Successful App Profiles for the Google Play and App Store Markets

February 8, 2021

1 Successful App Profiles for the Google Play and App Store Markets

In this project, I attempt to find what creates a profitable app – specifically, what traits are common in apps that maximize user engagement with advertisements and generate profit. I will be using Python for data exploration, cleaning, and analysis.

As of September 2018, there were about 2 million iOS apps available on the App Store, and 2.1 million Android apps on Google Play. I will collect and analyze data of mobile apps currently active on both the Google Play and App Store. I will analyze a sample of the total data (4 million apps total) by using two data sets via Kaggle.

The first data set will be a sample of approximately 10,000 Androids apps from the Play store from August 2018 – this can be downloaded [here][https://www.kaggle.com/lava18/google-play-store-apps] from Kaggle. The second data set will be a sample of approximately 7,000 iOS apps from the App Store – this can be downloaded [here][https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps].

We will go through 3 stages in the process: exploring the data, cleaning the data, and analyzing the data to generate actionable insights. For simplicity, we will be focusing only on free apps presented in the English language.

1.1 Exploring Our Data

We will begin by exploring our data and attempting to discern their configurations. Below is a `explore_data` that takes in a data set, start index, and end index to generate a portion of the data.

[18]:

```python
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]))
```

Below, we will open each file in order to continue our exploration.

[19]:

```python
open_file1 = open('/Users/natasharavinand/Downloads/my_datasets/Projects/
 ↪googleplaystore.csv')
from csv import reader
read_file1 = reader(open_file1)
playstore_data = list(read_file1)

open_file2 = open('/Users/natasharavinand/Downloads/my_datasets/Projects/
 ↪AppleStore.csv')
read_file2 = reader(open_file2)
appstore_data = list(read_file2)
```

We will print the header columns of both data sets to get a sense of the material they contain, as well as a few rows from each.

[20]:

```python
print(playstore_data[0])
print('\n')
print(appstore_data[0])
print('\n')
print('Play Store Rows: ', playstore_data[1:3])
print('\n')
print('App Store Rows: ', appstore_data[1:3])
```

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


['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']


Play Store Rows:  [['Photo Editor & Candy Camera & Grid & ScrapBook',
'ART_AND_DESIGN', '4.1', '159', '19M', '10,000+', 'Free', '0', 'Everyone', 'Art
& Design', 'January 7, 2018', '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']]


App Store Rows:  [['284882215', 'Facebook', '389879808', 'USD', '0.0',
'2974676', '212', '3.5', '3.5', '95.0', '4+', 'Social Networking', '37', '1',
'29', '1'], ['389801252', 'Instagram', '113954816', 'USD', '0.0', '2161558',
'1289', '4.5', '4.0', '10.23', '12+', 'Photo & Video', '37', '0', '29', '1']]
```

Some of the information in common include:

Column	Description
App Title	Title of the app
Category	The genre designation of the app

Column	Description
Rating	The average user rating
Installations	The total number of downloads of the app
Reviews	The total number of reviews

Some of the column names slightly differ between the two data sets. We can pick out a few criteria that may aid in designating a “popular” app, such as the amount of reviews or installations.

1.2 Cleaning our Data

In order to derive accurate and actionable insights, we must clean the data for a few reasons: we must remove duplicate content, ensure updated data, and use data that manages our criteria (free, English-language apps).

1.2.1 Removing erroneous data

We see from the documentation the App Store data set has mostly unique values. Upon inspection of the Play Store data set, we see that the element in row 10,473 does not have a value for the `category` column, which has shifted values downward. In order to fix this, we remove this row:

[21]:

```python
del playstore_data[10473]
```

Removing duplicate data We inspect the Play Store data set even more to find many duplicates of certain apps. For example, the app Twitter has three entries:

[22]:

```python
for app in playstore_data:
    if app[0] == 'Twitter':
        print(app)
```

```
['Twitter', 'NEWS_AND_MAGAZINES', '4.3', '11667403', 'Varies with device',
'500,000,000+', 'Free', '0', 'Mature 17+', 'News & Magazines', 'August 6, 2018',
'Varies with device', 'Varies with device']
['Twitter', 'NEWS_AND_MAGAZINES', '4.3', '11667403', 'Varies with device',
'500,000,000+', 'Free', '0', 'Mature 17+', 'News & Magazines', 'August 6, 2018',
'Varies with device', 'Varies with device']
['Twitter', 'NEWS_AND_MAGAZINES', '4.3', '11657972', 'Varies with device',
'500,000,000+', 'Free', '0', 'Mature 17+', 'News & Magazines', 'July 30, 2018',
'Varies with device', 'Varies with device']
```

We see this is because the data was collected at different times and thus have a different number of reviews.

[23]:

```python
duplicate_set = []
unique_set = []
duplicates = 0

for app in playstore_data:
    if app[0] in unique_set and app not in duplicate_set:
        duplicate_set.append(app[0])
        duplicates += 1
    elif app[0] not in unique_set:
        unique_set.append(app[0])

print(duplicate_set[:10])
print(duplicates)
```

```
['Quick PDF Scanner + OCR FREE', 'Box', 'Google My Business', 'ZOOM Cloud
Meetings', 'join.me - Simple Meetings', 'Box', 'Zenefits', 'Google Ads', 'Google
My Business', 'Slack']
1181
```

In fact, we see many duplicates in the set – 1,181 to be exact. Some duplicates are printed above.

Therefore, in order to have our final data set include data that is the most recent, we will be removing duplicates with the criterion that they have the highest review count. This will ensure we choose the most recent entry.

In order to remove the duplicates, we will use a dictionary that contains one entry per app, with each entry being the most recent one.

[24]:

```
reviews_max = {}

for app in playstore_data[1:]:
    n_reviews = float(app[3])
    name = app[0]
    if name in reviews_max and reviews_max[name] < n_reviews:
        reviews_max[name] = n_reviews
    elif name not in reviews_max:
        reviews_max[name] = n_reviews
```

Now that we have a dictionary of each entry, we can begin to create a "cleaned" list of data.

[25]:

```
android_clean = []
already_added = []

for app in playstore_data[1:]:
    name = app[0]
    reviews = float(app[3])
    if (reviews == reviews_max[name]) and (name not in already_added):
        android_clean.append(app)
        already_added.append(name)
```

Removing non-English apps In order to discern which apps have a non-English title, we will be checking for whether any characters in the title have ASCII codes that are not between 0-127, or those found in the English set.

We will begin by writing a function that takes in a string and returns whether it is probably

English; this function will check whether 3 or more characters are above 127 and will return `False` accordingly.

[26]:

```python
def isEnglish(string):
    counter = 0
    for character in string:
        if ord(character) > 127:
            counter += 1
        if counter > 3:
            return False
    return True
```

We will now apply the function `isEnglish` to the data sets to obtain only English-language apps.

[27]:

```python
android_clean2 = []

for app in android_clean:
    name = app[0]
    if isEnglish(name):
        android_clean2.append(app)

appstore_clean = []
for app in appstore_data:
    name = app[1]
    if isEnglish(name):
        appstore_clean.append(app)
```

Removing paid apps Lastly, we remove all apps whose type is "Paid" with a similar methodology from above.

[28]:

```python
final_play_data = []

for app in android_clean2:
    price = app[7]
    if price == "0":
        final_play_data.append(app)

final_app_data = []

for app in appstore_clean[1:]:
    price = float(app[4])
    if price == 0:
        final_app_data.append(app)
```

Now, we have two cleaned lists of data – `final_play_data` for the Play Store and `final_app_data` for the App Store – to analyze.

1.3 Analyzing the Data

1.3.1 Finding the most common genres

Now, we will analyze our cleaned data sets to find the most common genres of apps and generate a frequency table. The `genres` column and `category` from the Play Store data set and the `prime_genre` table from the App Store data set will help us generate accurate counts.

We will:

- Build a function to generate frequency tables that show percentages
- Build a function to display the percentages in a descending order

Now, we will focus on creating our frequency table. Below is a function to generate our frequency table as well as sort it in descending order of percentages.

[29]:

```python
def freq_table(data_set, integer):
    freq_dict = {}
    counter = 0
    for row in data_set:
        counter += 1
        val = row[integer]
        if val in freq_dict:
            freq_dict[val] += 1
        else:
            freq_dict[val] = 1

    freq_percentages = {}
    for key in freq_dict:
        percentage = freq_dict[key] / counter
        freq_percentages[key] = percentage

    return freq_percentages


def display_table(dataset, index):
    table = freq_table(dataset, index)
    table_display = []
    for key in table:
        key_val_as_tuple = (table[key], key)
        table_display.append(key_val_as_tuple)

    table_sorted = sorted(table_display, reverse = True)
    for entry in table_sorted:
        print(entry[1], ':', entry[0])
```

Using these functions, we'll now display the frequency table of the columns `prime_genre`, `Genres`, and `Category`.

[30]:

```python
display_table(final_app_data, 11) #prime_genre column
```

```
Games : 0.5816263190564867
Entertainment : 0.07883302296710118
Photo & Video : 0.04965859714463067
Education : 0.03662321539416512
Social Networking : 0.032898820608317815
Shopping : 0.0260707635009311
Utilities : 0.025139664804469275
Sports : 0.021415270018621976
Music : 0.020484171322160148
Health & Fitness : 0.020173805090006207
Productivity : 0.01738050900062073
Lifestyle : 0.015828677839851025
News : 0.01334574798261949
Travel : 0.012414649286157667
Finance : 0.0111731843575419
Weather : 0.008690254500310366
Food & Drink : 0.008069522036002483
Reference : 0.00558659217877095
Business : 0.005276225946617008
Book : 0.004345127250155183
Navigation : 0.00186219739292365
Medical : 0.00186219739292365
Catalogs : 0.0012414649286157666
```

We see that among the apps on the App Store (with the criteria of free and English-language), the most popular type of app are games. Games represent the majority of apps in this sample of the App Store, claiming 58.16% of the space. Entertainment, Photo & Video, Education, and Social Networking trail. The large volumes of Game and Entertainment apps on the App Store imply that many apps are created for non-practical purposes. However, this also suggests it may be difficult to launch a successful game application due to market saturation.

[31]:
```
display_table(final_play_data, 1)
```

```
FAMILY : 0.18907942238267147
GAME : 0.09724729241877256
TOOLS : 0.08461191335740072
BUSINESS : 0.04591606498194946
LIFESTYLE : 0.039034296028880866
PRODUCTIVITY : 0.03892148014440433
FINANCE : 0.03700361010830325
MEDICAL : 0.03531137184115524
SPORTS : 0.03395758122743682
PERSONALIZATION : 0.03316787003610108
COMMUNICATION : 0.032378158844765345
HEALTH_AND_FITNESS : 0.030798736462093863
PHOTOGRAPHY : 0.02944494584837545
NEWS_AND_MAGAZINES : 0.027978339350180504
SOCIAL : 0.026624548736462094
```

```
TRAVEL_AND_LOCAL : 0.023352888086642598
SHOPPING : 0.022450361010830325
BOOKS_AND_REFERENCE : 0.021435018050541516
DATING : 0.01861462093862816
VIDEO_PLAYERS : 0.017937725631768955
MAPS_AND_NAVIGATION : 0.013989169675090252
FOOD_AND_DRINK : 0.012409747292418772
EDUCATION : 0.011620036101083033
ENTERTAINMENT : 0.009589350180505414
LIBRARIES_AND_DEMO : 0.009363718411552346
AUTO_AND_VEHICLES : 0.009250902527075812
HOUSE_AND_HOME : 0.008235559566787004
WEATHER : 0.008009927797833934
EVENTS : 0.007107400722021661
PARENTING : 0.006543321299638989
ART_AND_DESIGN : 0.006430505415162455
COMICS : 0.006204873646209386
BEAUTY : 0.005979241877256318
```

The category of apps on the Play Store (with the criteria of free and English-language) that seems to be the most popular is the Family category, representing 18.90% of the sample. The game category is behind at 9.72%, followed by Tools, Business, Lifestyle, and Productivity. This data tells us there may be a greater market for family-oriented or practical applications on the Play Store, although non-practical applications like games or video players are still very abundant.

[32]:
```python
display_table(final_play_data, -4)
```

```
Tools : 0.08449909747292418
Entertainment : 0.06069494584837545
Education : 0.05347472924187725
Business : 0.04591606498194946
Productivity : 0.03892148014440433
Lifestyle : 0.03892148014440433
Finance : 0.03700361010830325
Medical : 0.03531137184115524
Sports : 0.03463447653429603
Personalization : 0.03316787003610108
Communication : 0.032378158844765345
Action : 0.03102436823104693
Health & Fitness : 0.030798736462093863
Photography : 0.02944494584837545
News & Magazines : 0.027978339350180504
Social : 0.026624548736462094
Travel & Local : 0.023240072202166066
Shopping : 0.022450361010830325
Books & Reference : 0.021435018050541516
Simulation : 0.020419675090252706
Dating : 0.01861462093862816
```

```
Arcade : 0.018501805054151624
Video Players & Editors : 0.017712093862815883
Casual : 0.01759927797833935
Maps & Navigation : 0.013989169675090252
Food & Drink : 0.012409747292418772
Puzzle : 0.01128158844765343
Racing : 0.009927797833935019
Role Playing : 0.009363718411552346
Libraries & Demo : 0.009363718411552346
Auto & Vehicles : 0.009250902527075812
Strategy : 0.009138086642599278
House & Home : 0.008235559566787004
Weather : 0.008009927797833934
Events : 0.007107400722021661
Adventure : 0.006768953068592058
Comics : 0.006092057761732852
Beauty : 0.005979241877256318
Art & Design : 0.005979241877256318
Parenting : 0.004963898916967509
Card : 0.004512635379061372
Casino : 0.0042870036101083035
Trivia : 0.004174187725631769
Educational;Education : 0.0039485559566787
Board : 0.003835740072202166
Educational : 0.0037229241877256318
Education;Education : 0.003384476534296029
Word : 0.002594765342960289
Casual;Pretend Play : 0.00236913357400722
Music : 0.002030685920577617
Racing;Action & Adventure : 0.0016922382671480144
Puzzle;Brain Games : 0.0016922382671480144
Entertainment;Music & Video : 0.0016922382671480144
Casual;Brain Games : 0.0013537906137184115
Casual;Action & Adventure : 0.0013537906137184115
Arcade;Action & Adventure : 0.0012409747292418773
Action;Action & Adventure : 0.0010153429602888086
Educational;Pretend Play : 0.0009025270758122744
Simulation;Action & Adventure : 0.00078971119133574
Parenting;Education : 0.00078971119133574
Entertainment;Brain Games : 0.00078971119133574
Board;Brain Games : 0.00078971119133574
Parenting;Music & Video : 0.0006768953068592057
Educational;Brain Games : 0.0006768953068592057
Casual;Creativity : 0.0006768953068592057
Art & Design;Creativity : 0.0006768953068592057
Education;Pretend Play : 0.0005640794223826715
Role Playing;Pretend Play : 0.0004512635379061372
Education;Creativity : 0.0004512635379061372
```

```
Role Playing;Action & Adventure : 0.0003384476534296029
Puzzle;Action & Adventure : 0.0003384476534296029
Entertainment;Creativity : 0.0003384476534296029
Entertainment;Action & Adventure : 0.0003384476534296029
Educational;Creativity : 0.0003384476534296029
Educational;Action & Adventure : 0.0003384476534296029
Education;Music & Video : 0.0003384476534296029
Education;Brain Games : 0.0003384476534296029
Education;Action & Adventure : 0.0003384476534296029
Adventure;Action & Adventure : 0.0003384476534296029
Video Players & Editors;Music & Video : 0.0002256317689530686
Sports;Action & Adventure : 0.0002256317689530686
Simulation;Pretend Play : 0.0002256317689530686
Puzzle;Creativity : 0.0002256317689530686
Music;Music & Video : 0.0002256317689530686
Entertainment;Pretend Play : 0.0002256317689530686
Casual;Education : 0.0002256317689530686
Board;Action & Adventure : 0.0002256317689530686
Video Players & Editors;Creativity : 0.0001128158844765343
Trivia;Education : 0.0001128158844765343
Travel & Local;Action & Adventure : 0.0001128158844765343
Tools;Education : 0.0001128158844765343
Strategy;Education : 0.0001128158844765343
Strategy;Creativity : 0.0001128158844765343
Strategy;Action & Adventure : 0.0001128158844765343
Simulation;Education : 0.0001128158844765343
Role Playing;Brain Games : 0.0001128158844765343
Racing;Pretend Play : 0.0001128158844765343
Puzzle;Education : 0.0001128158844765343
Parenting;Brain Games : 0.0001128158844765343
Music & Audio;Music & Video : 0.0001128158844765343
Lifestyle;Pretend Play : 0.0001128158844765343
Lifestyle;Education : 0.0001128158844765343
Health & Fitness;Education : 0.0001128158844765343
Health & Fitness;Action & Adventure : 0.0001128158844765343
Entertainment;Education : 0.0001128158844765343
Communication;Creativity : 0.0001128158844765343
Comics;Creativity : 0.0001128158844765343
Casual;Music & Video : 0.0001128158844765343
Card;Action & Adventure : 0.0001128158844765343
Books & Reference;Education : 0.0001128158844765343
Art & Design;Pretend Play : 0.0001128158844765343
Art & Design;Action & Adventure : 0.0001128158844765343
Arcade;Pretend Play : 0.0001128158844765343
Adventure;Education : 0.0001128158844765343
```

When we look at the genres for the Play Store (with the criteria of free and English-language), Tools has the greatest share of 8.44%, followed by Entertainment with 6.06% and Education with

5.34%. We can derive many of the applications are used for practical purposes. `Genres` seems to have many more labels than `category`, so in order to analyze the big picture, we will be using `category` for further analysis.

Some insights we can derive from these frequency tables include:

- The App Store has a greater volume of non-practical, fun and entertainment applications
- The Play Store has a variety of non-practical and practical applications
- The Play Store may have a specific market for family-oriented applications
- Compared to the App Store, the Play Store may have a more rounded market (practical vs non-practical)

1.3.2 Finding which apps have the most users

To find which genres are the most popular, we can calculate the average number of installs for each app genre. We have the number of installs per app in the `Installs` column of our Play Store data set, but this information is missing from our App Store data set. As a substitute, we will use the number of user ratings, which is located in the `rating_count_tot` column.

To calculate the average number of user ratings per app genre on the App Store, we must:

- Separate the apps of each genre
- Add up the user ratings for the apps of that genre.
- Divide the sum by the number of apps belonging to that genre

Most popular apps per genre on the App Store We will begin with the App Store and the `prime_genre` column. Below, we calculate the average number of user ratings per genre in the App Store data set.

[33]:

```python
app_genres = freq_table(final_app_data, -5)
for genre in app_genres:
    total = 0
    len_genre = 0
    for app in final_app_data:
        genre_app = app[-5]
        if genre_app == genre:
            n_ratings = float(app[5])
            total += n_ratings
            len_genre += 1
    avg_n_ratings = total / len_genre
    print(genre, ":" , avg_n_ratings)
```

```
Social Networking : 71548.34905660378
Photo & Video : 28441.54375
Games : 22788.6696905016
Music : 57326.530303030304
Reference : 74942.11111111111
Health & Fitness : 23298.015384615384
Weather : 52279.892857142855
Utilities : 18684.456790123455
```

```
Travel : 28243.8
Shopping : 26919.690476190477
News : 21248.023255813954
Navigation : 86090.33333333333
Lifestyle : 16485.764705882353
Entertainment : 14029.830708661417
Food & Drink : 33333.92307692308
Sports : 23008.898550724636
Book : 39758.5
Finance : 31467.944444444445
Education : 7003.983050847458
Productivity : 21028.410714285714
Business : 7491.117647058823
Catalogs : 4004.0
Medical : 612.0
```

Here, we see that Navigation apps are the most used, although this could be influenced by a few but very popular applications like Google Maps. Social Networking (with Instagram, Facebook, Twitter, etc) and Music (with Spotify, Pandora, etc) can also be responsible for their category's popularity.

Now, we will analyze our Play Store data similarly.

Most popular apps per genre on the Play Store We use our `display_table` function to look at the `Installs` of apps on the Play Store.

[35]:
```
display_table(final_play_data, 5)
```

```
1,000,000+ : 0.1572653429602888
100,000+ : 0.11552346570397112
10,000,000+ : 0.10548285198555957
10,000+ : 0.10198555956678701
1,000+ : 0.08393501805054152
100+ : 0.06915613718411552
5,000,000+ : 0.06825361010830325
500,000+ : 0.05561823104693141
50,000+ : 0.047721119133574005
5,000+ : 0.04512635379061372
10+ : 0.035424187725631766
500+ : 0.032490974729241874
50,000,000+ : 0.023014440433212997
100,000,000+ : 0.021322202166064983
50+ : 0.019178700361010830
5+ : 0.0078971119133574
1+ : 0.0050767148014440435
500,000,000+ : 0.002707581227436823
1,000,000,000+ : 0.002256317689530686
0+ : 0.0004512635379061372
0 : 0.0001128158844765343
```

We see that many of these are open-ended – for example, we don't know if 500+ installs means 500 installs or 1,000,000 installs. In order to remain conservative, we will treat these values as literal values (ex. 500+ installs will be treated as 500 installs). Our analysis, however, would be more accurate if we had access to the number of installations across these apps.

[41]:

```python
play_genres = freq_table(final_play_data, 1)

for category in play_genres:
    total = 0
    len_category = 0
    for app in final_play_data:
        category_app = app[1]
        if category_app == category:
            n_installs = app[5]
            installs = n_installs.replace('+', '')
            installs = installs.replace(',', '')
            total += float(installs)
            len_category += 1
    avg_installs = total / len_category
    print(category, ":", avg_installs)
```

```
ART_AND_DESIGN : 1986335.0877192982
AUTO_AND_VEHICLES : 647317.8170731707
BEAUTY : 513151.88679245283
BOOKS_AND_REFERENCE : 8767811.894736841
BUSINESS : 1712290.1474201474
COMICS : 817657.2727272727
COMMUNICATION : 38456119.167247385
DATING : 854028.8303030303
EDUCATION : 1833495.145631068
ENTERTAINMENT : 11640705.88235294
EVENTS : 253542.22222222222
FINANCE : 1387692.475609756
FOOD_AND_DRINK : 1924897.7363636363
HEALTH_AND_FITNESS : 4188821.9853479853
HOUSE_AND_HOME : 1331540.5616438356
LIBRARIES_AND_DEMO : 638503.734939759
LIFESTYLE : 1437816.2687861272
GAME : 15588015.603248259
FAMILY : 3695641.8198090694
MEDICAL : 120550.61980830671
SOCIAL : 23253652.127118643
SHOPPING : 7036877.311557789
PHOTOGRAPHY : 17840110.40229885
SPORTS : 3638640.1428571427
TRAVEL_AND_LOCAL : 13984077.710144928
TOOLS : 10801391.298666667
PERSONALIZATION : 5201482.6122448975
```

```
PRODUCTIVITY : 16787331.344927534
PARENTING : 542603.6206896552
WEATHER : 5074486.197183099
VIDEO_PLAYERS : 24727872.452830188
NEWS_AND_MAGAZINES : 9549178.467741935
MAPS_AND_NAVIGATION : 4056941.7741935486
```

We see that communication apps have the most downloads, but this may be the result of a few players in the market – ex. Messenger, WhatsApp – garnering most of the downloads. We see this here:

[43]:
```
for app in final_play_data:
    if (app[5] == '500,000,000+' or app[5] == '100,000,000+') and app[1] ==␣
 ↪'COMMUNICATION':
        print(app[0], ':', app[5])
```

```
imo beta free calls and text : 100,000,000+
Android Messages : 100,000,000+
Google Duo - High Quality Video Calls : 500,000,000+
imo free video calls and chat : 500,000,000+
Who : 100,000,000+
GO SMS Pro - Messenger, Free Themes, Emoji : 100,000,000+
LINE: Free Calls & Messages : 500,000,000+
Firefox Browser fast & private : 100,000,000+
UC Browser - Fast Download Private & Secure : 500,000,000+
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+
```

We see the same trend with video related applications, with some applications composing much of the market, as shown below.

[51]:
```
for app in final_play_data:
    if(app[5] == '500,000,000+' or app[5] == '100,000,000+') and app[1] ==␣
 ↪'VIDEO_PLAYERS':
        print(app[0], ':', app[5])
```

```
Motorola Gallery : 100,000,000+
VLC for Android : 100,000,000+
MX Player : 500,000,000+
Dubsmash : 100,000,000+
VivaVideo - Video Editor & Photo Movie : 100,000,000+
VideoShow-Video Editor, Video Maker, Beauty Camera : 100,000,000+
Motorola FM Radio : 100,000,000+
```

[53]:

```
app_genres = freq_table(final_app_data, -5)

for category in app_genres:
    total = 0
    len_category = 0
    for app in final_app_data:
        category_app = app[-5]
        if category_app == category:
            n_ratings = app[5]
            total += float(n_ratings)
            len_category += 1
    avg_ratings = total / len_category
    print(category, ":", avg_ratings)
```

```
Social Networking : 71548.34905660378
Photo & Video : 28441.54375
Games : 22788.6696905016
Music : 57326.530303030304
Reference : 74942.11111111111
Health & Fitness : 23298.015384615384
Weather : 52279.892857142855
Utilities : 18684.456790123455
Travel : 28243.8
Shopping : 26919.690476190477
News : 21248.023255813954
Navigation : 86090.33333333333
Lifestyle : 16485.764705882353
Entertainment : 14029.830708661417
Food & Drink : 33333.92307692308
Sports : 23008.898550724636
Book : 39758.5
Finance : 31467.944444444445
Education : 7003.983050847458
Productivity : 21028.410714285714
Business : 7491.117647058823
Catalogs : 4004.0
Medical : 612.0
```

We see that in the App Store, we similarly have high numbers of installations for Social Networking, Photo and Video, Games, Music, and Entertainment apps. Other genres that are popular include Reference and Weather.

We have seen that there is a high volume of games in the App and Play Stores, and that entertainmnent and communication are popularly downloaded applications. However, there might be an oversaturation of games and a difficulty to compete with industry giants that dominate the entertainment, communication, and video markets.

1.4 Conclusion

From what we can gather from a short analysis of the data, games, entertainment, communication, and video apps are all popular in both the Play and App Stores. The App Store market seems to concentrate many non-practical entertainment applications while the Play Store's market seems to be more rounded. The genres of applications that were most popularly downloaded across both stores included communication, music, and entertainment, but as we discussed above this could be due to the presence of a few major applications that carve out a significant piece of the market for their respective genre.