In [62]:

```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
```

## Loading the Dataset

In [63]:

```
# Q - 1
df = pd.read_csv('E:/Study/Python/googleplaystore.csv')
```

In [64]:

```
df.shape
```

Out[64]:

```
(10841, 13)
```

### Checking the Nulls

In [65]:

```
# Q - 2
df.isnull().sum()
```

Out[65]:

```
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
```

### Droping the Null Values

In [66]:

```
# Q - 3
df.dropna(inplace=True)
```

In [67]:

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

Out[67]:

```
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 [68]:

```
df.head()
```

Out[68]:

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

### Converting Size into Numeric from String & from MB to KB conversion

In [69]:

```
# Q - 4.1

converted_of_size = []

for i in df['Size']:
    if i[-1] == 'M':
        b = float(i[:-1])
        d = 1000
        c = b * d
        f = int(c)
        converted_of_size.append(f)
    elif i == 'Varies with device':
        converted_of_size.append(0)

    else:
        e = i[:-1]
        converted_of_size.append(e)

print(converted_of_size)

df['converted_of_size'] = converted_of_size
```

```
[19000, 14000, 8700, 25000, 2800, 5600, 19000, 29000, 33000, 3100, 28000,
12000, 20000, 21000, 37000, 5500, 17000, 39000, 31000, 14000, 12000, 4200,
23000, 6000, 25000, 6100, 4600, 4200, 9200, 5200, 11000, 11000, 4200, 920
0, 24000, 0, 11000, 9400, 15000, 10000, 0, 1200, 12000, 24000, 26000, 800
0, 7900, 25000, 56000, 57000, 0, 35000, 33000, 33000, 5600, 54000, '201',
3600, 5700, 17000, 8600, 2400, 27000, 2700, 2500, 0, 0, 7000, 35000, 1600
0, 16000, 0, 17000, 3400, 8900, 3900, 2900, 38000, 32000, 37000, 15000, 54
00, 18000, 0, 38000, 1100, 0, 0, 7900, 35000, 0, 17000, 19000, 14000, 220
0, 4500, 17000, 14000, 9800, 21000, 0, 52000, 14000, 25000, 9000, 0, 1200
0, 0, 35000, 6700, 30000, 2900, 17000, 2900, 0, 0, 2600, 4200, 7100, 5700
0, 22000, 24000, 21000, 3400, 6400, 3200, 8200, 2900, 23000, 4600, 3100,
0, 4900, 9500, 0, 0, 0, 0, 0, 4200, 5400, 0, 2800, 5000, 0, 5900, 13000, 7
100, 6700, 0, 17000, 19000, 6700, 21000, 0, 37000, 15000, 23000, 19000, 23
000, 73000, 4900, 6800, 0, 0, 2900, 3500, 4000, 21000, 2300, 0, 10000, 610
0, 2100, 42000, 30000, 0, 0, 0, 29000, 0, 0, 0, 35000, 9100, 25000, 3900,
18000, 12000, 21000, 0, 0, 0, 0, 55000, 3200, 0, 0, '23', 16000, 0, 14000,
37000, 0, 0, 11000, 25000, 7300, 6500, 25000, 3100, 0, 1500, 7500, 8600,
0, 0, 0, 0, 39000, 14000, 19000, 6800, 39000, 0, 14000, 20000, 0, 0, 2600
0, 51000, 41000, 20000, 0, 12000, 48000, 10000, 22000, 11000, 8500, 8600,
```

### Converting Reviews from String to Numeric

In [70]:

```
## Q - 4.2


reviews = df['Reviews']

new_reviews = []

for i in reviews:
    a = int(i)
    new_reviews.append(a)

new_reviews
df['new_reviews'] = new_reviews
df.head()
```

Out[70]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Photo Editor & Candy Camera & Grid & ScrapBook | ART_AND_DESIGN | 4.1 | 159 | 19M | 10,000+ | Free | 0 | Everyone | Art & Design |
| **1** | Coloring book moana | ART_AND_DESIGN | 3.9 | 967 | 14M | 500,000+ | Free | 0 | Everyone | Art & Design;Pretend Play |
| **2** | U Launcher Lite – FREE Live Cool Themes | ART_AND_DESIGN | 4.7 | 87510 | 8.7M | 5,000,000+ | Free | 0 | Everyone | Art & Design |

### Converting Installs from String to Numeric

In [71]:

```
# Q - 4.3

installs = df['Installs']
New_Installs = []

for i in installs:
    b = i.replace("+","")
    c = b.replace(",","")
    d = int(c)
    New_Installs.append(d)

df['New_Installs'] = New_Installs
df.head()
```

Out[71]:

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

### Removing "$" sign from Price and creating a new column

In [72]:

```
# Q - 4.4

price = df['Price']

new_price = []

for i in price:
    if i[0] == '$':
        n = float(i[1:])
        new_price.append(n)
    else:
        m = float(i)
        new_price.append(m)
df['new_price'] = new_price
df.head()
```

Out[72]:

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

### Checking the Min and Max Rating in the df dataset

In [73]:

```
# Q - 5.1

df.Rating.describe() # There are no outliers because Min = 1 / Max = 5
```

Out[73]:

```
count    9360.000000
mean        4.191838
std         0.515263
min         1.000000
25%         4.000000
50%         4.300000
75%         4.500000
max         5.000000
Name: Rating, dtype: float64
```

### Review should not be more than Installs

In [74]:

```
# Q - 5.2

install = df['New_Installs']
g = install - new_reviews
g
```

Out[74]:

```
0            9841
1          499033
2         4912490
3        49784356
4           99033
           ...
10834         493
10836        4962
10837          96
10839         886
10840     9601693
Name: New_Installs, Length: 9360, dtype: int64
```

In [75]:

```
final_reviews = []

for i in g:
    if i < 0:
        o = 0
        final_reviews.append(o)
    else:
        p = int(i)
        final_reviews.append(p)

df['calculated_reviews'] = final_reviews
df.sort_values(by = ['calculated_reviews'],inplace = True)
df = df.drop([6183,8591,6700,2454,7402,10697,5917,4663,7147])
df.shape
```

Out[75]:

```
(9351, 18)
```

In [76]:

```
df.head()
```

Out[76]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Conter Ratin |
|---|---|---|---|---|---|---|---|---|---|
| **2450** | Tablet Reminder | MEDICAL | 5.0 | 4 | 2.5M | 5+ | Free | 0 | Everyon |
| **5245** | AJ RETAILS | SHOPPING | 5.0 | 9 | 169k | 10+ | Free | 0 | Everyon |
| **7466** | 211:CK | GAME | 5.0 | 8 | 38M | 10+ | Paid | $0.99 | Tee |
| **6870** | BV Taxi Sudan | MAPS_AND_NAVIGATION | 3.5 | 8 | 26M | 10+ | Free | 0 | Everyon |
| **7321** | CG Jobs | FAMILY | 5.0 | 8 | 14M | 10+ | Free | 0 | Everyon |

In [77]:

```
# Q - 5.3

a = df['new_price']
new_type1 = []
for i in a:
    if i == 0:
        b = 'Free'
        new_type1.append(b)
    elif i > 0 :
        c = 'paid'
        new_type1.append(c)
    else:
        d = 0
        new_type1.append(d)

new_type1
df['new_type'] = new_type1
```

In [78]:

```
df[(df['Type'] == 'Free') & (df['new_price'] > 0)] # there are no outliers
```

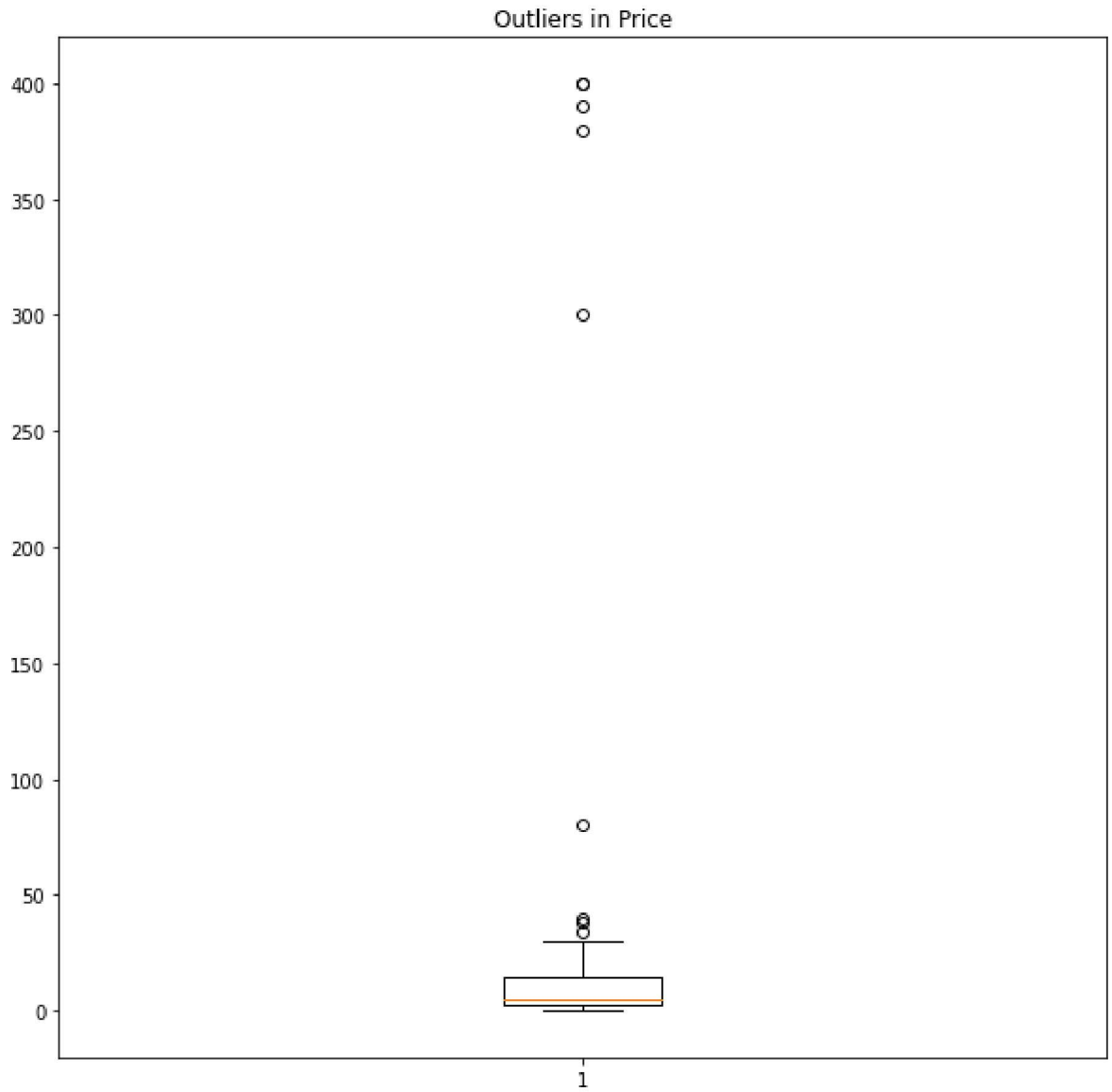
Out[78]:

| App | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres | Last Updated | Cur |
|---|---|---|---|---|---|---|---|---|---|---|---|

## Visualization

In [79]:

```
# Q - 5 = Boxplot for Price

plt.figure(figsize = (10,10))
box_price=df.new_price.unique()
box_price = sorted(box_price,reverse = True)
box_price

plt.title('Outliers in Price')

plt.boxplot(box_price)

plt.show()
```



In [80]:

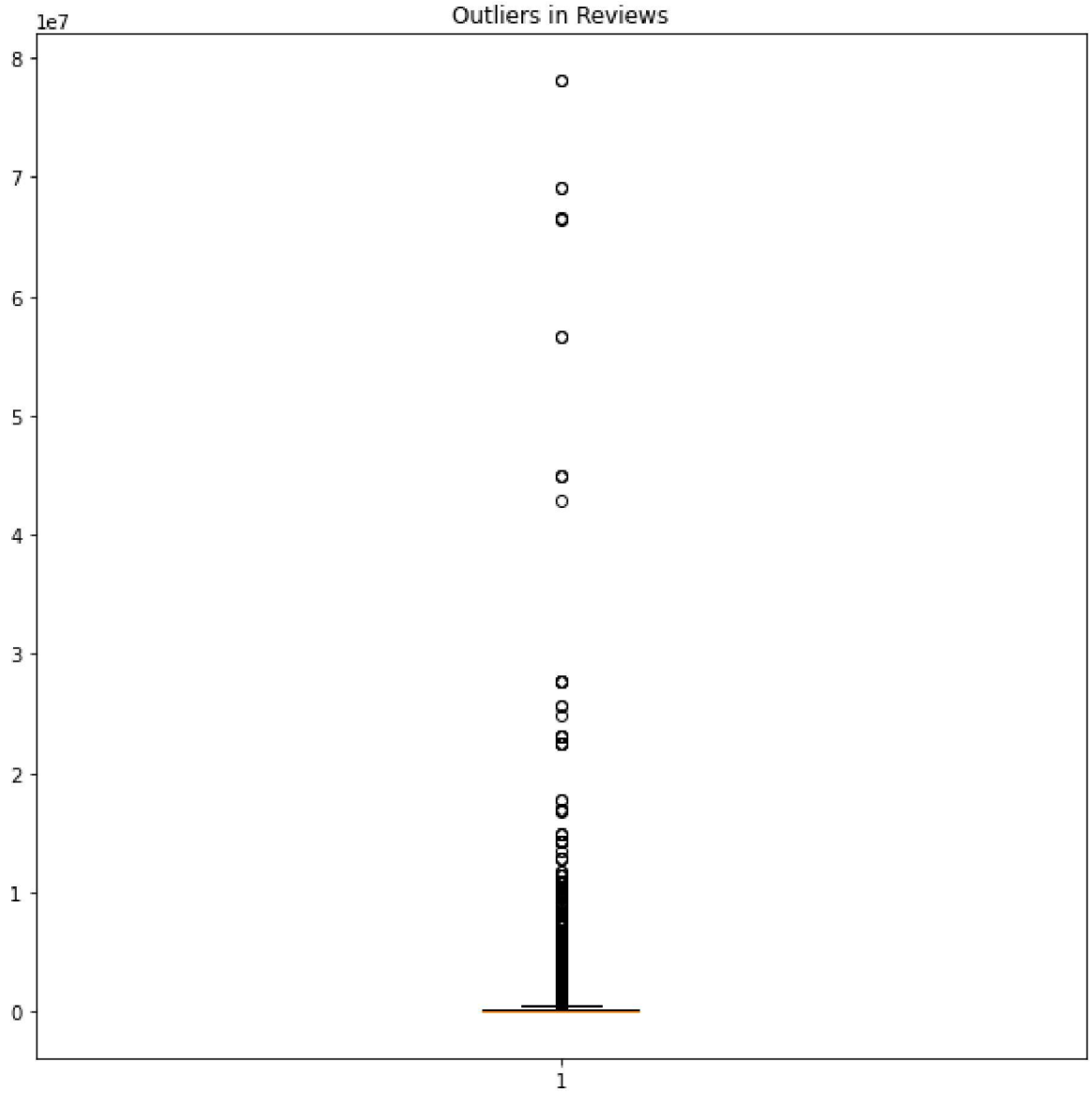
```
box_price # There are more than 200 are outliers
```

Out[80]:

```
[400.0,
 399.99,
 389.99,
 379.99,
 299.99,
 79.99,
 39.99,
 37.99,
 33.99,
 29.99,
 24.99,
 19.99,
 19.4,
 18.99,
 17.99,
 16.99,
 15.99,
 15.46,
```

In [81]:

```
# Q - 5 = Boxplot for Reviews

plt.figure(figsize = (10,10))

box_review=df.new_reviews.unique()
box_review = sorted(box_review,reverse = True)
box_review


plt.title('Outliers in Reviews')

plt.boxplot(box_review)

plt.show()
```



In [82]:

```
box_review
```

Out[82]:

```
[78158306,
 78128208,
 69119316,
 69109672,
 66577446,
 66577313,
 66509917,
 56646578,
 56642847,
 44893888,
 44891723,
 44881447,
 42916526,
 27725352,
 27724094,
 27723193,
 27722264,
 27711703.
```

In [83]:

```
# Q - 5 = Boxplot for Rating

plt.figure(figsize = (10,8))

box_rating=df.Rating.unique()
box_rating = sorted(box_rating,reverse = True)
box_rating


plt.title('Outliers in Rating')

plt.boxplot(box_rating)

plt.show()

## There are no Outliers
```

In [84]:

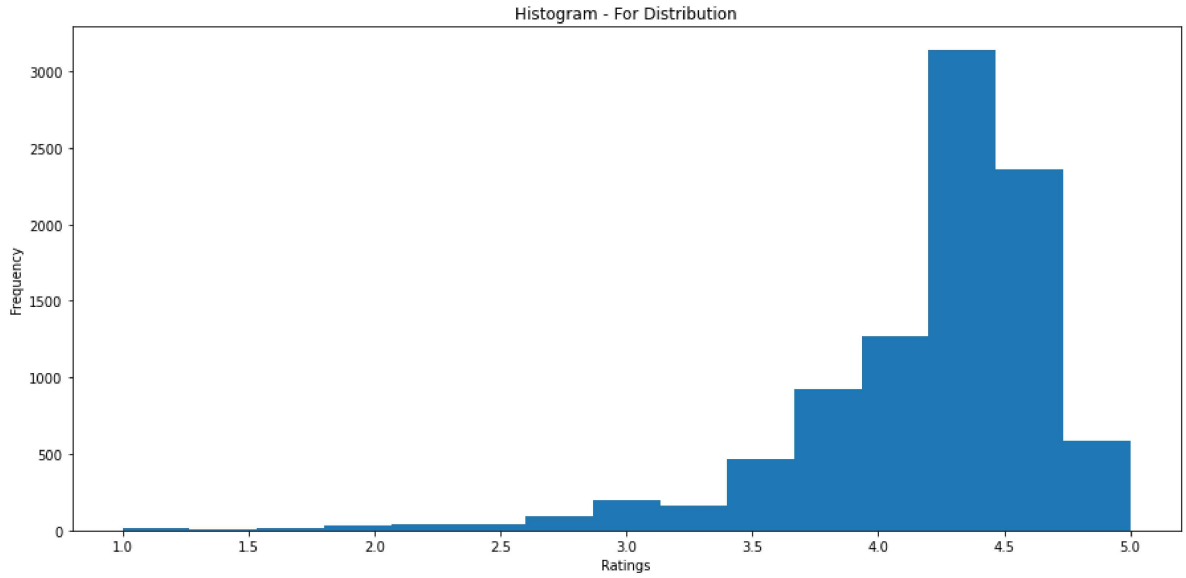
```
box_rating # No Outliers
```

Out[84]:

```
[5.0,
 4.9,
 4.8,
 4.7,
 4.6,
 4.5,
 4.4,
 4.3,
 4.2,
 4.1,
 4.0,
 3.9,
 3.8,
 3.7,
 3.6,
 3.5,
 3.4,
 3.3,
```

In [85]:

```
plt.figure(figsize =(15, 7))

plt.title('Histogram - For Distribution')

plt.hist(df['Rating'], bins = 15)

plt.xlabel('Ratings')

plt.ylabel('Frequency')

plt.show()
```




In [86]:

```
sales_across_years = df.groupby('Rating').Rating.sum()
sales_across_years.sort_values(ascending = False)

# 4 to 5 has more rating than others
```

Out[86]:

```
Rating
4.4    4875.2
4.5    4666.5
4.3    4626.8
4.2    3994.2
4.6    3785.8
4.1    2898.7
4.7    2345.3
4.0    2268.0
3.9    1505.4
5.0    1335.0
3.8    1151.4
4.8    1113.6
3.7     884.3
3.6     626.4
3.5     570.5
3.4     435.2
4.9     426.3
```

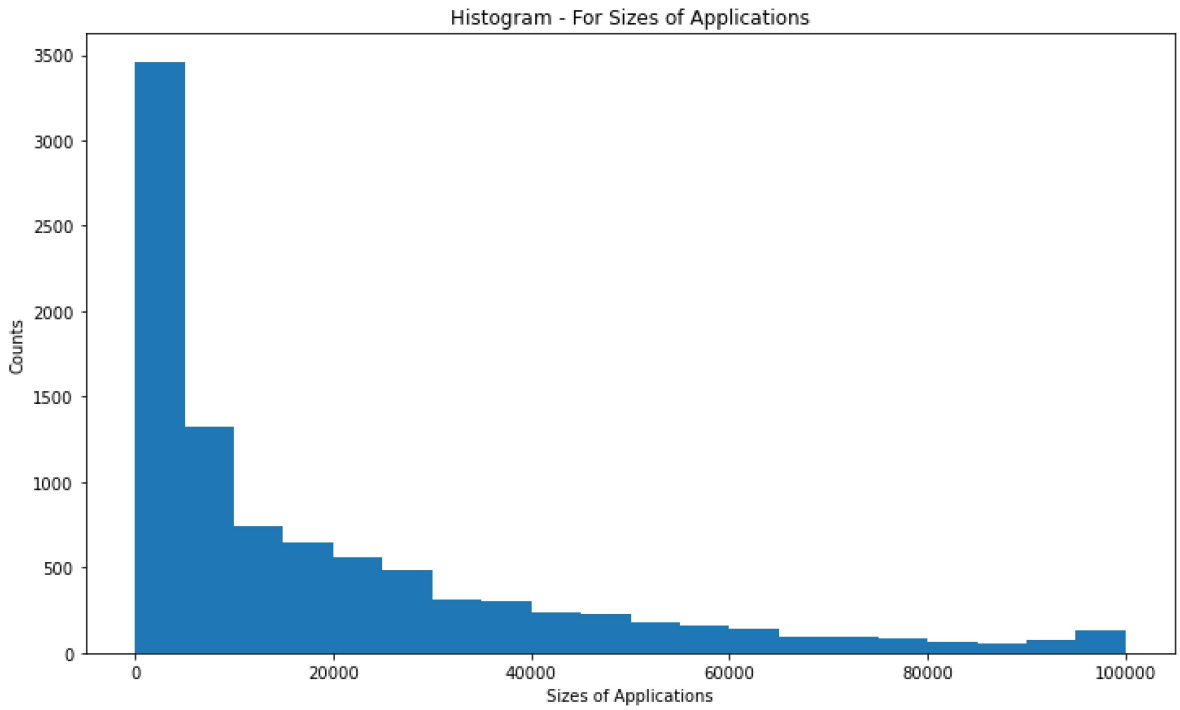
In [87]:

```
df['converted_of_size'] = df.converted_of_size.astype(float)
```

## Outliers Treatment

In [88]:

```
# Q - 5 = Histogram of Size

plt.figure(figsize = (12,7))

df['converted_of_size'] = df.converted_of_size.astype(float)

plt.title('Histogram - For Sizes of Applications')

plt.hist(df['converted_of_size'], bins = 20)

plt.xlabel('Sizes of Applications')

plt.ylabel('Counts')

plt.show()
```




### Above Histogram Shows that most of application's size have between 1 kb to 10000 kb

In [89]:

```
# Q - 6.1 = Removing outliers in Price which is more than 200

df.shape
```

Out[89]:

```
(9351, 19)
```

In [90]:

```
df = df.drop(index = df[df['new_price'] > 200].index)
df.shape
```

Out[90]:

```
(9336, 19)
```

In [91]:

```
# Q - 6.2 = Removing outliers in Reviews which is more than 2million

df.head(2)
```

Out[91]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres |
|---|---|---|---|---|---|---|---|---|---|---|
| **2450** | Tablet Reminder | MEDICAL | 5.0 | 4 | 2.5M | 5+ | Free | 0 | Everyone | Medical |
| **5245** | AJ RETAILS | SHOPPING | 5.0 | 9 | 169k | 10+ | Free | 0 | Everyone | Shopping |

In [92]:

```
df = df.drop(index = df[df['new_reviews'] > 2000000].index)
df.shape
```

Out[92]:

```
(8883, 19)
```

In [93]:

```
print(np.percentile(df['New_Installs'],10))
print(np.percentile(df['New_Installs'],25))
print(np.percentile(df['New_Installs'],50))
print(np.percentile(df['New_Installs'],70))
print(np.percentile(df['New_Installs'],90))
print(np.percentile(df['New_Installs'],95))
print(np.percentile(df['New_Installs'],96))
print(np.percentile(df['New_Installs'],97))
print(np.percentile(df['New_Installs'],98))
print(np.percentile(df['New_Installs'],99))
```

```
1000.0
10000.0
500000.0
1000000.0
10000000.0
10000000.0
50000000.0
50000000.0
50000000.0
100000000.0
```

In [94]:

```
df = df.drop(index = df[df['New_Installs'] > 50000000].index) # Deleting Outliers in Instal
df.shape
```

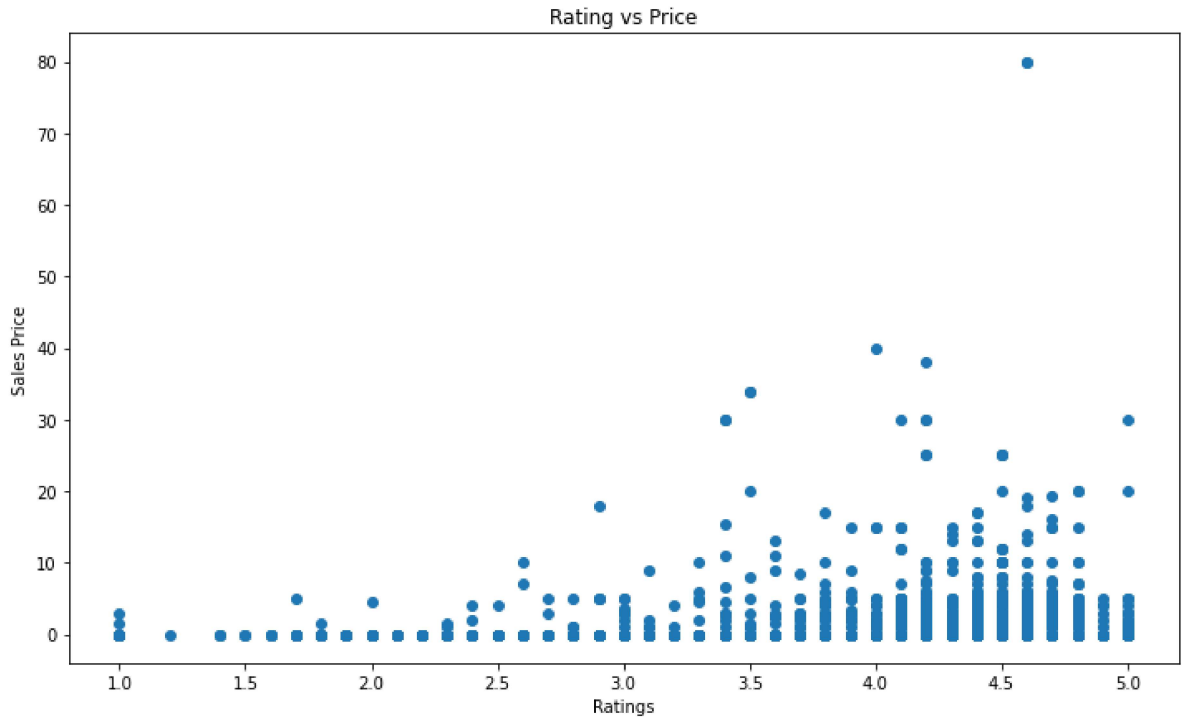
Out[94]:

```
(8741, 19)
```

## Bivariate Analysis : Visualization

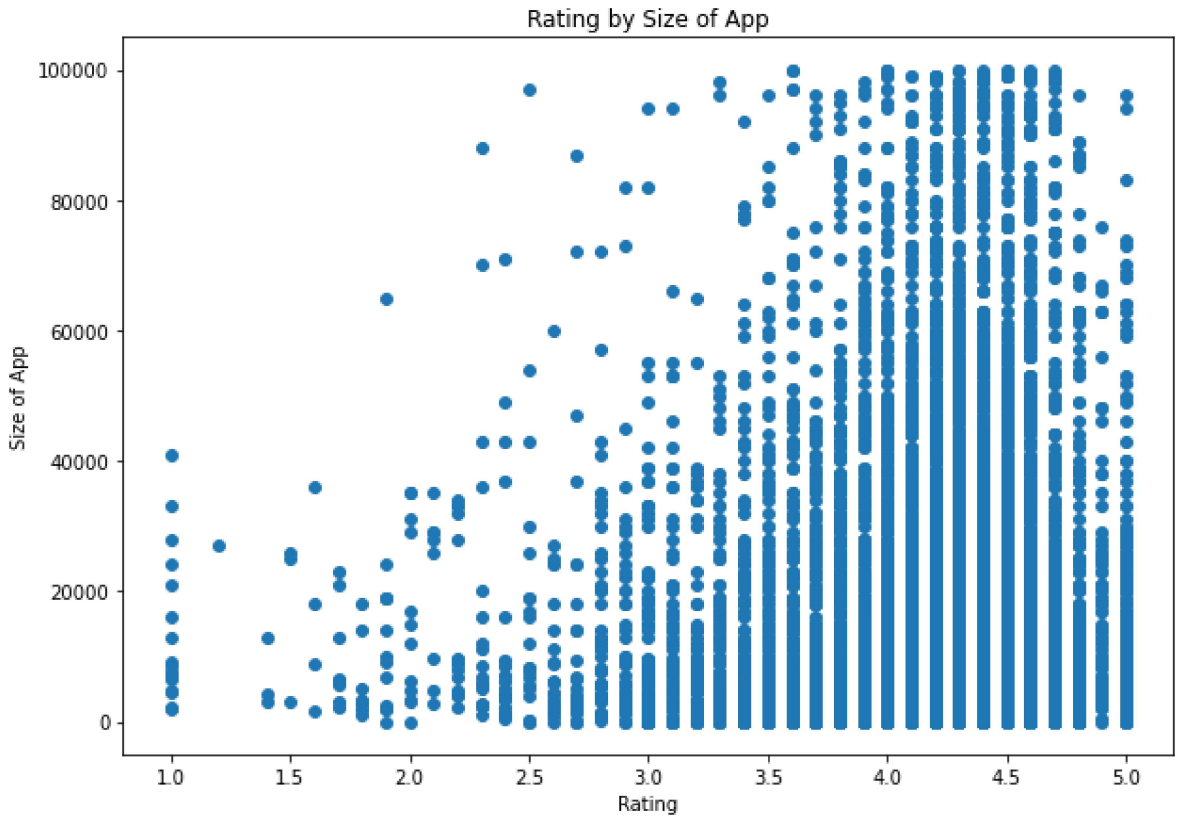
In [95]:

```
# Q = 7.1 = Scatterplot for Price of App and Rating
plt.figure(figsize = (12,7))

plt.xlabel('Ratings')
plt.ylabel('Sales Price')

plt.title('Rating vs Price')

plt.scatter(df['Rating'], df['new_price'])

plt.show()

# There are no such relationship between Rating and Price
```



In [96]:

```
# Q = 7.2 = Scatterplot for Size of App and Rating

plt.figure(figsize = (10,7))
plt.xlabel('Rating')
plt.ylabel('Size of App')

plt.title('Rating by Size of App')

plt.scatter(df['Rating'], df['converted_of_size'])

plt.show()


# Rating and size of App are connected : if rating is goin to high with that Size of App is
```



In [97]:

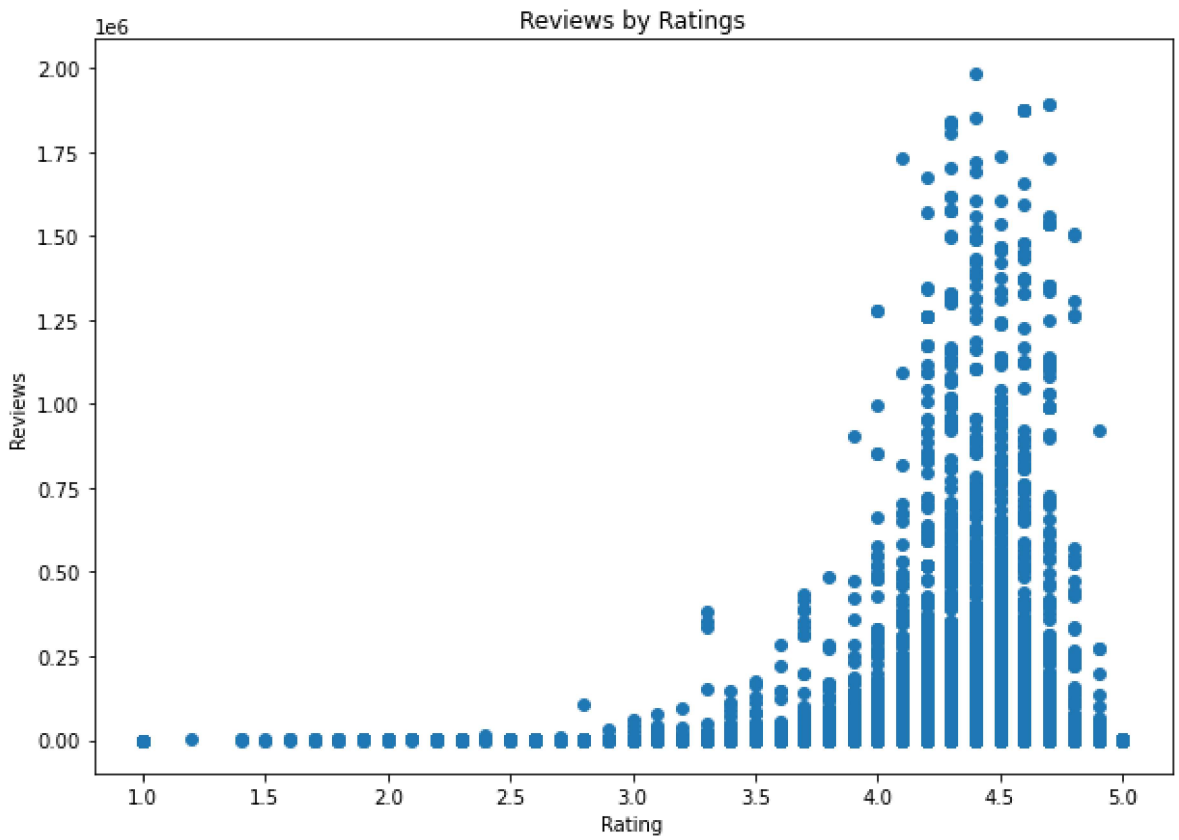
```
df.head(1)
```

Out[97]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres | Up |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **2450** | Tablet Reminder | MEDICAL | 5.0 | 4 | 2.5M | 5+ | Free | 0 | Everyone | Medical | A 3, |

In [98]:

```
# Q = 7.3 = Scatterplot for Reviews and Rating

plt.figure(figsize = (10,7))

plt.xlabel('Rating')
plt.ylabel('Reviews')

plt.title('Reviews by Ratings')

plt.scatter(df['Rating'], df['new_reviews'])

plt.show()

# There are some relation between Rating and Reviews. Between 4 to 5 ratings have maximum r
```

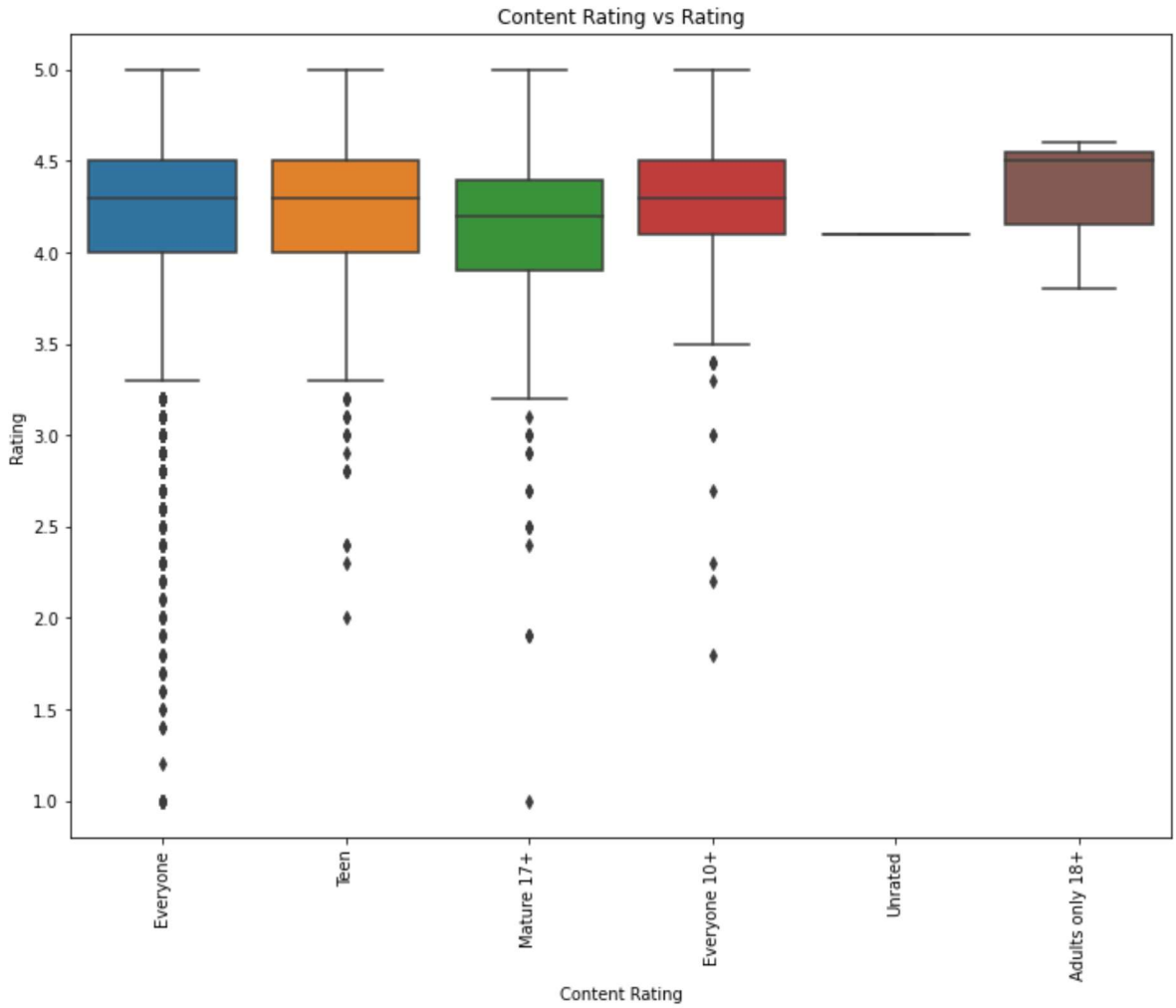


In [99]:

```
import seaborn as sns
```

In [100]:

```
# Q = 7.4 = Boxplot for Content Rating and Rating
plt.figure(figsize = (12,9))
sns.boxplot(x="Content Rating", y="Rating" ,data=df)
plt.xticks(rotation  = 90 )
plt.title('Content Rating vs Rating')

plt.show()
```



In [101]:

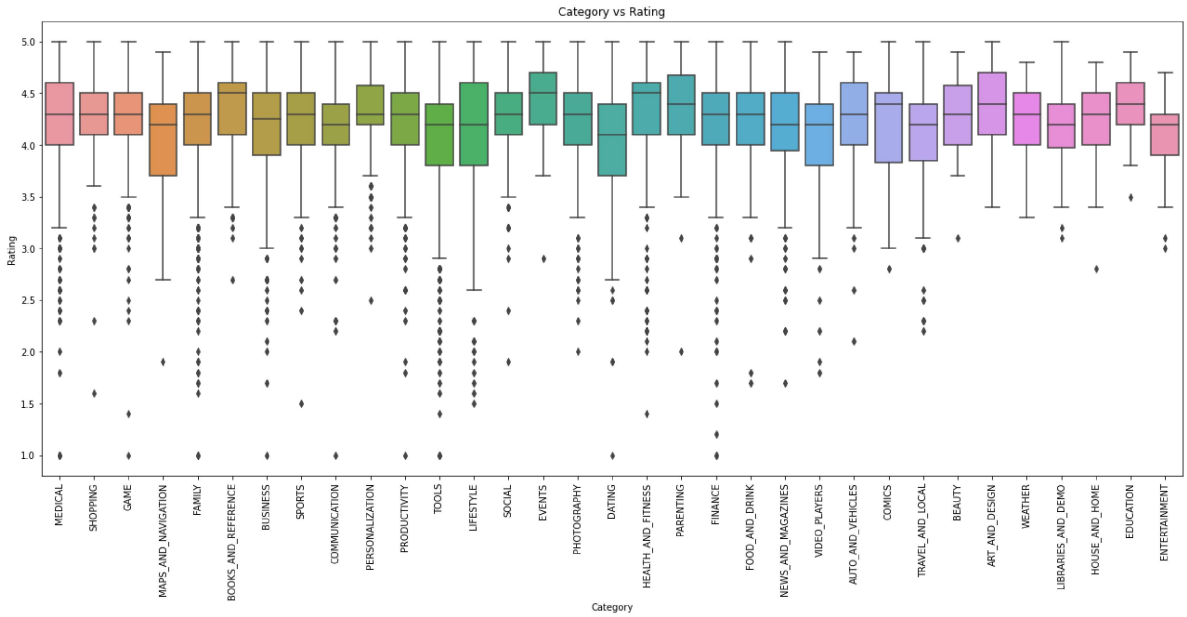
```
df.groupby(['Content Rating','Rating']).Rating.sum()
```

Out[101]:

```
Content Rating   Rating
Adults only 18+  3.8         3.8
                 4.5         4.5
                 4.6         4.6
Everyone         1.0        15.0
                 1.2         1.2
                            ...
Teen             4.7       239.7
                 4.8       110.4
                 4.9        34.3
                 5.0       105.0
Unrated          4.1         4.1
Name: Rating, Length: 119, dtype: float64
```

In [102]:

```
# Q = 7.5 = Boxplot for Category and Rating

plt.figure(figsize = (22,9))
sns.boxplot(x="Category", y="Rating", data=df)
plt.xticks(rotation = 90)
plt.title('Category vs Rating')
plt.show()
```




## Data Processing

In [103]:

```
inp1 = df.copy()
inp1.shape
```

Out[103]:

```
(8741, 19)
```

In [104]:

```
inp1.head()
```

Out[104]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Conter Ratin |
|---|---|---|---|---|---|---|---|---|---|
| **2450** | Tablet Reminder | MEDICAL | 5.0 | 4 | 2.5M | 5+ | Free | 0 | Everyon |
| **5245** | AJ RETAILS | SHOPPING | 5.0 | 9 | 169k | 10+ | Free | 0 | Everyon |
| **7466** | 211:CK | GAME | 5.0 | 8 | 38M | 10+ | Paid | $0.99 | Tee |
| **6870** | BV Taxi Sudan | MAPS_AND_NAVIGATION | 3.5 | 8 | 26M | 10+ | Free | 0 | Everyon |
| **7321** | CG Jobs | FAMILY | 5.0 | 8 | 14M | 10+ | Free | 0 | Everyon |

In [105]:

```
# np.log1p for Review

review = inp1['new_reviews']
review = np.log1p(review)
log_review = []

for i in review:
    a = round(i,4)
    log_review.append(a)

inp1['log_review'] = log_review
inp1.head()
```

Out[105]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Conter Ratin |
|---|---|---|---|---|---|---|---|---|---|
| **2450** | Tablet Reminder | MEDICAL | 5.0 | 4 | 2.5M | 5+ | Free | 0 | Everyon |
| **5245** | AJ RETAILS | SHOPPING | 5.0 | 9 | 169k | 10+ | Free | 0 | Everyon |
| **7466** | 211:CK | GAME | 5.0 | 8 | 38M | 10+ | Paid | $0.99 | Tee |
| **6870** | BV Taxi Sudan | MAPS_AND_NAVIGATION | 3.5 | 8 | 26M | 10+ | Free | 0 | Everyon |
| **7321** | CG Jobs | FAMILY | 5.0 | 8 | 14M | 10+ | Free | 0 | Everyon |

In [106]:

```
# np.log1p for Installs

installs = inp1['New_Installs']
installs = np.log1p(installs)
log_installs = []

for i in installs:
    b = round(i,4)
    log_installs.append(b)


inp1['log_installs'] = log_installs
inp1.head()
```

Out[106]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Conter Ratin |
|---|---|---|---|---|---|---|---|---|---|
| **2450** | Tablet Reminder | MEDICAL | 5.0 | 4 | 2.5M | 5+ | Free | 0 | Everyon |
| **5245** | AJ RETAILS | SHOPPING | 5.0 | 9 | 169k | 10+ | Free | 0 | Everyon |
| **7466** | 211:CK | GAME | 5.0 | 8 | 38M | 10+ | Paid | $0.99 | Tee |
| **6870** | BV Taxi Sudan | MAPS_AND_NAVIGATION | 3.5 | 8 | 26M | 10+ | Free | 0 | Everyon |
| **7321** | CG Jobs | FAMILY | 5.0 | 8 | 14M | 10+ | Free | 0 | Everyon |

5 rows × 21 columns

In [107]:

```
inp1.columns
```

Out[107]:

```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
       'Android Ver', 'converted_of_size', 'new_reviews', 'New_Installs',
       'new_price', 'calculated_reviews', 'new_type', 'log_review',
       'log_installs'],
      dtype='object')
```

In [108]:

```
# Removing the unwanted columns

inp1.drop(['App','Current Ver','Android Ver','Last Updated'],axis = 'columns', inplace = Tr

inp1.columns
```

Out[108]:

```
Index(['Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type', 'Price',
       'Content Rating', 'Genres', 'converted_of_size', 'new_reviews',
       'New_Installs', 'new_price', 'calculated_reviews', 'new_type',
       'log_review', 'log_installs'],
      dtype='object')
```

In [109]:

```
inp2 = inp1.copy()
```

In [110]:

```
inp2['Category'] = pd.get_dummies(inp2['Category'])
inp2['Genres'] = pd.get_dummies(inp2['Genres'])
inp2['Content Rating'] = pd.get_dummies(inp2['Content Rating'])
inp2.head(10)
```

Out[110]:

| | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres | converted_of_ |
|---|---|---|---|---|---|---|---|---|---|---|
| **2450** | 0 | 5.0 | 4 | 2.5M | 5+ | Free | 0 | 0 | 0 | 25 |
| **5245** | 0 | 5.0 | 9 | 169k | 10+ | Free | 0 | 0 | 0 | 1 |
| **7466** | 0 | 5.0 | 8 | 38M | 10+ | Paid | $0.99 | 0 | 0 | 380 |
| **6870** | 0 | 3.5 | 8 | 26M | 10+ | Free | 0 | 0 | 0 | 260 |
| **7321** | 0 | 5.0 | 8 | 14M | 10+ | Free | 0 | 0 | 0 | 140 |
| **9039** | 0 | 5.0 | 7 | 73M | 10+ | Paid | $0.99 | 0 | 0 | 730 |
| **2484** | 0 | 5.0 | 2 | 7.1M | 5+ | Free | 0 | 0 | 0 | 71 |
| **8146** | 0 | 5.0 | 2 | 1.4M | 5+ | Free | 0 | 0 | 0 | 14 |
| **7731** | 0 | 5.0 | 2 | 3.4M | 5+ | Free | 0 | 0 | 0 | 34 |
| **10562** | 0 | 1.5 | 2 | 26M | 5+ | Free | 0 | 0 | 0 | 260 |

In [111]:

```
inp2.drop(['Size','Installs','new_type','Reviews','New_Installs','Type','Price'],axis = 'co
inp2.head(10)
```

Out[111]:

| | Category | Rating | Content Rating | Genres | converted_of_size | new_reviews | new_price | calculat |
|---|---|---|---|---|---|---|---|---|
| **2450** | 0 | 5.0 | 0 | 0 | 2500.0 | 4 | 0.00 | |
| **5245** | 0 | 5.0 | 0 | 0 | 169.0 | 9 | 0.00 | |
| **7466** | 0 | 5.0 | 0 | 0 | 38000.0 | 8 | 0.99 | |
| **6870** | 0 | 3.5 | 0 | 0 | 26000.0 | 8 | 0.00 | |
| **7321** | 0 | 5.0 | 0 | 0 | 14000.0 | 8 | 0.00 | |
| **9039** | 0 | 5.0 | 0 | 0 | 73000.0 | 7 | 0.99 | |
| **2484** | 0 | 5.0 | 0 | 0 | 7100.0 | 2 | 0.00 | |
| **8146** | 0 | 5.0 | 0 | 0 | 1400.0 | 2 | 0.00 | |
| **7731** | 0 | 5.0 | 0 | 0 | 3400.0 | 2 | 0.00 | |
| **10562** | 0 | 1.5 | 0 | 0 | 26000.0 | 2 | 0.00 | |

## Train & Test Spliting the Dataset

In [112]:

```
from sklearn.model_selection import train_test_split
```

In [113]:

```
x = inp2.iloc[:,[0,2,3,4,5,6,7,8,9]]
y = inp2.iloc[:,[1]]
```

In [114]:

```
X_train,X_test,Y_train,Y_test = train_test_split(x,y,train_size = 0.70)
```

In [115]:

```
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(6118, 9)
(2623, 9)
(6118, 1)
(2623, 1)
```

## Model Building

In [116]:

```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import r2_score
```

In [117]:

```
lr = LinearRegression().fit(X_train,Y_train)
lr
```

Out[117]:

```
LinearRegression()
```

In [118]:

```
lr.intercept_
```

Out[118]:

```
array([4.60747446])
```

In [119]:

```
lr.coef_
```

Out[119]:

```
array([[ 2.78213272e-01,  1.05600705e-02, -1.03378670e-02,
        -1.14292082e-07, -6.99492998e-08, -8.06507998e-03,
         3.83466657e-09,  1.67420249e-01, -1.48290676e-01]])
```

In [120]:

```
Y_pred = lr.predict(X_test)
Y_pred
```

Out[120]:

```
array([[4.167509  ],
       [4.14417712],
       [4.09429319],
       ...,
       [4.26576518],
       [4.35387111],
       [4.29614698]])
```

In [121]:

```
r2_score(Y_test,Y_pred)
```

Out[121]:

```
0.12495521547377697
```

In [122]:

```
print('MAE:',metrics.mean_absolute_error(Y_test,Y_pred))
print('MSE:',metrics.mean_squared_error(Y_test,Y_pred))
print('RMSE:',np.sqrt(metrics.mean_absolute_error(Y_test,Y_pred)))
```

```
MAE: 0.34372188256697417
MSE: 0.24523297559529394
RMSE: 0.5862779908601159
```