Apple store Data Sets

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

Out[2]:

	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 [6]:
```
Appstore=readCSVdata(filepath)
filepath='DataFiles/AppleStore.csv'
def printDataFrameColumns(df):
    columns=df.columns
    for column in columns:
        print(column)
    return
printDataFrameColumns(Appstore)
```

```
Unnamed: 0
id
track_name
size_bytes
currency
price
rating_count_tot
rating_count_ver
user_rating
user_rating_ver
ver
cont_rating
prime_genre
sup_devices.num
ipadSc_urls.num
lang.num
vpp_lic
```

In [4]:
```
Appstore=readCSVdata(filepath)
# filepath='DataFiles/AppleStore.csv'
li=[]
def Prime_Genres(df):
    for i in range(len(df.values)):
        for j in range(12,len(df.columns)-4):
            d=df.values[i][j]
            if d not in li:
                li.append(d)
    print(li,end=" ")
Prime_Genres(Appstore)
```

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

In [7]:

```
Appstore=readCSVdata(filepath)
li={}
def highestnumberapps(df):
    for i in range(len(df.values)):
        for j in range(12,len(df.columns)-4):
            d=df.values[i][j]
            if d not in li.keys():
                li[d]=1
            else:
                li[d]+=1
    print(li)
    u=sorted(li.values(),reverse=True)
    print(u)
    print('\n')
    max1=max(u)
    for item in li.items():
        if item[1]==max1:
            print(item[0],':',max1)
highestnumberapps(Appstore)
```

```
{'Games': 3862, 'Productivity': 178, 'Weather': 72, 'Shopping': 122, 'Referenc
e': 64, 'Finance': 104, 'Music': 138, 'Utilities': 248, 'Travel': 81, 'Social N
etworking': 167, 'Sports': 114, 'Business': 57, 'Health & Fitness': 180, 'Enter
tainment': 535, 'Photo & Video': 349, 'Navigation': 46, 'Education': 453, 'Life
style': 144, 'Food & Drink': 63, 'News': 75, 'Book': 112, 'Medical': 23, 'Catal
ogs': 10}
[3862, 535, 453, 349, 248, 180, 178, 167, 144, 138, 122, 114, 112, 104, 81, 75,
72, 64, 63, 57, 46, 23, 10]


Games : 3862
```

In [8]:

```
Appstore=readCSVdata(filepath)
li={}
def lowestnumberapps(df):
    for i in range(len(df.values)):
        for j in range(12,len(df.columns)-4):
            d=df.values[i][j]
            if d not in li.keys():
                li[d]=1
            else:
                li[d]+=1
    print(li)
    u=sorted(li.values(),reverse=True)
    print(u)
    print('\n')
    min1=min(u)
    for item in li.items():
        if item[1]==min1:
            print(item[0],':',min1)
lowestnumberapps(Appstore)

```

```
{'Games': 3862, 'Productivity': 178, 'Weather': 72, 'Shopping': 122, 'Referenc
e': 64, 'Finance': 104, 'Music': 138, 'Utilities': 248, 'Travel': 81, 'Social N
etworking': 167, 'Sports': 114, 'Business': 57, 'Health & Fitness': 180, 'Enter
tainment': 535, 'Photo & Video': 349, 'Navigation': 46, 'Education': 453, 'Life
style': 144, 'Food & Drink': 63, 'News': 75, 'Book': 112, 'Medical': 23, 'Catal
ogs': 10}
[3862, 535, 453, 349, 248, 180, 178, 167, 144, 138, 122, 114, 112, 104, 81, 75,
72, 64, 63, 57, 46, 23, 10]


Catalogs : 10
```

In [10]:

```
Appstore = readCSVdata(filepath)

def ci(appdf,key):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key:
            return i

def ci1(appdf,key1):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key1:
            return i

def Highest_User_Rating(appdf,key,key1):
    li=[]
    CI=ci(appdf,key)
    CI1=ci1(appdf,key1)
    for i in appdf.values:
        li.append(i[CI])
    m=max(li)
    print(m)
    s=[]
    for i in appdf.values:
        if m==i[CI]:
            s.append(i[CI1])
    print(set(s))
    print(len(set(s)))

Highest_User_Rating(Appstore,'user_rating','prime_genre')
```

```
5.0
{'Utilities', 'Navigation', 'Education', 'News', 'Shopping', 'Music', 'Health &
Fitness', 'Food & Drink', 'Productivity', 'Book', 'Weather', 'Lifestyle', 'Cata
logs', 'Medical', 'Entertainment', 'Reference', 'Social Networking', 'Finance',
'Sports', 'Games', 'Travel', 'Photo & Video', 'Business'}
23
```

In [11]:

```
### App with highest downloads

Appstore = readCSVdata(filepath)
def ci(appdf,key):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key:
            return i

def ci1(appdf,key1):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key1:
            return i

def Highest_Downloads(appdf,key,key1):
    li=[]
    CI=ci(appdf,key)
    CI1=ci1(appdf,key1)
    for i in appdf.values:
        li.append(i[CI1])
    m=max(li)
    print(m)
    s=[]
    for i in appdf.values:
        if m==i[CI1]:
            s.append(i[CI])
    print(set(s))
    print(len(set(s)))

Highest_Downloads(Appstore,'track_name','rating_count_tot')
```

```
2974676
{'Facebook'}
1
```

In [12]:

```
### Category with highest average rating count

Appstore = readCSVdata(filepath)
def ci(appdf,key):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key:
            return i

def ci1(appdf,key1):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key1:
            return i

def Highest_Avg_Ratingcount(appdf,key,key1):
    li=[]
    CI=ci(appdf,key)
    CI1=ci1(appdf,key1)
    for i in appdf.values:
        li.append(i[CI])
    m=max(li)
    print(m)
    s=[]
    for i in appdf.values:
        if m==i[CI]:
            s.append(i[CI1])
    print(set(s))
    print(len(set(s)))

Highest_Avg_Ratingcount(Appstore,'rating_count_tot','prime_genre')
```

```
2974676
{'Social Networking'}
1
```

In [13]:

```
##### Average user rating for free apps

Appstore = readCSVdata(filepath)

def ci(appdf,key):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key:
            return i

def ci1(appdf,key1):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key1:
            return i
def Average_User_Rating_forFreeApps(appdf,key,key1):
    li=[]
    CI=ci(appdf,key)
    CI1=ci1(appdf,key1)
    for i in appdf.values:
        li.append(i[CI])
    m=min(li)
    print(m)
    s=[]
    for i in appdf.values:
        if m==i[CI]:
            s.append(i[CI1])
    a=sum(s)/len(s)
    print(a)
Average_User_Rating_forFreeApps(Appstore,'price','user_rating')
```

```
0.0
3.3767258382642997
```

In [14]:

```
### Average user rating for paid apps

Appstore = readCSVdata(filepath)

def ci(appdf,key):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key:
            return i

def ci1(appdf,key1):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key1:
            return i
def Average_User_Rating_forPaidApps(appdf,key,key1):
    li=[]
    CI=ci(appdf,key)
    CI1=ci1(appdf,key1)
    for i in appdf.values:
        if i[CI]>0:
            li.append(i[CI1])
    print(sum(li)/len(li))

Average_User_Rating_forPaidApps(Appstore,'price','user_rating')
```

```
3.720948742438714
```

In [15]:

```
### Category with highest  average  user rating for paid apps
Appstore = readCSVdata(filepath)

def ci(appdf,key):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key:
            return i

def ci1(appdf,key1):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key1:
            return i

def ci2(appdf,key2):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key2:
            return i
def Average_User_Rating_forPaidApps(appdf,key,key1,key2):
    li=[]
    CI=ci(appdf,key)
    CI1=ci1(appdf,key1)
    CI2=ci2(appdf,key2)
    for i in appdf.values:
        if i[CI]>0:
            li.append(i[CI1])
    m=max(li)
    print(m)
    s=[]
    for i in appdf.values:
        if m==i[CI1]:
            s.append(i[CI2])
    print(set(s))

Average_User_Rating_forPaidApps(Appstore,'price','user_rating','prime_genre'
```

```
5.0
{'Utilities', 'Navigation', 'Education', 'News', 'Shopping', 'Music', 'Health &
Fitness', 'Food & Drink', 'Productivity', 'Book', 'Weather', 'Lifestyle', 'Cata
logs', 'Medical', 'Entertainment', 'Reference', 'Social Networking', 'Finance',
'Sports', 'Games', 'Travel', 'Photo & Video', 'Business'}
```

In [16]:

```
### Most frequent Price point >0


Appstore = readCSVdata(filepath)

def ci(appdf,key):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key:
            return i

def Frequent_Price(appdf,key):
    li=[]
    d={}
    C=ci(appdf,key)
    for i in appdf.values:
        if i[C]>0:
            li.append(i[C])
    for i in li:
        if i not in d:
            d[i]=1
        else:
            d[i]+=1
    print(d)
    print('\n')
    v=max(d.values())
    for item in d.items():
        if item[1]==v:
            print("most frequent price point is",item[0],':',v)


Frequent_Price(Appstore,'price')

```

```
{3.99: 277, 0.99: 728, 9.99: 81, 4.99: 394, 7.99: 33, 2.99: 683, 1.99: 621, 5.9
9: 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, 99.9
9: 1, 29.99: 6, 34.99: 1, 18.99: 1, 22.99: 2}


most frequent price point is 0.99 : 728
```

In [3]:

```
#### Compare average user rating for paid vs free gaming apps


Appstore = readCSVdata(filepath)

def ci(appdf,key):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key:
            return i

def ci1(appdf,key1):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key1:
            return i

def ci2(appdf,key2):
    for i in range(len(appdf.values)):
        if appdf.columns[i]==key2:
            return i
def Compare_freeGammingapps(appdf,key,key1,key2):
    CI=ci(appdf,key)
    CI1=ci1(appdf,key1)
    CI2=ci2(appdf,key2)
    sum1=0
    count=0
    s=0
    c=0
    for i in range(len(appdf.values)):
        if appdf.values[i][CI]=='Games':
            if appdf.values[i][CI1]>0:
                sum1=sum1+appdf.values[i][CI2]
                count=count+1
            else:
                s=s+appdf.values[i][CI2]
                c=c+1
    avg=sum1/count
    avg1=s/c
    print(avg,avg1)
    print('\n')
    if avg>avg1:
        print("Paid apps are having the highest than free apps",':',avg)
    else:
        print("Free are greater than paid apps",':',avg1)


Compare_freeGammingapps(Appstore,'prime_genre','price','user_rating',)
```

```
3.9049844236760123 3.5285777580859548


Paid apps are having the highest than free apps : 3.9049844236760123
```

In [ ]:
```
1
```