In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df=pd.read_csv("clean_gpaydata.csv")
df.head()
```

Out[2]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone	
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone	De
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone	
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	Teen	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	Everyone	Des

In [3]:

```
df.shape
```

Out[3]:

```
(10840, 17)
```

In [4]:

```
df.head()
```

Out[4]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	Everyone

In [5]:

```
df_copy= df.copy()

```

In [6]:

```
df_copy.head()
```

Out[6]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone	
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone	De
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone	
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	Teen	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	Everyone	Des

In [7]:

```
df_copy.shape
```

Out[7]:

```
(10840, 17)
```

In [8]:

```
df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10840 entries, 0 to 10839
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10840 non-null  object 
 1   Category        10840 non-null  object 
 2   Rating          9366 non-null   float64
 3   Reviews         10840 non-null  int64  
 4   Size            9145 non-null   float64
 5   Installs        10840 non-null  int64  
 6   Type            10839 non-null  object 
 7   Price           10840 non-null  float64
 8   Content Rating  10840 non-null  object 
 9   Genres          10840 non-null  object 
 10  Last Updated    10840 non-null  object 
 11  Current Ver     10832 non-null  object 
 12  Android Ver     10838 non-null  object 
 13  day             10840 non-null  int64  
 14  date            10840 non-null  object 
 15  month           10840 non-null  int64  
 16  yrar            10840 non-null  int64  
dtypes: float64(3), int64(5), object(9)
memory usage: 1.4+ MB
```

In [9]:

```
df_copy.columns
```

Out[9]:

```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
       'Android Ver', 'day', 'date', 'month', 'yrar'],
      dtype='object')
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
App                  0
Category             0
Rating            1474
Reviews              0
Size              1695
Installs             0
Type                 1
Price                0
Content Rating       0
Genres               0
Last Updated         0
Current Ver          8
Android Ver          2
day                  0
date                 0
month                0
yrar                 0
dtype: int64
```

In [11]:

```
df.describe().T
```

Out[11]:

	count	mean	std	min	25%	50%	75%	
Rating	9366.0	4.191757e+00	5.152189e-01	1.000	4.0	4.3	4.5	5.000000
Reviews	10840.0	4.441529e+05	2.927761e+06	0.000	38.0	2094.0	54775.5	7.815831
Size	9145.0	2.151746e+01	2.258804e+01	0.011	4.9	13.0	30.0	1.000000
Installs	10840.0	1.546434e+07	8.502936e+07	0.000	1000.0	100000.0	5000000.0	1.000000
Price	10840.0	1.027368e+00	1.594970e+01	0.000	0.0	0.0	0.0	4.000000
day	10840.0	1.560904e+01	9.561621e+00	1.000	6.0	16.0	24.0	3.100000
month	10840.0	6.422325e+00	2.578388e+00	1.000	5.0	7.0	8.0	1.200000
yrar	10840.0	2.017400e+03	1.100914e+00	2010.000	2017.0	2018.0	2018.0	2.018000

In [12]:

```
# if you want to include all the columns
df.describe(include="all")
```

Out[12]:

	App	Category	Rating	Reviews	Size	Installs	Type	
count	10840	10840	9366.000000	1.084000e+04	9145.000000	1.084000e+04	10839	108
unique	9659	33	NaN	NaN	NaN	NaN	2	
top	ROBLOX	FAMILY	NaN	NaN	NaN	NaN	Free	
freq	9	1972	NaN	NaN	NaN	NaN	10039	
mean	NaN	NaN	4.191757	4.441529e+05	21.517458	1.546434e+07	NaN	
std	NaN	NaN	0.515219	2.927761e+06	22.588038	8.502936e+07	NaN	
min	NaN	NaN	1.000000	0.000000e+00	0.011000	0.000000e+00	NaN	
25%	NaN	NaN	4.000000	3.800000e+01	4.900000	1.000000e+03	NaN	
50%	NaN	NaN	4.300000	2.094000e+03	13.000000	1.000000e+05	NaN	
75%	NaN	NaN	4.500000	5.477550e+04	30.000000	5.000000e+06	NaN	
max	NaN	NaN	5.000000	7.815831e+07	100.000000	1.000000e+09	NaN	4

In [13]:

```
# to find duplicated values
df.duplicated().sum()
```

Out[13]:

```
483
```

In [14]:

```
#duplicated values in the frame
df[df.duplicated()]
```

Out[14]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
229	Quick PDF Scanner + OCR FREE	BUSINESS	4.2	80805	NaN	5000000	Free	0.0	Everyone
236	Box	BUSINESS	4.2	159872	NaN	10000000	Free	0.0	Everyone
239	Google My Business	BUSINESS	4.4	70991	NaN	5000000	Free	0.0	Everyone
256	ZOOM Cloud Meetings	BUSINESS	4.4	31614	37.0	10000000	Free	0.0	Everyone
261	join.me - Simple Meetings	BUSINESS	4.0	6989	NaN	1000000	Free	0.0	Everyone
...	...	...	...	...	...	...	...	...	...
8643	Wunderlist: To-Do List & Tasks	PRODUCTIVITY	4.6	404610	NaN	10000000	Free	0.0	Everyone
8654	TickTick: To Do List with Reminder, Day Planner	PRODUCTIVITY	4.6	25370	NaN	1000000	Free	0.0	Everyone
8658	ColorNote Notepad Notes	PRODUCTIVITY	4.6	2401017	NaN	100000000	Free	0.0	Everyone
10049	Airway Ex - Intubate. Anesthetize. Train.	MEDICAL	4.3	123	86.0	10000	Free	0.0	Everyone
10767	AAFP	MEDICAL	3.8	63	24.0	10000	Free	0.0	Everyone

483 rows × 17 columns

In [15]:

```
# drop duplicated values
df.drop_duplicates()
```

Out[15]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	(
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	E
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	E
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	E
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	E
...	...	...	...	...	...	...	...	...	
10835	Sya9a Maroc - FR	FAMILY	4.5	38	53.0	5000	Free	0.0	E
10836	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6	100	Free	0.0	E
10837	Parkinson Exercices FR	MEDICAL	NaN	3	9.5	1000	Free	0.0	E
10838	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	NaN	1000	Free	0.0	
10839	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19.0	10000000	Free	0.0	E

10357 rows × 17 columns

In [16]:

```
#check duplicate sum
df.drop_duplicates().sum()
```

Out[16]:

```
App                 Photo Editor & Candy Camera & Grid & ScrapBook...
Category            ART_AND_DESIGNART_AND_DESIGNART_AND_DESIGNART_...
Rating                                                        37238.6
Reviews                                                    4203954052
Size                                                        188000.95
Installs                                                 146631914527
Price                                                         10676.0
Content Rating      EveryoneEveryoneEveryoneTeenEveryoneEveryoneEv...
Genres              Art & DesignArt & Design;Pretend PlayArt & Des...
Last Updated        2018-01-072018-01-152018-08-012018-06-082018-0...
day                                                            161767
date                2018-01-072018-01-152018-08-012018-06-082018-0...
month                                                           66254
yrar                                                         20894035
dtype: object
```

In [17]:

```
df.shape
```

Out[17]:

```
(10840, 17)
```

In [18]:

```
display(df.drop_duplicates())
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	E
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	E
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	E
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	E
...	...	...	...	...	...	...	...	...	
10835	Sya9a Maroc - FR	FAMILY	4.5	38	53.0	5000	Free	0.0	E
10836	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6	100	Free	0.0	E
10837	Parkinson Exercices FR	MEDICAL	NaN	3	9.5	1000	Free	0.0	E
10838	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	NaN	1000	Free	0.0	
10839	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19.0	10000000	Free	0.0	E

10357 rows × 17 columns

In [19]:

```
#Permanently deletws duplicate values
df.drop_duplicates(keep=False, inplace=True)
```

In [20]:

```
df.shape
```

Out[20]:

```
(9947, 17)
```

Exploring the data

segregate the categorical and numerical Values

In [21]:

```
numerical_feature= [feature for feature in df.columns if (df[feature]).dtype != "O"]
numerical_feature
```

Out[21]:

```
['Rating', 'Reviews', 'Size', 'Installs', 'Price', 'day', 'month', 'yrar']
```

In [22]:

```
categorial_feature = [feature for feature in df.columns if (df[feature]).dtype =="O"]
categorial_feature
```

Out[22]:

```
['App',
 'Category',
 'Type',
 'Content Rating',
 'Genres',
 'Last Updated',
 'Current Ver',
 'Android Ver',
 'date']
```

In [23]:

```
df["App"].value_counts()
```

Out[23]:

```
ROBLOX                                               9
8 Ball Pool                                          7
Zombie Catchers                                      6
Bubble Shooter                                       6
Helix Jump                                           6
                                                    ..
Vienna U-Bahn                                        1
U-Haul                                               1
Kicker U                                             1
/u/app                                               1
iHoroscope - 2018 Daily Horoscope & Astrology        1
Name: App, Length: 9381, dtype: int64
```

In [24]:

```
9/9947
```

Out[24]:

```
0.0009047954157032271
```

In [25]:

```
len(df["App"].value_counts())
```

Out[25]:

```
9381
```

9381 categories are available in the above App field

In [26]:

```
#to get the percentages or prAPORTION OF THE VALUES
df["App"].value_counts(normalize=True)
```

Out[26]:

```
ROBLOX                                           0.000905
8 Ball Pool                                      0.000704
Zombie Catchers                                  0.000603
Bubble Shooter                                   0.000603
Helix Jump                                       0.000603
                                                   ...
Vienna U-Bahn                                    0.000101
U-Haul                                           0.000101
Kicker U                                         0.000101
/u/app                                           0.000101
iHoroscope - 2018 Daily Horoscope & Astrology    0.000101
Name: App, Length: 9381, dtype: float64
```

In [27]:

```
df["App"].value_counts(normalize=False)
```

Out[27]:

```
ROBLOX                                                9
8 Ball Pool                                           7
Zombie Catchers                                       6
Bubble Shooter                                        6
Helix Jump                                            6
                                                     ..
Vienna U-Bahn                                         1
U-Haul                                                1
Kicker U                                              1
/u/app                                                1
iHoroscope - 2018 Daily Horoscope & Astrology         1
Name: App, Length: 9381, dtype: int64
```

In [28]:

```
#MULTIPLY BY 100
df["App"].value_counts(normalize=True)*100
```

Out[28]:

```
ROBLOX                                                0.090480
8 Ball Pool                                           0.070373
Zombie Catchers                                       0.060320
Bubble Shooter                                        0.060320
Helix Jump                                            0.060320
                                                        ...
Vienna U-Bahn                                         0.010053
U-Haul                                                0.010053
Kicker U                                              0.010053
/u/app                                                0.010053
iHoroscope - 2018 Daily Horoscope & Astrology         0.010053
Name: App, Length: 9381, dtype: float64
```

In [29]:

```
# to see entire categorical data
df[categorial_feature]
```

Out[29]:

	App	Category	Type	Content Rating	Genres	Last Updated	Curren Ve
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	Free	Everyone	Art & Design	2018-01-07	1.0.
1	Coloring book moana	ART_AND_DESIGN	Free	Everyone	Art & Design;Pretend Play	2018-01-15	2.0.
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	Free	Everyone	Art & Design	2018-08-01	1.2.
3	Sketch - Draw & Paint	ART_AND_DESIGN	Free	Teen	Art & Design	2018-06-08	Varie wit devic
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	Free	Everyone	Art & Design;Creativity	2018-06-20	1.
...	...	...	...	...	...	...	.
10835	Sya9a Maroc - FR	FAMILY	Free	Everyone	Education	2017-07-25	1.4
10836	Fr. Mike Schmitz Audio Teachings	FAMILY	Free	Everyone	Education	2018-07-06	1.
10837	Parkinson Exercices FR	MEDICAL	Free	Everyone	Medical	2017-01-20	1.
10838	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	Free	Mature 17+	Books & Reference	2015-01-19	Varie wit devic
10839	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	Free	Everyone	Lifestyle	2018-07-25	Varie wit devic

9947 rows × 9 columns

In [30]:

```
df[numerical_feature]
```

Out[30]:

	Rating	Reviews	Size	Installs	Price	day	month	yrar
0	4.1	159	19.0	10000	0.0	7	1	2018
1	3.9	967	14.0	500000	0.0	15	1	2018
2	4.7	87510	8.7	5000000	0.0	1	8	2018
3	4.5	215644	25.0	50000000	0.0	8	6	2018
4	4.3	967	2.8	100000	0.0	20	6	2018
...	...	...	...	...	...	...	...	...
10835	4.5	38	53.0	5000	0.0	25	7	2017
10836	5.0	4	3.6	100	0.0	6	7	2018
10837	NaN	3	9.5	1000	0.0	20	1	2017
10838	4.5	114	NaN	1000	0.0	19	1	2015
10839	4.5	398307	19.0	10000000	0.0	25	7	2018

9947 rows × 8 columns

In [31]:

```
# Create new dataframe
num_df=df[numerical_feature]
cat_df=df[categorial_feature]
```

In [32]:

```
num_df
```

Out[32]:

	Rating	Reviews	Size	Installs	Price	day	month	yrar
0	4.1	159	19.0	10000	0.0	7	1	2018
1	3.9	967	14.0	500000	0.0	15	1	2018
2	4.7	87510	8.7	5000000	0.0	1	8	2018
3	4.5	215644	25.0	50000000	0.0	8	6	2018
4	4.3	967	2.8	100000	0.0	20	6	2018
...	...	...	...	...	...	...	...	...
10835	4.5	38	53.0	5000	0.0	25	7	2017
10836	5.0	4	3.6	100	0.0	6	7	2018
10837	NaN	3	9.5	1000	0.0	20	1	2017
10838	4.5	114	NaN	1000	0.0	19	1	2015
10839	4.5	398307	19.0	10000000	0.0	25	7	2018

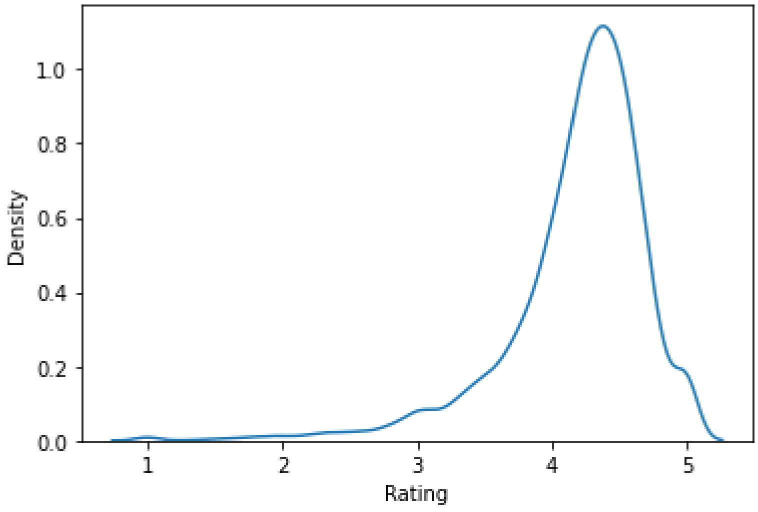
9947 rows × 8 columns

In [33]:

```
# check the distribution of numerical data
sns.kdeplot(num_df["Rating"])
```

Out[33]:

```
<AxesSubplot:xlabel='Rating', ylabel='Density'>
```

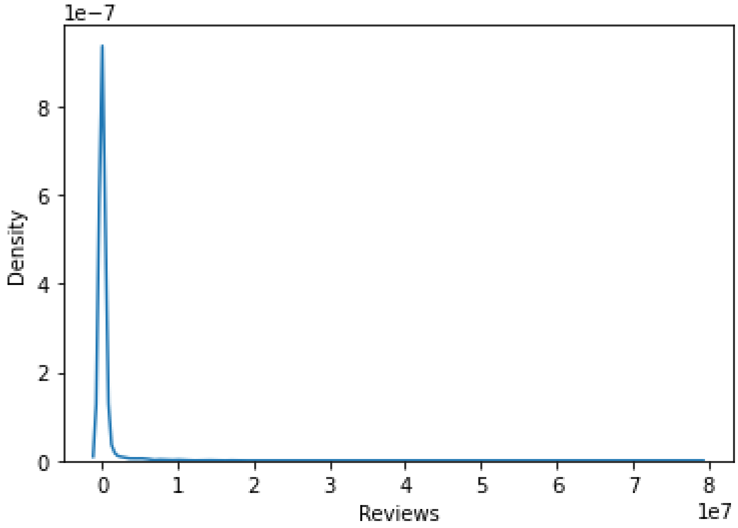


In [34]:

```
sns.kdeplot(df["Reviews"])
```

Out[34]:

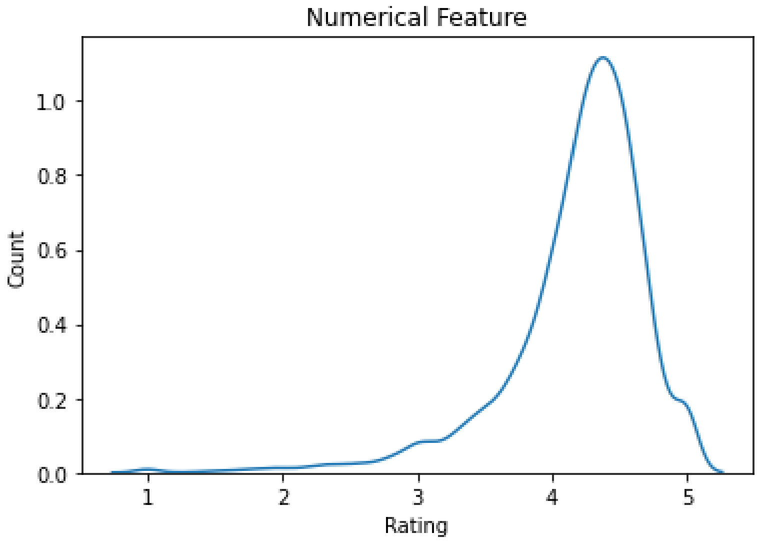
```
<AxesSubplot:xlabel='Reviews', ylabel='Density'>
```



In [35]:

```
# distribution related with each and every variable
for i in numerical_feature:

    sns.kdeplot(num_df[i])
    plt.xlabel(i)
    plt.ylabel("Count")
    plt.title("Numerical Feature")
    plt.show()
```



In [36]:

```
len(numerical_feature)
```

Out[36]:

```
8
```

In [37]:

```
(numerical_feature)
```

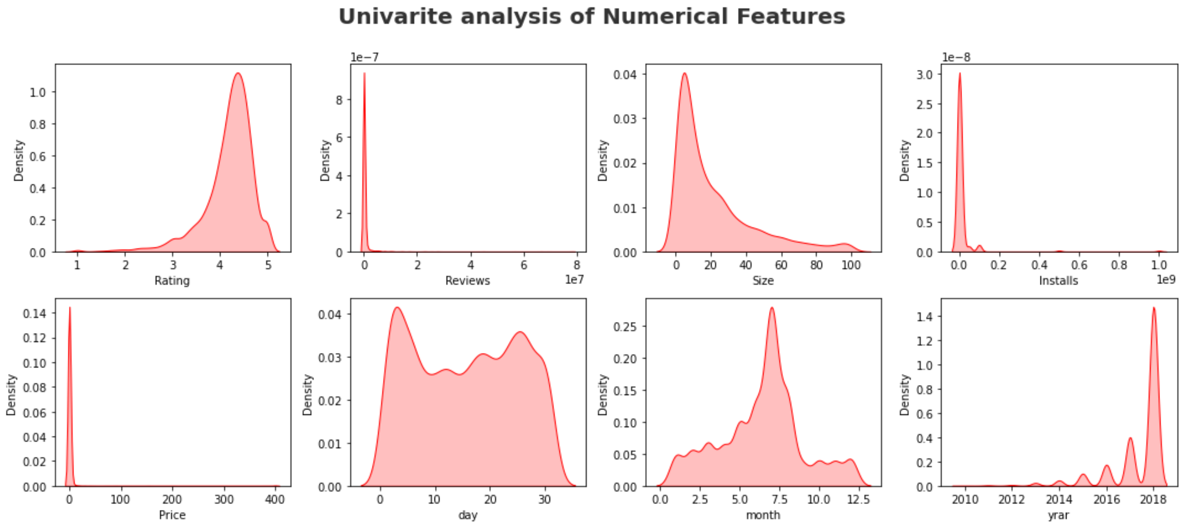
Out[37]:

```
['Rating', 'Reviews', 'Size', 'Installs', 'Price', 'day', 'month', 'yrar']
```

In [38]:

```python
# same code for numerical feature
plt.figure(figsize=(15,15))
plt.suptitle("Univarite analysis of Numerical Features" ,fontsize= 20,fontweight= "bold

for i in range (len(numerical_feature)):
    plt.subplot(5,4,i+1)
    sns.kdeplot(x=df[numerical_feature[i]],shade = True ,color= "r" )
    plt.xlabel(numerical_feature[i])
    plt.tight_layout()
    #https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.tight_layout.html
```



Observations

size, reviewers ,Price, Indtallsis are positive skewed and

rating and year is negative skewed(left skewed).

month is normally distributed

Analysis on Categorical columns

In [39]:

```

categorial_feature
```

Out[39]:

```
['App',
 'Category',
 'Type',
 'Content Rating',
 'Genres',
 'Last Updated',
 'Current Ver',
 'Android Ver',
 'date']
```

In [40]:

```
len(categorial_feature)
```

Out[40]:

```
9
```

In [41]:

```
cat_df
```

Out[41]:

	App	Category	Type	Content Rating	Genres	Last Updated	Curren Ve
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	Free	Everyone	Art & Design	2018-01-07	1.0.
1	Coloring book moana	ART_AND_DESIGN	Free	Everyone	Art & Design;Pretend Play	2018-01-15	2.0.
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	Free	Everyone	Art & Design	2018-08-01	1.2.
3	Sketch - Draw & Paint	ART_AND_DESIGN	Free	Teen	Art & Design	2018-06-08	Varie wit devic
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	Free	Everyone	Art & Design;Creativity	2018-06-20	1.
...	...	...	...	...	...	...	.
10835	Sya9a Maroc - FR	FAMILY	Free	Everyone	Education	2017-07-25	1.4
10836	Fr. Mike Schmitz Audio Teachings	FAMILY	Free	Everyone	Education	2018-07-06	1.
10837	Parkinson Exercices FR	MEDICAL	Free	Everyone	Medical	2017-01-20	1.
10838	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	Free	Mature 17+	Books & Reference	2015-01-19	Varie wit devic
10839	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	Free	Everyone	Lifestyle	2018-07-25	Varie wit devic

9947 rows × 9 columns

In [42]:

```
# check the different categories in "Type" Feature
cat_df["Type"].value_counts()

```

Out[42]:

```
Free    9215
Paid     731
Name: Type, dtype: int64
```

In [43]:

```
cat_df["Type"].unique()
```

Out[43]:

```
array(['Free', 'Paid', nan], dtype=object)
```

In [ ]:

```

```

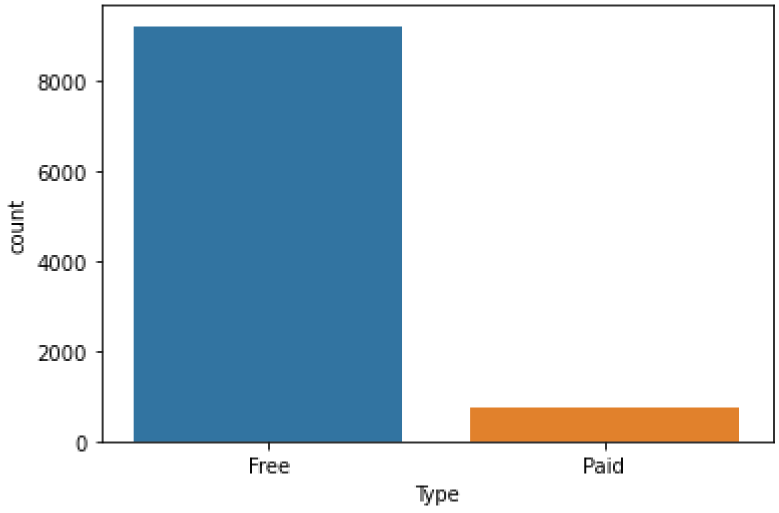
there are 9215 apps are free and 731 apps are paid

In [44]:

```
# plot this in count plot
sns.countplot(cat_df["Type"])
```

Out[44]:

```
<AxesSubplot:xlabel='Type', ylabel='count'>
```



In [45]:

```
# plot such graphs or type and Content Rating
cat_df["Content Rating"].value_counts()
```

Out[45]:

```
Everyone           8094
Teen               1099
Mature 17+          398
Everyone 10+        351
Adults only 18+       3
Unrated               2
Name: Content Rating, dtype: int64
```

In [46]:

```
cat_df["Content Rating"].unique()
```

Out[46]:

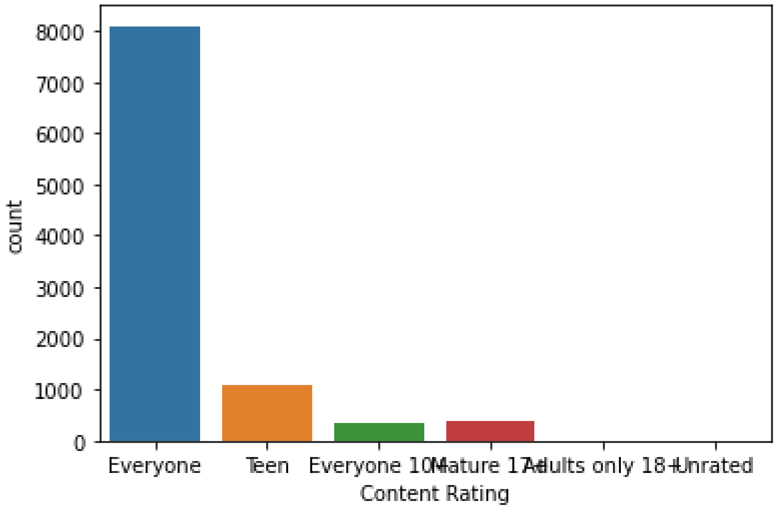
```
array(['Everyone', 'Teen', 'Everyone 10+', 'Mature 17+',
       'Adults only 18+', 'Unrated'], dtype=object)
```

In [47]:

```
sns.countplot(cat_df["Content Rating"])
```

Out[47]:

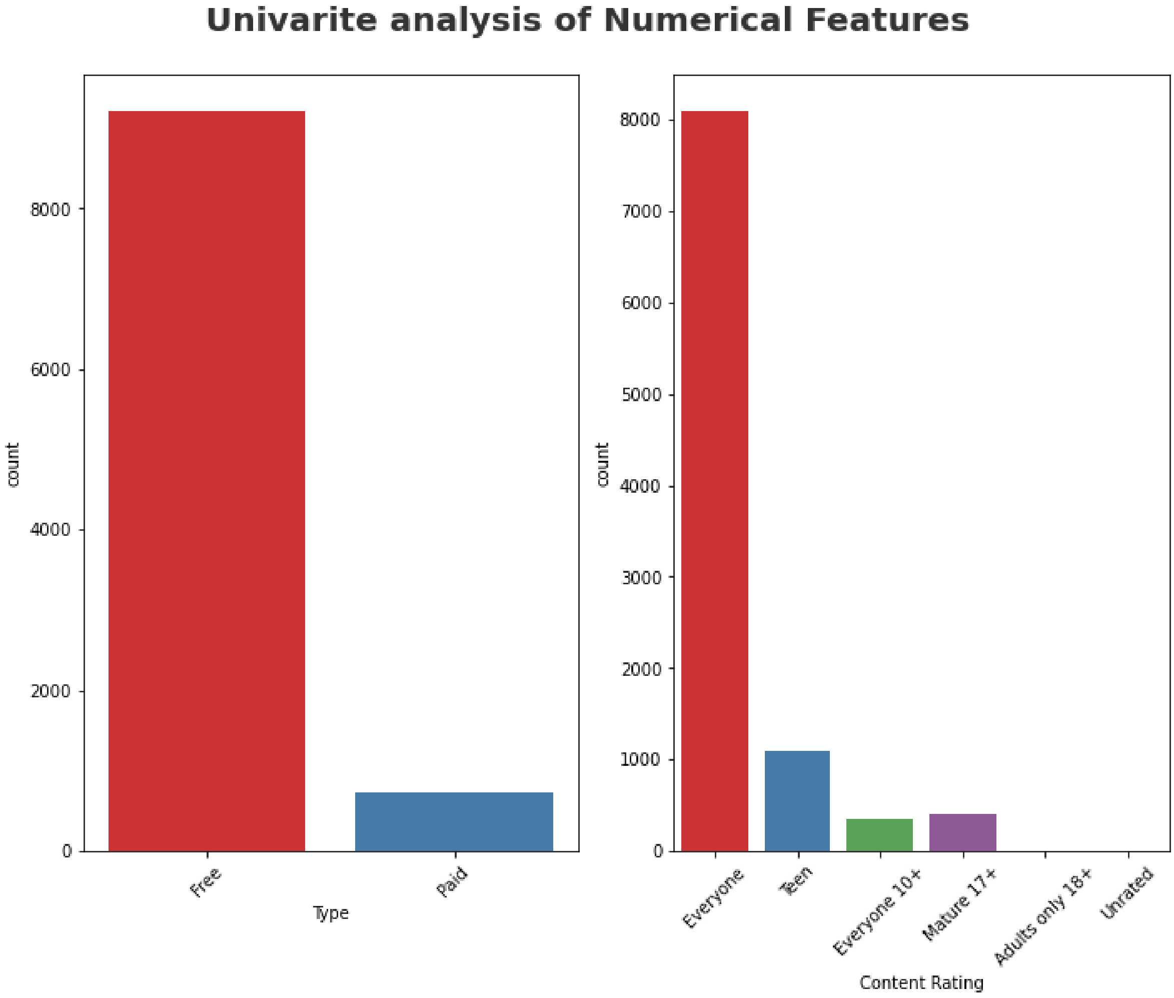
```
<AxesSubplot:xlabel='Content Rating', ylabel='count'>
```



In [48]:

```
plt.figure(figsize= (10,15))
plt.suptitle("Univarite analysis of Numerical Features", fontsize= 20,fontweight= "bold
category=["Type","Content Rating"]   # we ploted for these two features only

for i in range(0,len(category)):
    plt.subplot(2,2, i+1)
    sns.countplot(x= df[category[i]], palette="Set1")
    plt.xlabel( category[i])
    plt.xticks(rotation = 45)
    plt.tight_layout()
```



if the categories are 5-8 then its good to use count plot but if gategories are more then its complicated.

In [49]:

```
cat_df["Genres"].unique()
cat_df["Genres"].value_counts()
```

Out[49]:

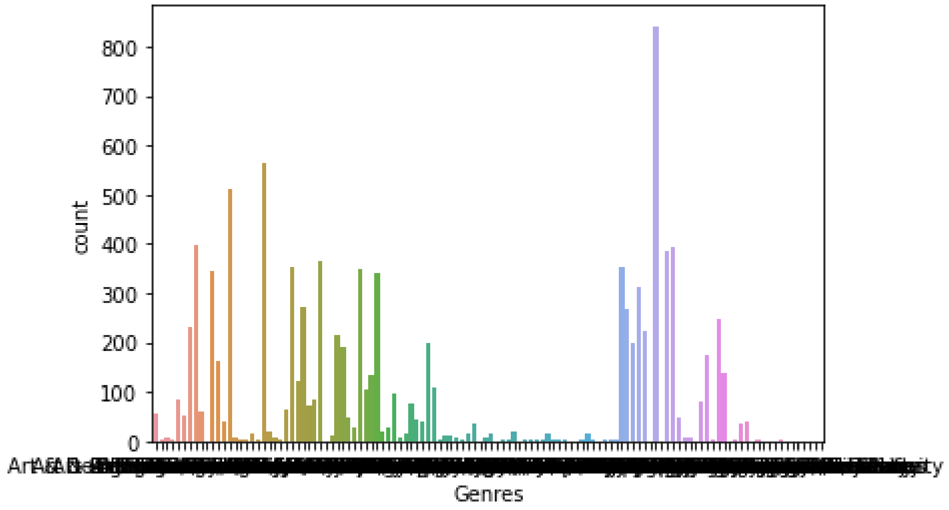
```
Tools                            842
Entertainment                    563
Education                        510
Business                         398
Productivity                     392
                                ... 
Health & Fitness;Education         1
Music & Audio;Music & Video        1
Arcade;Pretend Play                1
Entertainment;Education            1
Strategy;Creativity                1
Name: Genres, Length: 118, dtype: int64
```

In [50]:

```
# 118 categories
sns.countplot(cat_df["Genres"])
```

Out[50]:

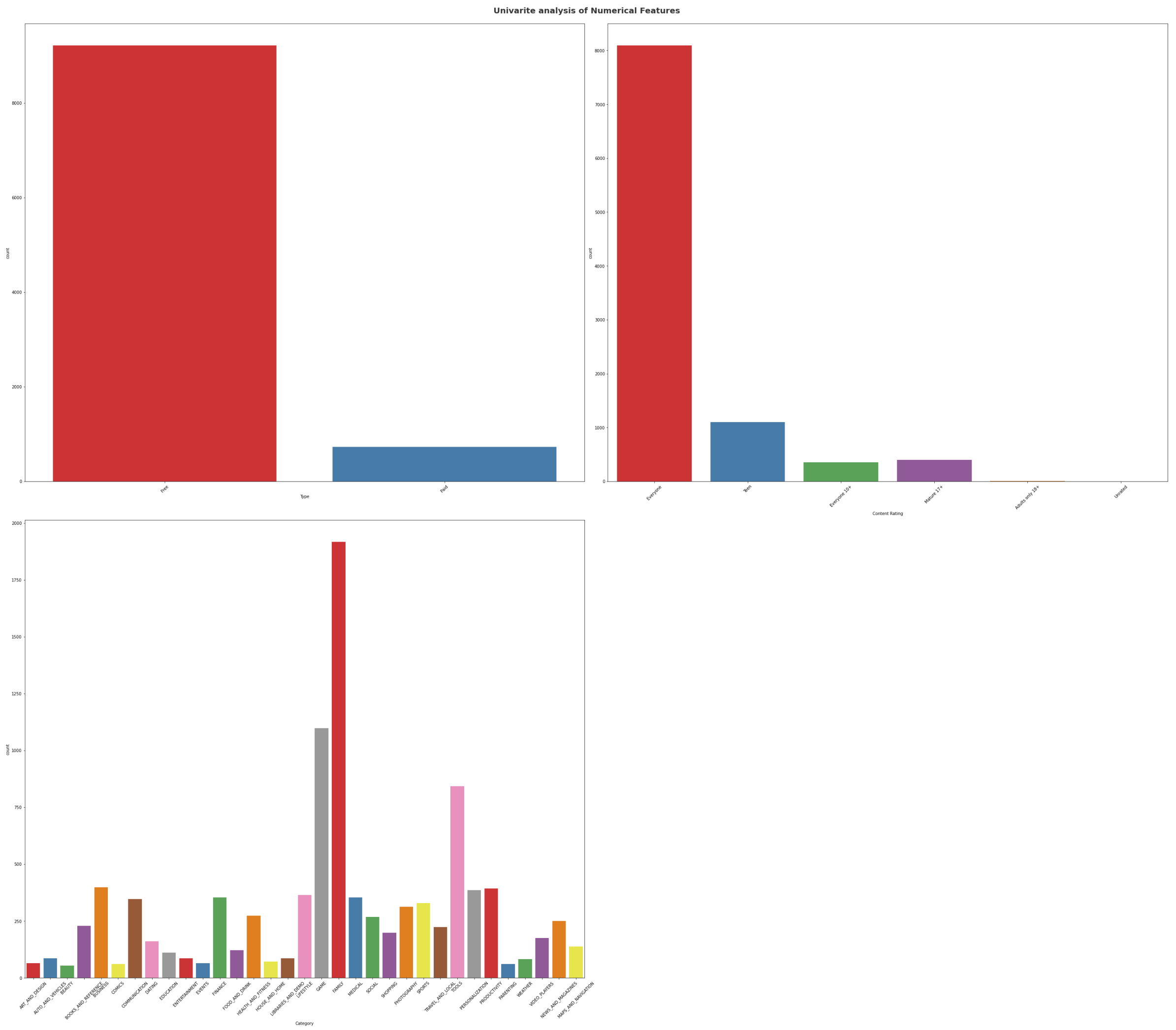
```
<AxesSubplot:xlabel='Genres', ylabel='count'>
```



In [51]:

```
plt.figure(figsize= (40,35))
plt.suptitle("Univarite analysis of Numerical Features", fontsize= 20,fontweight= "bold
category=["Type","Content Rating","Category"]   # we ploted for these two features only

for i in range(0,len(category)):
    plt.subplot(2,2, i+1)
    sns.countplot(x= df[category[i]], palette="Set1")
    plt.xlabel( category[i])
    plt.xticks(rotation = 45)
    plt.tight_layout()
```



qauestion: which one is most popular category

try to write a code

use Pie plot

(count plot and bar plot is same)

In [52]:

```
cat_df["Category"].value_counts()
# family is most popular category
```

Out[52]:

```
FAMILY                 1917
GAME                   1098
TOOLS                   843
BUSINESS                398
PRODUCTIVITY            392
PERSONALIZATION         385
LIFESTYLE               365
FINANCE                 354
MEDICAL                 354
COMMUNICATION           346
SPORTS                  328
PHOTOGRAPHY             312
HEALTH_AND_FITNESS      273
SOCIAL                  267
NEWS_AND_MAGAZINES      249
BOOKS_AND_REFERENCE     229
TRAVEL_AND_LOCAL        223
SHOPPING                199
VIDEO_PLAYERS           175
DATING                  160
MAPS_AND_NAVIGATION     137
FOOD_AND_DRINK          121
EDUCATION               111
AUTO_AND_VEHICLES        85
LIBRARIES_AND_DEMO       85
ENTERTAINMENT            85
WEATHER                  82
HOUSE_AND_HOME           72
ART_AND_DESIGN           65
EVENTS                   64
PARENTING                60
COMICS                   60
BEAUTY                   53
Name: Category, dtype: int64
```

In [53]:

```python
# Pie plot for family
sns.pieplot(cat_df["Category"])
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [53], in <cell line: 2>()
      1 # Pie plot for family
----> 2 sns.pieplot(cat_df["Category"])

AttributeError: module 'seaborn' has no attribute 'pieplot'
```

In [ ]:

```python
cat_df["Category"].value_counts().plot.pie()
```

In [ ]:

```python
# enhance Figure size
cat_df["Category"].value_counts().plot.pie(figsize=(15,20))
plt.suptitle("Pie chart", fontsize= 20,fontweight= "bold",alpha=0.8,y=0.5)
```

In [ ]:

```python
# enhance Figure size
plt.suptitle("Pie chart", fontsize= 20,fontweight= "bold",alpha=0.8,y=0.8)
cat_df["Category"].value_counts().plot.pie(figsize=(15,20))

```

In [ ]:

```python
# to get the percentages
plt.suptitle("Pie chart", fontsize= 20,fontweight= "bold",alpha=0.8,y=0.8)
cat_df["Category"].value_counts().plot.pie(figsize=(15,20),autopct='%.0f%%')

# plotting data on chart
#plt.pie(cat_df["Category"].value_counts(), colors=palette_color, autopct='%.0f%%')

# displaying chart
plt.show()
```

Q: Write a code to get top ten app categories

In [ ]:

```python
cat_df["Category"].value_counts()
```

In [ ]:

```python
# put above in new data frame
pd.DataFrame(cat_df["Category"].value_counts())
```

In [ ]:

```
# Take this data frame in new variable called category
category=pd.DataFrame(cat_df["Category"].value_counts())
```

In [ ]:

```
# we got top 10 categories
category.head(10)
```

In [ ]:

```
# rename category heading to count
category.rename(columns={"Category":"counts"},inplace= True)
category
```

plot the only 10 values

reset the index

In [ ]:

```
#  sns.countplot(x= category.index[:10],y= "counts",data=category[:10])
#sns do not gives output so use bar graph
sns.barplot(x= category.index[:10],y= "counts",data=category[:10])
```

In [ ]:

```
# unable to see properly so pass figure size
plt.figure(figsize=(20,20))
sns.barplot(x= category.index[:10],y= "counts",data=category[:10])
```

Q: which category has largest installation

In [ ]:

```
cat_df.head(10)
```

In [ ]:

```
# see here category and Installations
df.head()
```

In [ ]:

```
num_df.head()
```

In [ ]:

```
df.groupby(["Category"])["Installs"].sum()
```

In [ ]:

```
# arrange in sort manner
df.groupby(["Category"])["Installs"].sum().sort_values()
```

In [ ]:

```
# Arrange in desendings
df.groupby(["Category"])["Installs"].sum().sort_values(ascending= False)
```

In [ ]:

```
# another way to get top 10 largest values
df.groupby(["Category"])["Installs"].sum().nlargest(10)
```

we can plot this by bar plot or Pie plot

In [ ]:

```
df.groupby(["Category"])["Installs"].sum().nlargest(10).plot.pie(figsize= (12,12))
```

Bar Plot

In [ ]:

```
df.groupby(["Category"])["Installs"].sum().nlargest(10).plot.bar(figsize= (12,12))
#df.groupby(["Category"])["Installs"].sum().nlargest(10).plot.(kind= "bar", figsize= (1
```

qestions

1.how many apps are there on google play store which get 5 ratings?

2.does size of the application has any impact on its popularity?

3.what are the top 5 most installed apps in each popular category?

4.which category app users are reviewing the most?

5.which kind of app user are downloading the most free/paid?

liner regression

you need to create a model(linear regression) where all the features except Rating will be independent features and rating will be a dependent feature

####### create a model : https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html)

In [54]:

```
#1.how many apps are there on google play store which get 5 ratings?
df[df["Rating"]==5]  # get the data which has rating 5
df[df["Rating"]==5]["App"] # there are 268 app's whoes rating are 5
df[df["Rating"]==5]["App"].count()
a=df[df["Rating"]==5]["App"].count()
print("Total Apps on google play store which get 5 ratings are:",a)
```

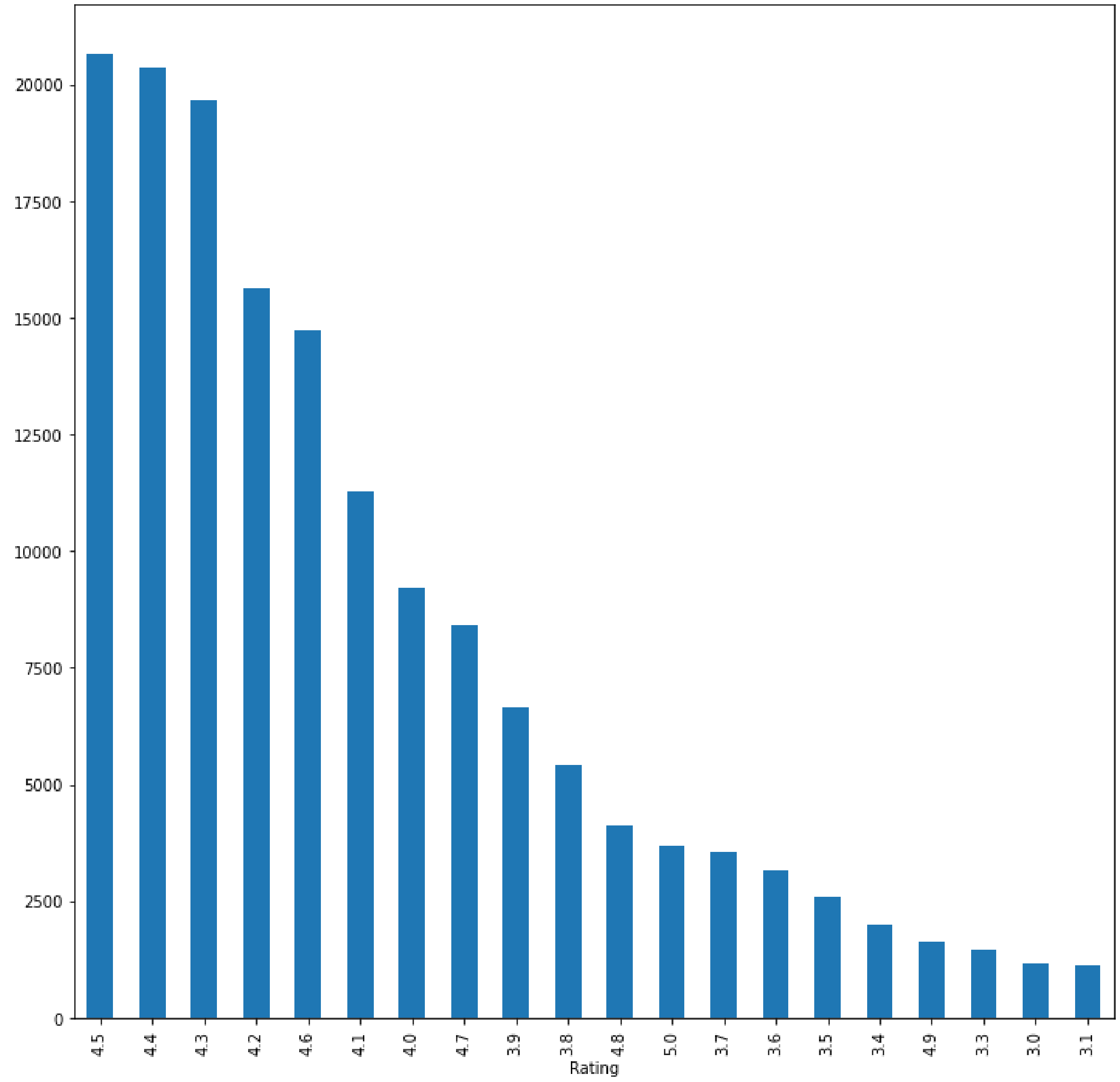
```
Total Apps on google play store which get 5 ratings are: 268
```

In [55]:

```
# 2.does size of the application has any impact on its popularity?
df.groupby(["Rating"])["Size"].sum().sort_values(ascending= False)
df.groupby(["Rating"])["Size"].sum().sort_values(ascending= False).nlargest(20).plot.ba
#Yes as size increases the rating increases
```

Out[55]:

```
<AxesSubplot:xlabel='Rating'>
```



In [56]:

```
df.groupby(["Rating"])["Size"].sum().sort_values(ascending= False)
```

Out[56]:

```
Rating
4.5    20669.600
4.4    20368.257
4.3    19641.206
4.2    15641.182
4.6    14742.759
4.1    11274.782
4.0     9208.417
4.7     8423.204
3.9     6659.470
3.8     5415.697
4.8     4110.963
5.0     3690.552
3.7     3554.864
3.6     3142.002
3.5     2595.239
3.4     1974.876
4.9     1630.434
3.3     1453.582
3.0     1171.716
3.1     1124.628
3.2      794.176
2.8      675.672
2.9      672.685
2.3      344.600
2.5      340.862
2.7      335.270
2.4      275.809
1.0      228.800
2.6      218.836
2.2      217.400
2.0      191.400
2.1      164.200
1.9      154.200
1.7       80.300
1.6       64.600
1.5       54.000
1.8       49.087
1.2       27.000
1.4       20.200
Name: Size, dtype: float64
```

In [57]:

```
#3.what are the top 5 most installed apps in each popular category?
df.groupby(["App","Category"])["Installs"].sum().sort_values(ascending= False).head(5)

```

Out[57]:

```
App                           Category
Hangouts                      COMMUNICATION       4000000000
Subway Surfers                GAME                4000000000
Google Photos                 PHOTOGRAPHY         4000000000
Maps - Navigate & Explore     TRAVEL_AND_LOCAL    3000000000
Google Chrome: Fast & Secure  COMMUNICATION       3000000000
Name: Installs, dtype: int64
```

In [ ]:

```
# there are communication, game,photography,TRAVEL_AND_LOCAL , COMMUNICATION    categor
```

In [60]:

```
#4.which category app users are reviewing the most?
df.groupby(["Category"])["Reviews"].sum().sort_values(ascending= False)

```

Out[60]:

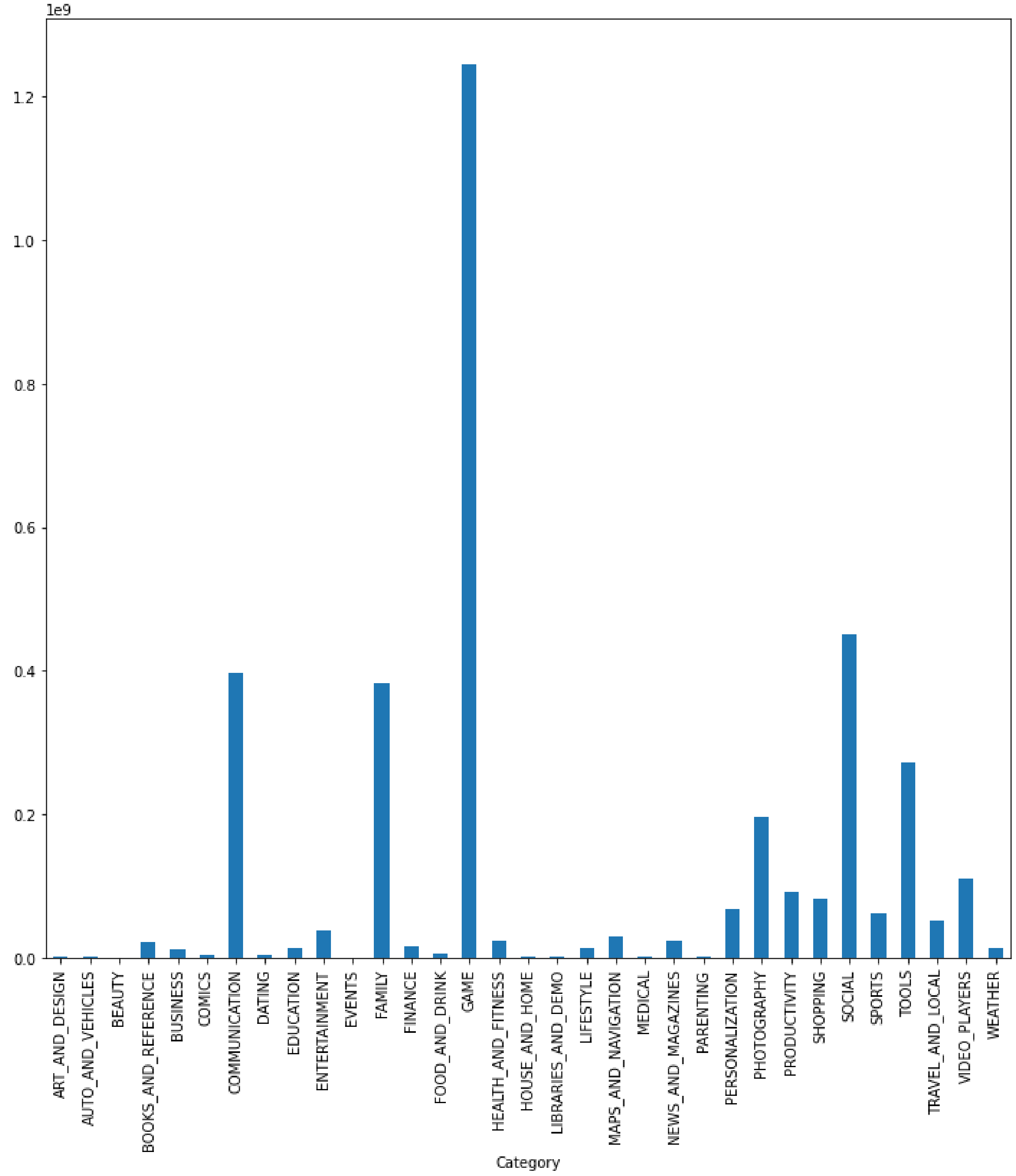
```
Category
GAME                   1245650951
SOCIAL                  450953900
COMMUNICATION           397569013
FAMILY                  383338162
TOOLS                   273185044
PHOTOGRAPHY             195466914
VIDEO_PLAYERS           110380188
PRODUCTIVITY             92374969
SHOPPING                 82317633
PERSONALIZATION          67506827
SPORTS                   61996740
TRAVEL_AND_LOCAL         51260122
ENTERTAINMENT            37854634
MAPS_AND_NAVIGATION      30659254
HEALTH_AND_FITNESS       23936793
NEWS_AND_MAGAZINES       22979922
BOOKS_AND_REFERENCE      21787385
FINANCE                  16449054
WEATHER                  14604735
EDUCATION                13988229
LIFESTYLE                12760163
BUSINESS                 11022817
FOOD_AND_DRINK            6459822
DATING                    3799655
COMICS                    3383276
ART_AND_DESIGN            1714440
HOUSE_AND_HOME            1613159
MEDICAL                   1207546
AUTO_AND_VEHICLES         1163666
LIBRARIES_AND_DEMO        1037118
PARENTING                  958331
BEAUTY                     396240
EVENTS                     161018
Name: Reviews, dtype: int64
```

In [59]:

```
df.groupby(["Category"])["Reviews"].sum().plot.bar(figsize= (12,12))
# Ans: App category GAME reviews a most which is equal to 1245650951
```

Out[59]:

```
<AxesSubplot:xlabel='Category'>
```



In [58]:

```
#5.which kind of app user are downloading the most free/paid?
# check the different categories in "Type" Feature
cat_df["Type"].value_counts()
#df.groupby(["Type","App"])["Installs"].sum().sort_values(ascending= False)
df.groupby(["Type","App"])["Installs"].value_counts().sort_values(ascending= False)
# Ans : below are the details of free an paid user app's with downloading detals
```

Out[58]:

```
Type  App                              Installs
Free  ROBLOX                           100000000    9
      8 Ball Pool                      100000000    7
      Zombie Catchers                  10000000     6
      Helix Jump                       100000000    6
      Angry Birds Classic              100000000    5
                                                   ..
      Dumb Ways to Die 2: The Games    50000000     1
      Dulquer Salmaan HD Wallpapers    100          1
      Dude Perfect                     10000        1
      Dubsmash                         100000000    1
Paid  💎 I'm rich                        10000        1
Name: Installs, Length: 9393, dtype: int64
```

In [ ]:

```
df.head(5)
```

In [ ]:

```
cat_df.head()
```

In [ ]:

```
# plot pie plot for type
df.value_counts("Type").plot(kind ="pie",figsize = (10,10))
```

In [ ]:

```
df.value_counts("Type").plot(kind ="bar",figsize = (10,10))
```

In [ ]:

```
# Linear Regression
num_df.head()
```

create linear regression model where all the features except rating will be a independent feature and rating

will be a dependent feature. so use

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html)

In [ ]:

```

```