DATA ANALYSIS OF GOOGLE APPS RATING

IMPORTING THE REQUIRED MODULES

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

READING THE DATA

In [2]:
```
google_data=pd.read_csv('googleplaystore.csv')
```

In [3]:
```
type(google_data)
```

Out[3]:
```
pandas.core.frame.DataFrame
```

In [4]:
```
google_data.head()   #inspecting the first 5 rows
```

Out[4]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
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	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
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	1.1	4.4 and up

In [5]:
```
google_data.shape
```

Out[5]:
```
(10841, 13)
```

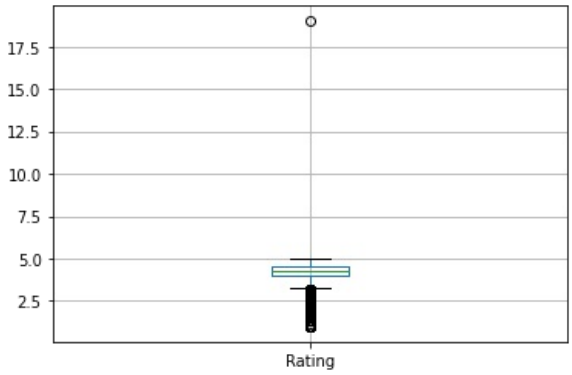
In [6]:
```
google_data.describe()
```

Out[6]:

	Rating
count	9367.000000
mean	4.193338
std	0.537431
min	1.000000
25%	4.000000
50%	4.300000
75%	4.500000
max	19.000000

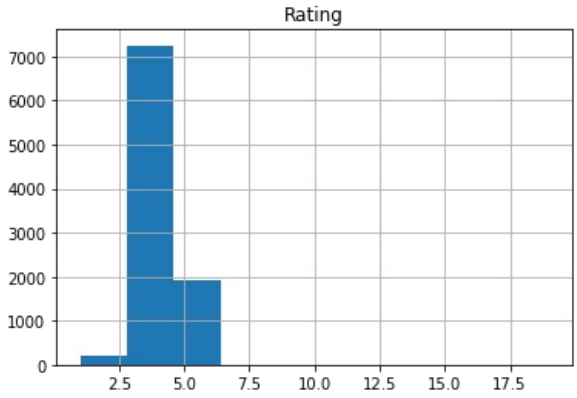
In [7]:
```
google_data.boxplot()
```

Out[7]:
```
<AxesSubplot:>
```



In [8]:
```
google_data.hist()
```

Out[8]:
```
array([[<AxesSubplot:title={'center':'Rating'}>]], dtype=object)
```



In [9]:
```
google_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10841 non-null  object 
 1   Category        10841 non-null  object 
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object 
 4   Size            10841 non-null  object 
 5   Installs        10841 non-null  object 
 6   Type            10840 non-null  object 
 7   Price           10841 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10841 non-null  object 
 10  Last Updated    10841 non-null  object 
 11  Current Ver     10833 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

DATA CLEANING

COUNT THE NUMBER OF MISSING VALUES IN THE DATAFRAME|

In [10]:
```
google_data.isnull()
```

Out[10]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False

4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10836	False	False	False	False	False	False	False	False	False	False	False	False	False
10837	False	False	False	False	False	False	False	False	False	False	False	False	False
10838	False	False	True	False	False	False	False	False	False	False	False	False	False
10839	False	False	False	False	False	False	False	False	False	False	False	False	False
10840	False	False	False	False	False	False	False	False	False	False	False	False	False

10841 rows × 13 columns

In [11]:
```
google_data.isnull().sum()
```

Out[11]:
```
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
```

CHECK HOW MANY RATINGS ARE MORE THAN 5-OUTLIERS

In [12]:
```
google_data[google_data.Rating>5]
```

Out[12]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0M	1,000+	Free	0	Everyone	NaN	February 11, 2018	1.0.19	4.0 and up	NaN

In [13]:
```
google_data.drop([10472],inplace=True)
```

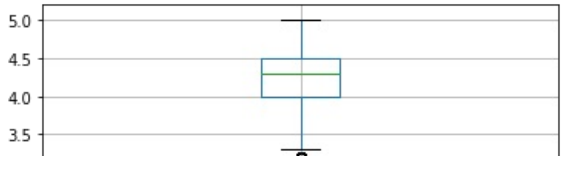
In [14]:
```
google_data[10470:10475]
```

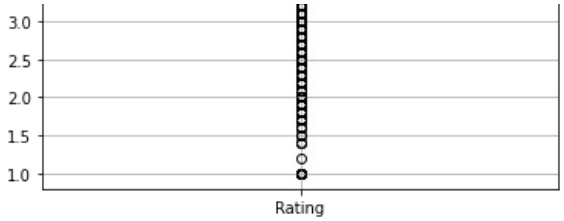
Out[14]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10470	Jazz Wi-Fi	COMMUNICATION	3.4	49	4.0M	10,000+	Free	0	Everyone	Communication	February 10, 2017	0.1	2.3 and up
10471	Xposed Wi-Fi-Pwd	PERSONALIZATION	3.5	1042	404k	100,000+	Free	0	Everyone	Personalization	August 5, 2014	3.0.0	4.0.3 and up
10473	osmino Wi-Fi: free WiFi	TOOLS	4.2	134203	4.1M	10,000,000+	Free	0	Everyone	Tools	August 7, 2018	6.06.14	4.4 and up
10474	Sat-Fi Voice	COMMUNICATION	3.4	37	14M	1,000+	Free	0	Everyone	Communication	November 21, 2014	2.2.1.5	2.2 and up
10475	Wi-Fi Visualizer	TOOLS	3.9	132	2.6M	50,000+	Free	0	Everyone	Tools	May 17, 2017	0.0.9	2.3 and up

In [15]:
```
google_data.boxplot()
```

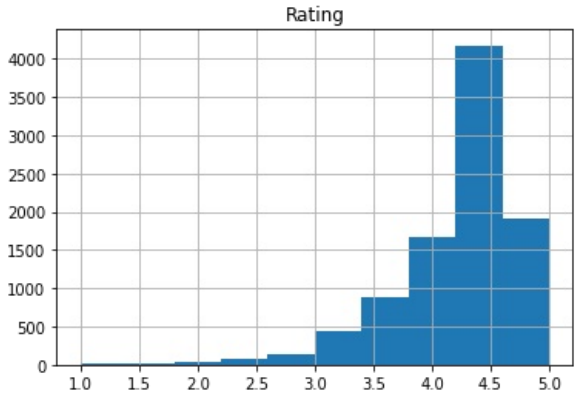
Out[15]:
```
<AxesSubplot:>
```





In [16]:
```
google_data.hist()
```

Out[16]:
```
array([[<AxesSubplot:title={'center':'Rating'}>]], dtype=object)
```



REMOVE COLUMS THAT ARE 90% EMPTY

In [17]:
```
threshold=len(google_data)*0.1  #10% of my rows*100
threshold
```

Out[17]:
```
1084.0
```

In [18]:
```
google_data.dropna(thresh=threshold, axis=1 , inplace=True)
```

In [19]:
```
print(google_data.isnull().sum())
```

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

In [20]:
```
google_data.shape
```

Out[20]:
```
(10840, 13)
```

DATA IMPUTATION AND MANIPULATION

FILLING THE NULL VALUES WITH APPROPRIATE VALUES USING AGGREGATE FUNCTIONS SUCH AS MEAN, MEDIAN OR MODE

In [21]:
```
#define a function impute_median
def impute_median(series):
    return series.fillna(series.median())
```

In [22]:
```
google_data.Rating=google_data['Rating'].transform( impute_median)
```

In [23]:
```
#cound the number of null values in each column
google_data.isnull().sum()
```

Out[23]:
```
App               0
Category          0
Rating            0
Reviews           0
Size              0
Installs          0
Type              1
Price             0
Content Rating    0
Genres            0
Last Updated      0
Current Ver       8
Android Ver       2
dtype: int64
```

In [24]:
```
#modes of categorical values
print(google_data['Type'].mode())
print(google_data['Current Ver'].mode())
print(google_data['Android Ver'].mode())
```

```
0    Free
dtype: object
0    Varies with device
dtype: object
0    4.1 and up
dtype: object
```

In [25]:
```
#Fill the missing categorical values with mode
google_data['Type'].fillna(str(google_data['Type'].mode().values[0]), inplace=True)
google_data['Current Ver'].fillna(str(google_data['Current Ver'].mode().values[0]), inplace=True)
google_data['Android Ver'].fillna(str(google_data['Android Ver'].mode().values[0]), inplace=True)
```

In [26]:
```
google_data.isnull().sum()
```

Out[26]:
```
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
```

In [27]:
```
#Lets's convert price, Reviews and Ratings into numerical values
google_data['Price']=google_data['Price'].apply(lambda x: str(x).replace('$', '') if '$' in str(x) else str(x))
google_data['Price']=google_data['Price'].apply(lambda x: float(x))
google_data['Reviews']=pd.to_numeric(google_data['Reviews'], errors='coerce')
```

In [28]:
```
google_data['Installs']=google_data['Installs'].apply(lambda x: str(x).replace('+', '') if '+' in str(x) else str
google_data['Installs']=google_data['Installs'].apply(lambda x: str(x).replace(',', '') if ',' in str(x) else str
google_data['Installs']=google_data['Installs'].apply(lambda x: float(x))
```

In [29]:
```
google_data.head(10)
```

Out[29]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera &	ART_AND_DESIGN	4.1	159	19M	10000.0	Free	0.0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up

	Grid & ScrapBook												
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500000.0	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5000000.0	Free	0.0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50000000.0	Free	0.0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100000.0	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5.6M	50000.0	Free	0.0	Everyone	Art & Design	March 26, 2017	1	2.3 and up
6	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178	19M	50000.0	Free	0.0	Everyone	Art & Design	April 26, 2018	1.1	4.0.3 and up
7	Infinite Painter	ART_AND_DESIGN	4.1	36815	29M	1000000.0	Free	0.0	Everyone	Art & Design	June 14, 2018	6.1.61.1	4.2 and up
8	Garden Coloring Book	ART_AND_DESIGN	4.4	13791	33M	1000000.0	Free	0.0	Everyone	Art & Design	September 20, 2017	2.9.2	3.0 and up
9	Kids Paint Free - Drawing Fun	ART_AND_DESIGN	4.7	121	3.1M	10000.0	Free	0.0	Everyone	Art & Design;Creativity	July 3, 2018	2.8	4.0.3 and up

In [30]:
```
google_data.describe()  #summary stats after cleaning
```

Out[30]:

	Rating	Reviews	Installs	Price
count	10840.000000	1.084000e+04	1.084000e+04	10840.000000
mean	4.206476	4.441529e+05	1.546434e+07	1.027368
std	0.480342	2.927761e+06	8.502936e+07	15.949703
min	1.000000	0.000000e+00	0.000000e+00	0.000000
25%	4.100000	3.800000e+01	1.000000e+03	0.000000
50%	4.300000	2.094000e+03	1.000000e+05	0.000000
75%	4.500000	5.477550e+04	5.000000e+06	0.000000
max	5.000000	7.815831e+07	1.000000e+09	400.000000

DATA VISUALIZATION

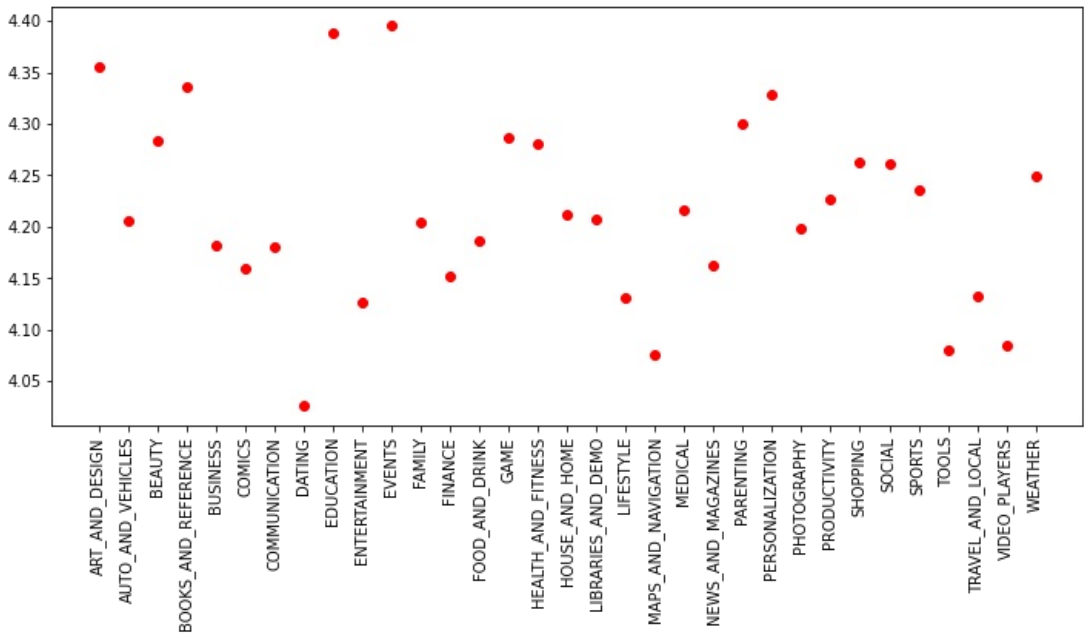
In [31]:
```
grp=google_data.groupby('Category')
x=grp['Rating'].agg(np.mean)
y=grp['Price'].agg(np.sum)
z=grp['Reviews'].agg(np.mean)
print(x)
print(y)
print(z)
```

```
Category
ART_AND_DESIGN          4.355385
AUTO_AND_VEHICLES       4.205882
BEAUTY                  4.283019
BOOKS_AND_REFERENCE     4.335498
BUSINESS                4.182391
COMICS                  4.160000
COMMUNICATION           4.180103
DATING                  4.025641
EDUCATION               4.388462
ENTERTAINMENT           4.126174
EVENTS                  4.395313
FAMILY                  4.204564
FINANCE                 4.151639
FOOD_AND_DRINK          4.185827
GAME                    4.286888
```

```
HEALTH_AND_FITNESS     4.280059
HOUSE_AND_HOME         4.211364
LIBRARIES_AND_DEMO     4.207059
LIFESTYLE              4.131414
MAPS_AND_NAVIGATION    4.075182
MEDICAL                4.216199
NEWS_AND_MAGAZINES     4.161837
PARENTING              4.300000
PERSONALIZATION        4.328827
PHOTOGRAPHY            4.197910
PRODUCTIVITY           4.226651
SHOPPING               4.263077
SOCIAL                 4.261017
SPORTS                 4.236458
TOOLS                  4.080071
TRAVEL_AND_LOCAL       4.132946
VIDEO_PLAYERS          4.084000
WEATHER                4.248780
Name: Rating, dtype: float64
Category
ART_AND_DESIGN            5.97
AUTO_AND_VEHICLES        13.47
BEAUTY                    0.00
BOOKS_AND_REFERENCE     119.77
BUSINESS                185.27
COMICS                    0.00
COMMUNICATION            83.14
DATING                   31.43
EDUCATION                17.96
ENTERTAINMENT             7.98
EVENTS                  109.99
FAMILY                 2434.78
FINANCE                2900.83
FOOD_AND_DRINK            8.48
GAME                    287.30
HEALTH_AND_FITNESS       67.34
HOUSE_AND_HOME            0.00
LIBRARIES_AND_DEMO        0.99
LIFESTYLE              2360.87
MAPS_AND_NAVIGATION      26.95
MEDICAL                1439.96
NEWS_AND_MAGAZINES        3.98
PARENTING                 9.58
PERSONALIZATION         153.96
PHOTOGRAPHY             134.21
PRODUCTIVITY            250.93
SHOPPING                  5.48
SOCIAL                   15.97
SPORTS                  100.00
TOOLS                   267.25
TRAVEL_AND_LOCAL         49.95
VIDEO_PLAYERS            10.46
WEATHER                  32.42
Name: Price, dtype: float64
Category
ART_AND_DESIGN         2.637600e+04
AUTO_AND_VEHICLES      1.369019e+04
BEAUTY                 7.476226e+03
BOOKS_AND_REFERENCE    9.506090e+04
BUSINESS               3.033598e+04
COMICS                 5.638793e+04
COMMUNICATION          2.107138e+06
DATING                 3.115931e+04
EDUCATION              2.538191e+05
ENTERTAINMENT          3.971688e+05
EVENTS                 2.515906e+03
FAMILY                 2.080255e+05
FINANCE                4.795281e+04
FOOD_AND_DRINK         6.994748e+04
GAME                   1.385859e+06
HEALTH_AND_FITNESS     1.111253e+05
HOUSE_AND_HOME         4.518619e+04
LIBRARIES_AND_DEMO     1.220139e+04
LIFESTYLE              3.372457e+04
MAPS_AND_NAVIGATION    2.237902e+05
MEDICAL                3.425432e+03
NEWS_AND_MAGAZINES     1.922292e+05
PARENTING              1.597218e+04
PERSONALIZATION        2.279238e+05
PHOTOGRAPHY            6.373631e+05
PRODUCTIVITY           2.691438e+05
SHOPPING               4.424662e+05
SOCIAL                 2.105903e+06
SPORTS                 1.844536e+05
TOOLS                  3.240629e+05
TRAVEL_AND_LOCAL       2.427051e+05
VIDEO_PLAYERS          6.307439e+05
WEATHER                1.781065e+05
Name: Reviews, dtype: float64
```

In [32]:
```
plt.figure(figsize=(12,5))
plt.plot(x , 'ro')
plt.xticks(rotation=90)
plt.show
```

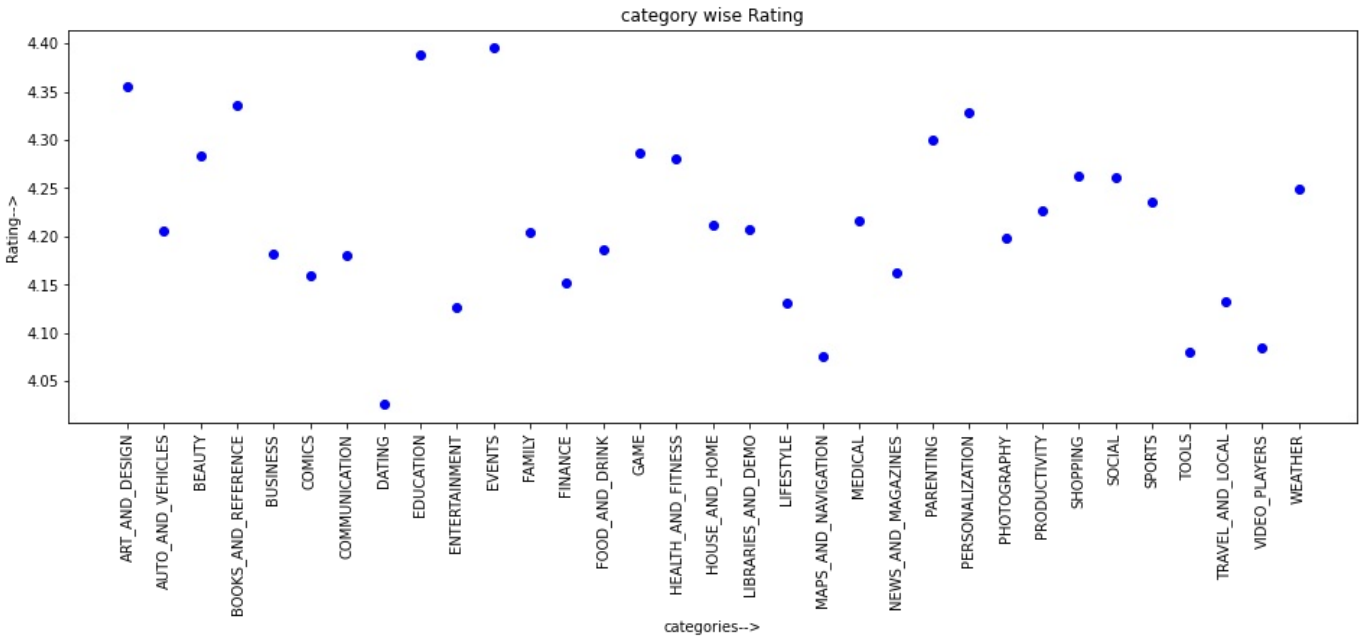
Out[32]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [33]:
```
plt.figure(figsize=(16,5))
plt.plot(x , 'ro', color='b')
plt.xticks(rotation=90)
plt.title('category wise Rating')
plt.xlabel('categories-->')
plt.ylabel('Rating-->')
plt.show
```

```
C:\Users\ARHAMQ~1\AppData\Local\Temp/ipykernel_2616/1012254525.py:2: UserWarning: color is redundantly defined by the 'color' keyword argument and the fmt string "ro" (-> color='r'). The keyword argument will take precedence.
  plt.plot(x , 'ro', color='b')
```

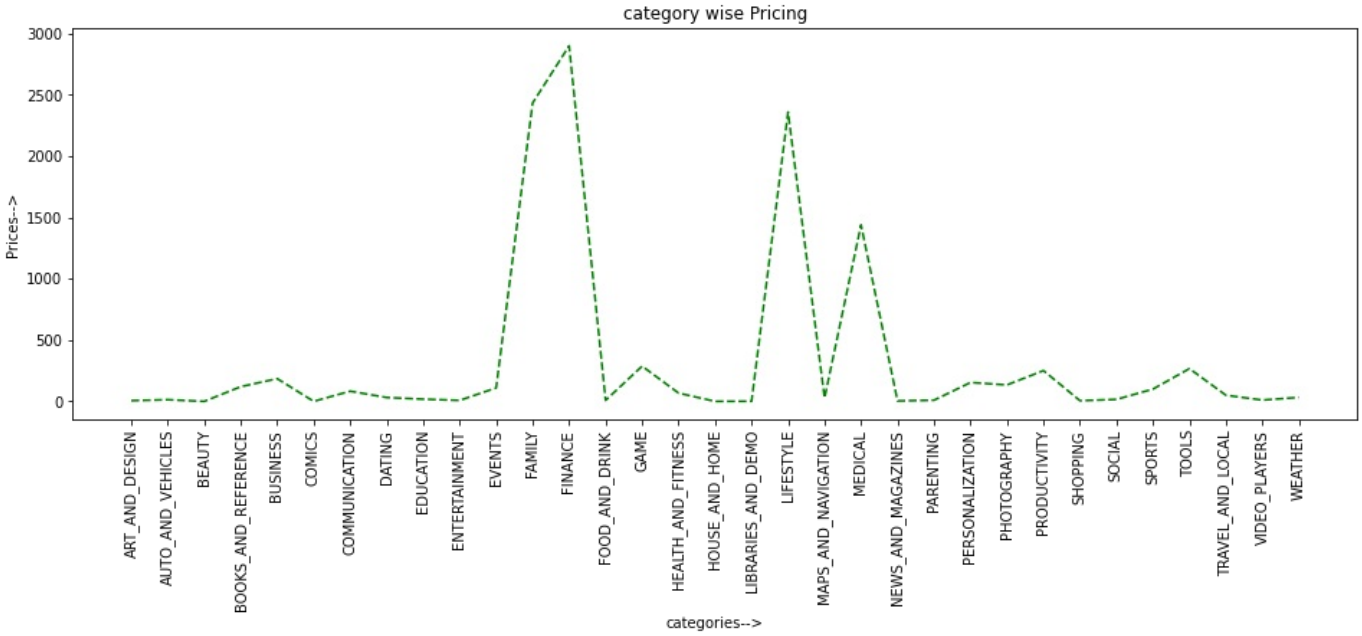
Out[33]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [34]:
```
plt.figure(figsize=(16,5))
```

```
plt.plot(y ,'r--', color= 'g')
plt.xticks(rotation=90)
plt.title('category wise Pricing')
plt.xlabel('categories-->')
plt.ylabel('Prices-->')
plt.show()
```

```
C:\Users\ARHAMQ~1\AppData\Local\Temp/ipykernel_2616/307299940.py:2: UserWarning: color is redundantly defined by
the 'color' keyword argument and the fmt string "r--" (-> color='r'). The keyword argument will take precedence.
  plt.plot(y ,'r--', color= 'g')
```



In [35]:

```
plt.figure(figsize=(16,5))
plt.plot(y ,'g^', color= 'b')
plt.xticks(rotation=90)
plt.title('category wise Reviews')
plt.xlabel('categories-->')
plt.ylabel('Reviews-->')
plt.show()
```

```
C:\Users\ARHAMQ~1\AppData\Local\Temp/ipykernel_2616/3073269057.py:2: UserWarning: color is redundantly defined by
the 'color' keyword argument and the fmt string "g^" (-> color='g'). The keyword argument will take precedence.
  plt.plot(y ,'g^', color= 'b')
```