In [36]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('googleplaystore.csv')
df.head()
```
Out[36]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Curre V
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varie wi devic
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1

Check for null value in data

In [37]:
```
df.isnull().sum(axis=0)
```
Out[37]:
```
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
```

Drop Records ith nulls in an of the columns

In [38]:
```
print("Frame Size before: ",df.shape)
df.dropna(subset=['Rating','Type','Content Rating','Current Ver','Android Ver'],axis=0, inpl
ace=True)
print("rame Size After: ",df.shape)
df.isnull().sum(axis=0)
```
```
Frame Size before:  (10841, 13)
rame Size After:  (9360, 13)
```
Out[38]:
```
App               0
Category          0
Rating            0
Reviews           0
Size              0
Installs          0
Type              0
Price             0
Content Rating    0
Genres            0
Last Updated      0
Current Ver       0
Android Ver       0
dtype: int64
```

Extract the numerical value from the column ,multiply the value by 1000 if size is mentioned in MB

In [39]:
```
j=df.columns.get_loc('Size')
for i in range(0,len(df)):
    if df.iloc[i,j].lower().endswith('k'):
        df.iloc[i,j]=float(df.iloc[i,j][0:-1])
    elif df.iloc[i,j].lower().endswith('m'):
        df.iloc[i,j]=float(df.iloc[i,j][0:-1])*1000
```
In [40]:
```
df.Size = pd.to_numeric(df.Size,errors='coerce')
df.dropna(subset=['Size'],inplace=True)
df.shape
```
Out[40]: (7723, 13)

Rewiews is a numeric field that is loaded as string field . Convert it into numeric (int/float)

In [41]:
```
df.Reviews =df.Reviews.astype('float64')
df.Reviews.dtypes
```
Out[41]: dtype('float64')

In [42]:
```
df.Installs=df.Installs.str.replace(',','').str.replace('+','').astype('int64')
df.Installs.dtype
```
Out[42]: dtype('int64')

Price field is string andhas $symbol. Remove$sign and convert it intpo umeric

In [43]:
```
df.Price=df.Price.str.replace('$','').astype('float64')
```
In [44]:
```
df.Price.dtype
```
Out[44]: dtype('float64')

Sanity Check:

Average rating shoiuld be between 1 and 5 as onl these values are allowed on the pla store. Drop the rows that have a value outside this range

In [45]:
```
filter=(df.Rating<1) | (df.Rating>5)
print("Data Frame size :",df.shape,"count of rows containing WRONG Rating:",filter.value_cou
nts())
```
```
Data Frame size : (7723, 13) count of rows containing WRONG Rating: False    7723
Name: Rating, dtype: int64
```

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 [46]:
```
rows=df[df.Installs < df.Reviews].index
df.drop(rows,axis=0,inplace=True)
df.shape
```
Out[46]: (7717, 13)

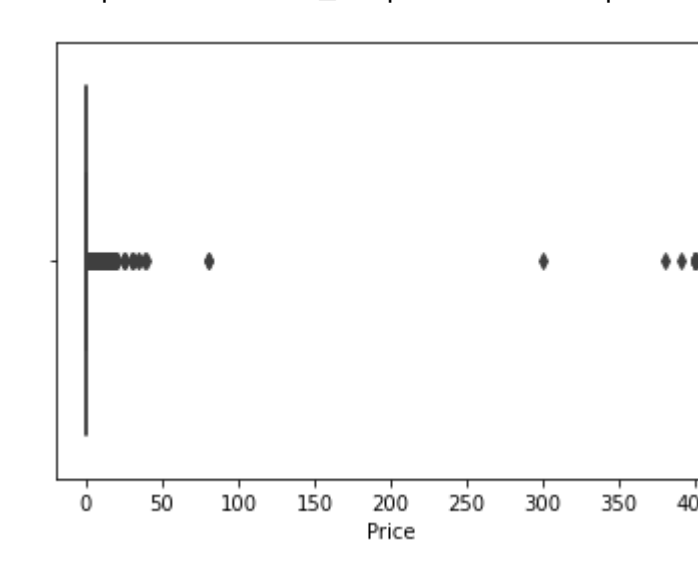
In [47]:
```
rows=df[(df.Type.str.lower()=='free' )& (df.Price > 0)].index
rows
```
Out[47]: Int64Index([], dtype='int64')

For free apps (tpes="free"), the price should not be >0. Drop Any such rows

In [48]:
```
rows=df[(df.Type.str.lower()=='free')& (df.Price >0)].index
rows
```
Out[48]: Int64Index([], dtype='int64')

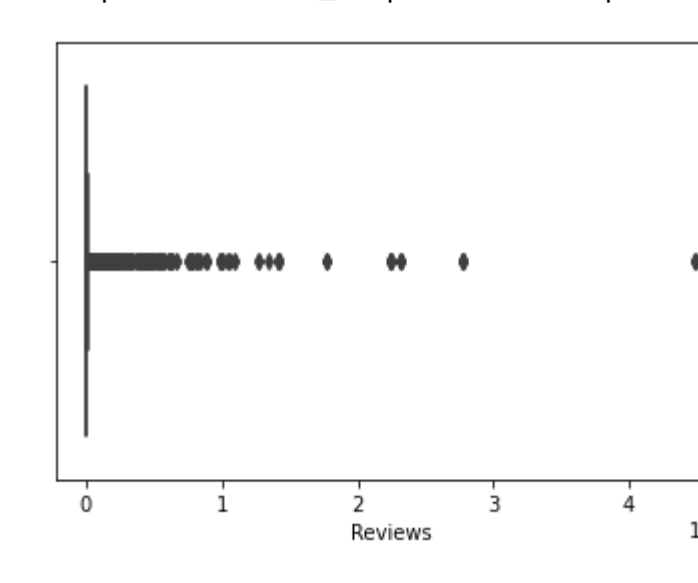
Boxplot for Price

In [49]:
```
import seaborn as sns
```
In [50]:
```
sns.boxplot(df.Price)
```
Out[50]: <matplotlib.axes._subplots.AxesSubplot at 0xd0bb940a00>



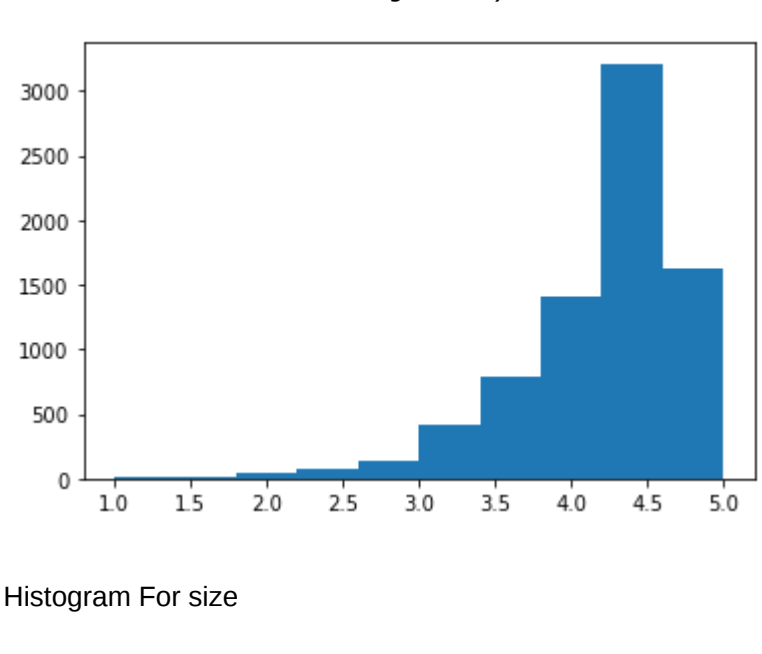
Boxplot For reviews

In [51]:
```
sns.boxplot(x='Reviews',data=df)
```
Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0xd0bbba3310>



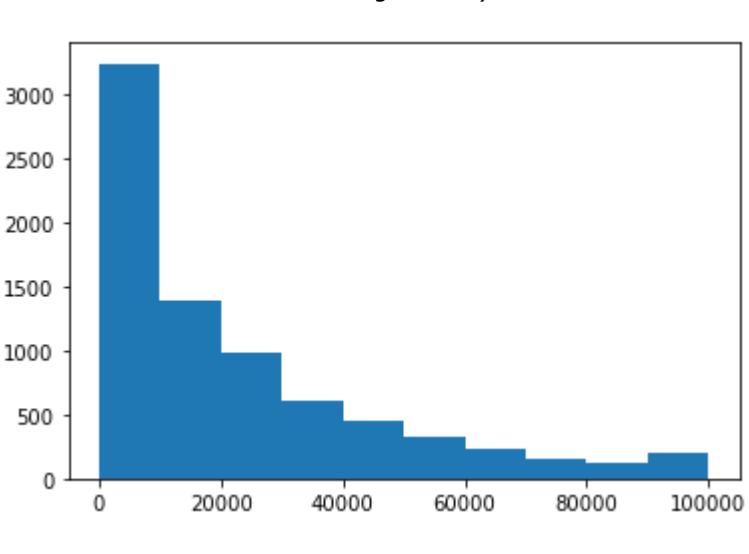
Histogram For rating

In [52]:
```
plt.hist(df.Rating)
```
Out[52]:
```
(array([  17.,   18.,   39.,   72.,  132.,  408.,  781., 1406., 3212.,
        1632.]),
 array([1. , 1.4, 1.8, 2.2, 2.6, 3. , 3.4, 3.8, 4.2, 4.6, 5. ]),
 <a list of 10 Patch objects>)
```



Histogram For size

In [53]:
```
plt.hist(df.Size)
```
Out[53]:
```
(array([3245., 1398.,  991.,  606.,  449.,  325.,  226.,  161.,  117.,
         199.]),
 array([8.500000e+00, 1.000765e+04, 2.000680e+04, 3.000595e+04,
        4.000510e+04, 5.000425e+04, 6.000340e+04, 7.000255e+04,
        8.000170e+04, 9.000085e+04, 1.000000e+05]),
 <a list of 10 Patch objects>)
```



Outlier Treatment:

Price: From the box plot , it seems like there are some apps with ver high price . A price of $200 for an application on the play store is very high and suspicious!

Check out the records with very high price is 200 indeed a high price ? Drop these as most seem to be junk apps

In [54]:
```
df.drop(df[df.Price>250].index, axis=0, inplace=True)
df.shape
```
Out[54]: (7702, 13)

Review :Vry few have very high number of reviews .these are all star apps that dont help with the analysis ans , in fact will skew it. Drop records having more than million reviews

In [55]:
```
df.drop(df[df.Reviews>200000].index, axis=0,inplace =True)
df.shape
```
Out[55]: (6792, 13)

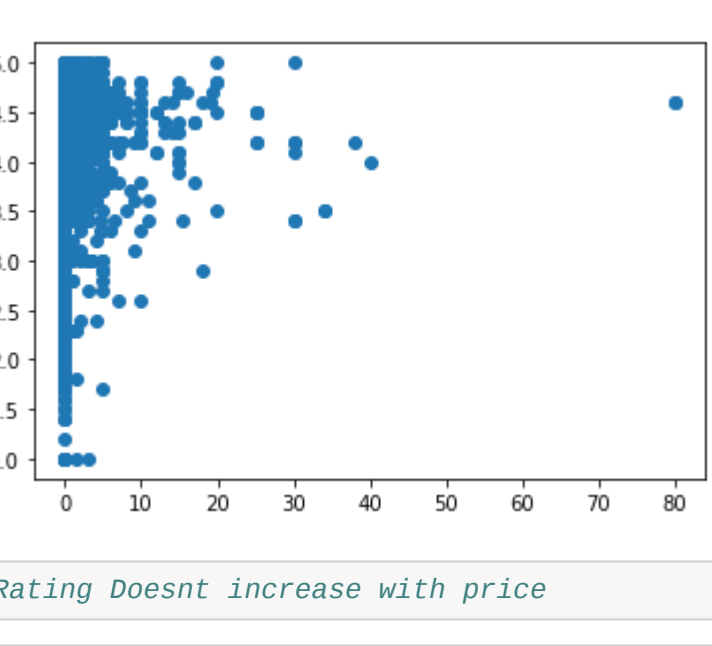
installs:There seems to be some outlier inthis field too . Apps having very high number of installs should be dropped from analysis

In [56]:
```
#find out the different percentiles -10,25,50,70,90,95,99 decide a threshold as cutoff for o
utlier ad drop records having value more than that
```
In [57]:
```
df.Installs.quantile([0.1,0.25,0.5,0.70,0.9,0.95,0.99])
```
Out[57]:
```
0.10        1000.0
0.25        5000.0
0.50      100000.0
0.70     1000000.0
0.90     5000000.0
0.95    10000000.0
0.99    10000000.0
Name: Installs, dtype: float64
```
In [58]:
```
df=df[df.Installs<10000000]
```
In [59]:
```
df.shape
```
Out[59]: (6406, 13)

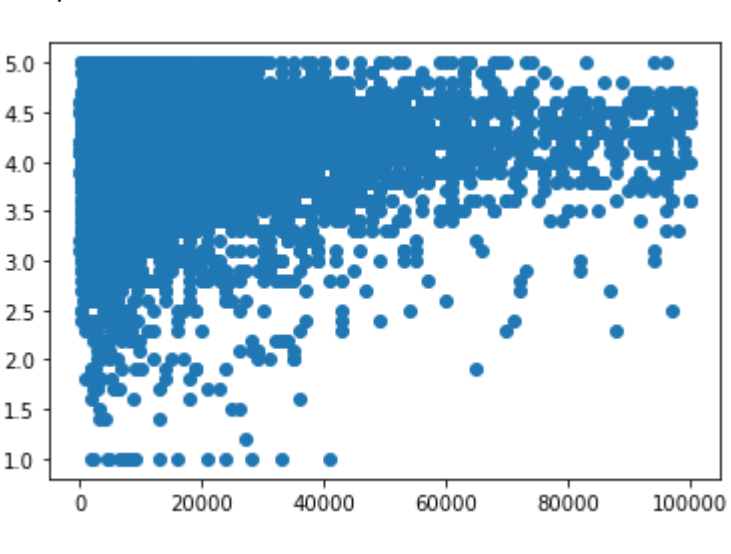
bivariat analysis: Lets look at how the available predictors relate to the variable of interest . i.e. our target variable rating .Make Scatter Plots and box plot (for character feature)to assess the relation between rating and the other feature

Make Scatter Plot/joinplo for rating Vs Price

In [60]:
```
plt.scatter(x=df.Price,y=df.Rating)
```
Out[60]: <matplotlib.collections.PathCollection at 0xd0bb60fac0>



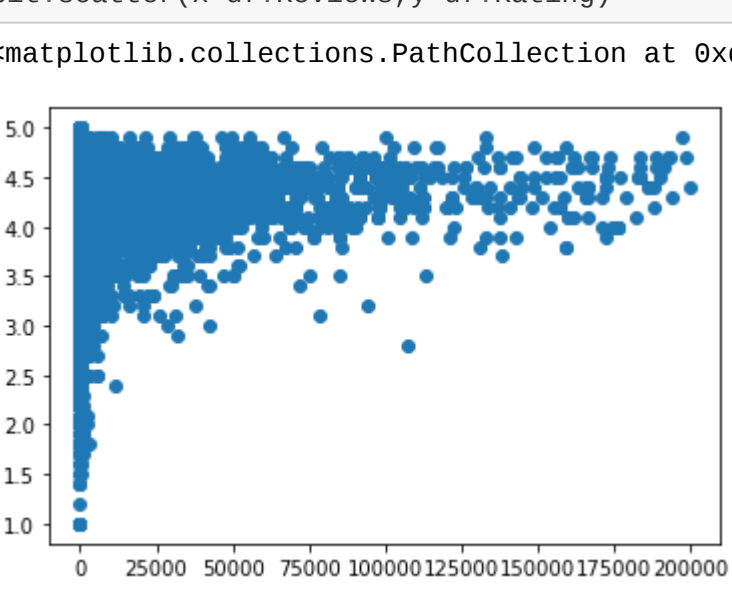
In [61]:
```
#Rating Doesnt increase with price
```
In [62]:
```
plt.scatter(x=df.Size,y=df.Rating)
```
Out[62]: <matplotlib.collections.PathCollection at 0xd0bccae700>



In [ ]:
```
#It can be obsereve that heavier apps are having higher rating
```

Make Scatter plot/joinplot for rating Vs Reviews

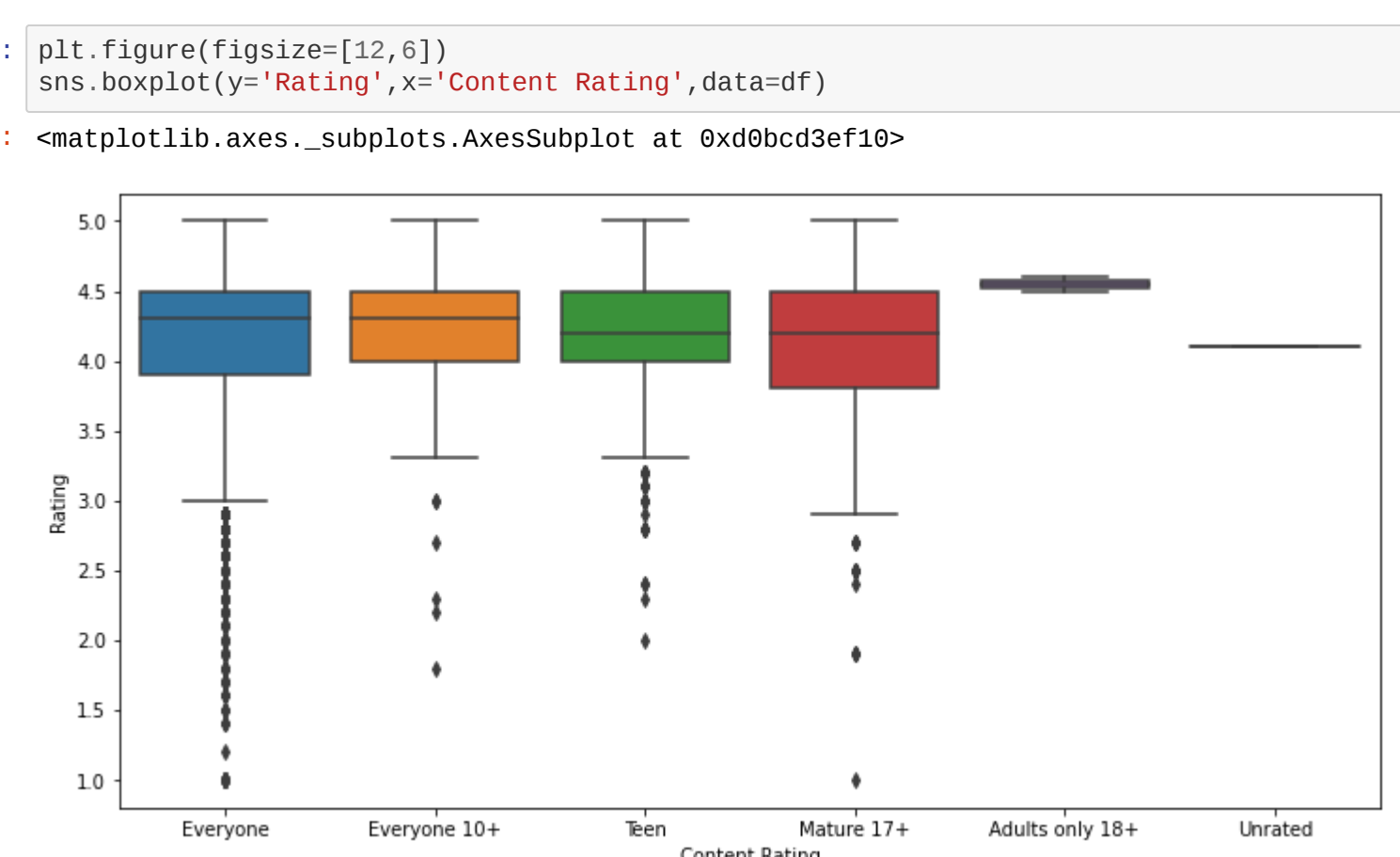
In [63]:
```
plt.scatter(x=df.Reviews,y=df.Rating)
```
Out[63]: <matplotlib.collections.PathCollection at 0xd0bcd03c40>



scatter plot indicats higher rating for apps having Max Reviews .But this cannit be always it could be outlier

Make boxplot for rating vs Content Rating

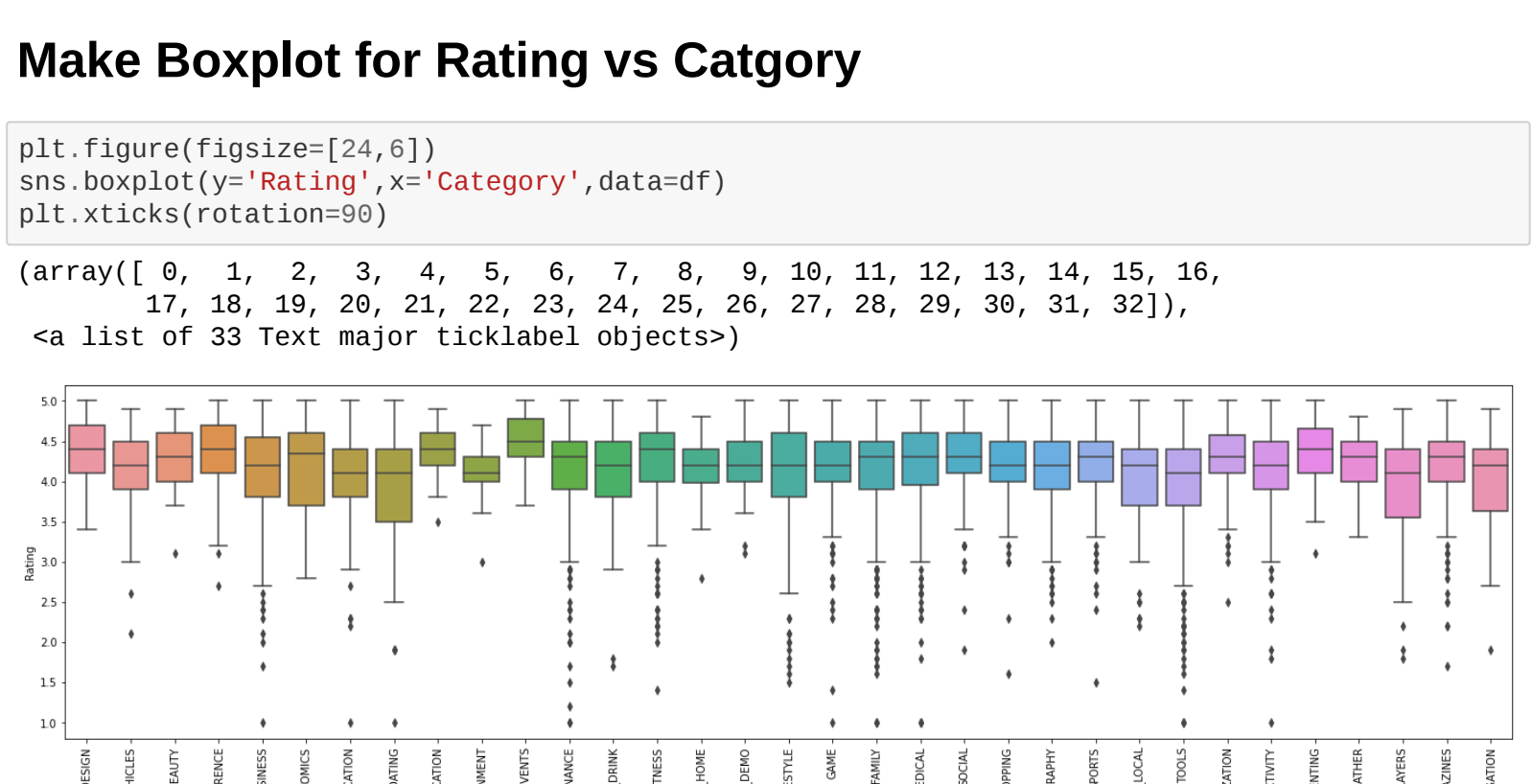
In [65]:
```
plt.figure(figsize=[12,6])
sns.boxplot(y='Rating',x='Content Rating',data=df)
```
Out[65]: <matplotlib.axes._subplots.AxesSubplot at 0xd0bcd3ef10>



In [66]:
```
#Not much conclusioncould be drawn as the plot is almost same for Contet Ratings, Except adu
lts and only 18+ & unrated
```

Make Boxplot for Rating vs Catgory

In [67]:
```
plt.figure(figsize=[24,6])
sns.boxplot(y='Rating',x='Category',data=df)
plt.xticks(rotation=90)
```
Out[67]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]),
 <a list of 33 Text major ticklabel objects>)
```



Data Preprocessing

In [70]:
```
inp1=df
inp1.reset_index(drop=True,inplace=True)
inp1.head()
```
Out[70]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Curren V
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159.0	19000.0	10000	Free	0.0	Everyone	Art & Design	January 7, 2018	1.0
1	Coloring book moana	ART_AND_DESIGN	3.9	967.0	14000.0	500000	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510.0	8700.0	5000000	Free	0.0	Everyone	Art & Design	August 1, 2018	1.2
3	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967.0	2800.0	100000	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	1.
4	Paper flowers instructions	ART_AND_DESIGN	4.4	167.0	5600.0	50000	Free	0.0	Everyone	Art & Design	March 26, 2017	1.

Review and install have some value that are still relatavel ver high . Before building a linear egression model, ou need to reduce the scre . Apply log transformation to review and install

In [72]:
```
inp1.Reviews=np.log1p(inp1.Reviews)
inp1.Installs=np.log1p(inp1.Reviews)
inp1.head()
```
Out[72]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Curr
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	5.075174	19000.0	1.804211	Free	0.0	Everyone	Art & Design	January 7, 2018	1
1	Coloring book moana	ART_AND_DESIGN	3.9	6.875232	14000.0	2.063723	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	2
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	11.379520	8700.0	2.516043	Free	0.0	Everyone	Art & Design	August 1, 2018	1
3	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	6.875232	2800.0	2.063723	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	
4	Paper flowers instructions	ART_AND_DESIGN	4.4	5.123964	5600.0	1.812210	Free	0.0	Everyone	Art & Design	March 26, 2017	

Drop Columns app last update , current ver, androd ver.These variable are not useful for our task

In [75]:
```
inp1.drop(['App','Last Updated','Current Ver','Android Ver'],axis=1,inplace=True)
inp1.head()
```
Out[75]:

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres
0	ART_AND_DESIGN	4.1	5.075174	19000.0	1.804211	Free	0.0	Everyone	Art & Design
1	ART_AND_DESIGN	3.9	6.875232	14000.0	2.063723	Free	0.0	Everyone	Art & Design;Pretend Play
2	ART_AND_DESIGN	4.7	11.379520	8700.0	2.516043	Free	0.0	Everyone	Art & Design
3	ART_AND_DESIGN	4.3	6.875232	2800.0	2.063723	Free	0.0	Everyone	Art & Design;Creativity
4	ART_AND_DESIGN	4.4	5.123964	5600.0	1.812210	Free	0.0	Everyone	Art & Design

In [76]:
```
inp1=pd.get_dummies(inp1,columns=['Category','Genres','Content Rating','Type'],drop_first=Tr
ue)
inp2=inp1.copy()
inp2.columns
```
Out[76]:
```
Index(['Rating', 'Reviews', 'Size', 'Installs', 'Price',
       'Category_AUTO_AND_VEHICLES', 'Category_BEAUTY',
       'Category_BOOKS_AND_REFERENCE', 'Category_BUSINESS', 'Category_COMICS',
       ...
       'Genres_Video Players & Editors;Creativity',
       'Genres_Video Players & Editors;Music & Video', 'Genres_Weather',
       'Genres_Word', 'Content Rating_Everyone', 'Content Rating_Everyone 10+',
       'Content Rating_Mature 17+', 'Content Rating_Teen',
       'Content Rating_Unrated', 'Type_Paid'],
      dtype='object', length=150)
```
In [83]:
```
from sklearn.model_selection import train_test_split
df_train,df_test=train_test_split(inp2,test_size=0.3,random_state=100)
```
In [84]:
```
y_train=df_train.pop('Rating')
X_train=df_train

y_test=df_test.pop('Rating')
X_test=df_test
```
In [90]:
```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(X_train,y_train)

from sklearn.metrics import r2_score
y_train_predict=lm.predict(X_train)
r2_score(y_train,y_train_predict)
```
Out[90]: 0.08622749639981775

In [91]:
```
X_test_prect=lm.predict(X_test)
r2_score(y_test,X_test_prect)
```
Out[91]: 0.05593838888438063

In [ ]:

In [ ]: