

Application Analysis to Determine a Profitable New Application for both Google Play and Apple App Stores

Author : Kenneth Lucas Kusima
Contributor: DataQuest

1. Project Origin:

This project was made per instructions from:

The DataQuest Course: Python for Data Science Fundamentals

2. Objective:

For this project, I pretend I am working as a data analyst for a company that builds Android and iOS mobile applications (apps). They make apps available both on Google Play and the App Store.

The company only build apps that are free to download and install, and the main source of revenue consists of in-app ads. This means that different companies can pay to have their ads appear in various parts of the application. This means that revenue for any given app is mostly influenced by the number of users who use the app — the more users that see and engage with the ads, the better. **My goal for this project is to analyze application data and assist the company developers understand what type of apps are likely to attract more users.**

After identifying and analysing the most attractive application types, the company hopes to follow the strategy outlined in the following steps:

- Use results from the analysis to generate a minimal Android version of the "perfect" app and add it to the Google Play Store.
- Assess and monitor the application's user response and further develop the app if good ratings are received.
- After six months, if substantial profit is seen, build an iOS version of the app and add it to the Apple App Store.

3. Dataset Definition:

According to Statista, in 2018, there were approximately 2 million iOS apps on the App Store and 2.1 million Android apps on the Google Play Store. Analysing data from more than 4 million apps across the Google Play and iOS App Stores. As a result, DataQuest offers data sets that encapsulate a select group of applications to which significant analysis can be conducted. Android apps data set holds data on approximately 10,000 Android apps collected in August 2018 from Google Play. Moreover, the data set for iOS Apps contains about 7,000 iOS apps collected in July 2017 from the App Store.

Table 1: Summary of Datasets to be used

Dataset	Location	Collection Date	Number of Apps
Android apps	Google Play Store	August 2018	10,841
iOS apps	Apple App Store	July 2017	7,197

4. Exploring the Datasets:

In [1]:

```
from csv import reader

#Apple Store Dataset
opened_file_1 = open('AppleStore.csv')
read_file_1 = reader(opened_file_1)
ios_apps_data = list(read_file_1)

#Google Play Store Dataset
opened_file_2 = open('googleplaystore.csv')
read_file_2 = reader(opened_file_2)
android_apps_data = list(read_file_2)

#Function for exploring the datasets inputed
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]))
        print('\n')

#Exploring the Apple Store Dataset:
print('The First 2 rows of iOS App dataset: \n')
explore_data(ios_apps_data,0,2,True)

#Exploring the Google Play Store Dataset:
print('The First 2 rows of Android dataset: \n')
explore_data(android_apps_data,0,2,True)
```

Number of rows: 7198
Number of columns: 16

The First 2 rows of Android dataset:

```
['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+', 'Free', '0', 'Everyone', 'Art & Desig
n', 'January 7, 2018', '1.0.0', '4.0.3 and up']
```

Number of rows: 10842
Number of columns: 13

In obtaining the first few rows of the two datasets, as see in the results above, a deeper understanding of the specific data entry types can be obtained . This can then be used to provide the basis and direction of our

analysis. Looking at the column headers allows us to pick the appropriate data entries to focus on in comparing, contrasting and discovering trends between the two datasets. A full view of the data recorded in the two datasets can be found in the links below:

Apple App Store (iOS Apps) : <https://dq-content.s3.amazonaws.com/350/AppleStore.csv> (<https://dq-content.s3.amazonaws.com/350/AppleStore.csv>)

Google Play Store (Android Apps) : <https://dq-content.s3.amazonaws.com/350/googleplaystore.csv> (<https://dq-content.s3.amazonaws.com/350/googleplaystore.csv>)

4.1 Column Header Analysis:

In looking at the column headers of the two data sets, similar columns that appear beneficial for different future analysis may be summarised:

Table 2: List of significant column headers guiding the type of data analysis to follow

iOS Apps headers	Column	Android Apps headers	Column
Track name	2	App	1
size_bytes	3	Size	2
price	5	Price	8
cont_rating	11	Content Rating	9
prime_genre	12	Genres	10
user_rating	8	Rating	3

5. Data Cleaning:

In order to provide accurate results, accurate data is needed. Consequently, data cleaning conducted before analysis provides assurance that results obtained would be free of human errors such as duplicates, typos and irrelevant data. The cleaning is done by removing or correcting wrong data, removing duplicate data and modifying the data so as to fit the purpose of the analysis.

5.1 Detecting and Deleting Wrong Data:

The google play dataset has been reported to have errors in a certain row. The entry (row 10,472) has been addressed to have "missing 'Rating' and a column shift happened for next columns". Since there are 10,842 apps, deleting this erroneous application entry would not provide much issues.

In [2]:

```
#Inspecting the suspicious rows
for row in android_apps_data[10472:10475]:
    print(row)
    print('\n') # adds a new (empty) line after each row
```

```
['Xposed Wi-Fi-Pwd', 'PERSONALIZATION', '3.5', '1042', '404k', '100,000+', 'Free', '0', 'Everyone', 'Personalization', 'August 5, 2014', '3.0.0', '4.0.3 and up']

['Life Made WI-Fi Touchscreen Photo Frame', '1.9', '19', '3.0M', '1,000+', 'Free', '0', 'Everyone', '', 'February 11, 2018', '1.0.19', '4.0 and up']

['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']
```

It is evident that the entry for the Android application **'Life Made WI-Fi Touchscreen Photo Frame'** (i.e. the row corresponding to `android_apps_data[10473]`) is erroneous. There appears to be missing data points. This application entry is therefore deleted as follows.

In [3]:

```
#Deleting the erroneous row
del android_apps_data[10473]

#Validation
for row in android_apps_data[10472:10475]:
    print(row)
    print('\n') # adds a new (empty) line after each row
```

```
['Xposed Wi-Fi-Pwd', 'PERSONALIZATION', '3.5', '1042', '404k', '100,000+', 'Free', '0', 'Everyone', 'Personalization', 'August 5, 2014', '3.0.0', '4.0.3 and up']

['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']

['Sat-Fi Voice', 'COMMUNICATION', '3.4', '37', '14M', '1,000+', 'Free', '0', 'Everyone', 'Communication', 'November 21, 2014', '2.2.1.5', '2.2 and up']
```

5.2 Clearing Out Duplicates:

Duplicates may result in an overestimation or underestimation of results. This is therefore need to investigate and correct for any repeated entries.

5.2.1 Explorinng the Duplicated Andorid Apps:

In [4]:

```
#Checking for duplicates in android dataset:
duplicate_apps_android = [] #Empty list for duplicate apps
unique_apps_android = [] #Empty list for unique apps

for app_android in android_apps_data: #Extracting rows from the android dataset
    name_android = app_android[0] #Isolating based on similar App name (i.e. name)
    if name_android in unique_apps_android: #Keeping track of duplicates
        duplicate_apps_android.append(name_android)
    else:
        unique_apps_android.append(name_android) #Isolating Unique entries

print('Number of duplicate android apps:', len(duplicate_apps_android))
print('\n')
print('Examples of duplicate android apps:', duplicate_apps_android[:10] )
```

Number of duplicate android apps: 1181

Examples of duplicate android 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']

In [5]:

```
#Exploring a few of the duplicates from the android dataset
print('Few duplicate data entrys in the android dataset: \n')

for app_android in android_apps_data:
    name_android = app_android[0]
    if name_android == 'Box':
        print(app_android)
    if name_android == 'Instagram':
        print(app_android)
```

Few duplicate data entrys in the android dataset:

```
['Box', 'BUSINESS', '4.2', '159872', 'Varies with device', '10,000,000+', 'Free', '0', 'Everyone', 'Business', 'July 31, 2018', 'Varies with device', 'Varies with device']
['Box', 'BUSINESS', '4.2', '159872', 'Varies with device', '10,000,000+', 'Free', '0', 'Everyone', 'Business', 'July 31, 2018', 'Varies with device', 'Varies with device']
['Box', 'BUSINESS', '4.2', '159872', 'Varies with device', '10,000,000+', 'Free', '0', 'Everyone', 'Business', 'July 31, 2018', 'Varies with device', 'Varies with device']
['Instagram', 'SOCIAL', '4.5', '66577313', 'Varies with device', '1,000,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies with device', 'Varies with device']
['Instagram', 'SOCIAL', '4.5', '66577446', 'Varies with device', '1,000,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies with device', 'Varies with device']
['Instagram', 'SOCIAL', '4.5', '66577313', 'Varies with device', '1,000,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies with device', 'Varies with device']
['Instagram', 'SOCIAL', '4.5', '66509917', 'Varies with device', '1,000,000,000+', 'Free', '0', 'Teen', 'Social', 'July 31, 2018', 'Varies with device', 'Varies with device']
```

The presence of 1181 duplicates pose a significant threat to the accuracy of further analysis. These duplicates need to be removed and this is done in the sections further below.

5.2.2 Explorinng the Duplicated iOS Apps:

Similarly for the iOS app dataset,

In [6]:

```

in iOS dataset:
    #Empty list for duplicate apps
    #Empty list for unique apps

data[1:]:          #Extracting rows from the iOS dataset
                  #Isolating based on similar App name (i.e column 0 from dataset)
duplicate_apps_ios: #Keeping track of duplicates
duplicate_apps_ios.append(name_ios)

duplicate_apps_ios.append(name_ios) #Isolating Unique entries

print('Number of duplicate iOS apps:', len(duplicate_apps_ios))

print('Examples of duplicate iOS apps:', duplicate_apps_ios[:40] )

```

Number of duplicate iOS apps: 0

Examples of duplicate iOS apps: []

There appears to be no duplicate apps in the iOS app dataset.

5.2.3 Removing the Duplicated Andorid Apps:

To remove duplicated data entries in the two application datasets, there needs to be a criteria to which only one of the entries is chosen out of the different duplicates. For the duplicated andorid apps, it can be found that a main difference among the duplicates is the number of reviews (the fourth column). They indicate that the duplicates were recorded at different times where the ones with the most reviews happen to be the most up to date entries. Consequently, to remove the android app duplicates, instead of randomly choosing applications to remove, focus is made to keep only the most up to date entry duplicate, that is, the (duplicated) row with the highest number of reviews.

In [7]:

```

reviews_max = {} #Dictionary to contain unique apps and the corresponding (max) number of reviews
for rows in android_apps_data[1:]:
    name = rows[0] #Isolating the name of every row entry
    n_reviews = float(rows[3]) #Isolating the reviews of the app entry

    if name in reviews_max and reviews_max[name] < n_reviews:
        reviews_max[name] = n_reviews

    if name not in reviews_max:
        reviews_max[name] = n_reviews

```

To validate that the duplicated entries have been removed, the inspection below can be done:

In [8]:

```
print('Expected length:', len(android_apps_data[1:]) - 1181)
print('Actual length:', len(reviews_max))
```

Expected length: 9659

Actual length: 9659

5.2.3.1 Isolating the non-duplicated android apps:

The dictionary : **'reviews_max'** now holds both the non-duplicated android applications and the corresponding ratings. This can then be used to create a new dataset of android apps with non_duplicates, named, **'android_apps_data_clean'** . Another list (**'already_added_android'**) can also be made to contain the names of all the apps in this new dataset.

In [9]:

```
clean = []
id = []

android_apps_data[1:]:

for row in android_apps_data[1:]:
    if (row['name'] == n_reviews) and (name not in already_added_android):
        android_apps_data_clean.append(row)
        already_added_android.append(name)

display(android_apps_data_clean, 0, 3, True) #Showing the final duplicate free android app dataset
```

```
['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', '50,000,000+',
 'Free', '0', 'Teen', 'Art & Design', 'June 8, 2018', 'Varies with device', '4.2 and up']
```

Number of rows: 9659

Number of columns: 13

5.3. Removing Non-English Apps

Our Company only uses English for the apps being developed therefore it is in our interest to remove any non-English applications in the datasets. The apps are filtered to only include those directed for an English-speaking audience.

The English language utilises characters which correspond to numbers that fall in the range 0 to 127 (i.e the ASCII range) . The corresponding number assigned to the different character can be found using the function 'ord()' . Hence, in order to distinguish the english applications, the function below filters the string input and checks if each character belongs in the ASCII range. All other characters and symbols (including chinese characters and emojis) fall outside of this range. As a result, to account for those english apps that have few non-english characters in their name, a maximum of 3 non-english characters is allowed in the characterisation of english applications. This means that any application with more than 3 characters not in the ASCII range will be considered non-English

In [10]:

```
def English_app(string):
    count = 0
    for char in string:
        val = (ord(char))
        if val > 127:
            count +=1
            if count > 3 :
                return False

    return True
```

5.3.1 Removing Non-English Andorid Apps:

In [11]:

```
android_apps_english_data = []
for row in android_apps_data_clean:
    name = row[0]
    if English_app(name):
        android_apps_english_data.append(row)

explore_data(android_apps_english_data,0,3)
print('Number of English Android Apps:',len(android_apps_english_data))
```

```
['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', '50,000,000+',
'Free', '0', 'Teen', 'Art & Design', 'June 8, 2018', 'Varies with device', '4.2 and up']
```

Number of English Android Apps: 9614

5.3.2 Removing Non-English iOS Apps:

In [12]:

```

iOS_apps_english_data = []
for row in ios_apps_data[1:]:
    name = row[1]
    if English_app(name):
        iOS_apps_english_data.append(row)

explore_data(iOS_apps_english_data,0,3)
print('Number of English iOS Apps:',len(iOS_apps_english_data))

```

```

['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', '128
9', '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 English iOS Apps: 6183

5.4. Separating the Free apps

Since the company's source of revenue is through in-app purchases, the applications developed are free applications. It is therefore critical to filter the datasets to include only free apps.

In [13]:

```

Android_dataset = []
iOS_dataset = []

for rows in android_apps_english_data:
    price = rows[7]
    if price == '0':
        Android_dataset.append(rows)

for rows in iOS_apps_english_data:
    prices = rows[4]
    if prices == '0.0':
        iOS_dataset.append(rows)

print('Actual Number of Android apps being developed:', len(Android_dataset))
print('Actual Number of iOS apps being developed:',len(iOS_dataset))

```

Actual Number of Android apps being developed: 8864

Actual Number of iOS apps being developed: 3222

6. Data Analysis:

Since the end goal is to find the most attractive app and add it to both the Google Play Store and the Apple App Store, the app profile being looked for should be profitable in both markets.

To start the analysis, we will begin by investigating what the most common genres for each market are. To do so, frequency tables are generated to see which the most prominent genres were. For android apps, the genres are listed in 'Genre' and 'Category' columns in the Android datasets and for iOS apps, the genres are listed in the 'prime_genre' columns in the iOS App Store datasets.

6.1 Application Frequency by Genre :

Below is the analysis of the most common free English applications in the Android Google Play and Apple App Stores. To do this, two functions are implemented:

- **Function 1: 'freq_table'** : To create genre based frequency tables showing the percentages - **Function 2: 'display_table'** : To reorder the variables and values in the frequency table from Function 1 to show the most common applications in the different app stores in descending order.

In [14]:

```
def freq_table(dataset, index):
    tables = {}
    totaling = 0

    for row in dataset:
        totaling += 1
        values = row[index]

        if values in tables:
            tables[values] += 1
        else:
            tables[values] = 1

    table_percentage = {}

    for keys in tables:
        percents = (tables[keys] / totaling) * 100
        table_percentage[keys] = percents

    return table_percentage

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])
```

6.1.1 Most Common Free English Applications in the iOS datasets by Genre:

Analysing the frequency table for the prime_genre column of the App Store data set.

In [15]:

```
display_table(iOS_dataset, 11) #prime_genre
```

```
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
```

From the frequency tables above, the top five most common free English application genres can be summarised as:

Table 3: List of top five most common free English applications in the Apple App Store.

Common Free English iOS Application	Percentage
Games	58.16%
Entertainment	7.88%
Photo & Video	4.96%
Education	3.66%
Social Networking	3.29%

From the data above it can be seen that the more social and entertaining applications like 'Games, Entertainment and Photo&Video' have a bigger prescence in the Apple Store amongst all the free english applications in the store. Despite there seems to be a larger offering of these application, this does not necessarily imply that these apps have the most demand and the highest number of users.

6.1.2 Most Common Free English Applications in the Android datasets by Genre:

Analysing the frequency table for the 'Genre' and 'Categories' column of the Android Google Play Store data set.

As Per Category Column of the Android Dataset

In [16]:

```
display_table(Android_dataset, 1) # Category
```

```
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
```

Unlike the Apple Store dataset, there seems to be a higher number of utilitarian and practical applications out of all the free English applications in the Android Google Play Store and can be summarised as seen in the table below.

Table 4: List of top five most common free English applications in the Google Play Store as per the Genres column.

Common Free English Android Application	Percentage
FAMILY	18.91%
GAME	9.72%
TOOLS	8.46%
BUSINESS	4.59%
LIFESTYLE	3.90%

As Per Genres Column of the Android Dataset

In [17]:

```
display_table(Android_dataset, 9) #Genres
Strategy;Action & Adventure : 0.01128158844765343
Simulation;Education : 0.01128158844765343
Role Playing;Brain Games : 0.01128158844765343
Racing;Pretend Play : 0.01128158844765343
Puzzle;Education : 0.01128158844765343
Parenting;Brain Games : 0.01128158844765343
Music & Audio;Music & Video : 0.01128158844765343
Lifestyle;Pretend Play : 0.01128158844765343
Lifestyle;Education : 0.01128158844765343
Health & Fitness;Education : 0.01128158844765343
Health & Fitness;Action & Adventure : 0.01128158844765343
Entertainment;Education : 0.01128158844765343
Communication;Creativity : 0.01128158844765343
Comics;Creativity : 0.01128158844765343
Casual;Music & Video : 0.01128158844765343
Card;Action & Adventure : 0.01128158844765343
Books & Reference;Education : 0.01128158844765343
Art & Design;Pretend Play : 0.01128158844765343
Art & Design;Action & Adventure : 0.01128158844765343
Arcade;Pretend Play : 0.01128158844765343
```

It can be seen that the 'Category' column has more applications, with more variety in comparison to the 'Genres' column. The relation between the two columns is unclear, however due to the increased variety, the Category column will be used as the column corresponding to the android application genres and then to be compared with the prie_genre column in the Apple App Store. As seen from the Genres column, the most common applications have a wider range, with more of them offering a practical purpose. The top five most common free english applications as per the category column can be summarised by the table below:

Table 5: List of top five most common free English applications in the Google Play Store as per the Category column.

Common Free English Android Application	Percentage of Android Apps
Tools	8.45%
Entertainment	6.07%
Education	5.35%
Business	4.59%
Productivity	3.89%

Similarly, the commonality of the most frequent genres in the Google Play Store do not correspond to the most in-demand/ most popular applications with the highest number of users.

6.2 Popular Applications by Genre :

A good way of determining which of the most in-demand applications is to analyse the number of installs, that is, how many times the applications were installed. The more installs an application has, the more popular it is. Since the data regarding the number of installs of an app is unavailable in the Apple App Store dataset, a work around implemented is to use the number of user ratings as a proxy. Focus then becomes to represent the average number of ratings for each app as a measure of popularity

6.2.1 Most Popular Free English Applications in the iOS datasets by Genre:

Since the App Store datasets has no number of installs recorded, the ratings are used to analyse which application genre are the most popular. To do so, we

- Isolate the apps of each genre.
- Sum up the user ratings for the apps of that genre.
- Divide the sum by the number of apps belonging to that genre (not by the total number of apps).

In [18]:

```
list_iOS = freq_table(iOS_dataset, 11) #prime_genre frequency table
for genre in list_iOS:
    total = 0
    len_genre = 0
    for row in iOS_dataset:
        genre_app = row[11]
        if genre_app == genre:
            rating = float(row[5])
            total += rating
            len_genre += 1

    avg_rating = total/len_genre
    print(genre, ': ', avg_rating)
```

```
Social Networking : 71548.34905660378
Photo & Video : 28441.54375
Games : 22788.6696905016
Music : 57326.530303030304
Reference : 74942.111111111111
Health & Fitness : 23298.015384615384
Weather : 52279.892857142855
Utilities : 18684.456790123455
Travel : 28243.8
Shopping : 26919.690476190477
News : 21248.023255813954
Navigation : 86090.333333333333
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
```

Table 6: List of top five most popular free English applications in the iOS App Store as per the highest average number of ratings.

Order	Popular iOS Application Genre	Average Rating
1	Navigation	86090
2	References	74942
3	Social Networking	71548
4	Music	57326

Order	Popular iOS Application Genre	Average Rating
5	Weather	52280

From data above, the Navigation genre receives the most ratings (86090 ratings) and appears to be the most popular in the iOS application store based on our criteria. This result however may prove misleading. This app is followed by 'References' (74942 ratings), 'Social Networking' (71548 ratings) and 'Music' (57326 ratings) applications. Out of the many applications in each genre, there may be a few applications responsible for the high number of ratings. Further analysis is done to support this:

In [19]:

```

print('Popular apps in the Navigation genre:')
print('Average ratings : 86090 \n')
for app in iOS_dataset:
    if app[-5] == 'Navigation':
        print(app[1], ': ', app[5]) # print name and number of ratings

print('\nPopular apps in the Reference genre:')
print('Average ratings : 74942 \n')
count = 0
for app in iOS_dataset:
    if app[-5] == 'Reference' and count < 6:
        print(app[1], ': ', app[5]) # print name and number of ratings
        count +=1

print('\nPopular apps in the Social Networking genre:')
print('Average ratings : 71548 \n')
count = 0
for app in iOS_dataset:
    if app[-5] == 'Social Networking' and count <6:
        print(app[1], ': ', app[5]) # print name and number of ratings
        count +=1

```

Popular apps in the Navigation genre:
Average ratings : 86090

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

Popular apps in the Reference genre:
Average ratings : 74942

Bible : 985920
Dictionary.com Dictionary & Thesaurus : 200047
Dictionary.com Dictionary & Thesaurus for iPad : 54175
Google Translate : 26786
Muslim Pro: Ramadan 2017 Prayer Times, Azan, Quran : 18418
New Furniture Mods - Pocket Wiki & Game Tools for Minecraft PC Edition : 17588

Popular apps in the Social Networking genre:
Average ratings : 71548

Facebook : 2974676
Pinterest : 1061624
Skype for iPhone : 373519
Messenger : 351466
Tumblr : 334293
WhatsApp Messenger : 287589

By analysing the individual ratings of the more popular applications, we can see that there are few giant apps whose presence dominates the specific genre group and in turn lead to the genre seeming more popular than it really is as per our criteria. Consequently, the average number of ratings seem to be skewed by a few apps which have hundreds of thousands of user ratings, while the other apps may struggle to get past the 10,000 threshold.

From the data above, the application genre that shows a potential for popularity is that which would incorporate the top 5 genres shown in Table 6. Nevertheless, despite their popularity, applications in the genres Navigation and weather are not known to have great retention. Usability of these applications do not foster and advocate for exploration and discovery hence proving less likely to be good hosts of in-app purchases(the model to which our company makes the most profit).

Consequently, an application I would recommend is that rooted under the References genre. An application that would allow users to access different books as well as corresponding video content, in a way that allows for the sharing of thoughts and ideas regarding the books. Furthermore, this application would include cook books and tutorials that foster reading skills and allow for the building of imagination as well as practical skills.

6.2.2 Most Popular Free English Applications in the Android datasets by Genre:

This dataset includes distinguishing number of installs, thus allowing us to know the exact number of people who installed the applications. However, the data entries are mostly estimates (eg. 1,000+) and so data cleaning is done below to assure better precision is achieved. The obtained number of ratings are then cleaned for each application and the average number of ratings is obtained for each application genre.

In [20]:

```
Names = freq_table(Android_dataset, 1)

for row in Names:
    total = 0
    length = 0
    for app in Android_dataset:
        category = app[1]
        if category == row:
            installs = app[5]
            installs = installs.replace(',', '')
            installs = installs.replace('+', '')
            total += float(installs)
            length += 1
    avg_installs = total / length
    print(row, ': ', 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
```

Table 7: List of top eleven most popular free English applications in the Google Play Store as per the highest average number of installs.

Order	Popular Android Application Genre	Average Number of Installations
1	COMMUNICATION	38,456,119
2	VIDEO_PLAYERS	24,727,872

Order	Popular Android Application Genre	Average Number of Installations
3	SOCIAL	23,253,652
4	PHOTOGRAPHY	17,840,110
5	PRODUCTIVITY	16,787,331
6	GAME	15,588,016
7	TRAVEL_AND_LOCAL	13,984,078
8	ENTERTAINMENT	11,640,705
9	TOOLS	10,801,391
10	NEWS_AND_MAGAZINES	9,549,178
11	BOOKS_AND_REFERENCE	8,767,812

In [21]:

```

print('\nPopular apps in the COMMUNICATION genre:')
print('Average Installs : 38,456,119 \n')
count = 0
for app in Android_dataset:
    if app[1] == 'COMMUNICATION' and count < 6:
        print(app[0], ': ', app[5]) # print name and number of ratings
        count +=1

print('\nPopular apps in the VIDEO_PLAYERS genre:')
print('Average Installs : 24,727,872 \n')
count = 0
for app in Android_dataset:
    if app[1] == 'VIDEO_PLAYERS' and count <6:
        print(app[0], ': ', app[5]) # print name and number of ratings
        count +=1

print('\nPopular apps in the BOOKS_AND_REFERENCE genre:')
print('Average Installs : 8,767,812 \n')
count = 0
for app in Android_dataset:
    if app[1] == 'BOOKS_AND_REFERENCE' and count <6:
        print(app[0], ': ', app[5]) # print name and number of ratings
        count +=1

```

Popular apps in the COMMUNICATION genre:
Average Installs : 38,456,119

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+

Popular apps in the VIDEO_PLAYERS genre:
Average Installs : 24,727,872

YouTube : 1,000,000,000+
All Video Downloader 2018 : 1,000,000+
Video Downloader : 10,000,000+
HD Video Player : 1,000,000+
Iqiyi (for tablet) : 1,000,000+
Video Player All Format : 10,000,000+

Popular apps in the BOOKS_AND_REFERENCE genre:
Average Installs : 8,767,812

E-Book Read – Read Book for free : 50,000+
Download free book with green book : 100,000+
Wikipedia : 10,000,000+
Cool Reader : 10,000,000+
Free Panda Radio Music : 100,000+
Book store : 1,000,000+

As seen in the data above, similar to the iOS applications application dataset, there appears to be few applications in the popular genres that have a many more ratings than the rest and hence causing the entire

corresponding genre to appear more prominent. Moreover, the data also shows that the application suggested in section 6.2.1 would have a positive result here. The book application explained would have features that touch on the popular applications genres. (Specifically 1, 2, 3, 5, 11)

- **1. Communications:** Ideally the book application would allow for virtual book clubs to be formed where people could meet via video conference calls e.t.c, to discuss various books and articles.
- **2. Video Player:** The application would have video player accessibility where people could post video content regarding books and articles. Content could range from tutorials to various book analysis.
- **3. Social:** The application would have options of connection, where people could post about their readings, their discoveries, their learning experience e.t.c
- **5. Productivity:** This app would allow for project creation, where each project can allow for corresponding books and video contents could be collected and used as a tool for efficient learning and/or research.
- **11. Books and References:** The proposed book application would result in housing a collection of different books and reference materials for users of all ages.

7. Conclusions:

The project successfully analysed mobile phone application datasets from both Google Play Store and Apple App Stores. This was to deduce an application genre that would prove profitable in both Google Play Store and Apple App Store, where primary mode of profit is through in-app purchases. Datasets were cleaned to include only free, English-based applications. The likelihood of the applications to be installed (Application Popularity) was assessed using average number of ratings per genre in the Google Play Store dataset and the average number of installs per genre in the Apple App Store dataset.

The resulting application genre chosen was that of a book/reference application with video content capabilities and with the intention of fostering a social environment. The application could be used as a means to learn new skills (through books and video tutorials) or simply to have exciting discussions 'book clubs' through virtual means such as video conferencing.

In []: