

Problem Set

App Store Dataset

- List of all unique Prime_Genres(categories) in the dataset
- Category with highest number of apps
- Category with lowest number of apps
- Category with highest user rating
- App with highest downloads
- Category with highest average rating count
- Average user rating for free apps
- Average user rating for paid apps
- Category with highest average user rating for paid apps
- Most frequent Price point > 0
- Compare average user rating for paid vs free gaming apps

```
In [1]: 1 import pandas as pd
        2 def readCSVfile(filepath):
        3     return pd.read_csv(filepath)
        4
        5 filepath='DataFiles/AppleStore.csv'
        6 readCSVfile(filepath)
```

Out[1]:

	Unnamed: 0	id	track_name	size_bytes	currency	price	rating_count_tot	rating_c
0	1	281656475	PAC-MAN Premium	100788224	USD	3.99	21292	
1	2	281796108	Evernote - stay organized	158578688	USD	0.00	161065	
2	3	281940292	WeatherBug - Local Weather, Radar, Maps, Alerts	100524032	USD	0.00	188583	
3	4	282614216	eBay: Best App to Buy, Sell, Save! Online Shop...	128512000	USD	0.00	262241	

In [2]:

```
1 appstore=readCSVfile(filepath)
2 appstore
```

38	USD	0.00	10	10	5.0	5.0	1.0	4+	Gai
38	USD	3.99	55	29	4.5	4.5	1.4	4+	Gai
36	USD	0.99	0	0	0.0	0.0	1.1	4+	Util
38	USD	0.00	14	4	4.0	3.5	1.3.0	4+	Gai
36	USD	0.00	41	19	4.5	4.5	1.3	4+	Gai

In [3]:

```
1 # Function to List of all unique Prime_Genres(categories) in the dataset
2 def getIndexPrimeGenre(ap,keyword):
3     for i in range(len(ap.columns)):
4         if ap.columns[i]==keyword:
5             return i
6
7 getIndexPrimeGenre(appstore,'prime_genre')
8
9 def uniqueDataPrime_Genre(df,key):
10     primeGameinedx=getIndexPrimeGenre(df,key)
11     li = []
12     [li.append(row[primeGameinedx]) for row in df.values]
13
14 #     for row in df.values:
15 #         li.append(row[primeGameinedx])
16     print(list((set(li))))
17     #l=set(li)
18     #print(len(l))
19
20 uniqueDataPrime_Genre(appstore,'prime_genre')
```

```
['Book', 'Food & Drink', 'Sports', 'Medical', 'Games', 'Health & Fitness', 'Photo & Video', 'Music', 'Shopping', 'Entertainment', 'Weather', 'Navigation', 'News', 'Reference', 'Lifestyle', 'Catalogs', 'Business', 'Travel', 'Social Networking', 'Finance', 'Productivity', 'Utilities', 'Education']
```

In [4]:

```

1  # Function to Category with highest number of apps
2  # li=[6,2,4,5, 'f', 'df',1,2,3,4,5,6, 'f']
3  #     unique = {}
4  #     for n in li:
5  #         if n in unique:
6  #             unique[n] += 1
7  #         else:
8  #             unique[n] = 1
9
10 def highestNoCategoryApp(df,key):
11     unique={}
12     appIndex=getIndexPrimeGenre(df,key)
13     for row in df.values:
14         if row[appIndex] in unique:
15             unique[row[appIndex]] +=1
16         else:
17             unique[row[appIndex]] = 1
18     print(unique)
19     val = max(unique.values())
20     for item in unique.items():
21         if val == item[1]:
22             print(item[0],':',item[1])
23     return
24
25 highestNoCategoryApp(appstore,'prime_genre')

```

```

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

```

In [6]:

```
1  # Function to Category with Lowest number of apps
2  def uniqueCategoryApps(df, key):
3      unique={}
4      appIndex=getIndexPrimeGenre(df, key)
5      for row in df.values:
6          if row[appIndex] in unique:
7              unique[row[appIndex]] +=1
8          else:
9              unique[row[appIndex]] = 1
10     return unique
11 uniqueCategoryApps(appstore, 'prime_genre')
12
13 def lowestCategoryApps(unique):
14     val = min(unique.values())
15     for item in unique.items():
16         if val == item[1]:
17             print(item[0], ': ', item[1])
18     return
19
20 unique=uniqueCategoryApps(appstore, 'prime_genre')
21 lowestCategoryApps(unique)
22
```

Catalogs : 10

```
In [47]: 1 # Category with highest user rating
2 # get user rating and category indices
3 def getIndexUserRating(df, key1):
4     for i in range(len(df.columns)):
5         if df.columns[i] == key1:
6             return i
7
8 def getIndexCategory(df, key2):
9     for i in range(len(df.columns)):
10        if df.columns[i] == key2:
11            return i
12
13 getIndexUserRating(appstore, 'user_rating')
14 getIndexCategory(appstore, 'prime_genre')
15
16 def highestCategory(df, key1, key2):
17     catIndex = getIndexCategory(df, key2)
18     usrIndex = getIndexUserRating(df, key1)
19     #l={}
20     l=[]
21     for row in df.values:
22         l.append(row[usrIndex])
23     val = max(l)
24     s=[]
25     for row in df.values:
26         if val == row[usrIndex]:
27             s.append(row[catIndex])
28
29     return set(s)
30
31 highestCategory(appstore, 'user_rating', 'prime_genre')
```

```
Out[47]: {'Book',
'Business',
'Catalogs',
'Education',
'Entertainment',
'Finance',
'Food & Drink',
'Games',
'Health & Fitness',
'Lifestyle',
'Medical',
'Music',
'Navigation',
'News',
'Photo & Video',
'Productivity',
'Reference',
'Shopping',
'Social Networking',
'Sports',
'Travel',
'Utilities',
'Weather'}
```

In [54]:

```
1  # App with highest downloads
2  # 1. Get index of "rating_count_tot"
3  def index(df, key):
4      for i in range(len(df.columns)):
5          if df.columns[i] == key:
6              return i
7  index(appstore, "rating_count_tot")
8
9  def index1(df, key2):
10     for i in range(len(df.columns)):
11         if df.columns[i] == key2:
12             return i
13  index1(appstore, "track_name")
14
15  def highestDownloadApp(df, key1, key2):
16     trackIndex = index1(df, key1)
17     ratingIndex = index(df, key2)
18     li = []
19     for row in df.values:
20         li.append(row[ratingIndex])
21     val = max(li)
22     app = []
23     for row in df.values:
24         if val == row[ratingIndex]:
25             app.append(row[trackIndex])
26     print(app)
27  highestDownloadApp(appstore, "track_name", "rating_count_tot")
28
29
30
```

['Facebook']

```
In [62]: 1 # Category with highest average rating count
2 # Get indices of rating_count_tot and prime_genre
3 def rateIndex(df,key):
4     for i in range(len(df.columns)):
5         if df.columns[i]==key:
6             return i
7 rateIndex(appstore,'rating_count_tot')
8
9 def primeGenreIndex(df,key):
10     for i in range(len(df.columns)):
11         if df.columns[i]==key:
12             return i
13 primeGenreIndex(appstore,'prime_genre')
14
15 def highestAvgRatingCount(df,key1,key2):
16     li = []
17     index1=rateIndex(df,key1)
18     index2=primeGenreIndex(df,key2)
19     for row in df.values:
20         li.append(row[index1])
21     val=max(li)
22     cat=[]
23     #for row in
24     [cat.append(row[index2]) for row in df.values if val==row[index1] ]
25     return [cat,val]
26
27 highestAvgRatingCount(appstore,'rating_count_tot','prime_genre')
```

```
Out[62]: [['Social Networking'], 2974676]
```

In [69]:

```

1  # Average user rating for free apps
2  def priceindex(df,key):
3      for i in range(len(df.columns)):
4          if df.columns[i]==key:
5              return i
6  priceindex(appstore,'price')
7
8  def userRatingIndex(df,key):
9      for i in range(len(df.columns)):
10         if df.columns[i]== key:
11             return i
12  userRatingIndex(appstore,"user_rating")
13
14  def freeAppAvgUserrating(df,key1,key2):
15      index1=priceindex(df,key1)
16      index2 = userRatingIndex(df,key2)
17      li=[]
18      for row in df.values:
19          li.append(row[index1])
20      val=min(li)
21      lis=[]
22      for row in df.values:
23          if val==row[index1]:
24              lis.append(row[index2])
25      print(sum(lis)/len(lis))
26
27  freeAppAvgUserrating(appstore,'price','user_rating')
28
29
30
31

```

3.3767258382642997

In [70]:

```

1  # Average user rating for paid apps
2  def paidAppAvgUserRating(df,key1,key2):
3      index1=priceindex(df,key1)
4      index2 = userRatingIndex(df,key2)
5      li=[]
6      for row in df.values:
7          li.append(row[index1])
8      val=min(li)
9      l=[]
10     for row in df.values:
11         if val!=row[index1]:
12             l.append(row[index2])
13     print(sum(l)/len(l))
14
15  paidAppAvgUserRating(appstore,'price','user_rating')
16
17

```

3.720948742438714


```

In [33]: 1 # Category with highest average user rating for paid apps
2 # 1.Get indices of user_rating,category i.e prime_genre, paid apps i.e price
3 def userratingIndex(df,key):
4     for i in range(len(df.columns)):
5         if df.columns[i]==key:
6             return i
7 userratingIndex(appstore,'user_rating')
8 def prime_genreIndex(df,key):
9     for i in range(len(df.columns)):
10         if df.columns[i]==key:
11             return i
12 prime_genreIndex(appstore,'prime_genre')
13 def paidIndex(df,key):
14     for i in range(len(df.columns)):
15         if df.columns[i]==key:
16             return i
17 paidIndex(appstore,'price')
18
19 def highestUserratingForPaidApps(df,key1,key2,key3):
20     li=[]
21     paidindex=paidIndex(df,key1)
22     usrIndex=userratingIndex(df,key2)
23     primeIndex=prime_genreIndex(df,key3)
24     for row in df.values:
25         li.append(row[paidindex])
26     val=min(li)
27     #print(val)
28     li1=[]
29     for row in df.values:
30         if val!=row[paidindex]:
31             li1.append(row[usrIndex])
32     val2=max(li1)
33     li2=[]
34     for row in df.values:
35         if (val2 == row[usrIndex] and val != row[paidindex]) :
36             li2.append(row[primeIndex])
37     print(set(li2))
38     print(len(set(li2)))
39
40
41 highestUserratingForPaidApps(appstore,'price','user_rating','prime_genre')
42

```

```

{'Book', 'Food & Drink', 'Sports', 'Reference', 'Navigation', 'Lifestyle', 'New
s', 'Utilities', 'Music', 'Entertainment', 'Photo & Video', 'Social Networkin
g', 'Travel', 'Finance', 'Games', 'Productivity', 'Health & Fitness', 'Weathe
r', 'Business', 'Education'}
20

```

```

In [44]: 1 # Most frequent Price point > 0
2 def mostFrequentPrice(df,key):
3     index=paidIndex(df,key)
4     print(index)
5     li={}
6     for row in df.values:
7         if row[index]!=0.0:
8             if row[index] in li:
9                 li[row[index]]+=1
10            else:
11                li[row[index]]=1
12        val=max(li.values())
13        print(li)
14        print(val)
15        for item in li.items():
16            if val==item[1]:
17                print('Most Frequent paid price greater than zero is -',item[0])
18
19    return
20 mostFrequentPrice(appstore,'price')
21
22

```

```

5
{3.99: 277, 0.99: 728, 9.99: 81, 4.99: 394, 7.99: 33, 2.99: 683, 1.99: 621, 5.99: 52, 12.99: 5, 21.99: 1, 249.99: 1, 6.99: 166, 74.99: 1, 19.99: 13, 8.99: 9, 24.99: 8, 13.99: 6, 14.99: 21, 16.99: 2, 47.99: 1, 11.99: 6, 59.99: 3, 15.99: 4, 27.99: 2, 17.99: 3, 299.99: 1, 49.99: 2, 23.99: 2, 20.99: 2, 39.99: 2, 99.99: 1, 29.99: 6, 34.99: 1, 18.99: 1, 22.99: 2}
728
Most Frequent paid price greater than zero is - 0.99

```

```
In [52]: 1 # Compare average user rating for paid vs free gaming apps
2 def gamingPriceAvgUserRating(df):
3     index1=paidIndex(df,'price')
4     index2=paidIndex(df,'user_rating')
5     index3=paidIndex(df,'prime_genre')
6     li=[]
7     li1=[]
8     for row in df.values:
9         if row[index3]=='Games':
10             if row[index1]==0.00:
11                 li.append(row[index2])
12             else:
13                 li1.append(row[index2])
14     freeAvg=sum(li)/len(li)
15     paidAvg=sum(li1)/len(li1)
16     print('freeApps user rating avg',freeAvg,'\n','paid apps usr rating avg')
17     if freeAvg>paidAvg:
18         print("free apps user rating avg is more than paid apps")
19     else:
20         print("paid apps user rating avg is more than free apps")
21
22 gamingPriceAvgUserRating(appstore)
23
24
```

```
freeApps user rating avg 3.5285777580859548
paid apps usr rating avg 3.9049844236760123
paid apps user rating avg is more than free apps
```

```
In [ ]: 1
```