In [1]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\dell\anaconda3\envs\pythonclassesjan
2023\lib\site-packages (1.24.1)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\dell\anaconda3\envs\pythonclassesja
n2023\lib\site-packages (1.5.3)
Requirement already satisfied: numpy>=1.20.3 in c:\users\dell\anaconda3\envs\pythoncl
assesjan2023\lib\site-packages (from pandas) (1.24.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dell\anaconda3\envs
\pythonclassesjan2023\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\envs\pythoncla
ssesjan2023\lib\site-packages (from pandas) (2022.7.1)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\envs\pythonclasses
jan2023\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\dell\anaconda3\envs\pythonclassesj
an2023\lib\site-packages (0.12.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\dell\anaconda3\env
s\pythonclassesjan2023\lib\site-packages (from seaborn) (3.6.3)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\users\dell\anaconda3\envs\p
ythonclassesjan2023\lib\site-packages (from seaborn) (1.24.1)
Requirement already satisfied: pandas>=0.25 in c:\users\dell\anaconda3\envs\pythoncla
ssesjan2023\lib\site-packages (from seaborn) (1.5.3)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\envs\pythoncla
ssesjan2023\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda3\envs\pyth
onclassesjan2023\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda3\envs\pytho
nclassesjan2023\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anaconda3\envs\p
ythonclassesjan2023\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\envs\pythoncl
assesjan2023\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (9.4.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dell\anaconda3\envs\pytho
nclassesjan2023\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.0.7)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\envs\python
classesjan2023\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (22.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anaconda3\envs\pyth
onclassesjan2023\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.38.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\envs\pythoncla
ssesjan2023\lib\site-packages (from pandas>=0.25->seaborn) (2022.7.1)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\envs\pythonclasses
jan2023\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seabor
n) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\envs\pythonclass
esjan2023\lib\site-packages (3.6.3)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anaconda3\envs\pyth
onclassesjan2023\lib\site-packages (from matplotlib) (4.38.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda3\envs\pytho
nclassesjan2023\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\envs\pythoncla
ssesjan2023\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dell\anaconda3\envs\pytho
nclassesjan2023\lib\site-packages (from matplotlib) (1.0.7)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda3\envs\pyth
onclassesjan2023\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\envs\pythoncl
assesjan2023\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anaconda3\envs\p
ythonclassesjan2023\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: numpy>=1.19 in c:\users\dell\anaconda3\envs\pythonclas
sesjan2023\lib\site-packages (from matplotlib) (1.24.1)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\envs\python
classesjan2023\lib\site-packages (from matplotlib) (22.0)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\envs\pythonclasses
jan2023\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

1.Load the data file using the pandas.

In [6]:
```
data = pd.read_csv('Dataset of App rating project/googleplaystore.csv')
```

In [7]:
```
data
```

Out[7]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Conte Ratir
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyor
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyor
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyor
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Tee
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyor
...	...	...	...	...	...	...	...	...	
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyor
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyor
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyor
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Matu 17
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyor

10841 rows × 13 columns

In [8]:
```
data.head()
```

Out[8]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	G
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & D
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Design;Pr
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & D
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & D
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Design;Cre

In [9]:
```
data.tail()
```

Out[9]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Conte Ratir
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyor
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyor
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyor
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Matu 17
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyor

In [10]:
```
data.shape
```

Out[10]:
```
(10841, 13)
```

2) check the null values in the data. Get the number of null values for each column.

In [11]:
```
# To check null values in the data.
data.isnull()
```

Out[11]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Curr
0	False	False	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...	
10836	False	False	False	False	False	False	False	False	False	False	False	F
10837	False	False	False	False	False	False	False	False	False	False	False	F
10838	False	False	True	False	False	False	False	False	False	False	False	F
10839	False	False	False	False	False	False	False	False	False	False	False	F
10840	False	False	False	False	False	False	False	False	False	False	False	F

10841 rows × 13 columns

In [12]:
```
# null values for each column.
data.isnull().sum()
```

Out[12]:
```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     1
Genres             0
Last Updated       0
Current Ver        8
Android Ver        3
dtype: int64
```

In [13]:
```
data.isnull().sum().sum()
```

Out[13]:
```
1487
```

3) Drop the Records with nulls in any of the columns.

In [14]:
```
data.dropna(inplace=True)
```

In [15]:
```
data.isnull().sum().sum()
```

Out[15]:
```
0
```

4.1) Size column has sizes in Kb as well as Mb. To analyze, you'll need to convert these to numeric.

4.1.1)Extract the numeric value from the column.

In [16]:
```
data.Size
```

Out[16]:
```
0                     19M
1                     14M
2                    8.7M
3                     25M
4                    2.8M
                ...
10834                2.6M
10836                 53M
10837                3.6M
10839    Varies with device
10840                 19M
Name: Size, Length: 9360, dtype: object
```

In [17]:
```
data['Size'] = data.Size.str.replace('M', '')
```

In [18]:
```
data.Size
```

Out[18]:
```
0                     19
1                     14
2                    8.7
3                     25
4                    2.8
                ...
10834                2.6
10836                 53
10837                3.6
10839    Varies with device
10840                 19
Name: Size, Length: 9360, dtype: object
```

In [19]:
```
data.drop(data[data.Size.str.len() > 4].index, inplace=True)
```

In [20]:
```
data.Size
```

Out[20]:
```
0         19
1         14
2        8.7
3         25
4        2.8
        ...
10833    619k
10834     2.6
10836      53
10837     3.6
10840      19
Name: Size, Length: 7723, dtype: object
```

4.1.2) Multiply the value by 1,000, if size is mentioned in Mb.

In [21]:
```
data['Size']=data[~data.Size.str.contains('k')]['Size'].astype(float) * 1000
```

In [22]:
```
data.Size
```

Out[22]:
```
0        19000.0
1        14000.0
2         8700.0
3        25000.0
4         2800.0
          ...
10833        NaN
10834     2600.0
10836    53000.0
10837     3600.0
10840    19000.0
Name: Size, Length: 7723, dtype: float64
```

In [23]:
```
data.isnull().mean()
# there is no need to drop null values in size as 30% null values in the column is the
```

Out[23]:
```
App               0.000000
Category          0.000000
Rating            0.000000
Reviews           0.000000
Size              0.033277
Installs          0.000000
Type              0.000000
Price             0.000000
Content Rating    0.000000
Genres            0.000000
Last Updated      0.000000
Current Ver       0.000000
Android Ver       0.000000
dtype: float64
```

4.2) Reviews is a numeric field that is loaded as a string field. Convert it to numeric (int/float).

In [24]:
```
data['Reviews']=data.Reviews.astype(float)
```

In [25]:
```
data.Reviews
```

Out[25]:
```
0           159.0
1           967.0
2         87510.0
3        215644.0
4           967.0
           ...
10833        44.0
10834         7.0
10836        38.0
10837         4.0
10840    398307.0
Name: Reviews, Length: 7723, dtype: float64
```

4.3) Installs field is currently stored as string and has values like 1,000,000+.

4.3.1) Treat 1,000,000+ as 1,000,000.

4.3.2) remove '+', ',' from the field, convert it to integer.

In [26]:
```
data['Installs']=data.Installs.str.replace('[,+]', '').astype(int)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_11632\1366402930.py:1: FutureWarning: The
default value of regex will change from True to False in a future version.
  data['Installs']=data.Installs.str.replace('[,+]', '').astype(int)
```

4.4) Price field is a string and has ' Dollar symbol. Remove 'Dollar' sign, and convert it to numeric.

In [27]:
```
data['Price']=data.Price.str.replace('$', '')
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_11632\3213578771.py:1: FutureWarning: The
default value of regex will change from True to False in a future version. In additio
n, single character regular expressions will *not* be treated as literal strings when
regex=True.
  data['Price']=data.Price.str.replace('$', '')
```

In [28]:
```
data['Price']=data.Price.astype(float)
```

In [29]:
```
data.Price
```

Out[29]:
```
0        0.0
1        0.0
2        0.0
3        0.0
4        0.0
        ...
10833    0.0
10834    0.0
10836    0.0
10837    0.0
10840    0.0
Name: Price, Length: 7723, dtype: float64
```

4.5) Sanity checks:

4.5.1) Average rating should be between 1 and 5 as only these values are allowed on the play store. Drop the rows that have a value outside this range.

In [30]:
```
data[(data['Rating'] > 1) & (data['Rating'] < 5)]
#Average rating which are between 1 and 5 as only these values are allowed on the play
```

Out[30]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Conten Ratin
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159.0	19000.0	10000	Free	0.0	Everyon
1	Coloring book moana	ART_AND_DESIGN	3.9	967.0	14000.0	500000	Free	0.0	Everyon
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510.0	8700.0	5000000	Free	0.0	Everyon
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644.0	25000.0	50000000	Free	0.0	Tee
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967.0	2800.0	100000	Free	0.0	Everyon
...	...	...	...	...	...	...	...	...	...
10832	FR Tides	WEATHER	3.8	1195.0	NaN	100000	Free	0.0	Everyon
10833	Chemin (fr)	BOOKS_AND_REFERENCE	4.8	44.0	NaN	1000	Free	0.0	Everyon
10834	FR Calculator	FAMILY	4.0	7.0	2600.0	500	Free	0.0	Everyon
10836	Sya9a Maroc - FR	FAMILY	4.5	38.0	53000.0	5000	Free	0.0	Everyon
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307.0	19000.0	10000000	Free	0.0	Everyon

7438 rows × 13 columns

In [31]:
```
data[(data['Rating'] < 1) & (data['Rating'] > 5)]
# there are no values outside the range.So, there are no values to drop.
```

Out[31]:

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	A

4.5.2) Reviews should not be more than installs as only those who installed can review the app. If there are any such records, drop them.

In [32]:
```
data[(data['Reviews']) > (data['Installs'])]
```

Out[32]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated
2454	KBA-EZ Health Guide	MEDICAL	5.0	4.0	25000.0	1	Free	0.00	Everyone	Medical	August 2, 2018
5917	Ra Ga Ba	GAME	5.0	2.0	20000.0	1	Paid	1.49	Everyone	Arcade	February 8, 2017
6700	Brick Breaker BR	GAME	5.0	7.0	19000.0	5	Free	0.00	Everyone	Arcade	July 23, 2018
7402	Trovami se ci riesci	GAME	5.0	11.0	6100.0	10	Free	0.00	Everyone	Arcade	March 11, 2017
8591	DN Blog	SOCIAL	5.0	20.0	4200.0	10	Free	0.00	Teen	Social	July 23, 2018
10697	Mu.F.O.	GAME	5.0	2.0	16000.0	1	Paid	0.99	Everyone	Arcade	March 3, 2017

In [33]:
```
data.drop(data[data['Reviews'] > data['Installs']].index,inplace=True)
```

In [34]:
```
data[(data['Reviews']) > (data['Installs'])]
# now as we dropped there no are reviews which are more than installs.
```

Out[34]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	A

4.5.3) For free apps (Type = “Free”), the price should not be >0. Drop any such rows.

In [35]:
```
data[(data.Type == 'Free') & (data.Price > 0)]
# no free apps has price greater than 0
```

Out[35]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	A

In [36]:
```
data
```

Out[36]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Conten Ratin
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159.0	19000.0	10000	Free	0.0	Everyon
1	Coloring book moana	ART_AND_DESIGN	3.9	967.0	14000.0	500000	Free	0.0	Everyon
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510.0	8700.0	5000000	Free	0.0	Everyon
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644.0	25000.0	50000000	Free	0.0	Tee
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967.0	2800.0	100000	Free	0.0	Everyon
...	...	...	...	...	...	...	...	...	.
10833	Chemin (fr)	BOOKS_AND_REFERENCE	4.8	44.0	NaN	1000	Free	0.0	Everyon
10834	FR Calculator	FAMILY	4.0	7.0	2600.0	500	Free	0.0	Everyon
10836	Sya9a Maroc - FR	FAMILY	4.5	38.0	53000.0	5000	Free	0.0	Everyon
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4.0	3600.0	100	Free	0.0	Everyon
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307.0	19000.0	10000000	Free	0.0	Everyon

7717 rows × 13 columns

5) Performing univariate analysis:

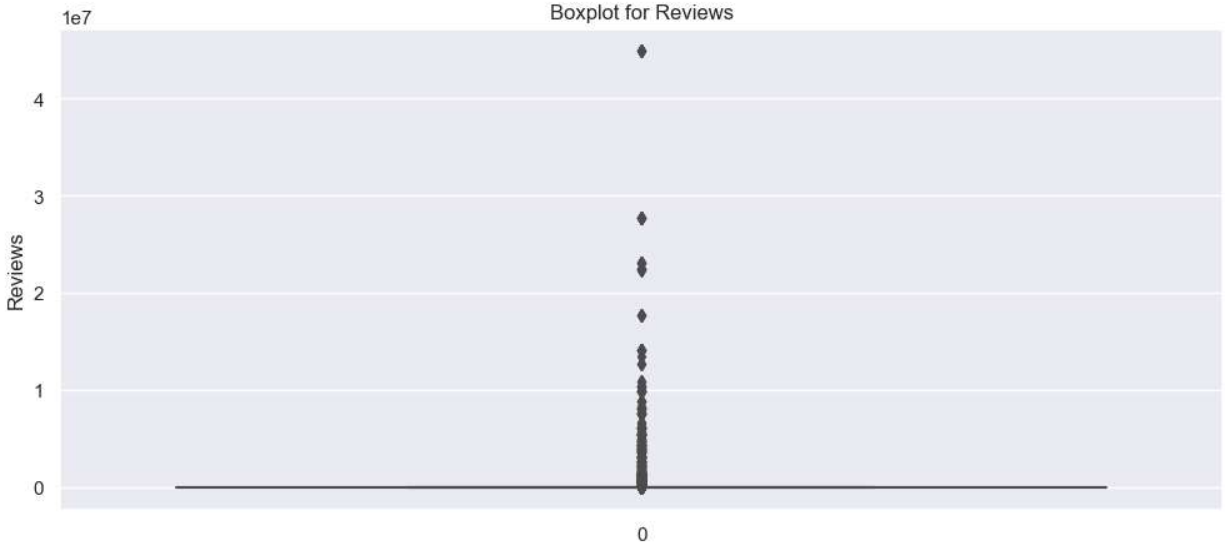
- Boxplot for Price
- Boxplot for Reviews

- Histogram for Rating
- Histogram for Size

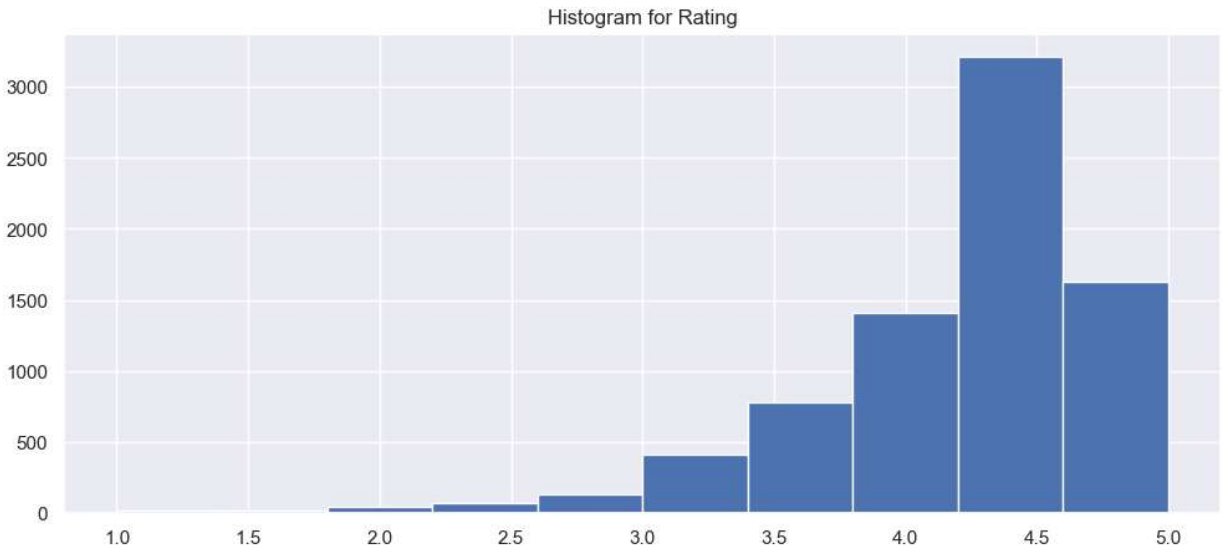
In [37]:
```
sns.set(rc={"figure.figsize":(12,5)})
```

In [38]:
```
sns.boxplot(data['Price'])
plt.title('Boxplot for Price')
plt.ylabel('Price');
```

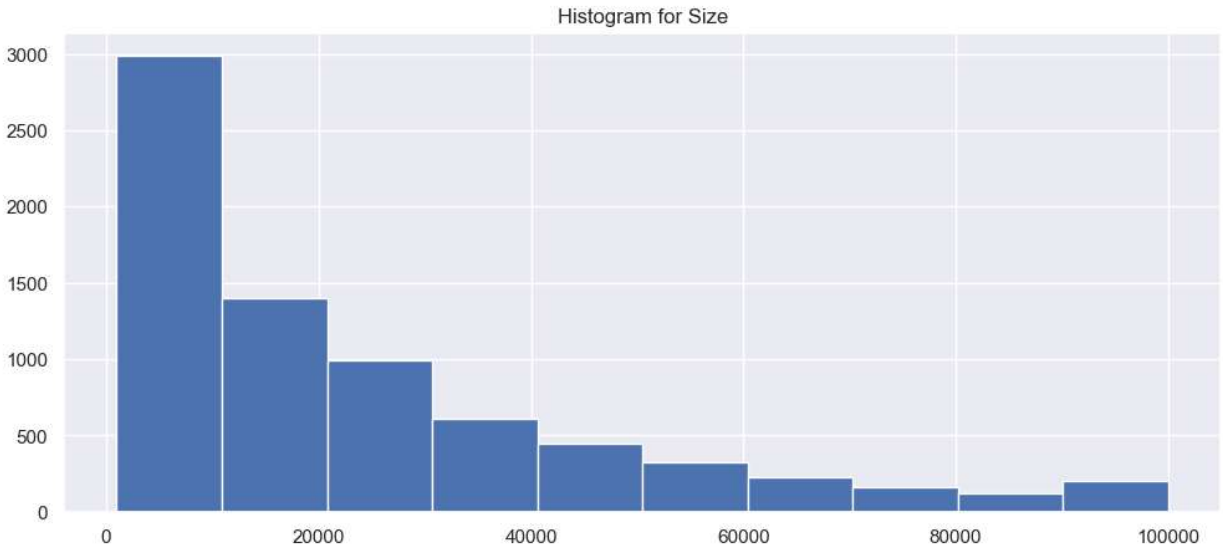
In [39]:
```
sns.boxplot(data['Reviews'])
plt.title('Boxplot for Reviews')
plt.ylabel('Reviews');
```



In [40]:
```
plt.hist(data.Rating);
plt.title('Histogram for Rating');
```



In [41]:
```
plt.hist(data['Size']);
plt.title('Histogram for Size');
```



6) Outlier treatment:

6.1) Price: From the box plot, it seems like there are some apps with very high price. A price of 200 dollars for an application on the Play Store is very high and suspicious!

6.1.1) Check out the records with very high price

6.2) Drop these as most seem to be junk apps

In [42]:
```
# checking the price of the apps which are more than 200 dollars.
# we have already removed dollar sign.
data[data['Price'] > 200]
```

Out[42]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genr
4197	most expensive app (H)	FAMILY	4.3	6.0	1500.0	100	Paid	399.99	Everyone	Entertainme
4362	💎 I'm rich	LIFESTYLE	3.8	718.0	26000.0	10000	Paid	399.99	Everyone	Lifesty
4367	I'm Rich - Trump Edition	LIFESTYLE	3.6	275.0	7300.0	10000	Paid	400.00	Everyone	Lifesty
5351	I am rich	LIFESTYLE	3.8	3547.0	1800.0	100000	Paid	399.99	Everyone	Lifesty
5354	I am Rich Plus	FAMILY	4.0	856.0	8700.0	10000	Paid	399.99	Everyone	Entertainme
5355	I am rich VIP	LIFESTYLE	3.8	411.0	2600.0	10000	Paid	299.99	Everyone	Lifesty
5356	I Am Rich Premium	FINANCE	4.1	1867.0	4700.0	50000	Paid	399.99	Everyone	Finan
5357	I am extremely Rich	LIFESTYLE	2.9	41.0	2900.0	1000	Paid	379.99	Everyone	Lifesty
5358	I am Rich!	FINANCE	3.8	93.0	22000.0	1000	Paid	399.99	Everyone	Finan
5359	I am rich(premium)	FINANCE	3.5	472.0	NaN	5000	Paid	399.99	Everyone	Finan
5362	I Am Rich Pro	FAMILY	4.4	201.0	2700.0	5000	Paid	399.99	Everyone	Entertainme
5364	I am rich (Most expensive app)	FINANCE	4.1	129.0	2700.0	1000	Paid	399.99	Teen	Finan
5366	I Am Rich	FAMILY	3.6	217.0	4900.0	10000	Paid	389.99	Everyone	Entertainme
5369	I am Rich	FINANCE	4.3	180.0	3800.0	5000	Paid	399.99	Everyone	Finan
5373	I AM RICH PRO PLUS	FINANCE	4.0	36.0	41000.0	1000	Paid	399.99	Everyone	Finan

In [43]:
```
data.drop(data[data['Price']>200].index,inplace=True)
```

In [44]:
```
data[data['Price'] > 200] # now we dropped the junk apps which are more than 200 dollo
```

Out[44]:

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	A

6.2) Reviews: Very few apps have very high number of reviews. These are all star apps that don't help with the analysis and, in fact, will skew it. Drop records having more than 2 million reviews.

In [45]:
```
#to check the apps with > 2000000 reviews
data[data['Reviews'] > 2000000]
# we can also use this command ....data[data['Reviews']>2000000]
```

Out[45]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
345	Yahoo Mail – Stay Organized	COMMUNICATION	4.3	4187998.0	16000.0	100000000	Free	0.0	Everyone
347	imo free video calls and chat	COMMUNICATION	4.3	4785892.0	11000.0	500000000	Free	0.0	Everyone
366	UC Browser Mini -Tiny Fast Private & Secure	COMMUNICATION	4.4	3648120.0	3300.0	100000000	Free	0.0	Teen
378	UC Browser - Fast Download Private & Secure	COMMUNICATION	4.5	17712922.0	40000.0	500000000	Free	0.0	Teen
383	imo free video calls and chat	COMMUNICATION	4.3	4785988.0	11000.0	500000000	Free	0.0	Everyone
...	...	...	...	...	...	...	...	...	...
9142	Need for Speed™ No Limits	GAME	4.4	3344300.0	22000.0	50000000	Free	0.0	Everyone 10+
9166	Modern Combat 5: eSports FPS	GAME	4.3	2903386.0	58000.0	100000000	Free	0.0	Mature 17+
10186	Farm Heroes Saga	FAMILY	4.4	7615646.0	71000.0	100000000	Free	0.0	Everyone
10190	Fallout Shelter	FAMILY	4.6	2721923.0	25000.0	10000000	Free	0.0	Teen
10327	Garena Free Fire	GAME	4.5	5534114.0	53000.0	100000000	Free	0.0	Teen

219 rows × 13 columns

In [46]:
```
data.drop(data[data['Reviews']>2000000].index,inplace=True)
# we can try in this way also
# data[data['Reviews']>2000000]
# data=data.dropna()
```

In [47]:
```
data[data['Reviews'] > 2000000]
# after dropping, now we cannot see the records having more than 2000000.
```

Out[47]:

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	A

6.3) Installs: There seems to be some outliers in this field too. Apps having very high number of installs should be dropped from the analysis.

-Find out the different percentiles – 10, 25, 50, 70, 90, 95, 99

-Decide a threshold as cutoff for outlier and drop records having values more than that.

In [48]:
```
data['Installs'].quantile([0.10, 0.25, 0.50, 0.70, 0.90, 0.95, 0.99])
```

Out[48]:
```
0.10        1000.0
0.25       10000.0
0.50      100000.0
0.70     1000000.0
0.90    10000000.0
0.95    10000000.0
0.99    50000000.0
Name: Installs, dtype: float64
```

In [49]:
```
import numpy as np
q1 = np.percentile(data['Installs'], 25)
q3 = np.percentile(data['Installs'], 75)
```

In [50]:
```
iqr=q3-q1
Threshold=1.5*iqr
q1,q3,iqr,Threshold
```

Out[50]:
```
(10000.0, 1000000.0, 990000.0, 1485000.0)
```

In [51]:
```
# Here our goal is to find the threshold(i.e is high value) as cutoff for outliers.
# we can define the threshold as any value that falls more than 1.5 times the IQR i.e
# below the first quartile or above the third quartile,and then remove any values that
```

In [52]:
```
Threshold=q3+1.5*iqr
Threshold
```

Out[52]:
```
2485000.0
```

In [53]:
```
data.drop(data[data['Installs']>2485000.0].index,inplace=True)
```

In [54]:
```
data.shape
```

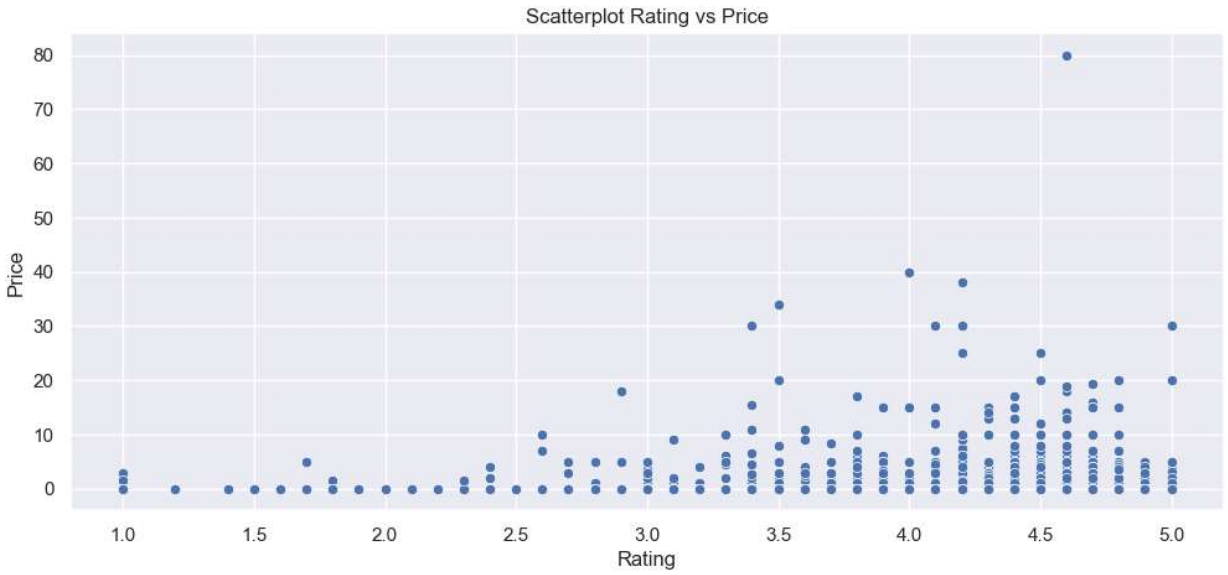
Out[54]:
```
(5954, 13)
```

7) Bivariate analysis: Let's look at how the available predictors relate to the variable of interest, i.e., our target variable rating. Make scatter plots (for numeric features) and box plots (for character features) to

assess the relations between rating and the other features.

7.1) Make scatter plot/joinplot for Rating vs. Price

In [55]:
```
sns.scatterplot(x='Rating', y='Price',data=data);
plt.title('Scatterplot Rating vs Price');
```

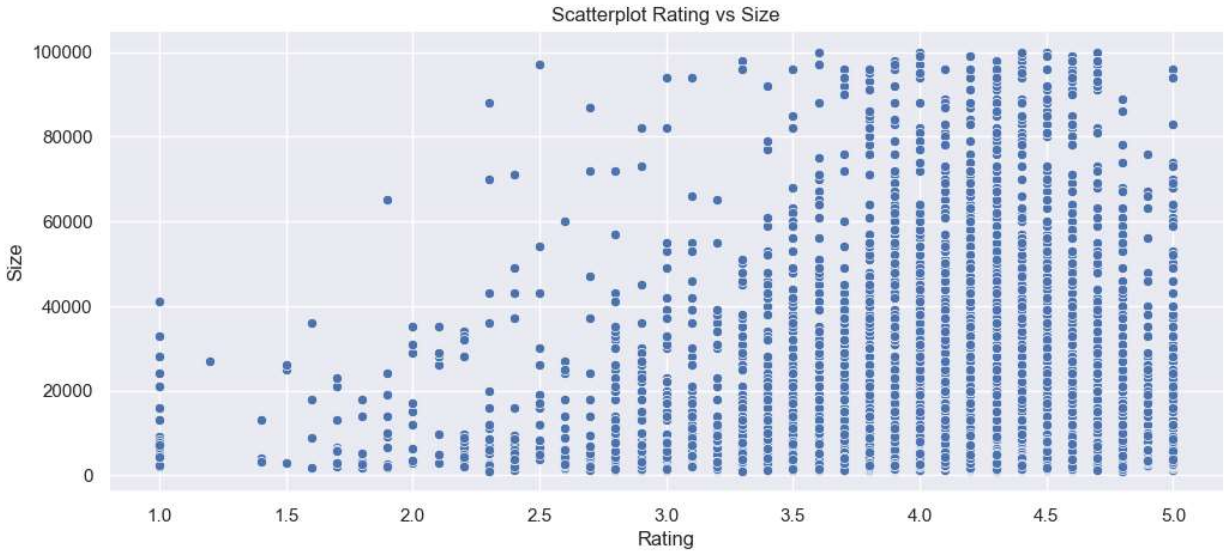


7.1.1)What pattern do you observe? Does rating increase with price?

Observation1 : We can understand that Rating increased with Price by observing the above plot.

7.2) Make scatter plot/joinplot for Rating vs. Size

In [56]:
```
sns.scatterplot(x='Rating', y='Size',data=data);
plt.title('Scatterplot Rating vs Size');
```

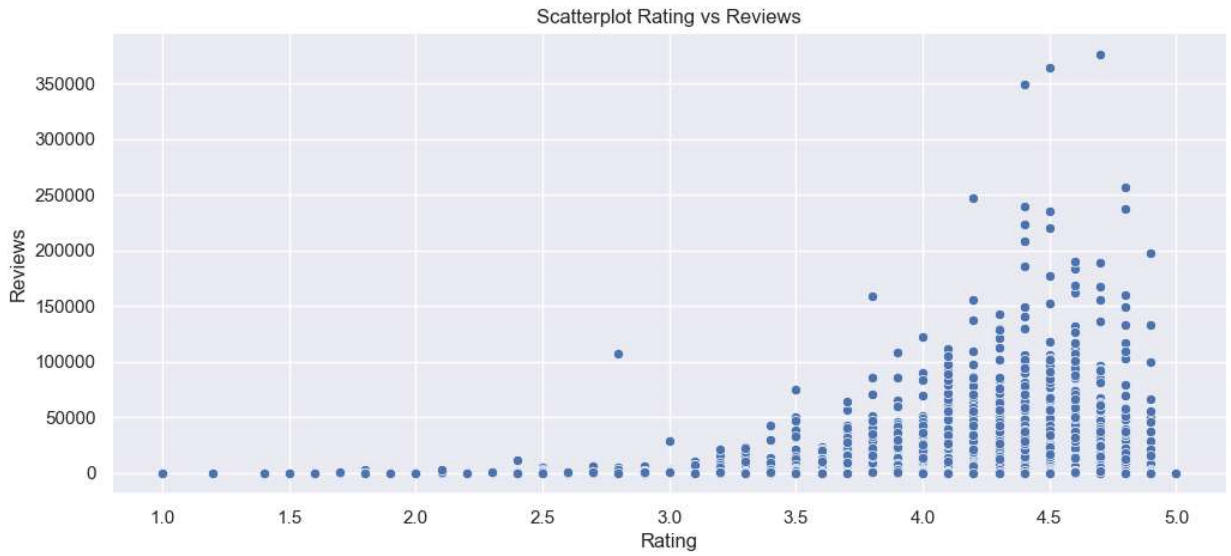


7.2.1) Are heavier apps rated better?

Observation2 : Yes,heavier apps rated better.

7.3) Make scatter plot/joinplot for Rating vs. Reviews

In [57]:
```
sns.scatterplot(x='Rating', y='Reviews',data=data);
plt.title('Scatterplot Rating vs Reviews');
```

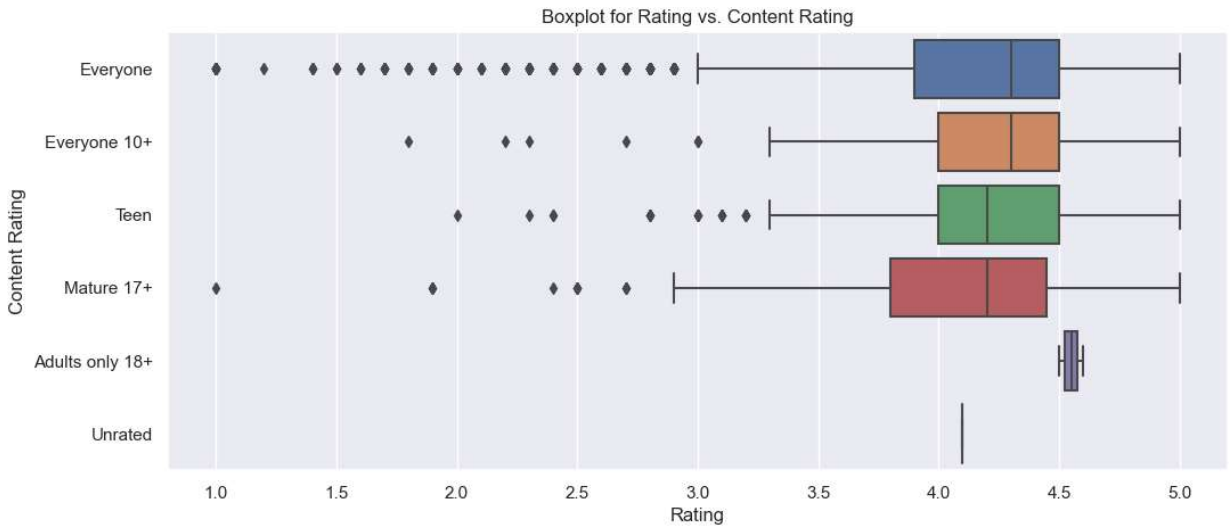


7.3.1) Does more review mean a better rating always?

Observation3 : Not necessarily,the number of reviews can provide an indication of the popularity and user engagement of an app, but it does not necessarily guarantee a better rating always. But in this case,by the above plot we understood that More reviews having better rating.

7.4) Make boxplot for Rating vs. Content Rating

In [58]:
```
sns.boxplot(x='Rating', y='Content Rating',data=data);
plt.title('Boxplot for Rating vs. Content Rating');
```

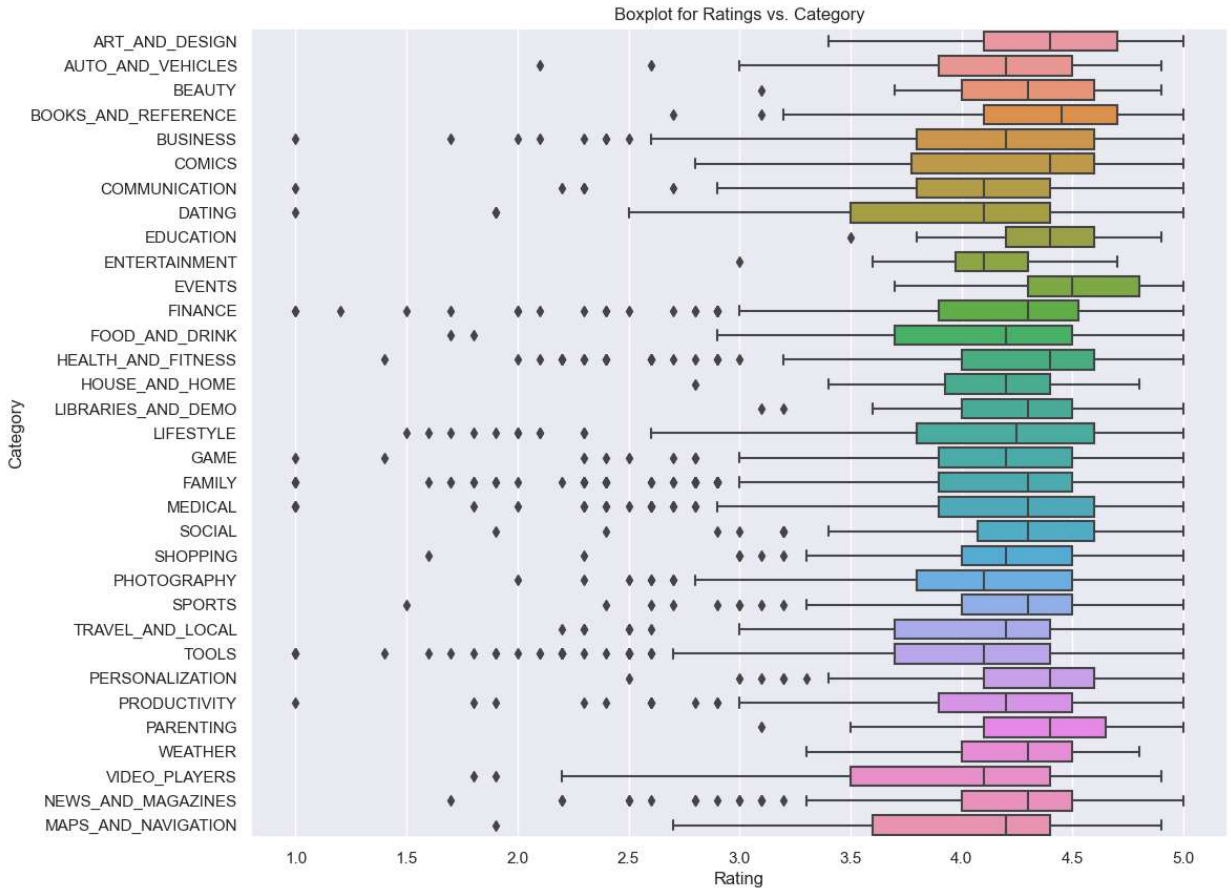


7.4.1) Is there any difference in the ratings? Are some types liked better?

Observation4 : No, there is not much difference between the ratings of most apps, except for those rated as "adults only 18+," and those apps have higher ratings.

7.5) Make boxplot for Ratings vs. Category

In [59]:
```
fig, ax = plt.subplots(figsize=(12,10))
sns.boxplot(x='Rating', y='Category',data=data, ax=ax);
plt.title('Boxplot for Ratings vs. Category');
```



7.5.1) Which genre has the best ratings?

Observation5 : By the above plot we can see that Events Category has Best ratings.

8) Data preprocessing

- For the steps below, create a copy of the dataframe to make all the edits. Name it inp1.

In [60]:
```
inp1 = data
```

In [61]:
```
inp1.head()
```

Out[61]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Ge
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159.0	19000.0	10000	Free	0.0	Everyone	Art & D
1	Coloring book moana	ART_AND_DESIGN	3.9	967.0	14000.0	500000	Free	0.0	Everyone	Design;Pre
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967.0	2800.0	100000	Free	0.0	Everyone	Design;Crea
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167.0	5600.0	50000	Free	0.0	Everyone	Art & D
6	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178.0	19000.0	50000	Free	0.0	Everyone	Art & D

8.1) Reviews and Install have some values that are still relatively very high. Before building a linear regression model, you need to reduce the skew. Apply log transformation (np.log1p) to Reviews and Installs.

In [62]:
```
inp1.skew()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_11632\3545313420.py:1: FutureWarning: The
default value of numeric_only in DataFrame.skew is deprecated. In a future version, i
t will default to False. In addition, specifying 'numeric_only=None' is deprecated. S
elect only valid columns or specify the value of numeric_only to silence this warnin
g.
  inp1.skew()
```

Out[62]:
```
Rating      -1.552545
Reviews      5.917079
Size         1.673972
Installs     1.079198
Price       14.222785
dtype: float64
```

In [63]:
```
inp1['Reviews'] = np.log1p(inp1['Reviews'])
inp1['Installs'] = np.log1p(inp1['Installs'])
```

In [64]:
```
inp1.skew()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_11632\3545313420.py:1: FutureWarning: The
default value of numeric_only in DataFrame.skew is deprecated. In a future version, i
t will default to False. In addition, specifying 'numeric_only=None' is deprecated. S
elect only valid columns or specify the value of numeric_only to silence this warnin
g.
  inp1.skew()
```

Out[64]:
```
Rating      -1.552545
Reviews     -0.043603
Size         1.673972
Installs    -0.507080
Price       14.222785
dtype: float64
```

8.2) Drop columns App, Last Updated, Current Ver, and Android Ver. These variables are not useful for our task.

In [65]:
```
inp1.drop(['App','Last Updated','Current Ver','Android Ver'],axis=1,inplace=True)
```

In [66]:
```
inp1
```

Out[66]:

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	ART_AND_DESIGN	4.1	5.075174	19000.0	9.210440	Free	0.0	Everyone	Art &
1	ART_AND_DESIGN	3.9	6.875232	14000.0	13.122365	Free	0.0	Everyone	Design;F
4	ART_AND_DESIGN	4.3	6.875232	2800.0	11.512935	Free	0.0	Everyone	Design;Cr
5	ART_AND_DESIGN	4.4	5.123964	5600.0	10.819798	Free	0.0	Everyone	Art &
6	ART_AND_DESIGN	3.8	5.187386	19000.0	10.819798	Free	0.0	Everyone	Art &
...	...	...	...	...	...	...	...	...	
10832	WEATHER	3.8	7.086738	NaN	11.512935	Free	0.0	Everyone	W
10833	BOOKS_AND_REFERENCE	4.8	3.806662	NaN	6.908755	Free	0.0	Everyone	B Ref
10834	FAMILY	4.0	2.079442	2600.0	6.216606	Free	0.0	Everyone	Edu
10836	FAMILY	4.5	3.663562	53000.0	8.517393	Free	0.0	Everyone	Edu
10837	FAMILY	5.0	1.609438	3600.0	4.615121	Free	0.0	Everyone	Edu

5954 rows × 9 columns

In [67]:
```
inp1.head()
```

Out[67]:

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres
0	ART_AND_DESIGN	4.1	5.075174	19000.0	9.210440	Free	0.0	Everyone	Art & Design
1	ART_AND_DESIGN	3.9	6.875232	14000.0	13.122365	Free	0.0	Everyone	Art & Design;Pretend Play
4	ART_AND_DESIGN	4.3	6.875232	2800.0	11.512935	Free	0.0	Everyone	Art & Design;Creativity
5	ART_AND_DESIGN	4.4	5.123964	5600.0	10.819798	Free	0.0	Everyone	Art & Design
6	ART_AND_DESIGN	3.8	5.187386	19000.0	10.819798	Free	0.0	Everyone	Art & Design

8.3) Get dummy columns for Category, Genres, and Content Rating. This needs to be done as the models do not understand categorical data, and all data should be numeric. Dummy encoding is one way to convert character fields to numeric. Name of dataframe should be inp2.

In [68]:
```
inp2 = inp1
```

In [69]:
```
inp2
```

Out[69]:

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	ART_AND_DESIGN	4.1	5.075174	19000.0	9.210440	Free	0.0	Everyone	Art &
1	ART_AND_DESIGN	3.9	6.875232	14000.0	13.122365	Free	0.0	Everyone	Design;F
4	ART_AND_DESIGN	4.3	6.875232	2800.0	11.512935	Free	0.0	Everyone	Design;Cr
5	ART_AND_DESIGN	4.4	5.123964	5600.0	10.819798	Free	0.0	Everyone	Art &
6	ART_AND_DESIGN	3.8	5.187386	19000.0	10.819798	Free	0.0	Everyone	Art &
...	...	...	...	...	...	...	...	...	
10832	WEATHER	3.8	7.086738	NaN	11.512935	Free	0.0	Everyone	W
10833	BOOKS_AND_REFERENCE	4.8	3.806662	NaN	6.908755	Free	0.0	Everyone	B Ref
10834	FAMILY	4.0	2.079442	2600.0	6.216606	Free	0.0	Everyone	Edu
10836	FAMILY	4.5	3.663562	53000.0	8.517393	Free	0.0	Everyone	Edu
10837	FAMILY	5.0	1.609438	3600.0	4.615121	Free	0.0	Everyone	Edu

5954 rows × 9 columns

In [80]:
```
import pandas as pd

# Create the dummy columns for Category, Genres, and Content Rating
category_dummies = pd.get_dummies(inp1['Category'], prefix='Category')
genres_dummies = pd.get_dummies(inp1['Genres'], prefix='Genres')
content_rating_dummies = pd.get_dummies(inp1['Content Rating'], prefix='Content_Rating

# Concatenate the original dataframe with the dummy columns
inp2 = pd.concat([inp1, category_dummies, genres_dummies, content_rating_dummies], axi
```

In [81]:
```
inp2
```

Out[81]:

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	ART_AND_DESIGN	4.1	5.075174	19000.0	9.210440	Free	0.0	Everyone	Art &
1	ART_AND_DESIGN	3.9	6.875232	14000.0	13.122365	Free	0.0	Everyone	Design;F
4	ART_AND_DESIGN	4.3	6.875232	2800.0	11.512935	Free	0.0	Everyone	Design;Cr
5	ART_AND_DESIGN	4.4	5.123964	5600.0	10.819798	Free	0.0	Everyone	Art &
6	ART_AND_DESIGN	3.8	5.187386	19000.0	10.819798	Free	0.0	Everyone	Art &
...	...	...	...	...	...	...	...	...	
10832	WEATHER	3.8	7.086738	NaN	11.512935	Free	0.0	Everyone	W
10833	BOOKS_AND_REFERENCE	4.8	3.806662	NaN	6.908755	Free	0.0	Everyone	B Ref
10834	FAMILY	4.0	2.079442	2600.0	6.216606	Free	0.0	Everyone	Edu
10836	FAMILY	4.5	3.663562	53000.0	8.517393	Free	0.0	Everyone	Edu
10837	FAMILY	5.0	1.609438	3600.0	4.615121	Free	0.0	Everyone	Edu

5954 rows × 154 columns

9) Train test split and apply 70-30 split. Name the new dataframes df_train and df_test.

10) Separate the dataframes into X_train, y_train, X_test, and y_test.

In [82]:
```
from sklearn.model_selection import train_test_split as tts
from sklearn.linear_model import LinearRegression as LR
from sklearn.metrics import mean_squared_error as mse
```

In [94]:
```
from sklearn.model_selection import train_test_split

# Split the data into training and testing sets
```

```
df_train, df_test = train_test_split(inp2, test_size=0.3, random_state=42)

# Separate the training data into X_train (features) and y_train (target variable)
X_train = df_train.drop(['Rating'], axis=1)
y_train = df_train['Rating']

# Separate the testing data into X_test (features) and y_test (target variable)
X_test = df_test.drop(['Rating'], axis=1)
y_test = df_test['Rating']
```

In [99]:
```
inp2.dtypes
```

Out[99]:
```
Category                          object
Rating                           float64
Reviews                          float64
Size                             float64
Installs                         float64
                                  ...
Content_Rating_Everyone            uint8
Content_Rating_Everyone 10+        uint8
Content_Rating_Mature 17+          uint8
Content_Rating_Teen                uint8
Content_Rating_Unrated             uint8
Length: 154, dtype: object
```