In [27]:
```
import pandas as pd
file_df = pd.read_csv("AS.csv")
```

In [28]:
```
import pandas as pd
apple_store_df = pd.read_csv("AS.csv")
```

In [29]:
```
apple_store_df
```

Out[29]:

	Unnamed: 0	id	track_name	size_bytes	currency	price	rating_count_tot	rating_cou
0	1	281656475	PAC-MAN Premium	100788224	USD	3.99	21292	
1	2	281796108	Evernote - stay organized	158578688	USD	0.00	161065	
2	3	281940292	WeatherBug - Local Weather, Radar, Maps, Alerts	100524032	USD	0.00	188583	
3	4	282614216	eBay: Best App to Buy, Sell, Save! Online Shop...	128512000	USD	0.00	262241	
4	5	282935706	Bible	92774400	USD	0.00	985920	
...	...	...	...	...	...	...	...	
7192	11081	1187617475	Kubik	126644224	USD	0.00	142	
7193	11082	1187682390	VR Roller-Coaster	120760320	USD	0.00	30	
7194	11087	1187779532	Bret Michaels Emojis + Lyric Keyboard	111322112	USD	1.99	15	
7195	11089	1187838770	VR Roller Coaster World - Virtual Reality	97235968	USD	0.00	85	
7196	11097	1188375727	Escape the Sweet Shop Series	90898432	USD	0.00	3	

7197 rows × 17 columns

```
In [30]: apple_store_df.columns
```

```
Out[30]: Index(['Unnamed: 0', 'id', 'track_name', 'size_bytes', 'currency', 'price',
                'rating_count_tot', 'rating_count_ver', 'user_rating',
                'user_rating_ver', 'ver', 'cont_rating', 'prime_genre',
                'sup_devices.num', 'ipadSc_urls.num', 'lang.num', 'vpp_lic'],
               dtype='object')
```

```
In [31]: needed_data = ["price", "currency", "rating_count_tot", "user_rating"]
```

```
In [32]: selected_data = apple_store_df[needed_data]
```

```
In [33]: selected_data
```

Out[33]:

	price	currency	rating_count_tot	user_rating
0	3.99	USD	21292	4.0
1	0.00	USD	161065	4.0
2	0.00	USD	188583	3.5
3	0.00	USD	262241	4.0
4	0.00	USD	985920	4.5
...	...	...	...	...
7192	0.00	USD	142	4.5
7193	0.00	USD	30	4.5
7194	1.99	USD	15	4.5
7195	0.00	USD	85	4.5
7196	0.00	USD	3	5.0

7197 rows × 4 columns

In [34]:
```
selected_data
```

Out[34]:

	price	currency	rating_count_tot	user_rating
0	3.99	USD	21292	4.0
1	0.00	USD	161065	4.0
2	0.00	USD	188583	3.5
3	0.00	USD	262241	4.0
4	0.00	USD	985920	4.5
...	...	...	...	...
7192	0.00	USD	142	4.5
7193	0.00	USD	30	4.5
7194	1.99	USD	15	4.5
7195	0.00	USD	85	4.5
7196	0.00	USD	3	5.0

7197 rows × 4 columns

In [35]:
```
row_1 = ["PAC-MAN Premium", "3.99", "USD", "21292", "4.0"]
row_2 = ["Evernote - stay organized", "0.00", "USD", "161065", "4.0"]
row_3 = ["WeatherBug - Local Weather, Radar, Maps, Alerts", "0.00", "USD", "18858
the_rows = [row_1, row_2, row_3]
rating_count_sum = float(row_1[3]) + float(row_2[3])
average = rating_count_sum / 2
print(round(average))
```

```
91178
```

In [36]:
```
ratings = [row_1[4], row_2[4], row_3[4]]
```

In [37]:
```
ratings
```

Out[37]:
```
['4.0', '4.0', '4.0']
```

In [38]:
```
sumation = 0
for row in the_rows:
    rating = float(row[3])
    sumation += rating
print(sumation / 3)
```

```
123646.66666666667
```

In [39]:
```
print(row_1)
```

```
['PAC-MAN Premium', '3.99', 'USD', '21292', '4.0']
```

In [40]:

```
from csv import reader
opened_files = open("AS.csv", encoding="UTF-8")
read_file = reader(opened_files)
list_file = list(read_file)
```

In [41]:

```
print(list_file[1][5])
```

```
3.99
```

In [42]:

```
list_file[0][5]
```

Out[42]:

```
'price'
```

In [43]:

```
results = []
sumate = 0
for row in list_file[1:]:
    track_name = row[2]
    price = float(row[5])
    user_rating = float(row[8])
    prime_genre = row[12]
    if prime_genre == "Games" and price == 0:
        results.append(user_rating)
        average = sum(results) / len(results)
print(average)
```

```
3.5285777580859548
```

In [44]:

```
list_file[12]
```

Out[44]:

```
['12',
 '284815117',
 'SCRABBLE Premium',
 '227547136',
 'USD',
 '7.99',
 '105776',
 '166',
 '3.5',
 '2.5',
 '5.19.0',
 '4+',
 'Games',
 '37',
 '0',
 '6',
 '1']
```

In [45]:

```
length = len(list_file)
list_file[0][12]
```

Out[45]:

```
'prime_genre'
```

In [46]:

```
suming = 0
for row in list_file[1:]:
    user_rating = float(row[8])
    suming += user_rating
    average# = suming / length
print(suming)
```

```
25383.5
```

In [47]:

```
results = []
#sumings = 0
for row in list_file[1:]:
    prime_genre = (row[12])
    price = float(row[5])
    if price > 0.0:
        results.append(track_name)
        #sumings += user_rating
print(results[:6])
```

```
['Escape the Sweet Shop Series', 'Escape the Sweet Shop Series', 'Escape the Sw
eet Shop Series', 'Escape the Sweet Shop Series', 'Escape the Sweet Shop Serie
s', 'Escape the Sweet Shop Series']
```

In [48]:

```
results = []
for row in list_file[1:]:
    user_rating = float(row[8])
    prime_genre = (row[12])
    price = float(row[5])
    if prime_genre == "Games" and price > 0:
        results.append(user_rating)
    average = (sum(results)) / len(results)
print(average)
```

```
3.9049844236760123
```

In [49]:

```
results = []
sumate = 0
for row in list_file[1:]:
    track_name = row[2]
    price = float(row[5])
    user_rating = float(row[8])
    prime_genre = row[12]
    if prime_genre != "Games" and price > 0:
        results.append(user_rating)
        average = sum(results) / len(results)
print(average)
```

```
3.5286458333333335
```

In [50]:

```
this = list_file[0]
```

In [51]:

```
float(this)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3596\1049700043.py in <module>
----> 1 float(this)

TypeError: float() argument must be a string or a number, not 'list'
```

In [52]:

```
ok = list_file
for row in list_file[1:]:
    track_name = row[2]
    price = float(row[5])
    user_rating = float(row[8])
    prime_genre = row[12]
    if price == 0.0:
        row.append("free")
    else:
        row.append("not free")
print(list_file[:2])
```

```
[['', 'id', 'track_name', 'size_bytes', 'currency', 'price', 'rating_count_to
t', 'rating_count_ver', 'user_rating', 'user_rating_ver', 'ver', 'cont_rating',
'prime_genre', 'sup_devices.num', 'ipadSc_urls.num', 'lang.num', 'vpp_lic'],
['1', '281656475', 'PAC-MAN Premium', '100788224', 'USD', '3.99', '21292', '2
6', '4', '4.5', '6.3.5', '4+', 'Games', '38', '5', '10', '1', 'not free']]
```

In [53]:

```
apps_data = [['Call of Duty: Zombies', 5.0], ['Facebook', 0.0], ['Instagram', 0.0

for app in apps_data:
    price = app[1]

    if price == 0.0:
        app.append('free')
    else:
        app.append('not free')

print(apps_data[:2])
```

```
[['Call of Duty: Zombies', 5.0, 'not free'], ['Facebook', 0.0, 'free']]
```

In [54]:

```
apps_data[0][1]
```

Out[54]:

```
5.0
```

In [55]:
```
list_file[1]
```

Out[55]:
```
['1',
 '281656475',
 'PAC-MAN Premium',
 '100788224',
 'USD',
 '3.99',
 '21292',
 '26',
 '4',
 '4.5',
 '6.3.5',
 '4+',
 'Games',
 '38',
 '5',
 '10',
 '1',
 'not free']
```

In [56]:
```
name = [['August 19, 1971', 'Santa Martha Acatitla', 'Mexico', 'Yes', 'Joel Davic
['Augus 19, 1971', 'Sana Martha Acatitla', 'Mexico', 'Ys', 'Joel Davd Kaplan Carl
```

In [57]:
```
index = 0
for names in name:
    print(names)
```

```
['August 19, 1971', 'Santa Martha Acatitla', 'Mexico', 'Yes', 'Joel David Kapla
n Carlos Antonio Contreras Castro']
['Augus 19, 1971', 'Sana Martha Acatitla', 'Mexico', 'Ys', 'Joel Davd Kaplan Ca
rlos Antonio Contreras Castro']
```

In [58]:
```
name = [["rilwan", "lanre", "jibola"], ["ade", "aba", "adigun"], ["wale", "biola'
```

In [59]:
```
index = 0
for names in name:
    name[index] = names[:-1]
    index += 1
```

In [60]:
```
name
```

Out[60]:
```
[['rilwan', 'lanre'], ['ade', 'aba'], ['wale', 'biola']]
```

In [61]:
```
file_df.columns
```

Out[61]:
```
Index(['Unnamed: 0', 'id', 'track_name', 'size_bytes', 'currency', 'price',
       'rating_count_tot', 'rating_count_ver', 'user_rating',
       'user_rating_ver', 'ver', 'cont_rating', 'prime_genre',
       'sup_devices.num', 'ipadSc_urls.num', 'lang.num', 'vpp_lic'],
      dtype='object')
```

In [62]:
```
file_df["cont_rating"].unique()
```
Out[62]:
```
array(['4+', '12+', '17+', '9+'], dtype=object)
```

In [63]:
```
list_file[2][11]
```
Out[63]:
```
'4+'
```

In [64]:
```
result = []
for file in list_file[1:]:
    matured = file[11]
    if matured == "12+":
        result.append(matured)
print(len(result))
```
```
1155
```

In [65]:
```
content_ratings = ['4+', '9+', '12+', '17+']
numbers = [4433, 987, 1155, 622]
content_rating_numbers = [['4+', '9+', '12+', '17+'], [4433, 987, 1155, 622]]
```

In [66]:
```
content_ratings = {"4+": 4433, "9+": 987, "12+": 1155, "17+":622}
```

In [67]:
```
content_ratings
```
Out[67]:
```
{'4+': 4433, '9+': 987, '12+': 1155, '17+': 622}
```

In [68]:
```
content_ratings["12+"]
```
Out[68]:
```
1155
```

In [69]:
```
family = {}
family["father"] = "Ridwan"
family["mother"] = "mariam"
family["daughter"] = "Ameenah"
```

In [70]:
```
family
```
Out[70]:
```
{'father': 'Ridwan', 'mother': 'mariam', 'daughter': 'Ameenah'}
```

In [71]:
```
family["father"]
```
Out[71]:
```
'Ridwan'
```

In [72]:
```
list_file[0]
```

Out[72]:
```
['',
 'id',
 'track_name',
 'size_bytes',
 'currency',
 'price',
 'rating_count_tot',
 'rating_count_ver',
 'user_rating',
 'user_rating_ver',
 'ver',
 'cont_rating',
 'prime_genre',
 'sup_devices.num',
 'ipadSc_urls.num',
 'lang.num',
 'vpp_lic']
```

In [73]:
```
maturity = {}
result_12 = []
result_4 = []
result_9 = []
result_17 = []
for file in list_file[1:]:
    matured = file[11]
    if matured == "12+":
      result_12.append(matured)
    elif matured == "4+":
        result_4.append(matured)
    elif matured == "9+":
       result_9.append(matured)
    elif matured == "17+":
      result_17.append(matured)
maturity["12+"] = len(result_12)
maturity["4+"] = len(result_4)
maturity["9+"] = len(result_9)
maturity["17+"] = len(result_17)
```

In [74]:
```
maturity
```

Out[74]:
```
{'12+': 1155, '4+': 4433, '9+': 987, '17+': 622}
```

In [75]:
```
d_1 = {'key_1': 'first_value',
 'key_2': 2,
 'key_3': 3.14,
 'key_4': True,
 'key_5': [4,2,1],
 'key_6': {'inner_key' : 6}
 }
```

In [76]:
```
'key_4' in d_1
```

Out[76]:
```
True
```

In [77]:
```
my_list = [5, 4, 7, 8, 9, 12, 5, 5, 8, 9, 9]
my_dict = {5: 0, 4: 0, 7: 0, 8: 0}
for listo in my_list:
    if listo in my_dict:
      my_dict[listo] += 1
```

In [78]:
```
my_dict
```

Out[78]:
```
{5: 3, 4: 1, 7: 1, 8: 2}
```

In [79]:
```
list_file[0][11]
```

Out[79]:
```
'cont_rating'
```

In [80]:
```
my_dictionary = {"4+": 0, "12+": 0, "17+": 0, "9+": 0}
for files in list_file[1:]:
    content_rating = files[11]
    if content_rating in my_dictionary:
       my_dictionary[content_rating] +=  1
```

In [81]:
```
my_dictionary
```

Out[81]:
```
{'4+': 4433, '12+': 1155, '17+': 622, '9+': 987}
```

In [82]:
```
the_dict = {}
for rate in list_file[1:]:
    content_rating = rate[11]
    if content_rating in the_dict:
       the_dict[content_rating] += 1
    else:
        the_dict[content_rating] = 1
```

In [83]:
```
the_dict
```

Out[83]:
```
{'4+': 4433, '12+': 1155, '17+': 622, '9+': 987}
```

In [84]:
```
freq_dict = {}
for rate in list_file[1:]:
    prime_genre = rate[12]
    if prime_genre in freq_dict:
        freq_dict[prime_genre] += 1
    else:
        freq_dict[prime_genre] = 1
```

In [85]:

```
freq_dict
```

Out[85]:

```
{'Games': 3862,
 'Productivity': 178,
 'Weather': 72,
 'Shopping': 122,
 'Reference': 64,
 'Finance': 104,
 'Music': 138,
 'Utilities': 248,
 'Travel': 81,
 'Social Networking': 167,
 'Sports': 114,
 'Business': 57,
 'Health & Fitness': 180,
 'Entertainment': 535,
 'Photo & Video': 349,
 'Navigation': 46,
 'Education': 453,
 'Lifestyle': 144,
 'Food & Drink': 63,
 'News': 75,
 'Book': 112,
 'Medical': 23,
 'Catalogs': 10}
```

In [86]:

```
import math
proportion_dict = {}
for key in freq_dict:
    proportion = (freq_dict[key]) / len(list_file)
    proportion = round(proportion, 3)
    proportion_dict[key] = proportion
```

In [87]:

```
proportion_dict
```

Out[87]:

```
{'Games': 0.537,
 'Productivity': 0.025,
 'Weather': 0.01,
 'Shopping': 0.017,
 'Reference': 0.009,
 'Finance': 0.014,
 'Music': 0.019,
 'Utilities': 0.034,
 'Travel': 0.011,
 'Social Networking': 0.023,
 'Sports': 0.016,
 'Business': 0.008,
 'Health & Fitness': 0.025,
 'Entertainment': 0.074,
 'Photo & Video': 0.048,
 'Navigation': 0.006,
 'Education': 0.063,
 'Lifestyle': 0.02,
 'Food & Drink': 0.009,
 'News': 0.01,
 'Book': 0.016,
 'Medical': 0.003,
 'Catalogs': 0.001}
```

In [88]:

```
import math
percentage_dict = {}
for keys in proportion_dict:
    percentage = (proportion_dict[keys]) * 100
    percentage = round(percentage, 3)
    percentage_dict[keys] = percentage
```

In [89]:
```
percentage_dict
```

Out[89]:
```
{'Games': 53.7,
 'Productivity': 2.5,
 'Weather': 1.0,
 'Shopping': 1.7,
 'Reference': 0.9,
 'Finance': 1.4,
 'Music': 1.9,
 'Utilities': 3.4,
 'Travel': 1.1,
 'Social Networking': 2.3,
 'Sports': 1.6,
 'Business': 0.8,
 'Health & Fitness': 2.5,
 'Entertainment': 7.4,
 'Photo & Video': 4.8,
 'Navigation': 0.6,
 'Education': 6.3,
 'Lifestyle': 2.0,
 'Food & Drink': 0.9,
 'News': 1.0,
 'Book': 1.6,
 'Medical': 0.3,
 'Catalogs': 0.1}
```

In [90]:
```
freq_dict = {}
for rate in list_file[1:]:
    size_byte = rate[3]
    if size_byte in freq_dict:
        freq_dict[size_byte] += 1
    else:
        freq_dicts[size_byte] = 1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3596\3951115270.py in <module>
      5         freq_dict[size_byte] += 1
      6     else:
----> 7         freq_dicts[size_byte] = 1

NameError: name 'freq_dicts' is not defined
```

In [91]:
```
freq_dicts
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3596\302237859.py in <module>
----> 1 freq_dicts

NameError: name 'freq_dicts' is not defined
```

In [92]:

```
freq_dicts = {'0 to 200 GB': 0, '200 to 400 GB': 0, '400 to 600 GB': 0, '600 to 8
for rate in list_file[1:]:
    size_byte = float(rate[3])
    if size_byte <= 200000000:
        freq_dicts["0 to 200 GB"] += 1
    elif 200000000 < size_byte <= 400000000:
        freq_dicts["200 to 400 GB"] += 1
    elif 400000000 < size_byte <= 600000000:
        freq_dicts["400 to 600 GB"] += 1
    elif 600000000 < size_byte <= 800000000:
        freq_dicts["600 to 800 GB"]
    elif 800000000 < size_byte <= 1000000000:
        freq_dicts["800 to 1000 GB"]
```

In [93]:

```
freq_dicts
```

Out[93]:

```
{'0 to 200 GB': 5610,
 '200 to 400 GB': 835,
 '400 to 600 GB': 250,
 '600 to 800 GB': 0,
 '800 to 1000 GB': 0}
```

In [94]:

```
ratings = ['4+', '4+', '4+', '9+', '12+', '12+', '17+', '17+']
def freq_function(ratings):
    freq_dico = {}
    for rating in ratings:
        if rating in freq_dico:
            freq_dico[rating] += 1
        else:
            freq_dico[rating] = 1
    return freq_dico
```

In [95]:

```
yes = freq_function(ratings)
```

In [96]:

```
yes
```

Out[96]:

```
{'4+': 3, '9+': 1, '12+': 2, '17+': 2}
```

In [97]:

```
def sqr_num(number):
    root = number * number
    return root
```

In [98]:

```
sqr_num(9)
```

Out[98]:

```
81
```

In [99]:

```
def dates(day, month, year):
    return "today's" + " " + "date" + " " + "is" + " " + str(month) + "/" + str(d
```

In [100]:

```
dates(12, 4, 2022)
```

Out[100]:

```
"today's date is 4/12/2022"
```

In [101]:
```
list_file[0]
```

Out[101]:
```
['',
 'id',
 'track_name',
 'size_bytes',
 'currency',
 'price',
 'rating_count_tot',
 'rating_count_ver',
 'user_rating',
 'user_rating_ver',
 'ver',
 'cont_rating',
 'prime_genre',
 'sup_devices.num',
 'ipadSc_urls.num',
 'lang.num',
 'vpp_lic']
```

In [102]:
```
list_file[0][5]
```

Out[102]:
```
'price'
```

In [103]:
```
def column_contents(the_file):
    track_name_contents = {}
    for track in the_file[1:]:
        track_name = track[5]
        if track_name in track_name_contents:
            track_name_contents[track_name] += 1
        else:
            track_name_contents[track_name] = 1
    return track_name_contents
```

In [104]:
```
column_contents = column_contents(list_file)
```

```
In [105]: column_contents
```

```
Out[105]: {'3.99': 277,
           '0': 4056,
           '0.99': 728,
           '9.99': 81,
           '4.99': 394,
           '7.99': 33,
           '2.99': 683,
           '1.99': 621,
           '5.99': 52,
           '12.99': 5,
           '21.99': 1,
           '249.99': 1,
           '6.99': 166,
           '74.99': 1,
           '19.99': 13,
           '8.99': 9,
           '24.99': 8,
           '13.99': 6,
           '14.99': 21,
           '16.99': 2,
           '47.99': 1,
           '11.99': 6,
           '59.99': 3,
           '15.99': 4,
           '27.99': 2,
           '17.99': 3,
           '299.99': 1,
           '49.99': 2,
           '23.99': 2,
           '20.99': 2,
           '39.99': 2,
           '99.99': 1,
           '29.99': 6,
           '34.99': 1,
           '18.99': 1,
           '22.99': 2}
```

```
In [106]: max(column_contents)
```

```
Out[106]: '99.99'
```

```
In [107]: column_contents.get("4.5")
```

```
In [108]: results = []   #calculating max and min of a dictionary
          for key in column_contents:
              values = column_contents[key]
              results.append(values)
```

```
In [109]: max(results)
```

```
Out[109]: 4056
```

In [168]:

```
highest = max(results)
```

In [169]:

```
highest in column_contents
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3596\940077535.py in <module>
----> 1 highest in column_contents

TypeError: argument of type 'function' is not iterable
```

In [170]:

```
def column_contents(the_file):  #frequency table
    track_name_contents = {}
    for track in the_file[1:]:
        track_name = track[5]
        if track_name in track_name_contents:
            track_name_contents[track_name] += 1
        else:
            track_name_contents[track_name] = 1
    return track_name_contents
```

In [171]:

```
a_list = [4444, 8897, 6340, 9896, 4835, 4324, 10, 6445,
          661, 1246, 1000, 7429, 1376, 8121, 647, 1280,
          3993, 4881, 9500, 6701, 1199, 6251, 4432, 37]
sumation = 0
for lists in a_list:
    sumation += lists
print(sumation)
```

```
103945
```

In [172]:

```
sum(a_list)
```

Out[172]:

```
103945
```

In [173]:

```
def extract(in_file):
    content_ratings = []
    for row in in_file[1:]:
        content_rating = row[11]
        content_ratings.append(content_rating)
    return content_ratings
```

In [174]:

```
def column_contents(the_file, index):  #frequency table
    track_name_contents = {}
    for track in the_file[1:]:
        track_name = track[index]
        if track_name in track_name_contents:
            track_name_contents[track_name] += 1
        else:
            track_name_contents[track_name] = 1
    return track_name_contents
```

In [175]:
```
column_contents(list_file, 12)
```

Out[175]:
```
{'Games': 3862,
 'Productivity': 178,
 'Weather': 72,
 'Shopping': 122,
 'Reference': 64,
 'Finance': 104,
 'Music': 138,
 'Utilities': 248,
 'Travel': 81,
 'Social Networking': 167,
 'Sports': 114,
 'Business': 57,
 'Health & Fitness': 180,
 'Entertainment': 535,
 'Photo & Video': 349,
 'Navigation': 46,
 'Education': 453,
 'Lifestyle': 144,
 'Food & Drink': 63,
 'News': 75,
 'Book': 112,
 'Medical': 23,
 'Catalogs': 10}
```

In [176]:
```
my_numbers = [4444, 8897, 6340, 9896, 4835, 4324, 10, 6445,
          661, 1246, 1000, 7429, 1376, 8121, 647, 1280,
          3993, 4881, 9500, 6701, 1199, 6251, 4432, 37]
def addition(my_numbers):
    results = 0
    for adding in my_numbers:
        results += adding
    return results

def lengther(my_numbers):
    lengths = 0
    for lengthing in my_numbers:
        lengths += 1
    return lengths

def means(my_numbers):
    summation = addition(my_numbers)
    length = lengther(my_numbers)
    mean = summation / length
    return mean
```

In [177]:
```
the_mean = means(my_numbers)
```

In [178]:
```
the_mean = round(the_mean)
```

In [179]:
```
the_mean
```

Out[179]:
```
4331
```

In [180]:
```
def pythagoras(a, b):
    first_square = a * a
    second_square = b * b
    third_square = (first_square + second_square) ** 2
    return first_square, second_square, third_square
```

In [181]:
```
pythagoras(5, 12)
```

Out[181]:
```
(25, 144, 28561)
```

In [182]:
```
def opening(this_file="AS.csv", header=True, encoding="UTF-8"):
    from csv import reader
    opened_files = open("AS.csv", encoding="UTF-8")
    read_file = reader(opened_files)
    list_files = list(read_file)
    if header:
        return list_files[1:]
    else:
        return list_files
```

In [183]:
```
def opening(this_file="AS.csv", header=True, encoding="UTF-8"):
    from csv import reader
    opened_files = open("AS.csv", encoding="UTF-8")
    read_file = reader(opened_files)
    list_files = list(read_file)
    if header:
        return list_files[1:], list_files[0]
    else:
        return list_files
```

In [184]:
```
def opening(this_file="./glo/googleplaystore.csv", header=True, encoding="UTF8"):
    from csv import reader
    opened_files = open("./glo/googleplaystore.csv", encoding="UTF-8")
    read_file = reader(opened_files)
    list_files = list(read_file)
    if header:
        return list_files[0], list_files[1:]
    else:
        return list_files
```

In [185]:
```
google_function = opening(this_file="./glo/googleplaystore.csv", header=True, enc
```

In [186]:

```
google_function
```

Out[186]:

```
(['App',
  'Category',
  'Rating',
  'Reviews',
  'Size',
  'Installs',
  'Type',
  'Price',
  'Content Rating',
  'Genres',
  'Last Updated',
  'Current Ver',
  'Android Ver'],
 [['Photo Editor & Candy Camera & Grid & ScrapBook',
   'ART_AND_DESIGN',
   '4.1',
   '159',
   '19M',
   '10,000+',
```

In [187]:

```
from csv import reader
opened_file = open("./glo/googleplaystore.csv", encoding="UTF-8")
read_fil = reader(opened_file)
list_google = list(read_fil)
```

In [188]:

```
from csv import reader
opened_filess = open("./glo/AppleStore.csv", encoding="UTF-8")
read_filll = reader(opened_filess)
list_apple = list(read_filll)
```

In [189]:

```
list_google
```

Out[189]:

```
[['App',
  'Category',
  'Rating',
  'Reviews',
  'Size',
  'Installs',
  'Type',
  'Price',
  'Content Rating',
  'Genres',
  'Last Updated',
  'Current Ver',
  'Android Ver'],
 ['Photo Editor & Candy Camera & Grid & ScrapBook',
  'ART_AND_DESIGN',
  '4.1',
  '159',
  '19M',
  '10,000+',
```

In [190]:
```
def explore_data(dataset, start, end, rows_and_columns=False):
    dataset_slice = dataset[start:end]
    for row in dataset_slice:
        print(row)
        print('\n') # adds a new (empty) line after each row

    if rows_and_columns:
        print('Number of rows:', len(dataset))
        print('Number of columns:', len(dataset[0]))
```

In [191]:
```
explore_data(list_google, 0, 3, rows_and_columns=False)
```

```
['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type', 'Price',
'Content Rating', 'Genres', 'Last Updated', 'Current Ver', 'Android Ver']


['Photo Editor & Candy Camera & Grid & ScrapBook', 'ART_AND_DESIGN', '4.1', '15
9', '19M', '10,000+', 'Free', '0', 'Everyone', 'Art & Design', 'January 7, 201
8', '1.0.0', '4.0.3 and up']


['Coloring book moana', 'ART_AND_DESIGN', '3.9', '967', '14M', '500,000+', 'Fre
e', '0', 'Everyone', 'Art & Design;Pretend Play', 'January 15, 2018', '2.0.0',
'4.0.3 and up']
```

In [192]:
```
explore_data(list_apple, 0, 3, rows_and_columns=False)
```

```
['id', 'track_name', 'size_bytes', 'currency', 'price', 'rating_count_tot', 'ra
ting_count_ver', 'user_rating', 'user_rating_ver', 'ver', 'cont_rating', 'prime
_genre', 'sup_devices.num', 'ipadSc_urls.num', 'lang.num', 'vpp_lic']


['284882215', 'Facebook', '389879808', 'USD', '0', '2974676', '212', '3.5', '3.
5', '95', '4+', 'Social Networking', '37', '1', '29', '1']


['389801252', 'Instagram', '113954816', 'USD', '0', '2161558', '1289', '4.5',
'4', '10.23', '12+', 'Photo & Video', '37', '0', '29', '1']
```

In [193]:
```
print(list_google[0])
```

```
['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type', 'Price',
'Content Rating', 'Genres', 'Last Updated', 'Current Ver', 'Android Ver']
```

In [194]:
```
print(list_apple[0])
```

```
['id', 'track_name', 'size_bytes', 'currency', 'price', 'rating_count_tot', 'ra
ting_count_ver', 'user_rating', 'user_rating_ver', 'ver', 'cont_rating', 'prime
_genre', 'sup_devices.num', 'ipadSc_urls.num', 'lang.num', 'vpp_lic']
```

In [195]:

```
len(list_google[1040])
```

Out[195]:

```
13
```

In [196]:

```
len(list_google[10472])
```

Out[196]:

```
13
```

In [197]:

```
len(list_google)    # do not run more than once
del(list_google[10472])
len(list_google)
```

Out[197]:

```
10841
```

In [198]:

```
len(list_google)
```

Out[198]:

```
10841
```

In [199]:

```
for apps in list_google:
    app = apps[0]
    if app == "Instagram":
        print(apps)
```

```
['Instagram', 'SOCIAL', '4.5', '66577313', 'Varies with device', '1,000,000,000
+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies with device', 'Vari
es with device']
['Instagram', 'SOCIAL', '4.5', '66577446', 'Varies with device', '1,000,000,000
+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies with device', 'Vari
es with device']
['Instagram', 'SOCIAL', '4.5', '66577313', 'Varies with device', '1,000,000,000
+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies with device', 'Vari
es with device']
['Instagram', 'SOCIAL', '4.5', '66509917', 'Varies with device', '1,000,000,000
+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies with device', 'Vari
es with device']
```

In [200]:

```
list_google[1][3]
```

Out[200]:

```
'159'
```

In [201]:

```
duplicate_list = []
unique_list = []
app_unique =[]
duplicate_app = []
for unique in list_google[1:]:
    name = unique[0]
    if name in app_unique:
        duplicate_app.append(name)
        duplicate_list.append(unique)

    else:
        app_unique.append(name)
        unique_list.append(unique)
print("length of duplicate app: ", len(duplicate_app))
print("\n")
print("exampale of duplicate app: ", duplicate_app[:15])
print("\n")
print(app_unique[0:2])
print("\n")
print(unique_list[0:2])
```

```
length of duplicate app:  1181


exampale of duplicate app:  ['Quick PDF Scanner + OCR FREE', 'Box', 'Google My
Business', 'ZOOM Cloud Meetings', 'join.me - Simple Meetings', 'Box', 'Zenefit
s', 'Google Ads', 'Google My Business', 'Slack', 'FreshBooks Classic', 'Insight
ly CRM', 'QuickBooks Accounting: Invoicing & Expenses', 'HipChat - Chat Built f
or Teams', 'Xero Accounting Software']


['Photo Editor & Candy Camera & Grid & ScrapBook', 'Coloring book moana']


[['Photo Editor & Candy Camera & Grid & ScrapBook', 'ART_AND_DESIGN', '4.1', '1
59', '19M', '10,000+', 'Free', '0', 'Everyone', 'Art & Design', 'January 7, 201
8', '1.0.0', '4.0.3 and up'], ['Coloring book moana', 'ART_AND_DESIGN', '3.9',
'967', '14M', '500,000+', 'Free', '0', 'Everyone', 'Art & Design;Pretend Play',
'January 15, 2018', '2.0.0', '4.0.3 and up']]
```

In [202]:

```
review_max = {}

for review in list_google[1:]:
    name = review[0]
    rate = float(review[3])
    if name in review_max and review_max[name] < rate:
        review_max[name] = rate
    elif name not in review_max:
        review_max[name] = rate
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3596\2815297727.py in <module>
      3 for review in list_google[1:]:
      4     name = review[0]
----> 5     rate = float(review[3])
      6     if name in review_max and review_max[name] < rate:
      7         review_max[name] = rate

ValueError: could not convert string to float: '3.0M'
```

In [203]:

```
review_max
```

Out[203]:

```
{'Photo Editor & Candy Camera & Grid & ScrapBook': 159.0,
 'Coloring book moana': 974.0,
 'U Launcher Lite – FREE Live Cool Themes, Hide Apps': 87510.0,
 'Sketch - Draw & Paint': 215644.0,
 'Pixel Draw - Number Art Coloring Book': 967.0,
 'Paper flowers instructions': 167.0,
 'Smoke Effect Photo Maker - Smoke Editor': 178.0,
 'Infinite Painter': 36815.0,
 'Garden Coloring Book': 13791.0,
 'Kids Paint Free - Drawing Fun': 121.0,
 'Text on Photo - Fonteee': 13880.0,
 'Name Art Photo Editor - Focus n Filters': 8788.0,
 'Tattoo Name On My Photo Editor': 44829.0,
 'Mandala Coloring Book': 4326.0,
 '3D Color Pixel by Number - Sandbox Art Coloring': 1518.0,
 'Learn To Draw Kawaii Characters': 55.0,
 'Photo Designer - Write your name with shapes': 3632.0,
 '350 Diy Room Decor Ideas': 27.0,
 'FlipaClip - Cartoon animation': 194216.0,
 'ibis Paint X': 224399.0,
```

In [204]:

```
list_google[0][0]
```

Out[204]:

```
'App'
```

In [205]:

```
len(list_google)
```

Out[205]:

```
10841
```

In [206]:
```
cleaned_list_google = []
already_added_list_google = []
for iterate in list_google[1:]:
    app_name = iterate[0]
    app_rating = float(iterate[3])
    if (review_max[app_name] == app_rating) and (app_name not in already_added_li
        cleaned_list_google.append(iterate)
        already_added_list_google.append(app_name)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3596\3388673217.py in <module>
      3 for iterate in list_google[1:]:
      4     app_name = iterate[0]
----> 5     app_rating = float(iterate[3])
      6     if (review_max[app_name] == app_rating) and (app_name not in alread
y_added_list_google):
      7         cleaned_list_google.append(iterate)

ValueError: could not convert string to float: '3.0M'
```

In [ ]:
```
len(cleaned_list_google)
```

In [ ]:
```
review_max['Photo Editor & Candy Camera & Grid & ScrapBook']
```

In [ ]:
```
len(android_clean)
```

In [ ]:
```
explore_data(cleaned_list_google, 0, 1, rows_and_columns=True)
```

In [207]:
```
def is_English(string):
    non_ascii = 0
    for character in string:
        if ord(character) > 127:
            non_ascii += 1

    if non_ascii > 3:
        return False
    else:
        return True

is_English("و برامج اخرى كثيرة")
```

Out[207]:
```
False
```

In [253]:
```
english_list_google = []
for values in cleaned_list_google[1:]:
    name = values[0]
    if is_English(name):
        english_list_google.append(values)
```

In [254]:
```
len(english_list_google)
```

Out[254]:
```
9254
```

In [210]:
```
free_list_google = []
for for_free in english_list_google:
    price = for_free[7]
    if price == "0":
        free_list_google.append(for_free)
```

In [211]:
```
free_list_google
```

Out[211]:
```
[['U Launcher Lite – FREE Live Cool Themes, Hide Apps',
  'ART_AND_DESIGN',
  '4.7',
  '87510',
  '8.7M',
  '5,000,000+',
  'Free',
  '0',
  'Everyone',
  'Art & Design',
  'August 1, 2018',
  '1.2.4',
  '4.0.3 and up'],
 ['Sketch - Draw & Paint',
  'ART_AND_DESIGN',
  '4.5',
  '215644',
  '25M',
  '50,000,000+',
  'Free',
```

In [212]:
```
def column_contents(the_file):  #frequency table
    track_name_contents = {}
    for track in the_file[1:]:
        track_name = track[9]
        if track_name in track_name_contents:
            track_name_contents[track_name] += 1
        else:
            track_name_contents[track_name] = 1
    return track_name_contents
```

In [213]:
```
new_value = max(alpha_dict, key=alpha_dict.get)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3596\3192844397.py in <module>
----> 1 new_value = max(alpha_dict, key=alpha_dict.get)

NameError: name 'alpha_dict' is not defined
```

In [214]:
```
genre = column_contents(free_list_google)
```

```
In [215]: sum_genre = genre.values()   #how to sum dictionary values

          total_genre = sum(sum_genre)

          print(total_genre)

          8529
```

```
In [216]: sorted_dict = dict( sorted(genre.items(),
                                     key=lambda item: item[1],
                                     reverse=True))
```

```
In [217]: sorted_dict

Out[217]: {'Tools': 723,
           'Entertainment': 521,
           'Education': 456,
           'Business': 392,
           'Productivity': 335,
           'Lifestyle': 324,
           'Finance': 319,
           'Medical': 309,
           'Health & Fitness': 271,
           'Communication': 270,
           'Personalization': 269,
           'Sports': 267,
           'Action': 258,
           'Photography': 253,
           'News & Magazines': 235,
           'Social': 229,
           'Shopping': 197,
           'Travel & Local': 195,
           'Books & Reference': 181,
           'Simulation': 179
```

```
In [218]: total_genre

Out[218]: 8529
```

In [219]:
```
genre
```

Out[219]:
```
{'Art & Design': 51,
 'Art & Design;Creativity': 6,
 'Auto & Vehicles': 79,
 'Beauty': 53,
 'Books & Reference': 181,
 'Business': 392,
 'Comics': 53,
 'Comics;Creativity': 1,
 'Communication': 270,
 'Dating': 165,
 'Education': 456,
 'Education;Creativity': 4,
 'Education;Education': 30,
 'Education;Pretend Play': 5,
 'Education;Brain Games': 3,
 'Entertainment': 521,
 'Entertainment;Brain Games': 7,
 'Entertainment;Creativity': 3,
 'Entertainment;Music & Video': 15,
 'Events': 61,
```

In [220]:
```
def proportion(the_file):  #proportion table
    track_name_contents = {}
    for track in the_file:
        proportion = the_file[track] / total_genre
        track_name_contents[track] = proportion
    return track_name_contents
```

In [221]:
```
proportion_file = proportion(genre)
```

In [222]:
```
proportion_file
```

Out[222]:
```
{'Art & Design': 0.005979599015124868,
 'Art & Design;Creativity': 0.0007034822370735139,
 'Auto & Vehicles': 0.009262516121467933,
 'Beauty': 0.006214093094149373,
 'Books & Reference': 0.02122171415171767,
 'Business': 0.045960839488802904,
 'Comics': 0.006214093094149373,
 'Comics;Creativity': 0.00011724703951225231,
 'Communication': 0.03165670066830813,
 'Dating': 0.019345761519521634,
 'Education': 0.05346465001758706,
 'Education;Creativity': 0.00046898815804900925,
 'Education;Education': 0.0035174111853675696,
 'Education;Pretend Play': 0.0005862351975612616,
 'Education;Brain Games': 0.00035174111853675694,
 'Entertainment': 0.061085707585883456,
 'Entertainment;Brain Games': 0.0008207292765857662,
 'Entertainment;Creativity': 0.00035174111853675694,
 'Entertainment;Music & Video': 0.0017587055926837848,
 'Events': 0.007152069410247301,
```

In [223]:
```
def percentage(the_fil):  #proportion table
    track_name_content = {}
    for track in the_fil:
        percentage = the_fil[track] * 100
        percentage = round(percentage, 1)
        track_name_content[track] = percentage
    return track_name_content
```

In [224]:
```
percentage_file = percentage(proportion_file)
```

In [225]:
```
percentage_file
```

Out[225]:
```
{'Art & Design': 0.6,
 'Art & Design;Creativity': 0.1,
 'Auto & Vehicles': 0.9,
 'Beauty': 0.6,
 'Books & Reference': 2.1,
 'Business': 4.6,
 'Comics': 0.6,
 'Comics;Creativity': 0.0,
 'Communication': 3.2,
 'Dating': 1.9,
 'Education': 5.3,
 'Education;Creativity': 0.0,
 'Education;Education': 0.4,
 'Education;Pretend Play': 0.1,
 'Education;Brain Games': 0.0,
 'Entertainment': 6.1,
 'Entertainment;Brain Games': 0.1,
 'Entertainment;Creativity': 0.0,
 'Entertainment;Music & Video': 0.2,
```

In [226]:
```
google_sorted_dict = dict( sorted(percentage_file.items(),
                           key=lambda item: item[1],
                           reverse=True))
```

In [227]:
```
google_sorted_dict
```

Out[227]:
```
{'Tools': 8.5,
 'Entertainment': 6.1,
 'Education': 5.3,
 'Business': 4.6,
 'Productivity': 3.9,
 'Lifestyle': 3.8,
 'Finance': 3.7,
 'Medical': 3.6,
 'Communication': 3.2,
 'Health & Fitness': 3.2,
 'Personalization': 3.2,
 'Sports': 3.1,
 'Action': 3.0,
 'Photography': 3.0,
 'News & Magazines': 2.8,
 'Social': 2.7,
 'Shopping': 2.3,
 'Travel & Local': 2.3,
 'Books & Reference': 2.1,
```

In [228]:
```
list_google[5][5]
```

Out[228]:
```
'100,000+'
```

In [229]:
```
len(cleaned_list_google)
```

Out[229]:
```
9299
```

In [230]:
```
install_dictionary = {}
for element in cleaned_list_google:
    the_app = element[0]
    the_installs = element[5]
    if the_app in install_dictionary:
        install_dictionary[the_app] = the_installs
    else:
        install_dictionary[the_app] = the_installs
```

In [231]:
```
install_dictionary
```

Out[231]:
```
{'Photo Editor & Candy Camera & Grid & ScrapBook': '10,000+',
 'U Launcher Lite – FREE Live Cool Themes, Hide Apps': '5,000,000+',
 'Sketch - Draw & Paint': '50,000,000+',
 'Pixel Draw - Number Art Coloring Book': '100,000+',
 'Paper flowers instructions': '50,000+',
 'Smoke Effect Photo Maker - Smoke Editor': '50,000+',
 'Infinite Painter': '1,000,000+',
 'Garden Coloring Book': '1,000,000+',
 'Kids Paint Free - Drawing Fun': '10,000+',
 'Text on Photo - Fonteee': '1,000,000+',
 'Name Art Photo Editor - Focus n Filters': '1,000,000+',
 'Tattoo Name On My Photo Editor': '10,000,000+',
 'Mandala Coloring Book': '100,000+',
 '3D Color Pixel by Number - Sandbox Art Coloring': '100,000+',
 'Learn To Draw Kawaii Characters': '5,000+',
 'Photo Designer - Write your name with shapes': '500,000+',
 '350 Diy Room Decor Ideas': '10,000+',
 'FlipaClip - Cartoon animation': '5,000,000+',
 'ibis Paint X': '10,000,000+',
```

In [232]:
```
sorted_installed_app = dict( sorted(install_dictionary.items(),
                           key=lambda item: item[1],
                           reverse=True))
```

In [233]:
```
sorted_installed_app
```

Out[233]:
```
{'Google Duo - High Quality Video Calls': '500,000,000+',
 'imo free video calls and chat': '500,000,000+',
 'LINE: Free Calls & Messages': '500,000,000+',
 'UC Browser - Fast Download Private & Secure': '500,000,000+',
 'Candy Crush Saga': '500,000,000+',
 'Pou': '500,000,000+',
 'My Talking Tom': '500,000,000+',
 'Temple Run 2': '500,000,000+',
 'Facebook Lite': '500,000,000+',
 'Snapchat': '500,000,000+',
 'Google Translate': '500,000,000+',
 'SHAREit - Transfer & Share': '500,000,000+',
 'Gboard - the Google Keyboard': '500,000,000+',
 'Microsoft Word': '500,000,000+',
 'Dropbox': '500,000,000+',
 'Google Calendar': '500,000,000+',
 'Cloud Print': '500,000,000+',
 'MX Player': '500,000,000+',
 'Twitter': '500,000,000+',
 'Flipboard: News For Our Time': '500,000,000+'
```

In [234]:

```
def column_contents(the_file, index):  #frequency table
    track_name_contents = {}
    for track in the_file:
        track_name = track[index]
        if track_name in track_name_contents:
            track_name_contents[track_name] += 1
        else:
            track_name_contents[track_name] = 1
    return track_name_contents
```

In [235]:

```
the_sorted_google = column_contents(cleaned_list_google, 5)
```

In [236]:

```
sorted_installed_google_app = dict( sorted(the_sorted_google.items(),
                           key=lambda item: item[1],
                           reverse=True))
```

In [237]:

```
sorted_installed_google_app
```

Out[237]:

```
{'1,000,000+': 1394,
 '100,000+': 1080,
 '10,000+': 985,
 '10,000,000+': 921,
 '1,000+': 842,
 '100+': 663,
 '5,000,000+': 600,
 '500,000+': 497,
 '50,000+': 449,
 '5,000+': 441,
 '10+': 345,
 '500+': 312,
 '50,000,000+': 202,
 '50+': 190,
 '100,000,000+': 188,
 '5+': 73,
 '1+': 58,
 '500,000,000+': 24,
 '1,000,000,000+': 20,
 '0+': 14,
 '0': 1}
```

In [238]:

```
def proportion(the_file):  #proportion table
    track_name_contents = {}
    for track in the_file:
        proportion = the_file[track] / total_genre
        track_name_contents[track] = proportion
    return track_name_contents
```

In [239]:

```
sorted_installed_google_app_proportion = proportion(sorted_installed_google_app)
```

In [240]:

```
def percentage(the_fil):  #proportion table
    track_name_content = {}
    for track in the_fil:
        percentage = the_fil[track] * 100
        percentage = round(percentage, 5)
        track_name_content[track] = percentage
    return track_name_content
```

In [241]:

```
sorted_installed_google_app_percentage = percentage(sorted_installed_google_app_p
```

sorted_installed_google_app_percentage

In [242]:

```
list_google[0][4]
```

Out[242]:

```
'Size'
```

In [243]:

```
unique_genre = {}
for iterarting in cleaned_list_google:
    the_genre = iterarting[1]
    if the_genre in unique_genre:
        unique_genre[the_genre] += 1
    else:
        unique_genre[the_genre] = 1
```

In [244]:

```
#unique_genre_average = {}
for iterate in unique_genre:
    total = 0
    length = 0
    for iterat in cleaned_list_google:
        category_app = iterat[1]
        if category_app == iterate:
            n_installs = iterat[5]
            n_installs = n_installs.replace(',', '')
            n_installs = n_installs.replace('+', '')
            total += float(n_installs)
            length += 1
    avg_n_installs = total / length
    print(iterate, ':', avg_n_installs)
```

```
ART_AND_DESIGN : 1856362.2950819673
AUTO_AND_VEHICLES : 647806.2195121951
BEAUTY : 513151.88679245283
BOOKS_AND_REFERENCE : 8124095.126829268
BUSINESS : 1721304.4296296297
COMICS : 817657.2727272727
COMMUNICATION : 37039172.218120806
DATING : 828971.2176470588
EDUCATION : 1775252.3364485982
ENTERTAINMENT : 11375402.298850575
EVENTS : 257623.40322580645
FINANCE : 1359097.2626865671
FOOD_AND_DRINK : 1891060.2767857143
HEALTH_AND_FITNESS : 4013724.603508772
HOUSE_AND_HOME : 1331540.5616438356
LIBRARIES_AND_DEMO : 518023.0120481928
LIFESTYLE : 1415237.2959770116
GAME : 14724224.331125828
FAMILY : 3380226.5750551876
MEDICAL : 97651.09207161126
SOCIAL : 23610201.73275862
SHOPPING : 6994243.925
PHOTOGRAPHY : 16582899.835164836
SPORTS : 3829866.9572953735
TRAVEL_AND_LOCAL : 13917036.278846154
TOOLS : 10098319.107365793
PERSONALIZATION : 4361699.58739255
PRODUCTIVITY : 15914800.01098901
PARENTING : 525351.8333333334
WEATHER : 5012340.277777778
VIDEO_PLAYERS : 24271003.765432097
NEWS_AND_MAGAZINES : 9824783.153526971
MAPS_AND_NAVIGATION : 3994086.4285714286
```

In [251]:

```
list_google[2][1]
```

Out[251]:

```
'ART_AND_DESIGN'
```

In [248]:
```
for app in cleaned_list_google:
    if app[1] == 'COMMUNICATION' and (app[5] == '1,000,000,000+'
                                      or app[5] == '500,000,000+'
                                      or app[5] == '100,000,000+'):
        print(app[0], ':', app[5])
```

```
WhatsApp Messenger : 1,000,000,000+
imo beta free calls and text : 100,000,000+
Android Messages : 100,000,000+
Google Duo - High Quality Video Calls : 500,000,000+
Messenger – Text and Video Chat for Free : 1,000,000,000+
imo free video calls and chat : 500,000,000+
Skype - free IM & video calls : 1,000,000,000+
Who : 100,000,000+
GO SMS Pro - Messenger, Free Themes, Emoji : 100,000,000+
LINE: Free Calls & Messages : 500,000,000+
Google Chrome: Fast & Secure : 1,000,000,000+
Firefox Browser fast & private : 100,000,000+
UC Browser - Fast Download Private & Secure : 500,000,000+
Gmail : 1,000,000,000+
Hangouts : 1,000,000,000+
Messenger Lite: Free Calls & Messages : 100,000,000+
Kik : 100,000,000+
KakaoTalk: Free Calls & Text : 100,000,000+
Opera Mini - fast web browser : 100,000,000+
Opera Browser: Fast and Secure : 100,000,000+
Telegram : 100,000,000+
Truecaller: Caller ID, SMS spam blocking & Dialer : 100,000,000+
UC Browser Mini -Tiny Fast Private & Secure : 100,000,000+
Viber Messenger : 500,000,000+
WeChat : 100,000,000+
Yahoo Mail – Stay Organized : 100,000,000+
BBM - Free Calls & Messages : 100,000,000+
```

In [252]:

```
for app in cleaned_list_google:
    if app[1] == 'COMMUNICATION':
        print(app[0], ':', app[5])
```

```
WhatsApp Messenger : 1,000,000,000+
Messenger for SMS : 10,000,000+
My Tele2 : 5,000,000+
imo beta free calls and text : 100,000,000+
Contacts : 50,000,000+
Call Free – Free Call : 5,000,000+
Web Browser & Explorer : 5,000,000+
Browser 4G : 10,000,000+
MegaFon Dashboard : 10,000,000+
ZenUI Dialer & Contacts : 10,000,000+
Cricket Visual Voicemail : 10,000,000+
TracFone My Account : 1,000,000+
Xperia Link™ : 10,000,000+
TouchPal Keyboard - Fun Emoji & Android Keyboard : 10,000,000+
Skype Lite - Free Video Call & Chat : 5,000,000+
My magenta : 1,000,000+
Android Messages : 100,000,000+
Google Duo - High Quality Video Calls : 500,000,000+
Seznam.cz : 1,000,000+
```

In [1]:

```
pip install git
```

```
Note: you may need to restart the kernel to use updated packages.

ERROR: Could not find a version that satisfies the requirement git (from versio
ns: none)
ERROR: No matching distribution found for git
```

In [ ]: