In [ ]:

```
                         Analysis of free User Centered Android and IOS Mobile Apps

This project is about building Android and IOS mobile apps that are free to download and install.  S
ince the main
source of revenueconsists of in-app ads, it shows that our revenue for any app is highly dependant o
n the number of users
who use our app.

Whenever more users see and engage with the ads, the more income would be generated.  Therefore our
goal for this
porject is to analyze data to help our developers understand what type of apps are likely to attract
more users.
```

In [7]:

```
#Opening googleplaystore.csv

from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)

print(opened_file.readable())
```

```
True
```

In [12]:

```
from csv import reader
opened_file = open('AppStore.csv', encoding= 'utf8')
read_file = reader(opened_file)
dataset = list(read_file)

print(opened_file.readable())
```

```
True
```

In [48]:

```
#Here we are exploring the two datasets using the explore_data() function

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 are printing the first few rows of each data set


from csv import reader
opened_file = open('AppStore.csv', encoding= 'utf8')
read_file = reader(opened_file)
dataset = list(read_file)
print(dataset[0:5])

print("\n")

from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)
print(googledataset[0:3])
```

```
[['', 'id', 'track_name', 'size_bytes', 'currency', 'price', 'rating_count_tot', 'rating_count_v
er', 'user_rating', 'user_rating_ver', 'ver', 'cont_rating', 'prime_genre', 'sup_devices.num',
'ipadSc_urls.num', 'lang.num', 'vpp_lic'], ['1', '281656475', 'PAC-MAN Premium', '100788224', 'U
SD', '3.99', '21292', '26', '4', '4.5', '6.3.5', '4+', 'Games', '38', '5', '10', '1'], ['2', '28
1796108', 'Evernote - stay organized', '158578688', 'USD', '0', '161065', '26', '4', '3.5', '8.
2.2', '4+', 'Productivity', '37', '5', '23', '1'], ['3', '281940292', 'WeatherBug - Local Weathe
r, Radar, Maps, Alerts', '100524032', 'USD', '0', '188583', '2822', '3.5', '4.5', '5.0.0', '4+',
'Weather', '37', '5', '3', '1'], ['4', '282614216', 'eBay: Best App to Buy, Sell, Save! Online S
hopping', '128512000', 'USD', '0', '262241', '649', '4', '4.5', '5.10.0', '12+', 'Shopping', '3
7', '5', '9', '1']]


[['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 & De
sign', '7-Jan-18', '1.0.0', '4.0.3 and up'], ['Coloring book moana', 'ART_AND_DESIGN', '3.9', '9
67', '14M', '500,000+', 'Free', '0', 'Everyone', 'Art & Design;Pretend Play', '15-Jan-18', '2.0.
0', '4.0.3 and up']]
```

In [61]:

```
# The number of rows and columns of each data set is as follows and the function assumes the argumen
t
# for the dataset parameter doesn't have a header row

from csv import reader
opened_file = open('AppStore.csv', encoding= 'utf8')
read_file = reader(opened_file)
dataset = list(read_file)
#print(dataset)

print('number of rows')
print(len(dataset))
print('\n')
print('number of columns')
len(dataset[0])
```

```
number of rows
7198


number of columns
```

Out[61]: 17

In [60]:

```
from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)
print('number of rows')
print(len(googledataset))
print('\n')
print('number of columns')
len(googledataset[0])
```

```
number of rows
10842


number of columns
```

Out[60]: 13

In [71]:

```
# Below we are prining the column names and trying to identify the columns that could help us with o
ur analysis

from csv import reader
opened_file = open('AppStore.csv', encoding= 'utf8')
read_file = reader(opened_file)
dataset = list(read_file)

column_names = dataset[0]

print(column_names)
print('\n')
print(print(column_names[1], ',', column_names[2], ',', column_names[4], ',', column_names[7], ',',
column_names[8], ',', column_names[9]))
```

```
['', 'id', 'track_name', 'size_bytes', 'currency', 'price', 'rating_count_tot', 'rating_count_ve
r', 'user_rating', 'user_rating_ver', 'ver', 'cont_rating', 'prime_genre', 'sup_devices.num', 'i
padSc_urls.num', 'lang.num', 'vpp_lic']


id , track_name , currency , rating_count_ver , user_rating , user_rating_ver
None
```

In [27]:

```
#Based on the discussion section, the error is at row number 10472 and index number is 2 and it is i
dneed incorrect as the rating should not
#exceed 5. In this case the rating is 19 so we have to delete the row.

from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)
googledataset = googledataset[1:]

Wrong_data = googledataset[10472]

print(Wrong_data)

index = Wrong_data.index('19')
print('\n')
print('The index of 19:', index)



print(len(googledataset))
del googledataset[10472]

print(len(googledataset))
```

```
['Life Made WI-Fi Touchscreen Photo Frame', '1.9', '19', '3.0M', '1,000+', 'Free', '0', 'Everyon
e', '', '11-Feb-18', '1.0.19', '4.0 and up', '']


The index of 19: 2
10841
10840
```

In [28]:

```
#As we can see below the Google Play data set has duplicate entries which counts 1181 and names of s
ome duplicate examples of app are alos listed
#Examples of duplicate apps: ['Quick PDF Scanner + OCR FREE', 'Box', 'Google My Business', 'ZOOM Clo
ud Meetings',
#'join.me - Simple Meetings', 'Box']

from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)
googledataset = googledataset[1:]

duplicate_apps = []
unique_apps = []

for app in googledataset:
    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('Examples of duplicate apps:', duplicate_apps[:6])
```

```
Number of duplicate apps: 1181


Examples of duplicate apps: ['Quick PDF Scanner + OCR FREE', 'Box', 'Google My Business', 'ZOOM
Cloud Meetings', 'join.me - Simple Meetings', 'Box']
10841
```

In [36]:

```
#Here we created a dictionary key which is unique app name with a corresponding dictionary value hav
ing the highest
#number of reviews of that app.  looping through the googledataset, excluding the header row and we
 use
#the third endex of the each row, by changing this particular review colmn (index 3) to float data t
ype.  After we check
#if name exists in the new dictionary or not.  Finally we check if the length of the dictionary is 9
659 rows after we subtract the
#duplicate entries as well as the actual length of the new dictionary we created is also 9659. Since
the newly created dictionary dos
#not have any duplicates it is supposed to be the 9659 in lenght.

from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)
googledataset = googledataset[1:]

Wrong_data = googledataset[10472]

print(Wrong_data)

index = Wrong_data.index('19')
print('\n')
print('The index of 19:', index)



print(len(googledataset))
del googledataset[10472]

reviews_max = {}
for app in googledataset:
    name = app[0]
    n_reviews = str(app[3])

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

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

print('Expected length:', len(googledataset) - 1181)
print('Actual length:', len(reviews_max))
```

```
['Life Made WI-Fi Touchscreen Photo Frame', '1.9', '19', '3.0M', '1,000+', 'Free', '0', 'Everyon
e', '', '11-Feb-18', '1.0.19', '4.0 and up', '']


The index of 19: 2
10841
Expected length: 9659
Actual length: 9659
```

In [14]:

```
#In this case the function takes in a string and returns False if the cahracter of the string is gre
ater than 127
#if not it returns True, which means it is a non English language.

from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)
googledataset = googledataset[1:]

def is_english(string):

    for character in string:
        if ord(character) > 127:
            return False

    return True

print(is_english('Facebook'))
print(is_english('爱奇艺PPS -《欢乐颂2》电视剧热播'))
print(is_english('Docs To Go™ Free Office Suite'))
print(is_english('Instachat ▯'))
```

```
True
False
False
False
```

In [15]:

```
#What we did above does not work for all the English apps as some app names use images or symbols an
d the like which
#falls outside of the ASCII range (more than three non-ASCII) characters. In addition to this we che
ck if these app names are '
#detected as English or non-English for each data set and explore the data sets .

from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)
googledataset = googledataset[1:]


def is_english(string):
    non_ascii = 0

    for character in string:
        if ord(character) > 127:
            non_ascii += 1

    if non_ascii > 3:
        return False
    else:
        return True

print(is_english('Docs To Go™ Free Office Suite'))
print(is_english('Instachat ▯'))
print(is_english('爱奇艺PPS -《欢乐颂2》电视剧热播'))
```

```
True
True
False
```

In [ ]:

```
#The code below shows loop through each dataset to isolate the free apps in a separate list and chec
ks
#the length of each data set to check how many apps are remaining.

from csv import reader
opened_file = open('googleplaystore.csv', encoding= 'utf8')
read_file = reader(opened_file)
googledataset = list(read_file)
googledataset = googledataset[1:]


from csv import reader
opened_file = open('AppStore.csv', encoding= 'utf8')
read_file = reader(opened_file)
dataset = list(read_file)
dataset = dataset[1:]

android_final = []
ios_final = []

for app in googledataset:
    price = str(app[7])
    if price == '0':
        android_final.append(app)

for app in dataset:
    price = app[4]
    if price == '0':
        ios_final.append(app)

print(len(android_final))
print(len(ios_final))
```

In [ ]:

```
#The reason why we want to find an app profile that fits both the App Store and Google Play is to de
termine the kinds of apps
that are likely to attract more users as our revenue is highly influenced and depends on the number
of people using our apps.
Based on the code above, which isolates both dataset in a sparate list we can inspect both data sets
to identify the columns
we can also generate frequency tables to find out what the most common geners in each market are.  S
o according to the results shown
below the App Store is dominated by apps designed for fun (game), while google play shows more balan
cedland scape of both practical and
fun (game) apps.  Therefore we would use the prime_genre column for the AppStore data set and the Ge
ners and Category columns for the
googleplaystore data set.

opened_file = open('googleplaystore.csv')
from csv import reader
read_file = reader(opened_file)
googledataset = list(read_file)

opened_file = open('AppStore.csv')
from csv import reader
read_file = reader(opened_file)
dataset = list(read_file)

googledataset_free_apps = []
dataset_free_apps = []

for app in googledataset:
    price = app[7]
    if price == '0':
        googledataset_free_apps.append(free_app)

for app in dataset:
    price = app[4]
    if price == '0':
        dataset_free_apps.append(free_app)

display_table(dataset_free_apps, -5)

Games : 55.64595660749507
Entertainment : 8.234714003944774
Photo & Video : 4.117357001972387
Social Networking : 3.5256410256410255
Education : 3.2544378698224854
Shopping : 2.983234714003945
Utilities : 2.687376725838264
Lifestyle : 2.3175542406311638
Finance : 2.0710059171597637
Sports : 1.947731755424063
Health & Fitness : 1.8737672583826428
Music : 1.6518737672583828
Book : 1.6272189349112427
Productivity : 1.5285996055226825
News : 1.4299802761341223
Travel : 1.3806706114398422
Food & Drink : 1.0601577909270217
Weather : 0.7642998027613412
Reference : 0.4930966469428008
Navigation : 0.4930966469428008
Business : 0.4930966469428008
Catalogs : 0.22189349112426035
Medical : 0.19723865877712032

display_table(dataset_free_apps, 1)


FAMILY : 17.739043824701195
GAME : 10.56772908366534
TOOLS : 7.6195219123505975
BUSINESS : 4.442231075697211
PRODUCTIVITY : 3.944223107569721
LIFESTYLE : 3.6155378486055776
SPORTS : 3.5856573705179287
COMMUNICATION : 3.5856573705179287
MEDICAL : 3.5258964143426295
FINANCE : 3.4760956175298805
HEALTH_AND_FITNESS : 3.237051792828685
PHOTOGRAPHY : 3.117529880478088
PERSONALIZATION : 3.0776892430278884
SOCIAL : 2.908366533864542
NEWS_AND_MAGAZINES : 2.7988047808764938
SHOPPING : 2.5697211155378485
TRAVEL_AND_LOCAL : 2.450199203187251
DATING : 2.2609561752988045
BOOKS_AND_REFERENCE : 2.0219123505976095
VIDEO_PLAYERS : 1.7031872509960162
EDUCATION : 1.5139442231075697
ENTERTAINMENT : 1.4641434262948207
MAPS_AND_NAVIGATION : 1.3147410358565739
FOOD_AND_DRINK : 1.245019920318725
HOUSE_AND_HOME : 0.8764940239043826
LIBRARIES_AND_DEMO : 0.8366533864541833
AUTO_AND_VEHICLES : 0.8167330677290837
WEATHER : 0.7370517928286853
EVENTS : 0.6274900398406374
ART_AND_DESIGN : 0.6175298804780877
COMICS : 0.5976095617529881
PARENTING : 0.5776892430278884
BEAUTY : 0.5278884462151394
```


In [ ]:

```
#The reason why we want to find an app profile that fits both the App Store and Google Play is to de
termine the kinds of apps
that are likely to attract more users as our revenue is highly influenced and depends on the number
of people using our apps.
Based on the code above, which isolates both dataset in a sparate list we can inspect both data sets
to identify the columns
we can also generate frequency tables to find out what the most common geners in each market are.  S
o according to the results shown
below the App Store is dominated by apps designed for fun (game), while google play shows more balan
cedland scape of both practical and
fun (game) apps.  Therefore we would use the prime_genre column for the AppStore data set and the Ge
ners and Category columns for the
googleplaystore data set.

opened_file = open('googleplaystore.csv')
from csv import reader
read_file = reader(opened_file)
googledataset = list(read_file)

opened_file = open('AppStore.csv')
from csv import reader
read_file = reader(opened_file)
dataset = list(read_file)

googledataset_free_apps = []
dataset_free_apps = []

for app in googledataset:
    price = app[7]
    if price == '0':
        googledataset_free_apps.append(free_app)

for app in dataset:
    price = app[4]
    if price == '0':
        dataset_free_apps.append(free_app)

display_table(dataset_free_apps, -5)

Games : 55.64595660749507
Entertainment : 8.234714003944774
Photo & Video : 4.117357001972387
Social Networking : 3.5256410256410255
Education : 3.2544378698224854
Shopping : 2.983234714003945
Utilities : 2.687376725838264
Lifestyle : 2.3175542406311638
Finance : 2.0710059171597637
Sports : 1.947731755424063
Health & Fitness : 1.8737672583826428
Music : 1.6518737672583828
Book : 1.6272189349112427
Productivity : 1.5285996055226825
News : 1.4299802761341223
Travel : 1.3806706114398422
Food & Drink : 1.0601577909270217
Weather : 0.7642998027613412
Reference : 0.4930966469428008
Navigation : 0.4930966469428008
Business : 0.4930966469428008
Catalogs : 0.22189349112426035
Medical : 0.19723865877712032

display_table(dataset_free_apps, 1)


FAMILY : 17.739043824701195
GAME : 10.56772908366534
TOOLS : 7.6195219123505975
BUSINESS : 4.442231075697211
PRODUCTIVITY : 3.944223107569721
LIFESTYLE : 3.6155378486055776
SPORTS : 3.5856573705179287
COMMUNICATION : 3.5856573705179287
MEDICAL : 3.5258964143426295
FINANCE : 3.4760956175298805
HEALTH_AND_FITNESS : 3.237051792828685
```