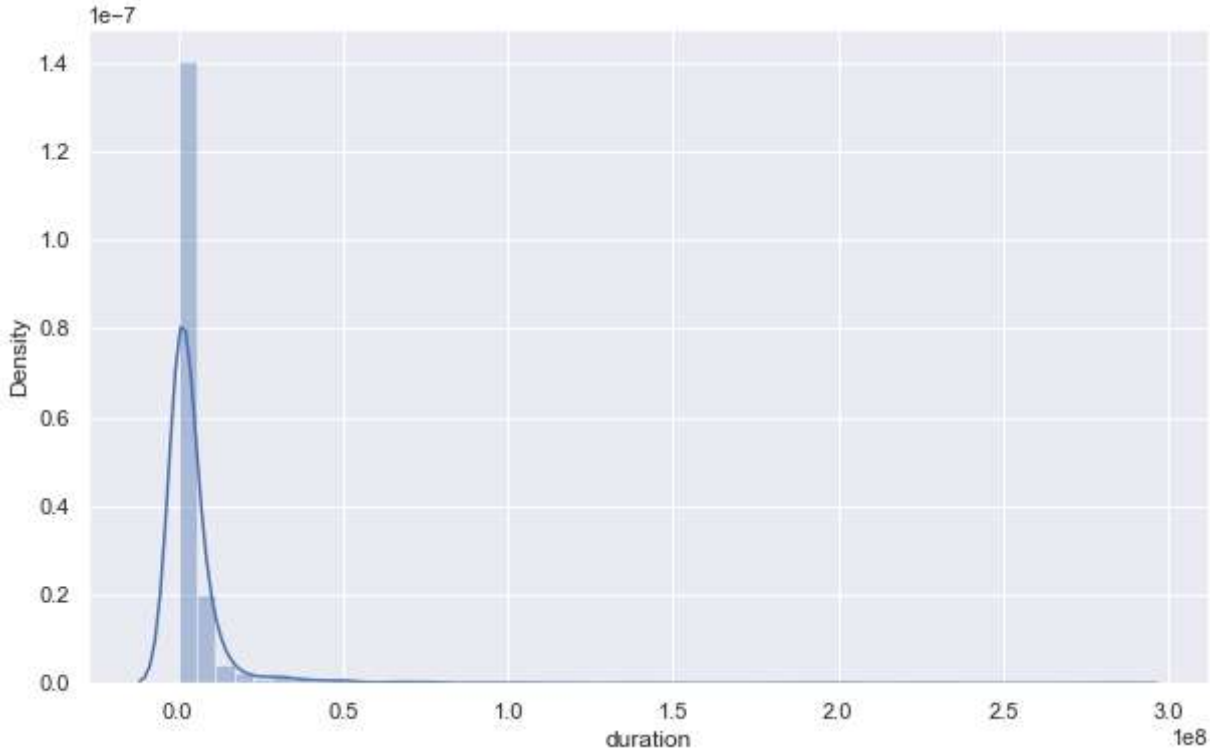
Out[18]:

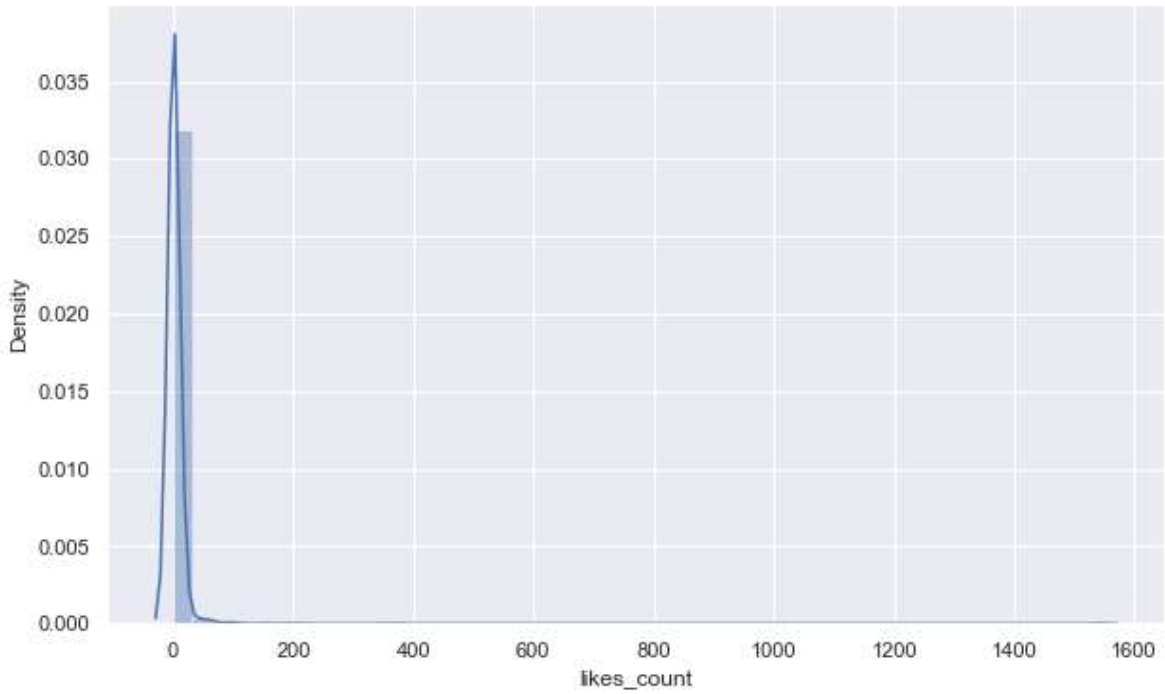
	id	artwork_url	created_at	description	duration	genre	label_name
0	1	https://i1.sndcdn.com/artworks-000000009550-e6...	2008-04-29T22:14:25Z	A collection of old sounds	826086	NaN	NaN
1	18	https://i1.sndcdn.com/artworks-000067273270-ui...	2008-04-30T01:21:15Z	My 2003 debut album made on the road with a la...	3247119	Electronic	sonarkollektiv
2	32	NaN	2008-05-01T04:12:43Z	artist / noe\r\n	354742	ambient pop	Floor Limi

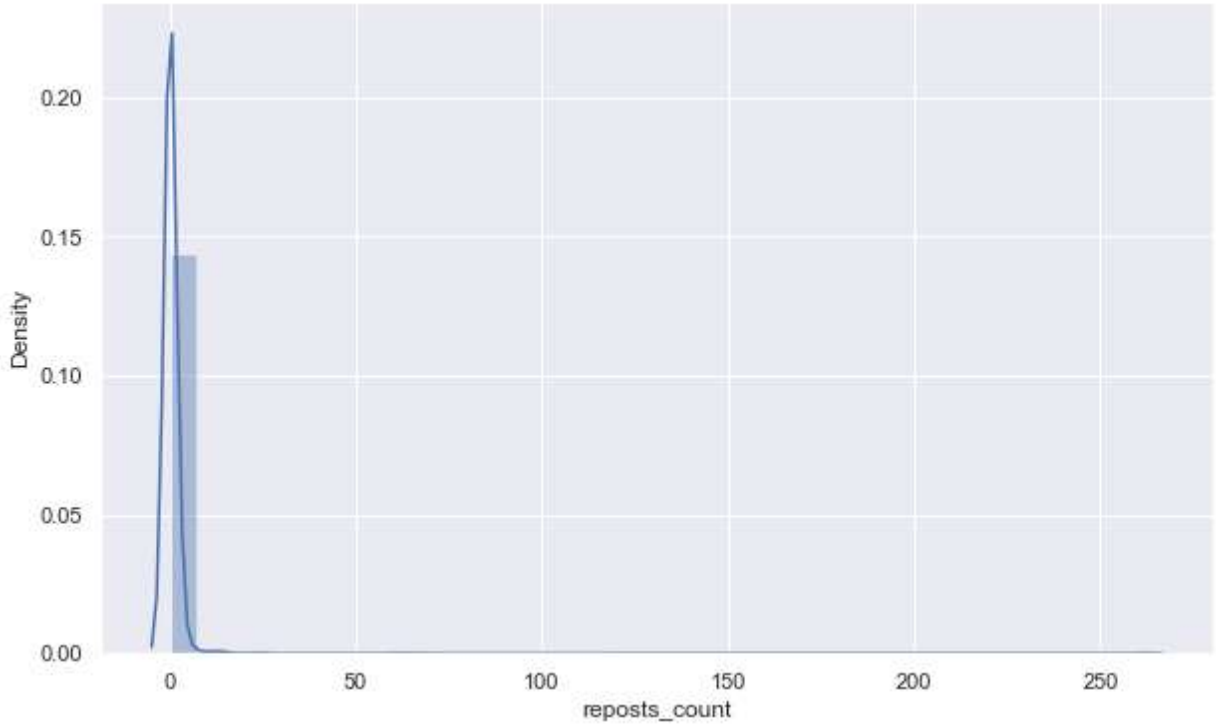
2.4. Khám phá và phân tích dữ liệu

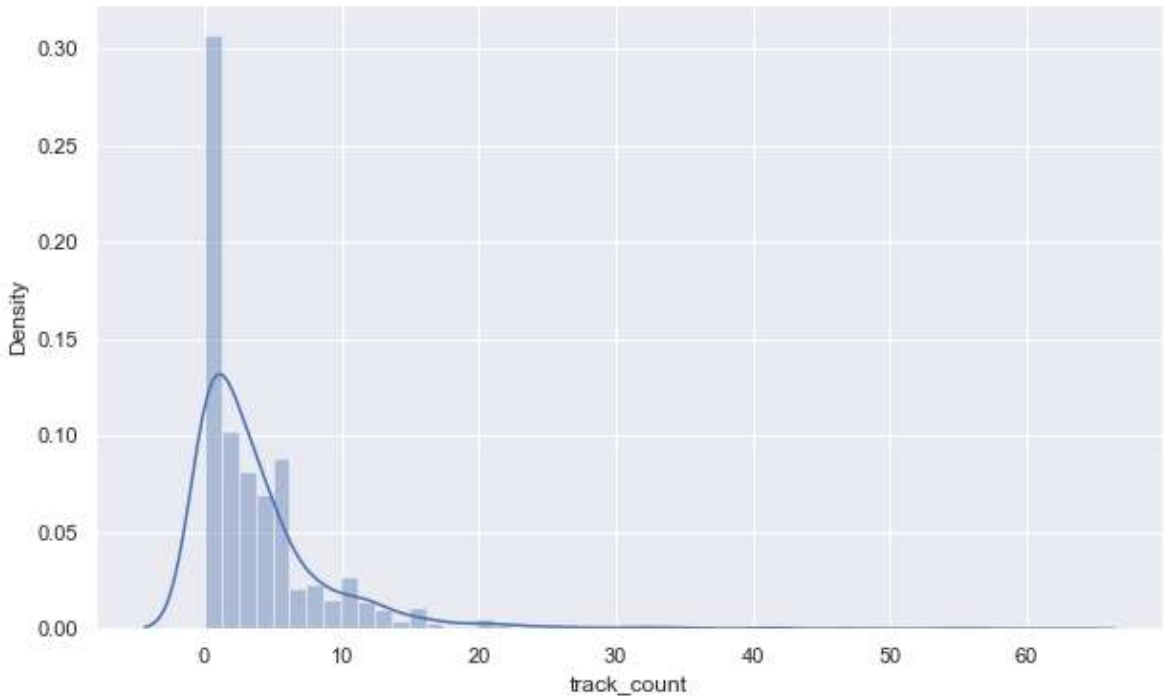
- Mối quan hệ đơn biến:
  - Để thể hiện mối quan hệ đơn biến của các biến, tụi em sử dụng biểu đồ dạng `distplot` .
  - Ở mỗi biểu đồ, trục hoành thể hiện các giá trị của cột dữ liệu đang xét, trục tung thể hiện phần trăm của các giá trị có trong bộ dữ liệu.
- Kết quả của việc dự đoán sự phân bố của các biến chính: nhìn chung các sự phân bố này khá gần với mong đợi được nêu ở trên, tuy nhiên, khoảng các giá trị thấp hơn trung bình nhiều hơn tụi em đã mong đợi.

In [19]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)
sb.set(rc={"figure.figsize": (10, 6)})
cols=df.select_dtypes('number').columns
for col in cols:
    sb.distplot(df[col])
    plt.show()
#sb.distplot(df['likes_count'])
```




0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
0
200
400
600
800
1000
1200
1400
1600
likes_count


0.20
0.15
0.10
0.05
0.00
Density
0
50
100
150
200
250
reposts_count



- Các tương quan đa biến:
  - Ở đây, tụi em sử dụng `correlation matrix` để tìm ra sự tương quan giữa các biến chính, sau đó sử dụng biểu đồ dạng `heatmap` để mô hình hoá ma trận vừa tìm được.

In [20]:
```
corr_matrix=df.corr()
corr_matrix
```

Out[20]:

	duration	likes_count	reposts_count	is_album	track_count
duration	1.000000	0.090257	0.078174	-0.100378	0.439010
likes_count	0.090257	1.000000	0.994259	0.046156	0.121882
reposts_count	0.078174	0.994259	1.000000	0.045895	0.101707
is_album	-0.100378	0.046156	0.045895	1.000000	0.046134
track_count	0.439010	0.121882	0.101707	0.046134	1.000000

In [21]:
```
plt.figure(figsize=[20,10])
sb.heatmap(df.corr(),annot=True)
```

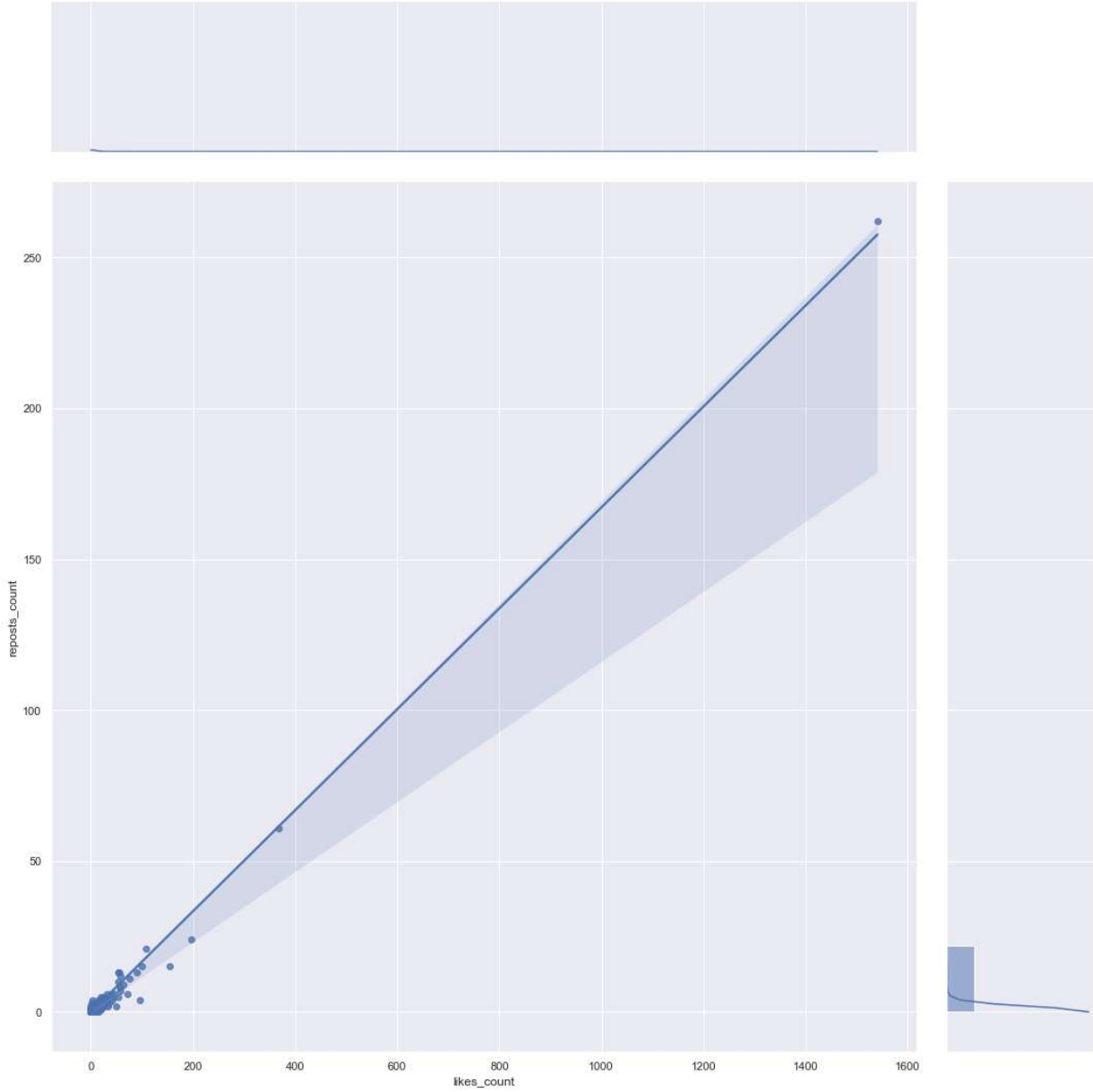
Out[21]: <AxesSubplot:>

- Dựa vào ma trận quan hệ trên, ta có thể rút ra kết luận cho các câu hỏi đã đặt ra bên trên như sau:
  - `reposts_count - likes_count` : có mối quan hệ tương quan dương và hệ số tương quan lớn (0.99), gần như đạt được đồng biến tuyệt đối, có nghĩa rằng `reposts_count - likes_count` quan hệ đồng biến với nhau.
  - `track_count - duration` : có mối quan hệ tương quan dương và hệ số tương quan tương đối lớn (0.44), có nghĩa rằng `track_count - duration` quan hệ đồng biến với nhau nhưng ở mức độ trung bình - khá.

2.5. Trả lời câu hỏi:

Câu hỏi 1: Lượt like và repost của một playlist có liên quan tới nhau?

In [22]:
```
data = df[["likes_count", "reposts_count"]]
sb.jointplot(data=data, x="likes_count", y="reposts_count",kind = "reg", height =
plt.show()
```



- Nhìn vào biểu đồ trên, ta có thể hình dung được rằng `likes_count` và `reposts_count` có quan hệ với nhau, nhưng do giá trị của các đặc trưng có trong bộ dữ liệu đang xét nhỏ, nên ta thấy rằng các điểm mô tả tập trung gần về giá trị gốc $(0, 0)$. Ở đây, tụi em chọn thêm dạng phát hoạ là "reg" tương ứng với regression (hồi quy) để thể hiện rõ hơn tương quan giữa cặp biến này. Theo kết quả trực quan hoá nhận được, ta có thể kết luận rằng `likes_count` và `reposts_count` tỉ lệ thuận với nhau.

Câu hỏi 2: Tổng số playlists qua từng thời kì (tháng, năm)?

In [23]:
```
#Tien xu ly du lieu
month_year=pd.to_datetime(df['display_date'])
month_year = month_year.apply(lambda x: x.strftime('%Y-%B'))
month_year
```

Out[23]:
```
0          2008-April
1          2008-April
2            2008-May
3            2008-May
4            2008-May
            ...
1503    2009-February
1504    2009-February
1505    2009-February
1506    2009-February
1507    2009-February
Name: display_date, Length: 1508, dtype: object
```

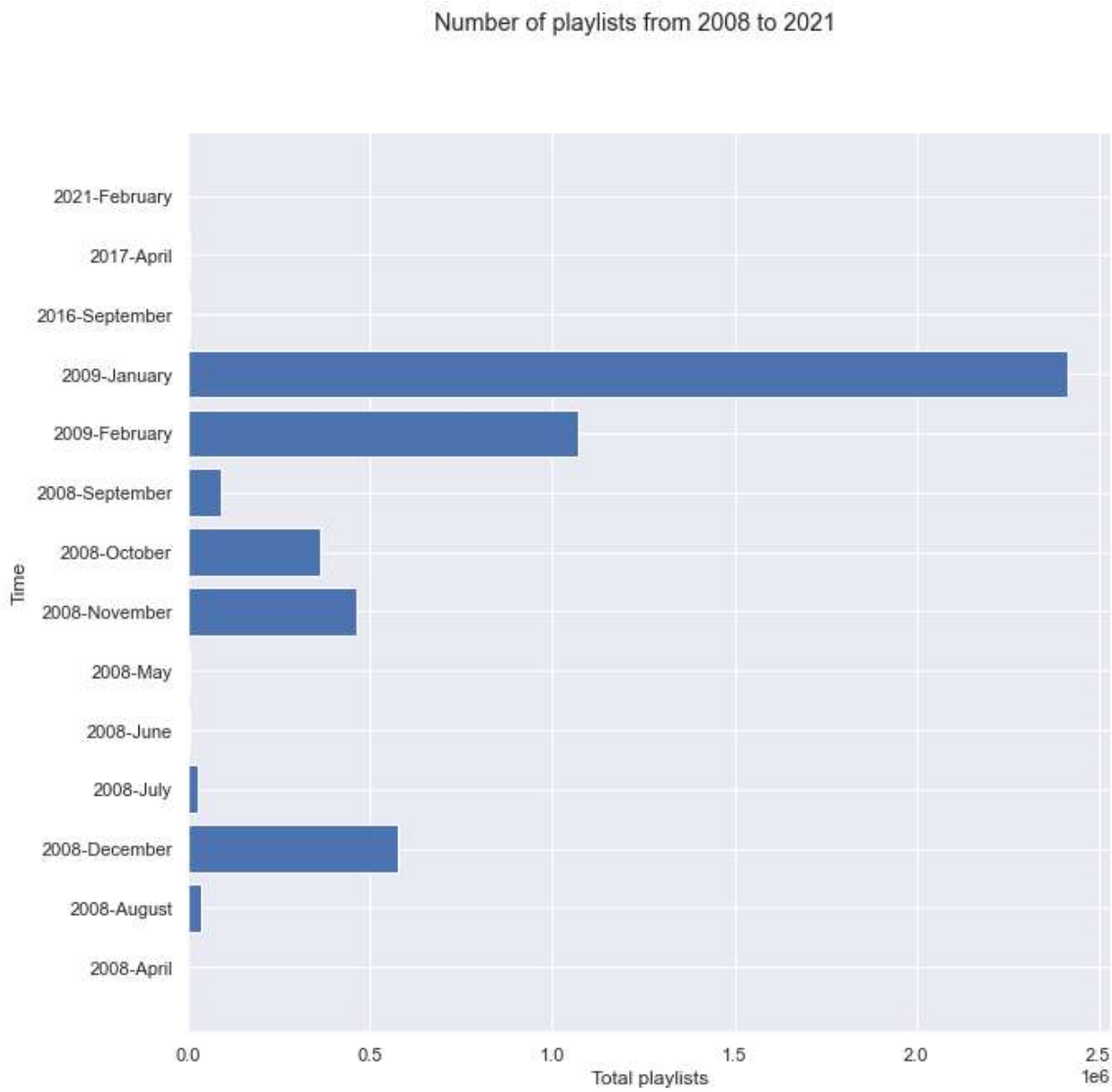
In [24]:
```
#Khai thac du lieu
survey_year_playlist = pd.DataFrame(np.array([np.array(df['id']),month_year]).T,c

survey_year_playlist = survey_year_playlist.groupby(['month_year']).sum()
survey_year_playlist
```

Out[24]:

month_year	count_playlists
2008-April	19
2008-August	36758
2008-December	575850
2008-July	25188
2008-June	2066
2008-May	1345
2008-November	460292
2008-October	362021
2008-September	89965
2009-February	1070157
2009-January	2413927
2016-September	3223
2017-April	3527
2021-February	345

In [25]:
```
#Ve bieu do
plt.figure(figsize=(10,10))
x=survey_year_playlist.index.get_level_values('month_year')
y=survey_year_playlist['count_playlists']
plt.barh(x,y)
plt.ylabel("Time")
plt.xlabel("Total playlists")
plt.suptitle('Number of playlists from 2008 to 2021')
plt.show()
```



- Trong bộ dữ liệu này, tổng số playlist được tạo tăng theo thời gian và đạt nhiều nhất vào tháng 1-2009. Tuy nhiên vẫn có khoảng thời gian ở giữa giai đoạn này mà số lượng playlist

được tạo thấp hơn những tháng trước đó. Đặc biệt sau tháng 1-2009, số lượng playlist rất ít và có xu hướng giảm dần. Do đó có thể thấy số lượng playlist không hoàn toàn tỉ lệ thuận với thời gian.

Câu hỏi 3: Số track càng nhiều thì thời lượng playlist càng lâu?

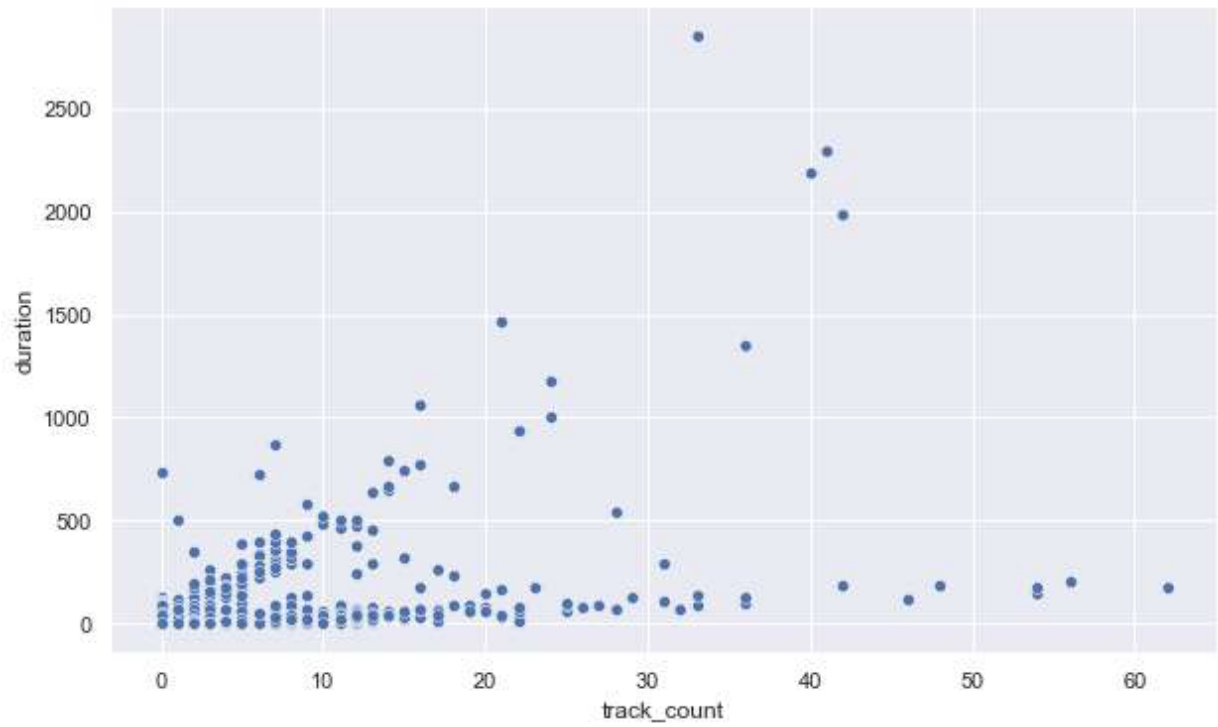
Tiền xử lý dữ liệu:

- Thực hiện chuyển giá trị của cột `duration` từ millisecond thành second.

In [26]:
```
import copy
df_2 = copy.deepcopy(df)
df_2["duration"] = df_2["duration"]/100000
```

Vẽ biểu đồ

In [27]:
```
timing_df = df_2[["track_count", "duration"]]
sb.scatterplot(data=timing_df, x="track_count", y="duration")
plt.show()
```



Dựa vào biểu đồ, ta thấy phần lớn các điểm mô tả tập trung gần về toạ độ gốc, chứng tỏ rằng các playlist có số lượng track nhỏ thì thời gian cũng nhỏ. Tuy nhiên, biểu đồ vẫn có các điểm mà tại đó thời lượng rất lớn nhưng số track lại nằm ở mức trung bình. Do đó, ta thấy rằng `track_count` và `duration` có tỉ lệ thuận với nhau, nhưng vẫn còn tồn tại các ngoại lệ.

Phần 3: Phân tích dữ liệu tracks

Thông tin cơ bản về dữ liệu

- **id**: Mã định danh của track
- **artwork_url**: Đường dẫn đến avatar của track
- **caption**: Tiêu đề của track
- **comment_count**: Số lượng bình luận
- **download_count**: Số lượng download của track
- **last_modified**: Lần chỉnh sửa cuối cùng về thông tin của track
- **permalink_url**: Đường dẫn trực tiếp đến track trên soundcloud
- **duration**: Thời lượng của track
- **title**: Tên track trên soundcloud
- **genre**: Thể loại của track
- **likes_count**: Số lượng likes cho track

Nội dung thực hiện

1. Tiền xử lí dữ liệu

- Thay thế "None" bằng Nan
- Xóa các bộ dữ liệu có tỉ lệ missing lớn hơn 25%

2. Câu hỏi

- Tương quan giữa số lượng lượt likes và comment trong một bài nhạc
- Dòng nhạc nào đang được ưa chuộng qua các năm

In [28]:
```
df=pd.read_csv("track.csv")
df.head()
```

Out[28]:

	id	artwork_url	caption	commentable	comment_count	created_at	d
0	43	https://i1.sndcdn.com/artworks-000000009541-24...	None	True	43	2007-08-03T20:34:54Z	Y o a
1	274	https://i1.sndcdn.com/artworks-000260720873-yk...	None	True	53	2007-09-20T12:23:26Z	a
2	53787294	None	None	True	81	2012-07-23T07:45:52Z	s in co
3	290	https://i1.sndcdn.com/artworks-000067273298-mj...	None	True	104	2007-09-22T14:24:38Z	
4	291	https://i1.sndcdn.com/artworks-000067273301-we...	None	True	176	2007-09-22T14:32:59Z	

5 rows × 33 columns

3.1 Tiền xử lý dữ liệu:

Chuyển các dữ liệu "None" về Nan

In [29]:
```
def isNaN(num):
    return num!=num
#convert none into nan
df=df.replace("None",np.nan)
```

Xóa các dòng dữ liệu có dữ liệu bị thiếu trên 25%

In [30]:
```
def delMissingRow(Data, limit):
    data_temp=copy.deepcopy(Data)
    percent=0
    count=0
    end = len(data_temp['id'])
    i=0
    missingList=[]
    while i<end:
        count=0
        for key in data_temp:
            if isNaN(data_temp[key][i]):
                count+=1
        if(count/len(data_temp.columns))>limit/100:
            missingList.append(i)
        i+=1
    data_temp=data_temp.drop(labels=missingList,axis=0)
    return data_temp
df=delMissingRow(df,25)
df.head()
```

Out[30]:

	id	artwork_url	caption	commentable	comment_count	created_at	des
0	43	https://i1.sndcdn.com/artworks-000000009541-24...	NaN	True	43	2007-08-03T20:34:54Z	Yet old a l
1	274	https://i1.sndcdn.com/artworks-000260720873-yk...	NaN	True	53	2007-09-20T12:23:26Z	aro liv n Ir
3	290	https://i1.sndcdn.com/artworks-000067273298-mj...	NaN	True	104	2007-09-22T14:24:38Z	S c
4	291	https://i1.sndcdn.com/artworks-	NaN	True	176	2007-09-	F S

3.2. Trả lời câu hỏi:

3.2.1. Tương quan giữa lượt like và comment

- Thực hiện: phân tích tương quan của số lượt like với trung bình số comment
- Ý tưởng thực hiện:
  - Tạo list số lượt like có tồn tại trong dữ liệu, sắp xếp theo thứ tự tăng dần
  - Tính trung bình số lượt comment của số lượt like đó. Ví dụ: với các track có số lượt like là 10, sẽ tính trung bình số lượt comment của các track đó. Lưu tất cả về 1 list
  - Vẽ biểu đồ line để xem xu hướng phát triển của dữ liệu
  - Thực hiện khám phá và vẽ thêm các biểu đồ khác để đánh giá tốt hơn dữ liệu
- Ý nghĩa: xem mỗi lượt like là lượt người thực sự yêu thích, lượt comment là người thực sự quan tâm, muốn chia sẻ về track đó.

- Đánh giá được mức độ quan tâm của người dùng khi nghe nhạc trên Soundcloud.
- Trang web có thực sự được nhiều lượt quan tâm không hay chỉ là quan tâm ảo.
- Khi lượt like tăng thì lượt comment có thực sự tăng theo hay không?

In [31]:
```
# tạo một dataframe tạm để lưu dữ liệu likes_count và comment_count
df_temp = pd.DataFrame(df[['likes_count','comment_count']])
df_temp
```

Out[31]:

	likes_count	comment_count
0	58	43
1	119	53
3	331	104
4	1065	176
5	594	68
...	...	...
2717	883	208
2724	38	17
2726	29	2
2730	2	5
2733	53	11

989 rows × 2 columns

In [32]:
```
# tạo list số lượng lượt like likes_count_list
# sắp xếp list theo thứ tự tăng dần để dễ cho quá trình trực quan phía sau
likes_count_list = df_temp['likes_count'].unique().tolist()
likes_count_list.remove(np.nan)
int_map = map(int, likes_count_list)
likes_count_list = list(int_map)
likes_count_list.sort()
likes_count_list
```

Out[32]:
```
[1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
```

In [33]:
```
# tạo list số lượng lượt comment comment_count_list
# vì tính tương ứng theo số lượt like, nên thứ tự list này tương ứng với thứ tự l
comment_count_list = []
for i in range(len(likes_count_list)):
    comment_count_temp = df_temp.loc[df_temp['likes_count'] == str(likes_count_li
    temp = 0
    for j in comment_count_temp:
        if isNaN(j) ==False:
            temp += int(j)
    temp = round((temp/len(comment_count_temp)),2)
    comment_count_list.append(temp)
comment_count_list
```

Out[33]:
```
[1.22,
 1.75,
 1.84,
 1.92,
 2.09,
 3.27,
 3.31,
 7.52,
 2.79,
 3.6,
 4.12,
 4.18,
 1.92,
 4.07,
 3.0,
 7.5,
 5.56,
 4.58,
 6.17,
```

In [34]:
```
# tạo dataframe để dùng cho quá trình trực quan, vẽ biểu đồ
df_final = pd.DataFrame({'likes count' : likes_count_list,'comment count' : comme
df_final
```

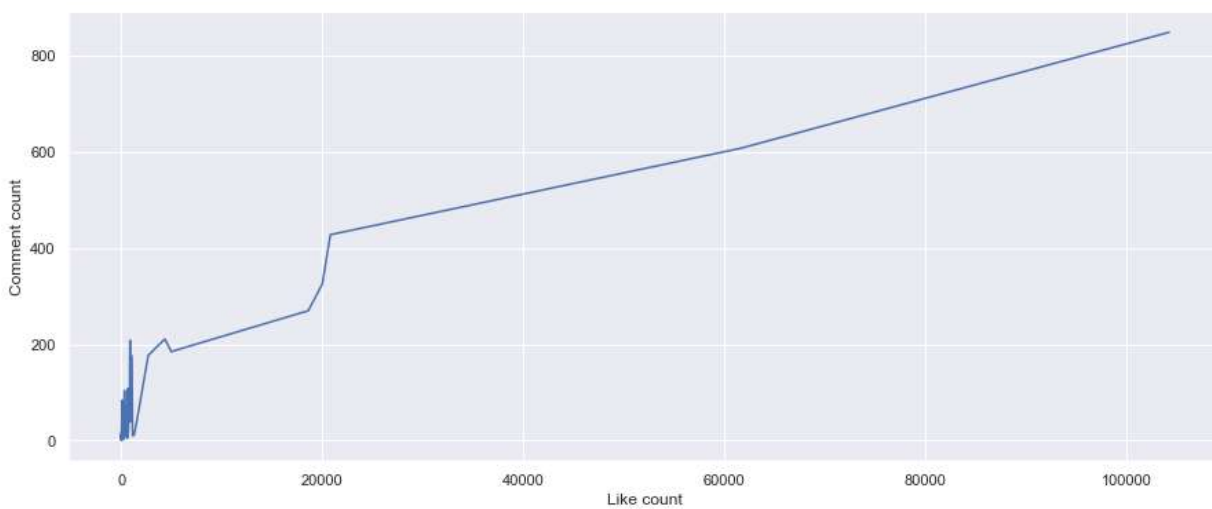
Out[34]:

	likes count	comment count
0	1	1.22
1	2	1.75
2	3	1.84
3	4	1.92
4	5	2.09
...	...	...
184	18612	270.00
185	20017	326.00
186	20801	428.00
187	61672	608.00
188	104229	849.00

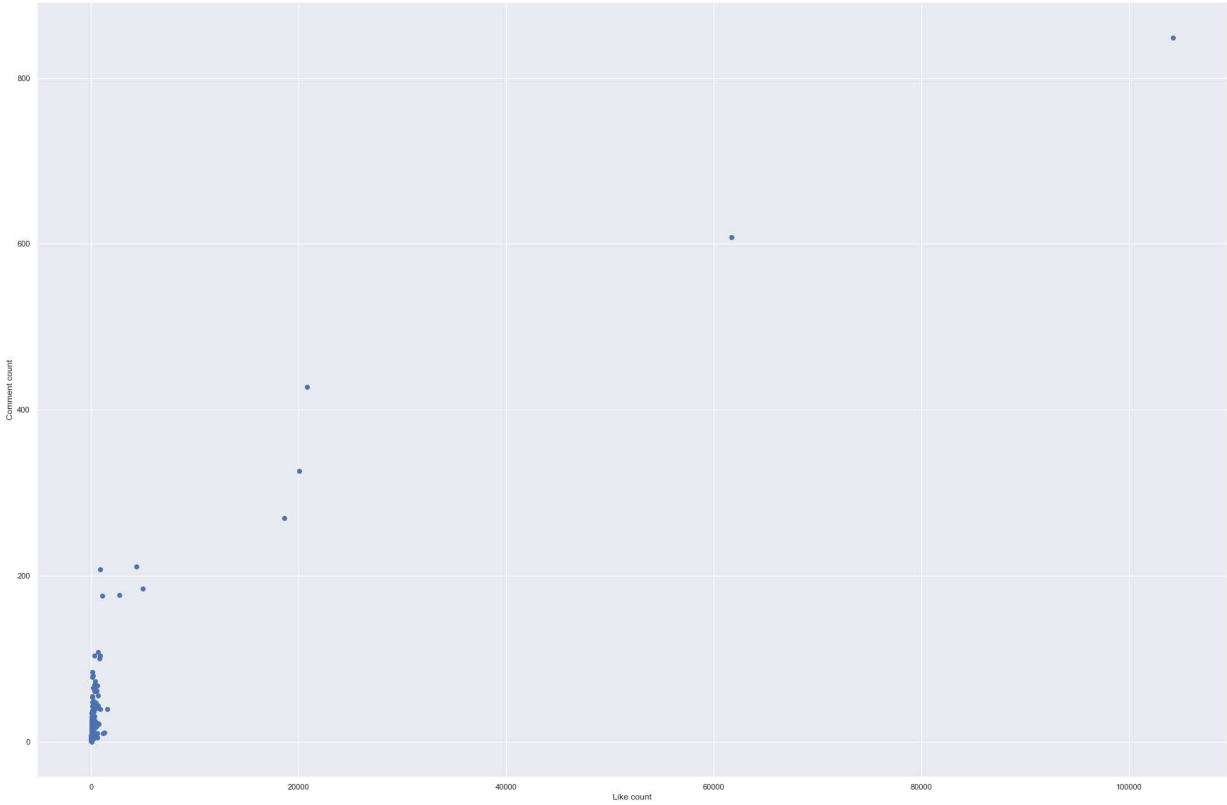
189 rows × 2 columns

Trực quan hóa dữ liệu

In [35]:
```
# vẽ một biểu đồ đường để nhìn tổng quan dữ liệu
plt.figure(figsize=(15,6))
plt.xlabel("Like count")
plt.ylabel("Comment count")
plt.plot(df_final['likes count'], df_final['comment count'])
plt.show()
```



In [36]:
```
# vẽ biểu đồ phân tán của dữ liệu
plt.figure(figsize=(30,20))
plt.xlabel("Like count")
plt.ylabel("Comment count")
plt.scatter(df_final['likes count'], df_final['comment count'])
plt.show()
```

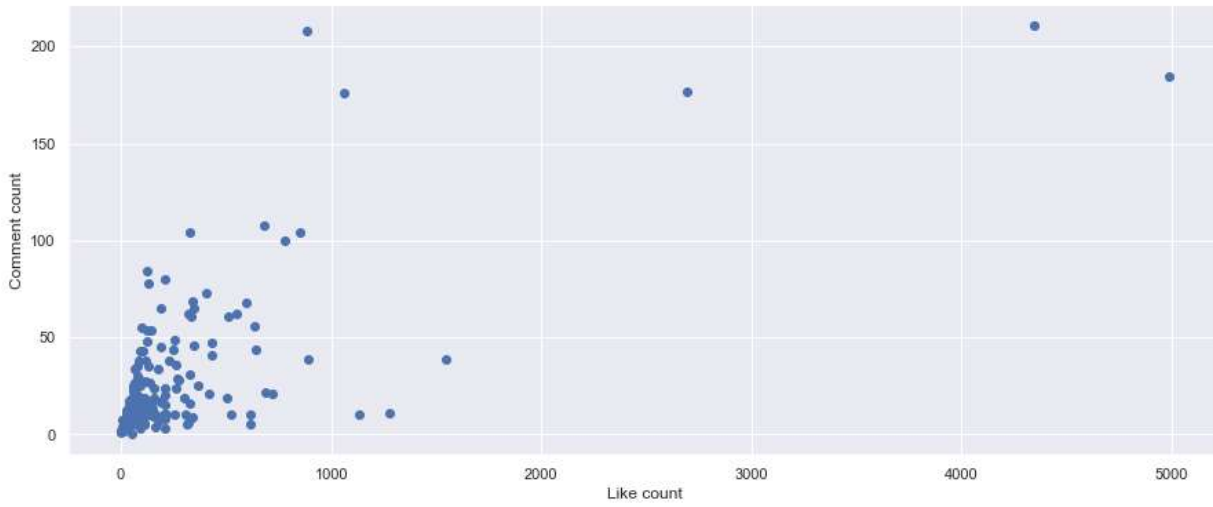


Dễ nhận thấy một số điểm sau ở biểu đồ trên

- Dữ liệu phân tán chủ yếu ở đoạn lượt like dưới 10000
- Có một vài dữ liệu nằm ngoài đoạn phân tán. Ví dụ lượt like là 104229 với số comment là 849 -> đây có thể là một track được nhiều người biết tới, được phổ biến và có rất nhiều người quan tâm.

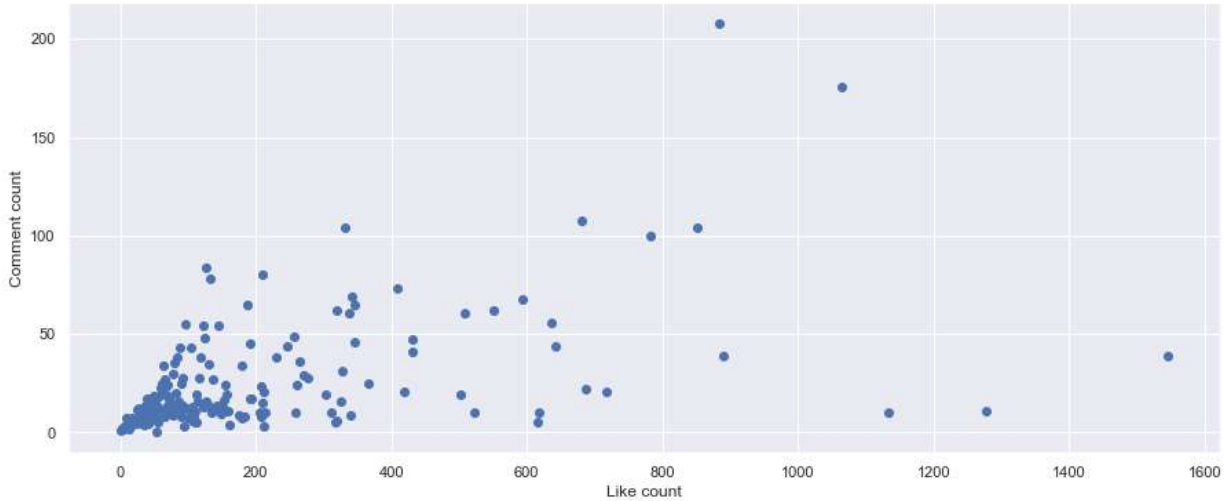
Một số dữ bộ dữ liệu ở cuối dataframe cũng tương tự, vì vậy sẽ loại bỏ các dữ liệu này để vẽ ở khu vực dữ liệu tập trung nhiều (lượt like dưới 10000)

In [37]:
```
# vẽ biểu đồ với dữ liệu like count nhỏ hơn 10000
plt.figure(figsize=(15,6))
plt.xlabel("Like count")
plt.ylabel("Comment count")
plt.scatter(df_final['likes count'].loc[df_final['likes count'] < 10000],
            df_final['comment count'].loc[df_final['likes count'] < 10000])
plt.show()
```



Với dữ liệu này, tiếp tục có một số dữ liệu lớn, nằm ngoài vùng tập trung chính, vì vậy, ta sẽ tiếp tục vẽ thêm 1 biểu đồ nữa, với dữ liệu lượt like dưới 2000

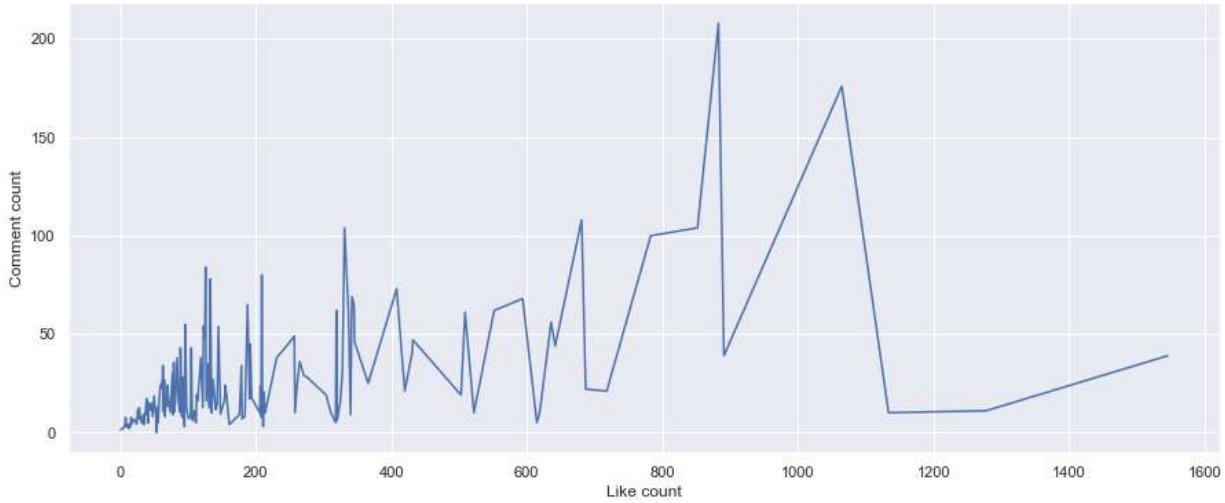
In [38]:
```
# vẽ biểu đồ với dữ liệu like count nhỏ hơn 2000
plt.figure(figsize=(15,6))
plt.xlabel("Like count")
plt.ylabel("Comment count")
plt.scatter(df_final['likes count'].loc[df_final['likes count'] < 2000],
            df_final['comment count'].loc[df_final['likes count'] < 2000])
plt.show()
```



Có thể thấy

- Dữ liệu tập trung ở khoảng dưới 400 là chính -> sự tương tác của người dùng với dữ liệu tương đối lớn, trung bình lượt comment cũng khá cao (xấp xỉ 50)
- Dữ liệu không có xu hướng tăng hay giảm cụ thể, mà có sự biến đổi không theo quy luật (sẽ thấy rõ hơn ở biểu đồ phía sau)

In [39]:
```
# vẽ biểu đồ đường của 2 dữ liệu
plt.figure(figsize=(15,6))
plt.xlabel("Like count")
plt.ylabel("Comment count")
plt.plot(df_final['likes count'].loc[df_final['likes count'] < 2000],
            df_final['comment count'].loc[df_final['likes count'] < 2000])
plt.show()
```



- Dữ liệu không đi theo xu hướng, mà có sự thay đổi lên xuống thất thường
- Ở số lượt like dưới 400, lượt comment trung bình trong khoảng 0 - 100
- Với các lượt like lớn hơn, nhìn chung các biến đổi khá thất thường -> có một vài track có lượt quan tâm rất lớn

Kết luận

- Người dung Soundcloud có nhiều tương tác với các track có trên hệ thống
- Không có một xu hướng cố định về lượt comment trung bình -> hành vi của người dùng khi nghe nhạc không cố định, phụ thuộc khá nhiều vào nội dung track, người đăng là ai,...

3.2.2. Dòng nhạc được thịnh hành qua các năm

- Thống kê các thể loại âm nhạc qua các năm, từ đó phần nào biết được xu hướng âm nhạc hiện Việc thống kê dựa trên việc phân tích dữ liệu của hai cột là `genre` và `release_date`

In [40]:
```
list_years=[]
list_genre=[]
df_years = df['release_date'].tolist()
df_genre=df['genre'].tolist()

for i in range(0,len(df['id'])):
    if isNaN(df_genre[i])==False and isNaN(df_years[i])==False:
        temp=df_genre[i].split('/')
        for j in temp:
            j=j.lower()
            list_genre.append(j)
            year=df_years[i][:4]
            list_years.append(year)
```

In [41]:
```
df1 = pd.DataFrame(list(zip(list_years, list_genre)), columns =['Year', 'Genre'])
res= df1.groupby(["Year", "Genre"]).size().reset_index(name="Time")
res['Year']
```

Out[41]:
```
0      1993
1      1995
2      1997
3      1997
4      1998
       ...
273    2019
274    2019
275    2020
276    2021
277    2021
Name: Year, Length: 278, dtype: object
```

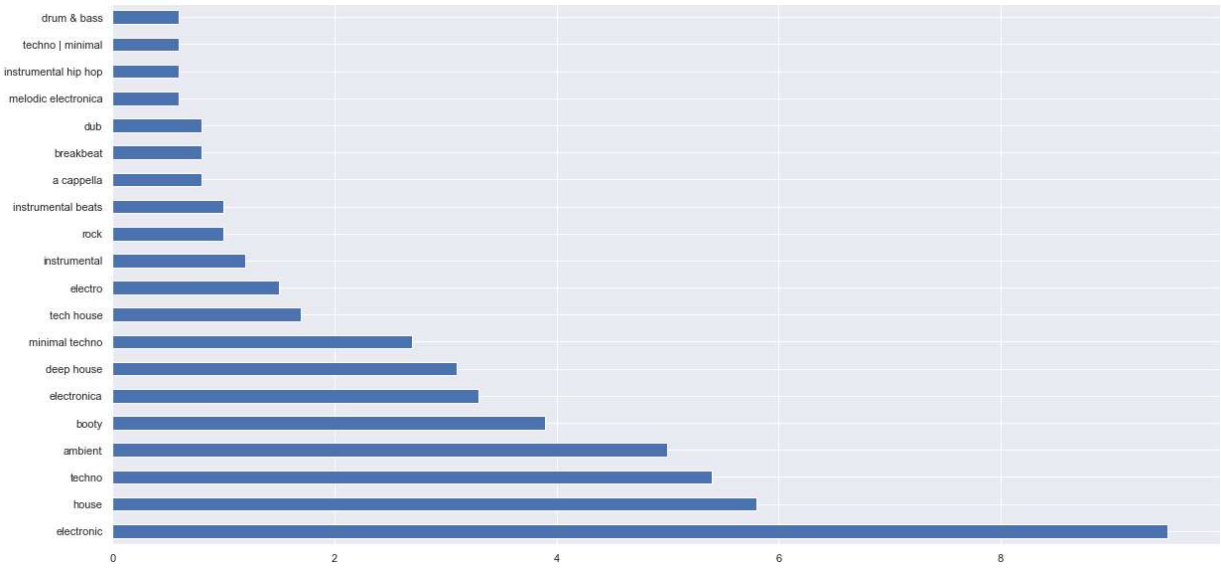
In [42]:
```
t=df1['Genre'].value_counts(normalize=True)*100
t=t.round(1)
t.sort_values(ascending=False)
t=t[:20]
```

Tương quan một biến

Mức độ phân bố của các dòng nhạc qua các năm

In [45]:
```
t.plot(figsize=(20,10),kind='barh')
```

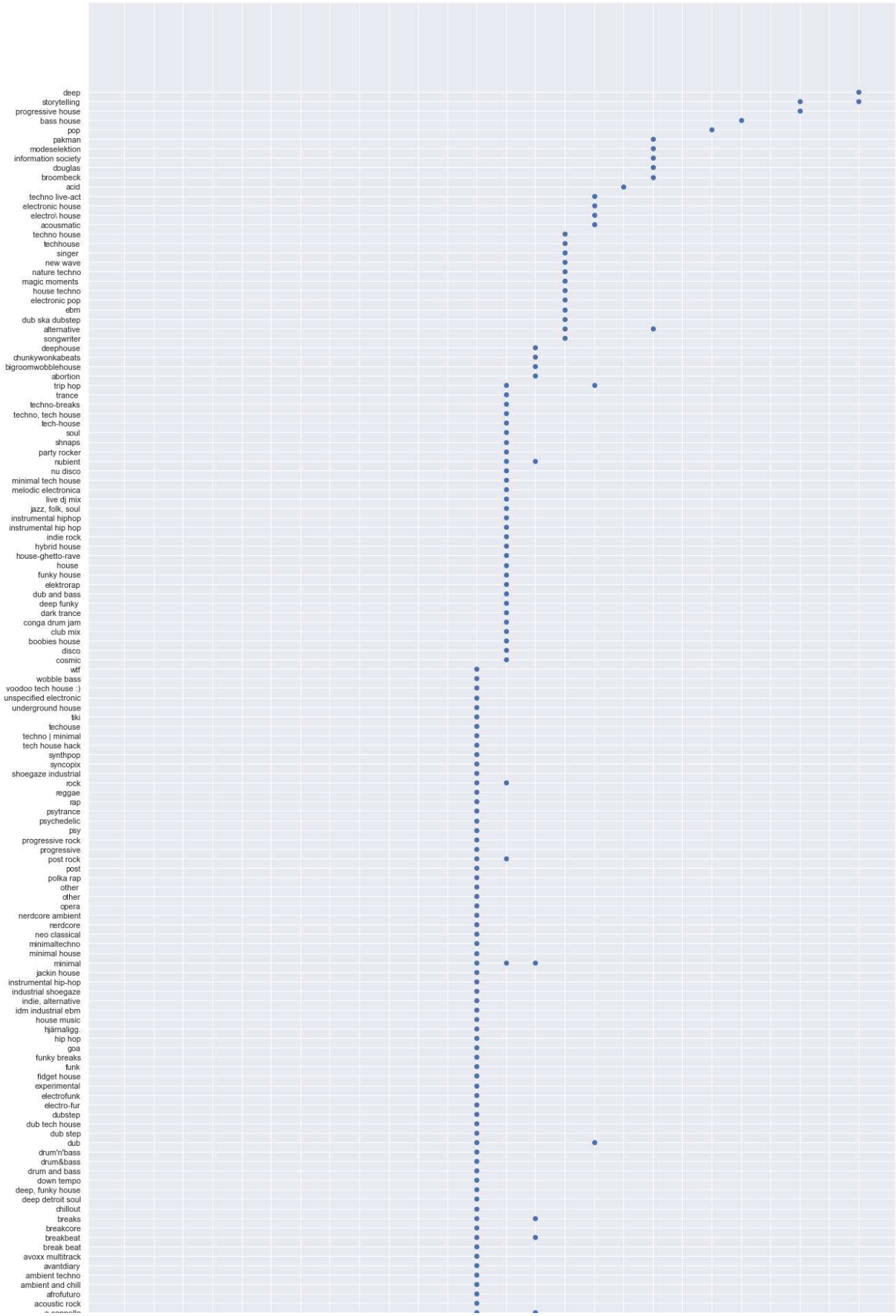
Out[45]: <AxesSubplot:>

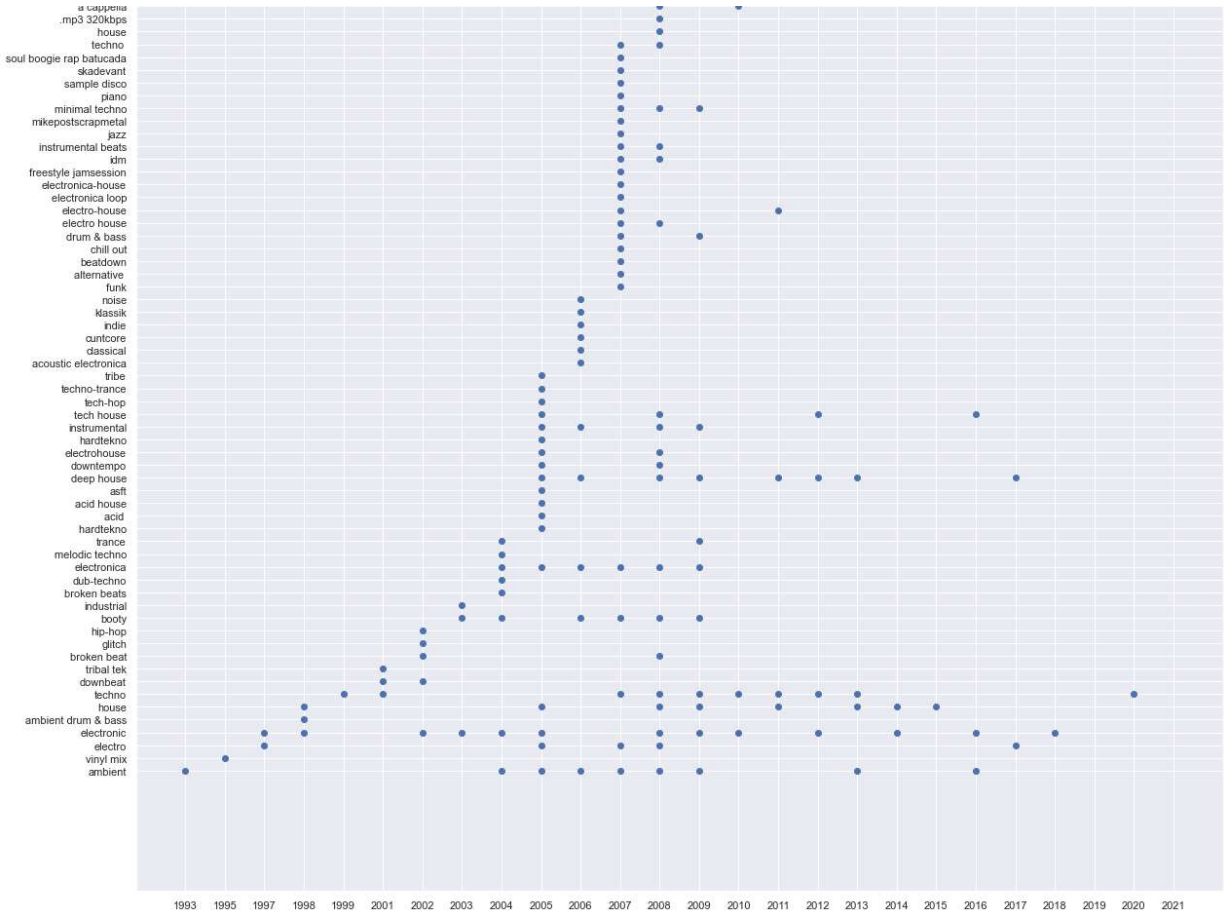


Phân bố dòng nhạc thịnh hành qua các năm

In [44]:
```
plt.figure(figsize=(20,50))
plt.scatter(res['Year'],res['Genre'])
```

Out[44]: <matplotlib.collections.PathCollection at 0x13518fafee0>





Kết luận:

Dòng nhạc qua các năm rất phong phú và không ngừng phát triển, trong dó có thể thấy các dòng nhạc điện tử ngày càng được ưa chuộng.