# Data Analysis: Popular Android and iOS Apps

# Goal: Determine what app profiles are likely to attract more users.

Background: Applications that are available for free downloads create revenue through in-app ads. This revenue is directly related to the amount of users who view these adds. Companies who build apps are likely to build more revenue by understanding which apps attract more viewers.

### Dataset Information: 1

As of Septermber 2018 there are about 2 million iOS apps on the Appp Store and 2.1 million Android apps on Google Play. We will be using a sample data set that contains data on ~10,000 Google Play apps and one that contains ~7,000 iOS apps.

We will be analyzing two public data sets found on kaggle.com.

Google Play Store App Data (https://www.kaggle.com/lava18/google-play-store-apps/home)

<u>Mobile App Store (Apple iOS)Data (https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps/home)</u>

```
In [3]: #import csv
from csv import reader
opened_file=open('AppleStore.csv')
read_file=reader(opened_file)
```

```
read_file=reader(opened_file)
ios=list(read_file)
ios_header=ios[0]
ios_body=ios[1:]
opened_file=open('googleplaystore.csv')
read_file=reader(opened_file)
android=list(read_file)
android_header=android[0]
android_body=android[1:]
```

The explore\_data function will make searching through the two datasets easier to understand.

This function takes in a dataset, a starting index, and an ending index. It prints the rows between that indices and also prints the number of rows and columns in dataset.

```
In [4]:
```

```
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]))
```

We will explore the header and first rows of each data set and determine which columns will be beneficial for our analysis

```
In [5]: print('ios_header: ', ios_header)
        print('\n')
        explore data(ios body, 1, 4, True)
          ios_header: ['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_url
          s.num', 'lang.num', 'vpp_lic']
          ['389801252', 'Instagram', '113954816', 'USD', '0.0', '2161558', '1289
           , '4.5', '4.0', '10.23', '12+', 'Photo & Video', '37', '0', '29', '1'
          1
          ['529479190', 'Clash of Clans', '116476928', 'USD', '0.0', '2130805',
           '579', '4.5', '4.5', '9.24.12', '9+', 'Games', '38', '5', '18', '1']
          ['420009108', 'Temple Run', '65921024', 'USD', '0.0', '1724546', '3842
           , '4.5', '4.0', '1.6.2', '9+', 'Games', '40', '5', '1', '1']
          Number of rows: 7197
          Number of columns: 16
```

In the ios applications we have 7197 rows.

The columns that seem related to our analysis seems to be:'track\_name', 'currency', 'price', 'rating\_count\_tot', 'rating\_count\_ver', and 'prime\_genre'

```
In [6]: print('\n')
```

```
print('android_header: ', android_header)
print('\n')
explore_data(android_body, 1, 4, True)
```

```
android_header: ['App', 'Category', 'Rating', 'Reviews', 'Size', 'Ins
talls', 'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', '
Current Ver', 'Android Ver']
```

```
['Coloring book moana', 'ART_AND_DESIGN', '3.9', '967', '14M', '500,00
0+', 'Free', '0', 'Everyone', 'Art & Design;Pretend Play', 'January 15
, 2018', '2.0.0', '4.0.3 and up']
```

```
['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', '5
0,000,000+', 'Free', '0', 'Teen', 'Art & Design', 'June 8, 2018', 'Var
ies with device', '4.2 and up']
```

```
Number of rows: 10841
Number of columns: 13
```

The number of rows in the Android data set is 10841

Looking at the header titles, the most relevent data for our analysis will include: 'App', 'Category', 'Reviews', 'Installs', 'Type', 'Price', and 'Genres'.

# **Data Cleaning:**

#### Deleting Error for Row in Android Data

The Google Play data <u>discussion section (https://www.kaggle.com/lava18/google-play-store-apps/discussion)</u> mentions that row 10472 may have an <u>error</u> (<u>https://www.kaggle.com/lava18/google-play-store-apps/discussion/66015</u>)</u>.

I will print the header row with the correct label for each column, a row with correct information, and the row that supposedly contains incorrect data in order to check this error.

```
In [7]: print(android_header) #header
         print('\n')
         print(android body[0]) #correct row
         print('\n')
         print(android_body[10472]) #error row
            ['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
            Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver', 'An
            droid Ver']
            ['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']
            ['Life Made WI-Fi Touchscreen Photo Frame', '1.9', '19', '3.0M', '1,00
            0+', 'Free', '0', 'Everyone', '', 'February 11, 2018', '1.0.19', '4.0
            and up']
          CONCLUSION: Row 10472 in the android, Google Play data set does have an error. The 'Rating'
          column should not have a value above 5. The value for 'Rating' in column 10472 = 19. We must
          delete this row to continue with our analysis This Row is also missing information for 'Category'
```

#### In [10]:

```
print(len(android_body))
#del(android_body[10472]) #dont run again
print(len(android_body))
print(android_body[10472])
```

```
10840
10840
['osmino Wi-Fi: free WiFi', 'TOOLS', '4.2', '134203', '4.1M', '10,000,
000+', 'Free', '0', 'Everyone', 'Tools', 'August 7, 2018', '6.06.14',
'4.4 and up']
```

When examining the new row 10472 in android\_body, I see that this row is missing the 'Category' column. After searching through the data set's discussion section, I see that many people have run into the same problem. I will run the row #del(android\_body[10472]) once again to get rid of this row as well.

### **Deleting Duplicate Entries**

#### Part 1:

When further exploring the Google Play data set, discussion section I found that there are some apps which have duplicate entries

```
In [11]: # example of duplicate entries for instagram
for app in android_body:
    name= app[0]
    if name == "Instagram":
        print(app)

    ['Instagram', 'SOCIAL', '4.5', '66577313', 'Varies with device', '1,00
    0,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies w
    ith device', 'Varies with device']
    ['Instagram', 'SOCIAL', '4.5', '66577446', 'Varies with device', '1,00
    0,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies w
    ith device', 'Varies with device']
    ['Instagram', 'SOCIAL', '4.5', '66577313', 'Varies with device', '1,00
    0,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies w
    ith device', 'Varies with device']
    ['Instagram', 'SOCIAL', '4.5', '66509917', 'Varies with device', '1,00
    0,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies w
    ith device', 'Varies with device']
    ['Instagram', 'SOCIAL', '4.5', '66509917', 'Varies with device', '1,00
    0,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies w
    ith device', 'Varies with device']
```

Next, I will create a list of duplicate apps and unique appls by looping through the android data set.

```
In [12]:
duplicate_apps = []
for app in android:
    name = app[0]
    if name in unique_apps:
        duplicate_apps.append(name)
    else:
        unique_apps.append(name)
print('Number of duplicate apps: ', len(duplicate_apps))
print('\n')
print('\n')
print('Examples of duplicate apps: ', duplicate_apps[:15])
```

```
Mumbers of Jun Plants and 1101
```

Examples of duplicate apps: ['Quick PDF Scanner + OCR FREE', 'Box', ' Google My Business', 'ZOOM Cloud Meetings', 'join.me - Simple Meetings ', 'Box', 'Zenefits', 'Google Ads', 'Google My Business', 'Slack', 'Fr eshBooks Classic', 'Insightly CRM', 'QuickBooks Accounting: Invoicing & Expenses', 'HipChat - Chat Built for Teams', 'Xero Accounting Softwa re']

In order to have an accurate account of the Android apps we are going to have to remove the duplicate rows for the same apps. However, we need to find the a method for chosing which row to keep.

When looking at the 4 duplicate instagram data rows, we see that the main difference does not lie in the 'Ratings' column, but it lies in the 'Reviews' column. We probably have duplicate rows becaue data was taken for the same app at different times.

Our goal in this project is to find which apps attract the most ratings. Therefore, we will want to keep the app data rows for the app which has the most ratings. This will allow us to have the most accurate analysis possible.

#### Part 2:

```
In [13]: print('Expected Length: ', len(android_body)-1181)
```

Expected Length: 9659

We have determined that there are 1,181 duplicate data rows in the Google Play app data set. After deleting these rows we should expect to have 9658 rows remaining.

In order to remove the duplicates I will create a dictionary where the key is the unique app name and the value is the highest number of reviews of that app. I will create a new data set which has only one entery per app which is the entry with the highest number of reviews.

```
In [14]: reviews_max={}
```

```
for app in android_body:
    name=app[0]
    n_reviews=float(app[3])

    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</pre>
```

```
In [15]: print(len(reviews_max))
```

9659

Type Markdown and LaTeX:  $\alpha^2$ 

In [16]: #Using the dictionary reviews\_max to remove the duplicate rows
android\_clean=[] #store new cleaned data
already\_added=[] #store app names

```
for app in android_body:
    name=app[0]
    n_reviews=float(app[3])
    if (reviews_max[name]==n_reviews) and (name not in already_added):
        android_clean.append(app)
        already_added.append(name)
print('length of clean data: ', len(android clean))
```

length of clean data: 9659

I will now use the explore\_data function to verify that android\_clean is the correct length with the same amount of rows as our header.

In [17]: explore\_data(android\_clean, 0, 15, True)

['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']

['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', '5 0,000,000+', 'Free', '0', 'Teen', 'Art & Design', 'June 8, 2018', 'Var ies with device', '4.2 and up']

['Pixel Draw - Number Art Coloring Book', 'ART\_AND\_DESIGN', '4.3', '96 7', '2.8M', '100,000+', 'Free', '0', 'Everyone', 'Art & Design;Creativ ity', 'June 20, 2018', '1.1', '4.4 and up']

['Paper flowers instructions', 'ART\_AND\_DESIGN', '4.4', '167', '5.6M', '50,000+', 'Free', '0', 'Everyone', 'Art & Design', 'March 26, 2017', '1.0', '2.3 and up']

['Smoke Effect Photo Maker - Smoke Editor', 'ART\_AND\_DESIGN', '3.8', '
178', '19M', '50,000+', 'Free', '0', 'Everyone', 'Art & Design', 'Apri
1 26, 2018', '1.1', '4.0.3 and up']

['Infinite Painter', 'ART\_AND\_DESIGN', '4.1', '36815', '29M', '1,000,0 00+', 'Free', '0', 'Everyone', 'Art & Design', 'June 14, 2018', '6.1.6 1.1', '4.2 and up']

['Garden Coloring Book', 'ART\_AND\_DESIGN', '4.4', '13791', '33M', '1,0 00,000+', 'Free', '0', 'Everyone', 'Art & Design', 'September 20, 2017 ', '2.9.2', '3.0 and up']

['Kids Paint Free - Drawing Fun', 'ART\_AND\_DESIGN', '4.7', '121', '3.1
M', '10,000+', 'Free', '0', 'Everyone', 'Art & Design;Creativity', 'Ju
ly 3, 2018', '2.8', '4.0.3 and up']

['Text on Photo - Fonteee', 'ART\_AND\_DESIGN', '4.4', '13880', '28M', ' 1,000,000+', 'Free', '0', 'Everyone', 'Art & Design', 'October 27, 201 7', '1.0.4', '4.1 and up']

['Name Art Photo Editor - Focus n Filters', 'ART AND DESIGN', '4.4', '

```
8788', '12M', '1,000,000+', 'Free', '0', 'Everyone', 'Art & Design', '
July 31, 2018', '1.0.15', '4.0 and up']
['Tattoo Name On My Photo Editor', 'ART_AND_DESIGN', '4.2', '44829', '
20M', '10,000,000+', 'Free', '0', 'Teen', 'Art & Design', 'April 2, 20
18', '3.8', '4.1 and up']
['Mandala Coloring Book', 'ART_AND_DESIGN', '4.6', '4326', '21M', '100
,000+', 'Free', '0', 'Everyone', 'Art & Design', 'June 26, 2018', '1.0
.4', '4.4 and up']
['3D Color Pixel by Number - Sandbox Art Coloring', 'ART_AND_DESIGN',
'4.4', '1518', '37M', '100,000+', 'Free', '0', 'Everyone', 'Art & Desi
gn', 'August 3, 2018', '1.2.3', '2.3 and up']
['Learn To Draw Kawaii Characters', 'ART_AND_DESIGN', '3.2', '55', '2.
7M', '5,000+', 'Free', '0', 'Everyone', 'Art & Design', 'June 6, 2018'
, 'NaN', '4.2 and up']
Number of rows: 9659
Number of columns: 13
```

The length of the cleaned android data matches our expected result from the begining of part 2.

# **Removing Non English Apps**

#### Part 1:

Certain app developers in the US are focused on targeting an English-speaking audience. I will continue to clean the data by searching for apps with non-English names.

In order to find all non-English names, I will isolate names with ASCII characters that do not fall in the normal English text range of 0-127.

```
In [18]: def is_english(string):
    for letter in string:
        if ord(letter) > 127:
            return False
    return True
In [19]: #testing function
    print(is_english('Instagram'))
    print(is_english('ğ奇艺PPS - 《欢乐颂2》电视剧热播'))
    print(is_english('bocs To Go™ Free Office Suite'))
    #print(is_english(Instachat ♀))
    True
    False
    False
In [20]: is_english(Instachat ♀)
    File "<ipython-input-20-b89d0ddafab6>", line 1
```

is\_english(Instachat 😜)

This function seems to work well for the first two strings with the most common English and non-English characters. I will further investigate the last two strings to improve the is\_english function.

```
In [21]: print(ord('<sup>m'</sup>))
print(ord('<sup>©'</sup>))

8482
128540
```

### Part 2:

To improve the is\_english function, I will alter the function to only return false if the string has more than 3 characters that fall outside of the ASCII English range. Adding this adjustment will make sure we do not lose valuable data due to a couple characters outside of the normal ASCII range.

```
In [22]: def is_english(string):
    n=0
    for letter in string:
        if ord(letter) > 127:
            n+=1
    if n>3:
        return False
    else:
        return True

print(is_english('Instagram'))
print(is_english('ğ奇艺PPS - 《欢乐颂2》电视剧热播'))
print(is_english('gocs To Go<sup>m</sup> Free Office Suite'))
print(is_english('Instachat ♀'))
True
```

False True True

This function is not 100% efficient because it may filter out English apps that have more than 3 characters that fall outside of the normal ASCII English range. However, we will still be left with the majority of the English app data.

I will run the is\_english function of the android\_clean and the ios\_body data to continue the data filtering process.

In [23]: android\_english=[]

for app in android\_clean:
 name=app[0]
 if is\_english(name):
 android\_english.append(app)

explore\_data(android\_english, 0, 3, True)

['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']

['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', '5 0,000,000+', 'Free', '0', 'Teen', 'Art & Design', 'June 8, 2018', 'Var ies with device', '4.2 and up']

Number of rows: 9614 Number of columns: 13

```
In [24]: ios_english=[]
for app in ios_body:
    name=app[1]
    if is_english(name):
        ios_english.append(app)
explore_data(ios_english, 0, 3, True)
    ['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']
    ['529479190', 'Clash of Clans', '116476928', 'USD', '0.0', '2130805',
    '579', '4.5', '4.5', '9.24.12', '9+', 'Games', '38', '5', '18', '1']
    Number of rows: 6183
    Number of columns: 16
```

### **Isolating the Free Apps**

The app developers for whom this analysis will pertain are only interested in the free applications. I will finish cleaning the android and ios data sets by deleting all apps which are not free for download.

```
In [25]: #checking price indices
            print('iOS Header :', ios_header) #index 4
            print('\n')
            print('Android Header: ', android_header) #index 7
             iOS Header : ['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_url
             s.num', 'lang.num', 'vpp_lic']
             Android Header: ['App', 'Category', 'Rating', 'Reviews', 'Size', 'Ins talls', 'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver', 'Android Ver']
In [26]: android_final=[]
            ios_final=[]
            for app in android english:
                 price= app[7]
                 if price == '0':
                      android_final.append(app)
            for app in ios_english:
                price= app[4]
                 if price == '0.0':
                      ios final.append(app)
            print('Android Final Length: ', len(android_final))
            print('iOS Final Length: ', len(ios_final))
```

```
Android Final Length: 8864
iOS Final Length: 3222
```

After completing the data cleaning process, I am left with 8864 Android apps and 3222 iOS apps. This is still a large enough sample to continue with the analysis.

# Most Common Apps By Genre

### Part 1:

Our goal in this project is to determine which types of english, free applications attract the most users. The more users for the applications means more revenue.

In order to minimize risk, we want to create applications with the highest user ratings and apps that are succesful in both the Google Play and App Store markets.

Our validation srategy is as follows:

 Build a minimal Android version of the app, and add it to Goog le Play.
 If the app has a good response from users, we develop it furth er.
 If the app is profitable after six months, we build an iOS ver sion of the app and add it to the App Stor

I will begin the analysis by finding the most popular genres for both applications. I will take count of these genres using a frequency table.

```
In [27]: #finding genre indices
print('iOS Header :', ios_header) #index 11 or -5 and
print('\n')
print('Android Header: ', android_header) #index 1 and -4
```

iOS Header : ['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\_url
s.num', 'lang.num', 'vpp\_lic']

Android Header: ['App', 'Category', 'Rating', 'Reviews', 'Size', 'Ins talls', 'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver', 'Android Ver']

### Part 2:

Creating two functions to build and analyze the genre frequency tables

```
In [43]: def freq_table(dataset, index):
             table = {}
             total = 0
             for row in dataset:
                 total += 1
                 value = row[index]
                 if value in table:
                     table[value] += 1
                 else:
                     table[value] = 1
             table_percentages = {}
             for key in table:
                 percentage = (table[key] / total) * 100
                 table_percentages[key] = percentage
             return table_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])
```

I will now generate a frequency table for the columns 'prime\_genre', 'genre' and 'category'.

In

[29]:	#prime_genres
	<pre>display_table(ios_final, 11)</pre>
	Games : 58.16263190564867
	Entertainment : 7.883302296710118
	Photo & Video : 4.9658597144630665
	Education : 3.662321539416512
	Social Networking : 3.2898820608317814
	Shopping : 2.60707635009311
	Utilities : 2.5139664804469275
	Sports : 2.1415270018621975
	Music : 2.0484171322160147
	Health & Fitness : 2.0173805090006205
	Productivity : 1.7380509000620732
	Lifestyle : 1.5828677839851024
	News : 1.3345747982619491
	Travel : 1.2414649286157666
	Finance : 1.1173184357541899
	Weather : 0.8690254500310366
	Food & Drink : 0.8069522036002483
	Reference : 0.5586592178770949
	Business : 0.5276225946617008
	Book : 0.4345127250155183
	Navigation : 0.186219739292365
	Medical : 0.186219739292365
	Catalogs : 0.12414649286157665

When analyzing the 'prime\_genre' column of the App store data set, we see that the most common genres for free apps are 'Games' and 'Entertainment'. Our frequency table shows that the Games category takes up over half of all apps (~58.16%). Entertainement category makes up about -7.8% of the apps. The runner-up genres would include 'Photo and Video' and 'Education'.

The general pattern that emerges from this from this frequency table is that most of the free apps available in the iOS app store are designed for entertainment rather than utility.

Before recommending an app based on a frequncy table I would like to point out that even though gaming apps are very common, this does not garuntee the popularity of this genre.

In [30]: #Category

display\_table(android\_final, 1)

```
FAMILY : 18.907942238267147

GAME : 9.724729241877256

TOOLS : 8.461191335740072

BUSINESS : 4.591606498194946

LIFESTYLE : 3.9034296028880866

PRODUCTIVITY : 3.892148014440433

FINANCE : 3.7003610108303246

MEDICAL : 3.531137184115524

SPORTS : 3.395758122743682

PERSONALIZATION : 3.3167870036101084

COMMUNICATION : 3.2378158844765346

HEALTH_AND_FITNESS : 3.0798736462093865
```

PHOTOGRAPHY : 2.944494584837545 NEWS AND MAGAZINES : 2.7978339350180503 SOCIAL : 2.6624548736462095 TRAVEL AND LOCAL : 2.33528880866426 SHOPPING : 2.2450361010830324 BOOKS AND REFERENCE : 2.1435018050541514 DATING : 1.861462093862816 VIDEO PLAYERS : 1.7937725631768955 MAPS AND NAVIGATION : 1.3989169675090252 FOOD\_AND\_DRINK : 1.2409747292418771 EDUCATION : 1.1620036101083033 ENTERTAINMENT : 0.9589350180505415 LIBRARIES AND DEMO : 0.9363718411552346 AUTO AND VEHICLES : 0.9250902527075812 HOUSE AND HOME : 0.8235559566787004 WEATHER : 0.8009927797833934 EVENTS : 0.7107400722021661 PARENTING : 0.6543321299638989 ART AND DESIGN : 0.6430505415162455 COMICS : 0.6204873646209386 BEAUTY : 0.5979241877256317

The top category for free android apps is 'Family' which makes up 18.9% of the free android apps available. The second most frequent category is 'Game' which makes up 9.72% of available apps. The next three categories include Tools, Business, and Lifestyle.

The general pattern seen here is that the free android apps available seem to target families and are oriented towards boosting productivity.

#### In [31]: #genres

```
display table(android final, -4)
Tools : 8.449909747292418
Entertainment : 6.069494584837545
Education : 5.347472924187725
Business : 4.591606498194946
Productivity : 3.892148014440433
Lifestyle : 3.892148014440433
Finance : 3.7003610108303246
Medical : 3.531137184115524
Sports : 3.463447653429603
Personalization : 3.3167870036101084
Communication : 3.2378158844765346
Action : 3.1024368231046933
Health & Fitness : 3.0798736462093865
Photography : 2.944494584837545
News & Magazines : 2.7978339350180503
Social : 2.6624548736462095
Travel & Local : 2.3240072202166067
Shopping : 2.2450361010830324
Books & Reference : 2.1435018050541514
```

When analyzing the most frequent genres in the Google Play Market for free apps, Tools and Enterntainment make up the most available genres. Following these two genres are Education, Business, and Productivity.

This data reinforces that the most common apps available in the Google Play market are made for utility purposes.

As stated before, It is important to understand that this data only displays the frequency of available apps for each category or genre. The amount of available apps for each genre does not

reflect the popularity or amount of users each grenre attracts.

The frequency tables generated show that the Apple Store has more apps designed for entertainement while the Google Play Market has more applications designed for utility.

In order to make a reccomendation to app developers, I will need to find which genre of apps are most popular in each market.

. .

### Most Popular Apps by Genre on the App Store

To find the most popular genres, it is important to calculate the average number of user ratings per app genre. First, I will Isolate the apps of each genre. Then, I will sum up the user ratings for the apps of that genre and divide the sum by the number of apps belonging to that genre.

```
In [40]: prime_genre=freq_table(ios_final, 11)
```

```
for genre in prime_genre:
    total=0
    len_genre=0
    for app in ios_final:
        genre_app=app[11]
        if genre_app=genre:
            rating= float(app[5])
            total+=rating
            len_genre+=1
        avg_rating=total/len_genre
        print(genre,': ', avg_rating)
```

```
Medical : 612.0
Weather : 52279.892857142855
Finance : 31467.94444444445
Photo & Video : 28441.54375
Social Networking : 71548.34905660378
Lifestyle : 16485.764705882353
Utilities : 18684.456790123455
Food & Drink : 33333.92307692308
Health & Fitness : 23298.015384615384
Education : 7003.983050847458
Business : 7491.117647058823
Shopping : 26919.690476190477
Reference : 74942.1111111111
Games : 22788.6696905016
Productivity : 21028.410714285714
Music : 57326.530303030304
Travel : 28243.8
Navigation : 86090.33333333333
Book : 39758.5
Sports : 23008.898550724636
Catalogs : 4004.0
News : 21248.023255813954
Entertainment : 14029.830708661417
```

The table above displays the most popular app genre for iOS devices are Navigation and Social Networking. I would like to further investigate this to see which apps dominate in these genres and the number of ratings each of these has.

```
In [38]: for app in ios_final:
    if app[11] == 'Navigation':
        print(app[1], ':', app[5]) # print name and number of ratings
Waze - GPS Navigation, Maps & Real-time Traffic : 345046
Google Maps - Navigation & Transit : 154911
Geocaching® : 12811
CoPilot GPS - Car Navigation & Offline Maps : 3582
ImmobilienScout24: Real Estate Search in Germany : 187
Railway Route Search : 5
```

Further analyzing the number of ratings each app in the navigation genre allows us to see that most people use either Waze or Google Maps. Therefore, it may not be smart to try to compete with these apps.

To my knoweldge, Social Networking will also follow this pattern. Social Networking apps are dominated by the big companies like Facebook, Instagram, and Twitter.

I will attempt to find a popular genre whos user ratings are about equal for each app. This gives us a higher chance to succeed in that genre of apps.

I see that the Reference genre has the next average number of highest ratings (~74,942)

```
ds Installer Tools : 4693
          GUNS MODS for Minecraft PC Edition - Mods Tools : 1497
          Guides for Pokémon GO - Pokemon GO News and Cheats : 826
          WWDC : 762
          Horror Maps for Minecraft PE - Download The Scariest Maps for Minecraf
          t Pocket Edition (MCPE) Free : 718
          VPN Express : 14
          Real Bike Traffic Rider Virtual Reality Glasses : 8
          教えて!goo : 0
          Jishokun-Japanese English Dictionary & Translator : 0
In [42]: for app in ios_final:
             if app[11] == 'Book':
                 print(app[1], ':', app[5]) # print name and number of ratings
          Kindle - Read eBooks, Magazines & Textbooks : 252076
          Audible - audio books, original series & podcasts : 105274
          Color Therapy Adult Coloring Book for Adults : 84062
          OverDrive - Library eBooks and Audiobooks : 65450
          HOOKED - Chat Stories : 47829
          BookShout: Read eBooks & Track Your Reading Goals : 879
          Dr. Seuss Treasury - 50 best kids books : 451
          Green Riding Hood : 392
          Weirdwood Manor : 197
          MangaZERO - comic reader : 9
          ikouhoushi : 0
          MangaTiara - love comic reader : 0
          謎解き:0
          謎解き2016 : 0
```

The Reference and Book Category both are among the most popular of genres and have a more balanced number of ratings amongst the different apps in those categories.

I suggest creating an app which lies within both of these categories. For example, we could create a book application which is also linked to games or interactive activities. These apps could also have dictionaries linked within the app as well as history background available for reference. We could interact news and fun games to keep the readers up to date on relevant information and make the app more than just a place to read books.

By combinging other popular app genres into the Book/Refernce category, the app will attract more viewers and in turn, create more revenue.

This idea seems to fit with the overall theme of iOS apps because it is also interactive and entertaining.

### Most Popular Apps by Genre on Google Play

Unlike the ios\_final data, the android\_final data contains information on the number of installs each genre has. In order to understand which genres are the most popular in the Google Play market, I will compute the average number of installs per genre.

```
In [44]: display_table(android_final, 5) #Installs column
```

```
1,000,000+ : 15.726534296028879
100,000+ : 11.552346570397113
10,000,000+ : 10.548285198555957
10,000+ : 10.198555956678701
1,000+ : 8.393501805054152
100+ : 6.915613718411552
```

5,000,000+ : 6.825361010830325 500,000+ : 5.561823104693141 50,000+ : 4.7721119133574 5,000+ : 4.512635379061372 10+ : 3.5424187725631766 500+ : 3.2490974729241873 50,000,000+ : 2.3014440433213 100,000,000+ : 2.1322202166064983 50+ : 1.917870036101083 5+ : 0.78971119133574 1+ : 0.5076714801444043 500,000,000+ : 0.2707581227436823 1,000,000,000+ : 0.22563176895306858 0+ : 0.04512635379061372 0 : 0.01128158844765343

```
In [72]: category_freq =freq_table(android_final, 1) #Category Column
for category in category_freq:
    tot_installs=0
    len_category=0
    for app in android_final:
        category_app=app[1]
        if category=category_app:
            n_installs= app[5]
            n_installs=n_installs.replace('+', '')
            n_installs=n_installs.replace(',', '')
            len_category +=1
            tot_installs=tot_installs/len_category
        if avg_installs >= 900000:
            print(category, ':', avg_installs)
```

TOOLS : 10801391.298666667 HEALTH AND FITNESS : 4188821.9853479853 SOCIAL : 23253652.127118643 BUSINESS : 1712290.1474201474 VIDEO PLAYERS : 24727872.452830188 FINANCE : 1387692.475609756 TRAVEL AND LOCAL : 13984077.710144928 BOOKS AND REFERENCE : 8767811.894736841 SHOPPING : 7036877.311557789 LIFESTYLE : 1437816.2687861272 PHOTOGRAPHY : 17840110.40229885 NEWS AND MAGAZINES : 9549178.467741935 ART\_AND\_DESIGN : 1986335.0877192982 EDUCATION : 1833495.145631068 ENTERTAINMENT : 11640705.88235294 FOOD AND DRINK : 1924897.7363636363 HOUSE AND HOME : 1331540.5616438356 WEATHER : 5074486.197183099 MAPS\_AND\_NAVIGATION : 4056941.7741935486 PERSONALIZATION : 5201482.6122448975 SPORTS : 3638640.1428571427 PRODUCTIVITY : 16787331.344927534 GAME : 15588015.603248259 FAMILY : 3695641.8198090694 COMMUNICATION : 38456119.167247385

We know that the Google Play market is oriented towards families and utility. The information above shows that the top downloaded apps are: Social, Video Players, Photography, Productivity, Game, Communication. However, the Books and References category is also high on the list.

```
In [70]: for app in android_final:
             if app[1] == 'PRODUCTIVITY' and (app[5] == '1,000,000,000+'
                                                      or app[5] == '500,000,000+'
                                                      or app[5] == '100,000,000+')
                 print(app[0], ':', app[5])
          Microsoft Word : 500,000,000+
          Microsoft Outlook : 100,000,000+
          Microsoft OneDrive : 100,000,000+
          Microsoft OneNote : 100,000,000+
          Google Keep : 100,000,000+
          ES File Explorer File Manager : 100,000,000+
          Dropbox : 500,000,000+
          Google Docs : 100,000,000+
          Microsoft PowerPoint : 100,000,000+
          Samsung Notes : 100,000,000+
          SwiftKey Keyboard : 100,000,000+
          Google Drive : 1,000,000,000+
          Adobe Acrobat Reader : 100,000,000+
          Google Sheets : 100,000,000+
          Microsoft Excel : 100,000,000+
          WPS Office - Word, Docs, PDF, Note, Slide & Sheet : 100,000,000+
          Google Slides : 100,000,000+
          ColorNote Notepad Notes : 100,000,000+
          Evernote - Organizer, Planner for Notes & Memos : 100,000,000+
          Google Calendar : 500,000,000+
          Cloud Print : 500,000,000+
          CamScanner - Phone PDF Creator : 100,000,000+
In [55]: for app in android_final:
             if app[1] == 'BOOKS_AND_REFERENCE' and (app[5] == '1,000,000+'
                                                      or app[5] == '5,000,000+'
                                                      or app[5] == '10,000,000+'
                                                      or app[5] == '50,000,000+'):
                 print(app[0], ':', app[5])
          Wikipedia : 10,000,000+
          Cool Reader : 10,000,000+
          Book store : 1,000,000+
          FBReader: Favorite Book Reader : 10,000,000+
          Free Books - Spirit Fanfiction and Stories : 1,000,000+
          AlReader -any text book reader : 5,000,000+
          FamilySearch Tree : 1,000,000+
          Cloud of Books : 1,000,000+
          ReadEra - free ebook reader : 1,000,000+
          Ebook Reader : 5,000,000+
          Read books online : 5,000,000+
          eBoox: book reader fb2 epub zip : 1,000,000+
          All Maths Formulas : 1,000,000+
          Ancestry : 5,000,000+
          HTC Help : 10,000,000+
          Moon+ Reader : 10,000,000+
          English-Myanmar Dictionary : 1,000,000+
          Golden Dictionary (EN-AR) : 1,000,000+
          All Language Translator Free : 1,000,000+
          Aldiko Book Reader : 10,000,000+
          Dictionary - WordWeb : 5,000,000+
          50000 Free eBooks & Free AudioBooks : 5,000,000+
          Al-Quran (Free) : 10,000,000+
```

....

Al Quran Indonesia : 10,000,000+

Al'Quran Bahasa Indonesia : 10,000,000+ Al Quran Al karim : 1,000,000+ Al Quran : EAlim - Translations & MP3 Offline : 5,000,000+ Koran Read &MP3 30 Juz Offline : 1,000,000+ Hafizi Quran 15 lines per page : 1,000,000+ Quran for Android : 10,000,000+ Satellite AR : 1,000,000+ Oxford A-Z of English Usage : 1,000,000+ Dictionary.com: Find Definitions for English Words : 10,000,000+ English Dictionary - Offline : 10,000,000+ Bible KJV : 5,000,000+ NOOK: Read eBooks & Magazines : 10,000,000+ Brilliant Quotes: Life, Love, Family & Motivation : 1,000,000+ Stats Royale for Clash Royale : 1,000,000+ Dictionary : 10,000,000+ wikiHow: how to do anything : 1,000,000+ EGW Writings : 1,000,000+ My Little Pony AR Guide : 1,000,000+ Spanish English Translator : 10,000,000+ Dictionary - Merriam-Webster : 10,000,000+ JW Library : 10,000,000+ Oxford Dictionary of English : Free : 10,000,000+ English Hindi Dictionary : 10,000,000+ English to Hindi Dictionary : 5,000,000+

The data above shows that there are a lot of e-reading apps that have been downloaded. We know that the Books and References category, with 8,767,811.89 downloads, does have potential.

In order to stand out from other apps we have to make a Book and References app which has unique features that we know android users are attracted too. We know that android users like to download entertaining apps (11,640,705.88 downloads) as well as Gaming Apps (15,588,015.60 dowloads). This data suggests adding games, quizes, or other entertaining features will create an overlapp between the Entertainment and Book categories.

# **Conclusion:**

In this project I analyzed data for free, english apps for the Apple Store and the Google Play market. The goal in this analysis was to determine which app genres are most popular and would succeed in both markets.

It is important that we create apps in a category which is not dominated by Big Name apps. This way, we can minimize risk and have an opportunity to grow.

I determined that creating Book Apps with interesting features could be an optimal choice. Some features could include: games, quizess, updates and news, and community chat centers. These unique features are categories in which the data shows users in both markets are most attracted to. Creating an app which falls in the Book category yet overlapps with key components which we know users are attracted too shows the most potential.

#### In [ ]: