Google Play Store Data EDA

Importing Libraries

In [359...
```python
import pandas as pd
import numpy as np
```

Importing Google Play Store Dataset

In [360...
```python
df=pd.read_csv("playstore-analysis.csv")
```

Sample data of first two rows

In [361...
```python
df.head(2)
```

Out[361...

	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	19000.0	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	14000.0	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

In [362...
```python
print(f"Total row and columns of data :\n{df.shape}")
```

```
Total row and columns of data :
(10841, 13)
```

Basic Information about dataset

In [363...
```python
print(df.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  float64
 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(2), object(11)
memory usage: 1.1+ MB
None
```

Part 1: Data clean up – Missing value treatment

Que)a. Drop records where rating is missing since rating is our target/study variable

In [364...
```python
print(f"Count of Missing Values in rating Columns:{df['Rating'].isnull().sum()}")
```

```
Count of Missing Values in rating Columns:1474
```

Que)b. Droping Missing value in dataset

In [365...
```python
df.dropna(subset=['Rating'],inplace=True)
```

In [366...
```python
print(f"Rechecking Missing Values in rating Columns:\n Result:->{df['Rating'].isnull().sum()}\n \nWe have succe
```

```
Rechecking Missing Values in rating Columns:
 Result:->0

We have successfully removed missing value in rating
```

Que)c. Check the null values for the Android Ver column

In [367...
```python
df.loc[df['Android Ver'].isnull()]
```

Out[367...

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	A
4453	[substratum] Vacuum: P	PERSONALIZATION	4.4	230	11000.000000	1,000+	Paid	$1.49	Everyone	Personalization	July 20, 2018	4.4	
4490	Pi Dark [substratum]	PERSONALIZATION	4.5	189	2100.000000	10,000+	Free	0	Everyone	Personalization	March 27, 2018	1.1	
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0M	21516.529524	Free	0	Everyone	NaN	February 11, 2018	1.0.19	4.0 and up	

In [368...
```python
df1=df.drop([10472], axis=0)
```

In [369...
```python
print(f"{df1.loc[df1['Android Ver'].isnull()]}\n\n We removed' Life Made WI-Fi Touchscreen Photo Frame row 'fro
```

```
                         App         Category  Rating Reviews     Size  \
4453  [substratum] Vacuum: P  PERSONALIZATION     4.4     230  11000.0
4490    Pi Dark [substratum]  PERSONALIZATION     4.5     189   2100.0

     Installs  Type  Price Content Rating           Genres    Last Updated  \
4453   1,000+  Paid  $1.49       Everyone  Personalization   July 20, 2018
4490  10,000+  Free      0       Everyone  Personalization  March 27, 2018

     Current Ver Android Ver
4453         4.4         NaN
4490         1.1         NaN

 We removed' Life Made WI-Fi Touchscreen Photo Frame row 'from data
```

Que)c. Replace remaining missing values with the mode

In [370...
```python
df1['Android Ver'].fillna(df1['Android Ver'].mode()[0],inplace=True)
```

In [371...
```python
print(f"Missing Value in Android Version= {df1['Android Ver'].isnull().sum()}")
```

```
Missing Value in Android Version= 0
```

Part 2. Data clean up – correcting the data types

Que)a. Which all variables need to be brought to numeric types?

In [372...
```python
print(f"{df1.dtypes}\n \nFrom above we can say columns Reviews,Installs,Price needs to brought into numeric dat
```

```
App                object
Category           object
Rating            float64
Reviews            object
Size              float64
Installs           object
Type               object
Price              object
Content Rating     object
Genres             object
Last Updated       object
Current Ver        object
Android Ver        object
dtype: object

From above we can say columns Reviews,Installs,Price needs to brought into numeric datatype
```

In [373...
```python
df1['Reviews'] = df1['Reviews'].astype(int)
df1['Size'] = df1['Size'].astype(int)
#df1['Installs']=df1['Installs'].astype(int)

# Additional conversion
df1['Last Updated'] = df1['Last Updated'].astype('datetime64[ns]')
```

In [374...
```python
print(f"Change Datatype Checking..\n\n {df1.dtypes}")
```

```
Change Datatype Checking..

 App                        object
Category                   object
Rating                    float64
Reviews                     int32
Size                        int32
Installs                   object
Type                       object
Price                      object
Content Rating             object
Genres                     object
Last Updated       datetime64[ns]
Current Ver                object
Android Ver                object
dtype: object
```

Que)b. Price variable – remove $ sign and convert to float

In [375...
```python
df1['Price']  = df1['Price'] .replace({'\$':''}, regex = True)
```

In [376...
```python
df1.drop(labels=df1[df1['Price']=='Everyone'].index, inplace = True)

df1['Price']= df1['Price'].astype('float')
```

In [378...
```python
df1.Price.value_counts()
```

Out[378...
```
0.00      8719
2.99       114
0.99       107
4.99        70
1.99        59
3.99        58
1.49        31
2.49        21
5.99        18
9.99        16
6.99        13
399.99      11
14.99       10
4.49         9
3.49         7
7.99         7
29.99        6
12.99        5
19.99        5
24.99        5
11.99        5
8.99         4
10.00        3
16.99        3
5.49         3
79.99        2
7.49         2
33.99        2
10.99        2
1.00         2
1.70         2
9.00         2
17.99        2
3.95         2
1.29         1
4.77         1
1.76         1
389.99       1
2.59         1
3.28         1
3.88         1
2.56         1
4.84         1
1.97         1
3.04         1
39.99        1
2.00         1
14.00        1
1.75         1
2.50         1
1.50         1
3.02         1
1.20         1
4.60         1
3.08         1
19.40        1
4.59         1
15.46        1
8.49         1
15.99        1
18.99        1
379.99       1
2.95         1
13.99        1
37.99        1
6.49         1
4.29         1
400.00       1
299.99       1
1.59         1
1.61         1
3.90         1
2.90         1
Name: Price, dtype: int64
```

Finally !!! $ symbol removed from Price columns

Que)c. Installs – remove ',' and '+' sign, convert to integer

In [379...
```python
df1['Installs']=df1['Installs'].str.replace('\+' ,"")
df1['Installs']=df1['Installs'].str.replace('\,' ,"")
```

In [380...
```python
df1[df1['Installs']=="+"]
```

Out[380...

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

In [381...
```python
df1[df1['Installs']==","]
```

Out[381...

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

In [382...
```python
df1['Installs']=df1['Installs'].astype(int)
```

In [383...
```python
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9366 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   App             9366 non-null   object
 1   Category        9366 non-null   object
 2   Rating          9366 non-null   float64
 3   Reviews         9366 non-null   int32
 4   Size            9366 non-null   int32
 5   Installs        9366 non-null   int32
 6   Type            9366 non-null   object
 7   Price           9366 non-null   float64
 8   Content Rating  9366 non-null   object
 9   Genres          9366 non-null   object
 10  Last Updated    9366 non-null   datetime64[ns]
 11  Current Ver     9362 non-null   object
 12  Android Ver     9366 non-null   object
dtypes: datetime64[ns](1), float64(2), int32(3), object(7)
memory usage: 1.2+ MB
```

Yup!!we removed + and , sign from colmn Installs as well as change datatype to int as per requirement

Part 3. Sanity checks – check for the following and handle accordingly

Que)a. Avg. rating should be between 1 and 5, as only these values are allowed on the play store.

In [384...
```python
df1.Rating.value_counts()
```

Out[384...
```
4.4    1109
4.3    1076
4.5    1038
4.2     952
4.6     823
4.1     708
4.0     568
4.7     499
3.9     386
3.8     303
5.0     274
3.7     239
4.8     234
3.6     174
3.5     163
3.4     128
3.3     102
4.9      87
3.0      83
3.1      69
3.2      64
2.9      45
2.8      42
2.6      25
2.7      25
2.5      21
2.3      20
2.4      19
1.0      16
2.2      14
1.9      13
2.0      12
1.7       8
2.1       8
1.8       8
1.6       4
1.4       3
1.5       3
1.2       1
Name: Rating, dtype: int64
```

Above result shows that Value is between 1 to 5

Que)b Reviews should not be more than installs as only those who installed can review the app.

- i. Are there any such records? Drop if so.

In [385...
```python
df1.loc[df1['Reviews'] > df1['Installs']]
```

Out[385...

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
2454	KBA-EZ Health Guide	MEDICAL	5.0	4	25000	1	Free	0.00	Everyone	Medical	2018-08-02	1.0.72	4.0.3 and up
4663	Alarmy (Sleep If U Can) - Pro	LIFESTYLE	4.8	10249	21516	10000	Paid	2.49	Everyone	Lifestyle	2018-07-30	Varies with device	Varies with device
5917	Ra Ga Ba	GAME	5.0	2	20000	1	Paid	1.49	Everyone	Arcade	2017-02-08	1.0.4	2.3 and up
6700	Brick Breaker BR	GAME	5.0	7	19000	5	Free	0.00	Everyone	Arcade	2018-07-23	1.0	4.1 and up
7402	Trovami se ci riesci	GAME	5.0	11	6100	10	Free	0.00	Everyone	Arcade	2017-03-11	0.1	2.3 and up
8591	DN Blog	SOCIAL	5.0	20	4200	10	Free	0.00	Teen	Social	2018-07-23	1.0	4.0 and up
10697	Mu.F.O.	GAME	5.0	2	16000	1	Paid	0.99	Everyone	Arcade	2017-03-03	1.0	2.3 and up

In [386...
```python
df2=(df1.loc[df1['Reviews'] > df1['Installs']]).index
```

In [387...
```python
df1.drop(df2,inplace=True)
df1.head(2)
```

Out[387...

	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	19000	10000	Free	0.0	Everyone	Art & Design	2018-01-07	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14000	500000	Free	0.0	Everyone	Art & Design;Pretend Play	2018-01-15	2.0.0	4.0.3 and up

In [388...
```python
df1.loc[df1['Reviews'] > df1['Installs']]
```

Out[388...

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

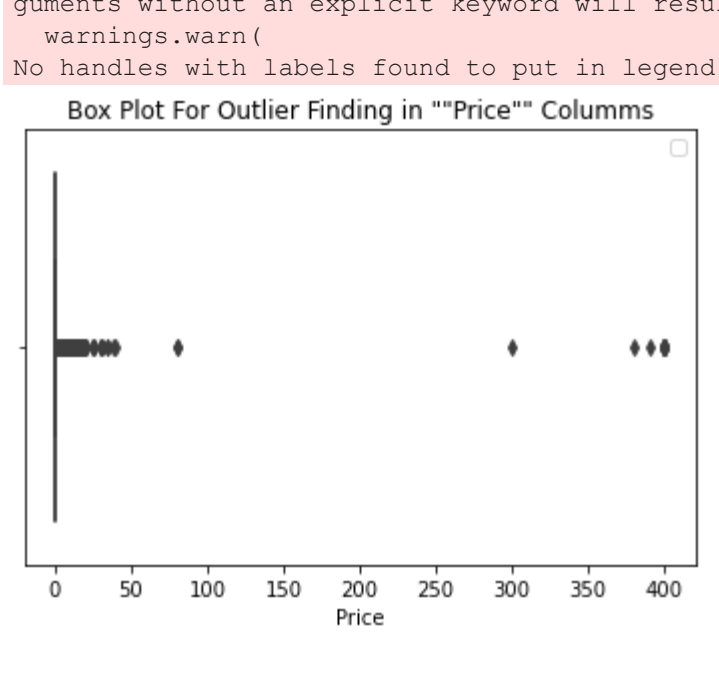
Part 4. Identify and Handle outliers – a. Price column

Que)a. Make suitable plot to identify outliers in price

In [389...
```python
import matplotlib.pyplot as plt
import seaborn as sns
```

In [390...
```python
sns.boxplot(df1['Price'])
plt.title('Box Plot For Outlier Finding in ""Price"" Columms')
plt.legend()
plt.show()
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
No handles with labels found to put in legend.
```



Que) b. Do you expect apps on the play store to cost $200? Check out these case

In [391...
```python
df1['Price'].value_counts().sort_index()
```

Out[391...
```
0.00      8715
0.99       106
1.00         2
1.20         1
1.29         1
1.49        30
1.50         1
1.59         1
1.61         1
1.70         2
1.75         1
1.76         1
1.97         1
1.99        59
2.00         1
2.49        20
2.50         1
2.56         1
2.59         1
2.90         1
2.95         1
2.99       114
3.02         1
3.04         1
3.08         1
3.28         1
3.49         7
3.88         1
3.90         1
3.95         2
3.99        58
4.29         1
4.49         9
4.59         1
4.60         1
4.77         1
4.84         1
4.99        70
5.49         3
5.99        18
6.49         1
6.99        13
7.49         2
7.99         7
8.49         1
8.99         4
9.00         2
9.99        16
10.00        3
10.99        2
11.99        5
12.99        5
13.99        1
14.00        1
14.99       10
15.46        1
15.99        1
16.99        3
17.99        2
18.99        1
19.40        1
19.99        5
24.99        5
29.99        6
33.99        2
37.99        1
39.99        1
79.99        2
299.99       1
379.99       1
389.99       1
399.99      11
400.00       1
Name: Price, dtype: int64
```

In [392...
```python
df1[df1['Price']>200]
```

Out[392...

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
4197	most expensive app (H)	FAMILY	4.3	6	1500	100	Paid	399.99	Everyone	Entertainment	2018-07-16	1.0	7.0 and up
4362	💎 I'm rich	LIFESTYLE	3.8	718	26000	10000	Paid	399.99	Everyone	Lifestyle	2018-03-11	1.0.0	4.4 and up
4367	I'm Rich - Trump Edition	LIFESTYLE	3.6	275	7300	10000	Paid	400.00	Everyone	Lifestyle	2018-05-03	1.0.1	4.1 and up
5351	I am rich	LIFESTYLE	3.8	3547	1800	100000	Paid	399.99	Everyone	Lifestyle	2018-01-12	2.0	4.0.3 and up
5354	I am Rich Plus	FAMILY	4.0	856	8700	10000	Paid	399.99	Everyone	Entertainment	2018-05-19	3.0	4.4 and up
5355	I am rich VIP	LIFESTYLE	3.8	411	2600	10000	Paid	299.99	Everyone	Lifestyle	2018-07-21	1.1.1	4.3 and up
5356	I Am Rich Premium	FINANCE	4.1	1867	4700	50000	Paid	399.99	Everyone	Finance	2017-11-12	1.6	4.0 and up
5357	I am extremely Rich	LIFESTYLE	2.9	41	2900	1000	Paid	379.99	Everyone	Lifestyle	2018-07-01	1.0	4.0 and up
5358	I am Rich!	FINANCE	3.8	93	22000	1000	Paid	399.99	Everyone	Finance	2017-12-11	1.0	4.1 and up
5359	I am rich(premium)	FINANCE	3.5	472	965	5000	Paid	399.99	Everyone	Finance	2017-05-01	3.4	4.4 and up
5362	I Am Rich Pro	FAMILY	4.4	201	2700	5000	Paid	399.99	Everyone	Entertainment	2017-05-30	1.54	1.6 and up
5364	I am rich (Most expensive app)	FINANCE	4.1	129	2700	1000	Paid	399.99	Teen	Finance	2017-12-06	2	4.0.3 and up
5366	I Am Rich	FAMILY	3.6	217	4900	10000	Paid	389.99	Everyone	Entertainment	2018-06-22	1.5	4.2 and up
5369	I am Rich	FINANCE	4.3	180	3800	5000	Paid	399.99	Everyone	Finance	2018-03-22	1.0	4.2 and up
5373	I AM RICH PRO PLUS	FINANCE	4.0	36	41000	1000	Paid	399.99	Everyone	Finance	2018-06-25	1.0.2	4.1 and up

Yes there are case where price is more than $200

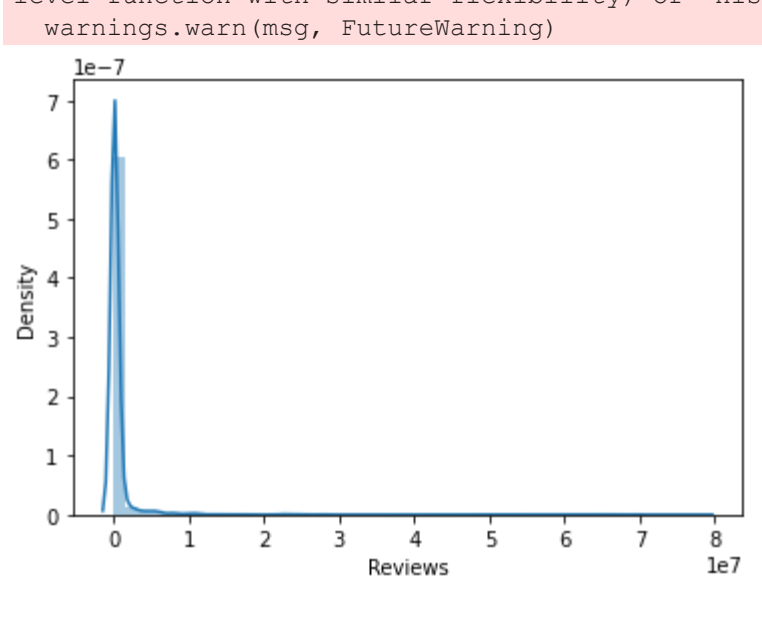
In [393...
```python
temp = df1[df1['Price'] > 30].index
df1.drop(labels=temp, inplace=True)
```

Que)b. Reviews column

i. Make suitable plot ii. Limit data to apps with < 1 Million reviews

In [394...
```python
sns.distplot(df1['Reviews'])
plt.show()
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



- Limit data to apps with < 1 Million reviews

In [395...
```python
temp1 = df1[df1['Reviews'] > 1000000 ].index
df1.drop(labels=temp1, inplace=True)
print(temp1.value_counts().sum(),'cols dropped')
```

```
704 cols dropped
```

In [396...
```python
df1.head(2)
```

Out[396...

	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	19000	10000	Free	0.0	Everyone	Art & Design	2018-01-07	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14000	500000	Free	0.0	Everyone	Art & Design;Pretend Play	2018-01-15	2.0.0	4.0.3 and up

Que)c Installs

- i. What is the 95th percentile of the installs?
- ii. Drop records having a value more than the 95th percentile

In [398...
```python
val=df1['Installs'].quantile(0.95) # 95th percentile
print(val)
```

```
10000000.0
```

In [399...
```python
# Verification
x = df1.Installs.quantile() > val
print (x)
```

```
False
```

Part 5: Data analysis to answer business questions

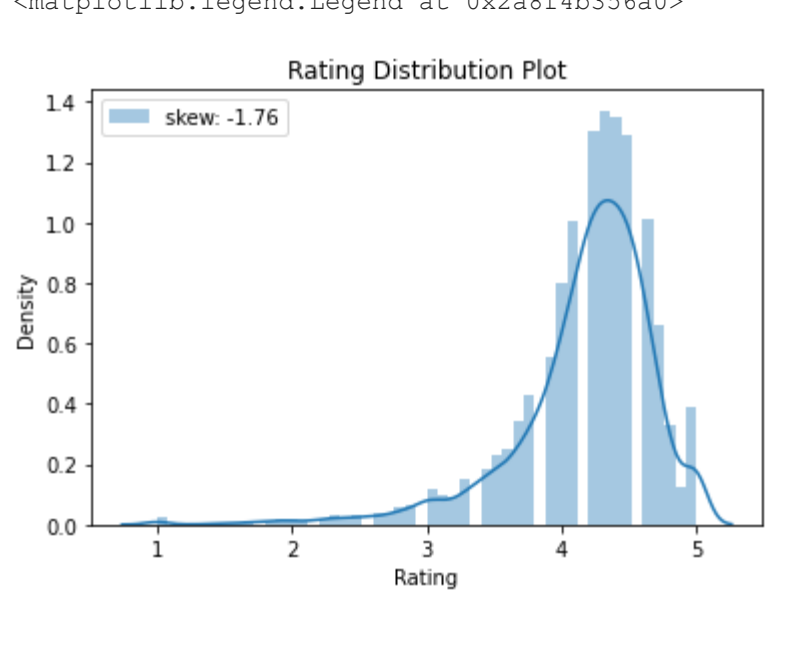
- What is the distribution of ratings like? (use Seaborn) More skewed towards higher/lower values?

In [400...
```python
sns.distplot(df1['Rating'], label="skew: " + str(np.round(df1['Rating'].skew(),2)))

plt.title('Rating Distribution Plot')
plt.legend()
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[400...
```
<matplotlib.legend.Legend at 0x2a8f4b356a0>
```



5. What is the distribution of ratings like? (use Seaborn) More skewed towards higher/lower values

In [401...
```python
print('The Median of this distribution {} It is greater than mean {} of this distribution'.format(df1.Rating.me
```

```
The Median of this distribution 4.3 It is greater than mean 4.172492471623822 of this distribution
```

- Since mode>= median > mean, the distribution of Rating is Negatively Skewed.
- Therefore distribution of Rating is more Skewed towards lower values.

Rating values are left skewed distribution or negatively skewed distribution as it's long tail is on the negative direction.

- From the distribution we can find that most of the rating is in between 4 to 5.

6. What are the top Content Rating values?

In [402...
```python
df1['Content Rating'].value_counts()
```

Out[402...
```
Everyone           6943
Teen                933
Mature 17+          417
Everyone 10+        337
Adults only 18+       3
Unrated               1
Name: Content Rating, dtype: int64
```

Top Content Rating values : Content Rating person are Everyone and Its count is 7414

6a) Are there any values with very few records?

Adults only 18+ and Unrated has less count in Content Rating so we drop them.

6b) If yes, drop those as they won't help in the analysis

In [403...
```python
temp3=df1.loc[df1['Content Rating']=='Adults only 18+'].index
```

In [404...
```python
temp4=df1.loc[df1['Content Rating']=='Unrated'].index
```

In [405...
```python
df1.drop(temp3,inplace=True)
```

In [406...
```python
df1.drop(temp4,inplace=True)
```

In [407...
```python
df1.loc[df1['Content Rating']=='Adults only 18+']
```

Out[407...

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

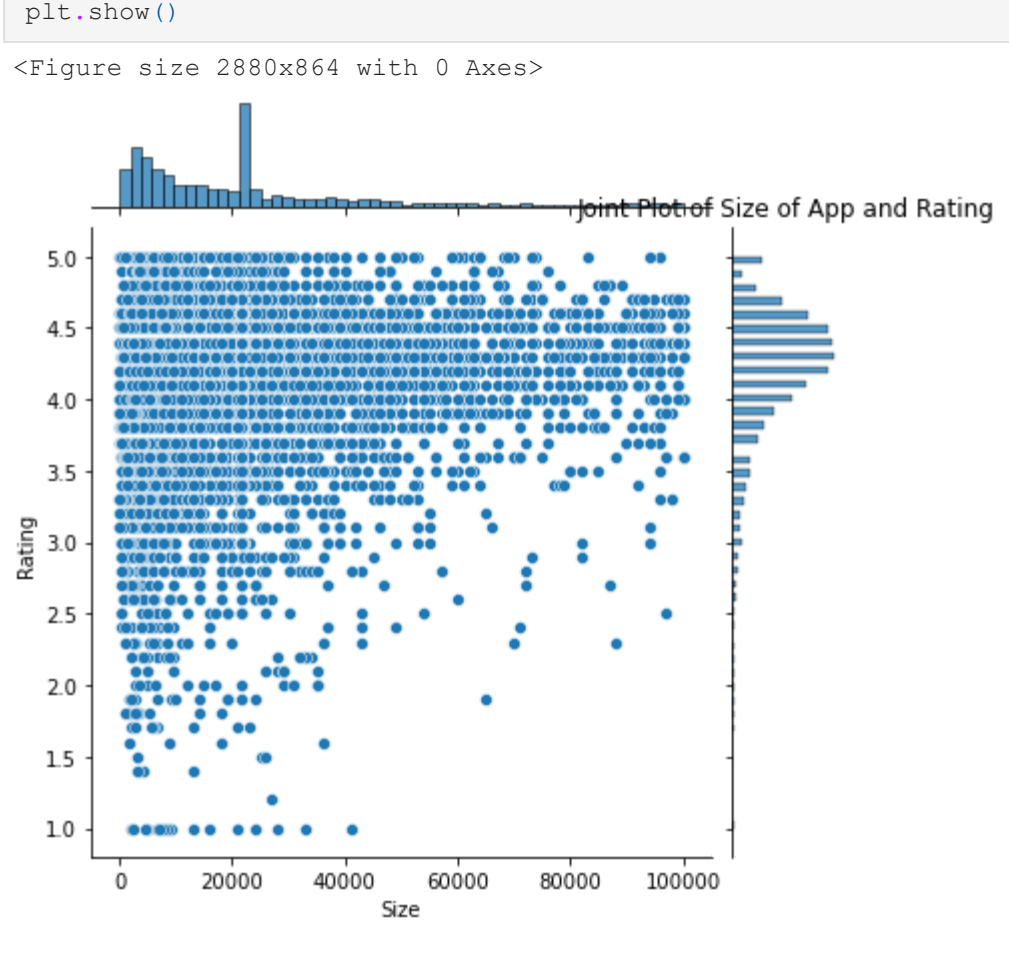
Part 7 Effect of size on rating

- a. Make a joinplot to understand the effect of size on rating

In [408...
```python
plt.figure(figsize=(40,12))
sns.jointplot(x=df1['Size'],y=df1['Rating'],data=df1)
plt.title("Joint Plot of Size of App and Rating")

plt.show()
```

```
<Figure size 2880x864 with 0 Axes>
```



7b. Do you see any patterns?

yes

7c. How do you explain the pattern?

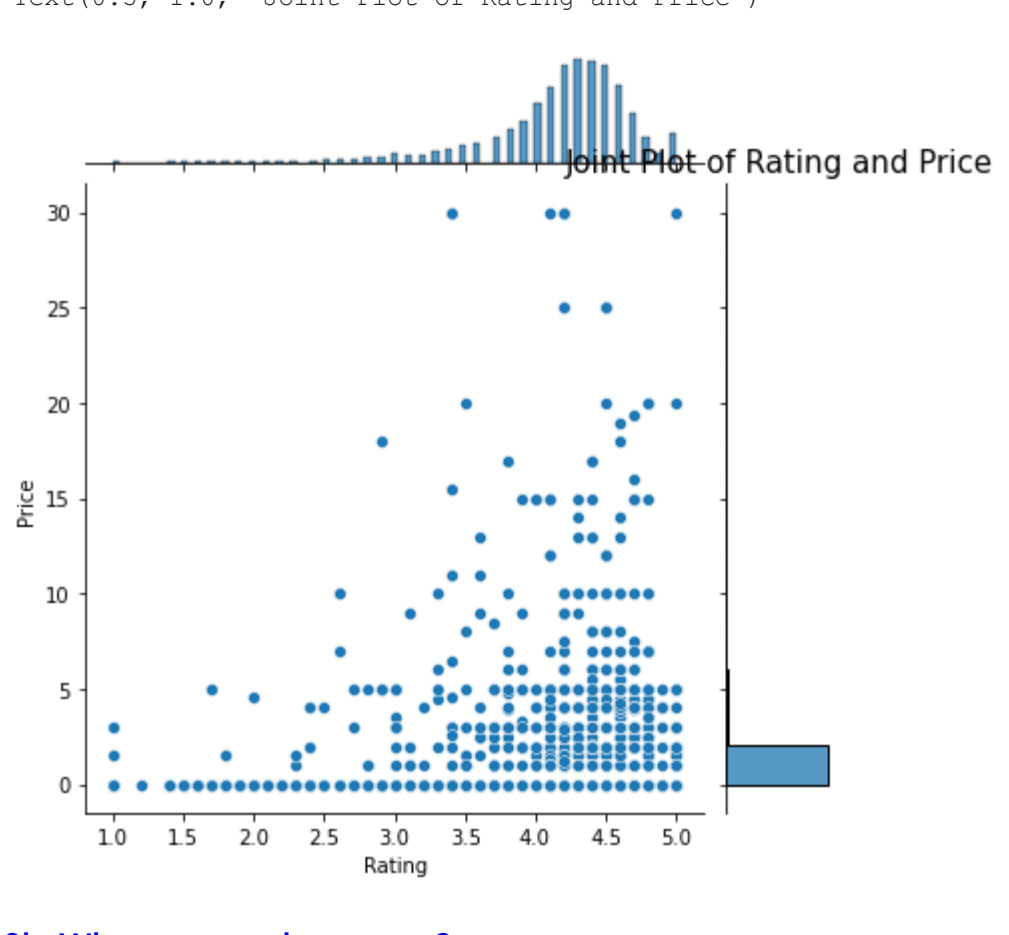
We observed that the maximum data point gather in between Rating 4.0-4.5 and size 0-40000(40 mb). Meaning is that the application size which is in between 20000-40000(40 mb) get good rating on play store

Part 8. Effect of price on rating

8a. Make a jointplot (with regression line)

In [409...
```python
#plt.figure(figsize=(40,12))
sns.jointplot(x ="Rating" , y = "Price" ,data = df1)
plt.title("Joint Plot of Rating and Price", fontsize = 15)
```

Out[409...
```
Text(0.5, 1.0, 'Joint Plot of Rating and Price')
```



8b. What pattern do you see?

Generally on increasing the Price, Rating remains almost constant greater than 4.

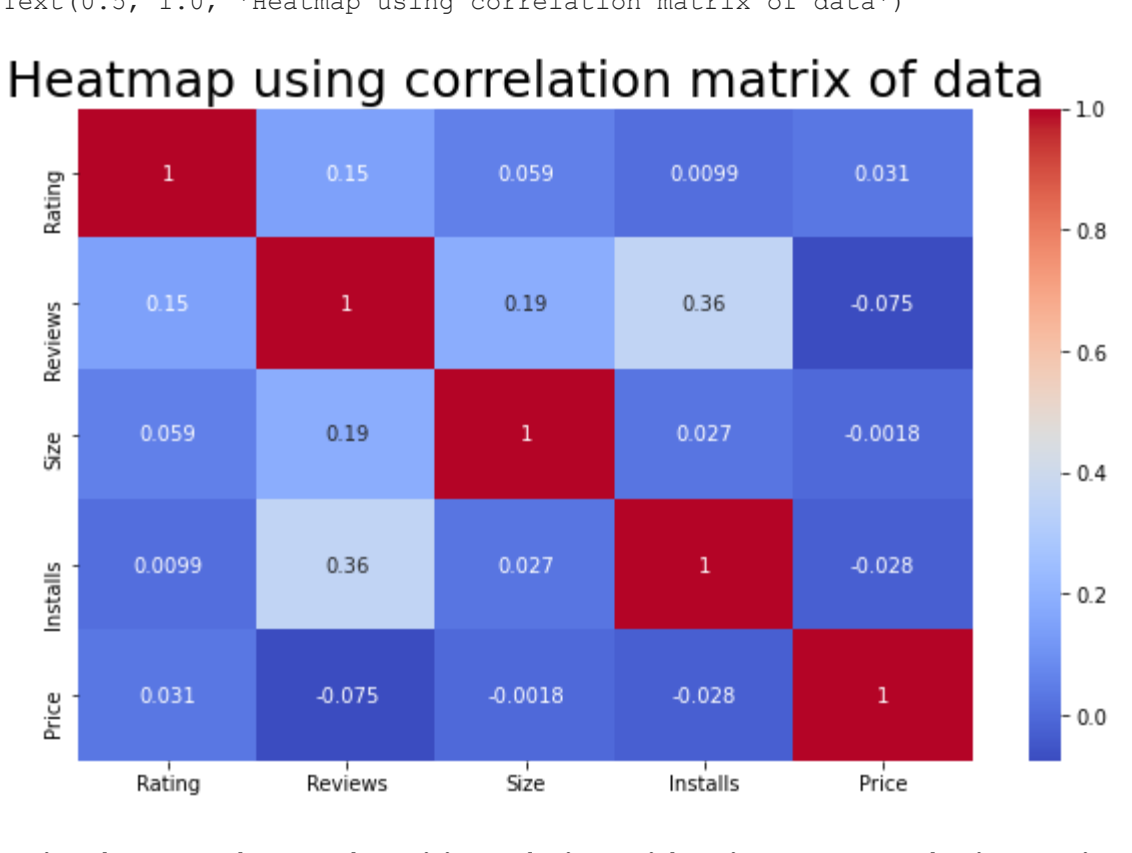
8c. How do you explain the pattern?

In [410...
```python
# Heatmap
plt.figure(figsize=(10,6)) # heatmap size in ratio 16:9

sns.heatmap(df1.corr(), annot = True, cmap ='coolwarm') # show heatmap

plt.title("Heatmap using correlation matrix of data", fontsize = 25) # title of heatmap
```

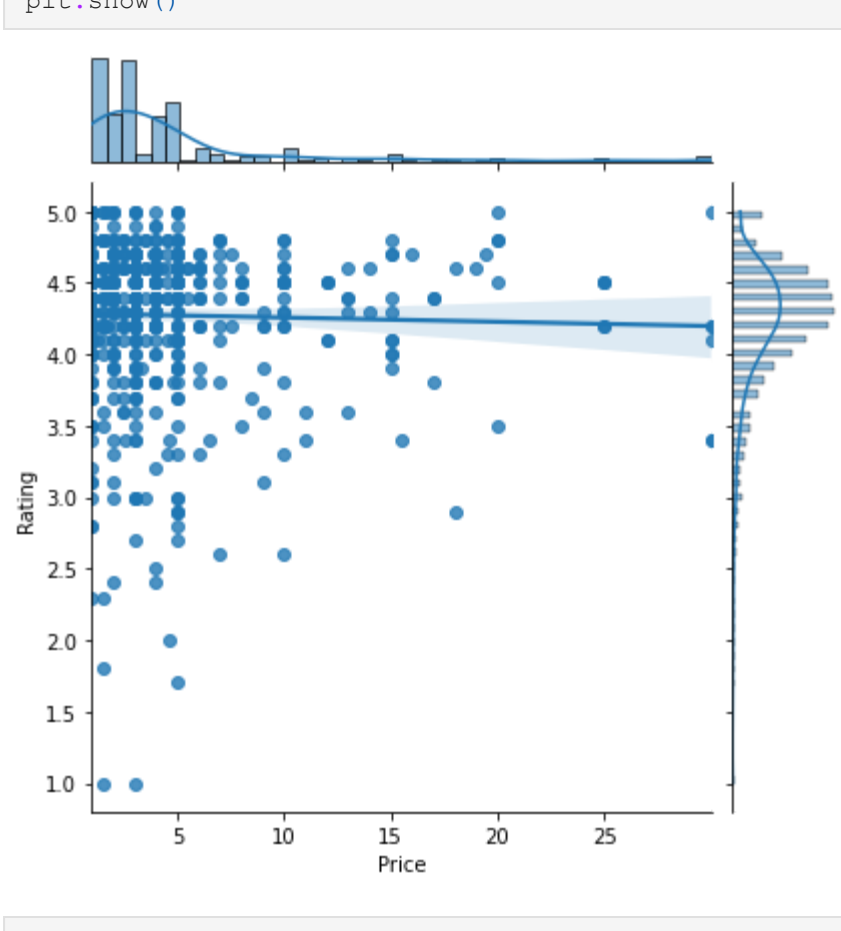
Out[410… Text(0.5, 1.0, 'Heatmap using correlation matrix of data')



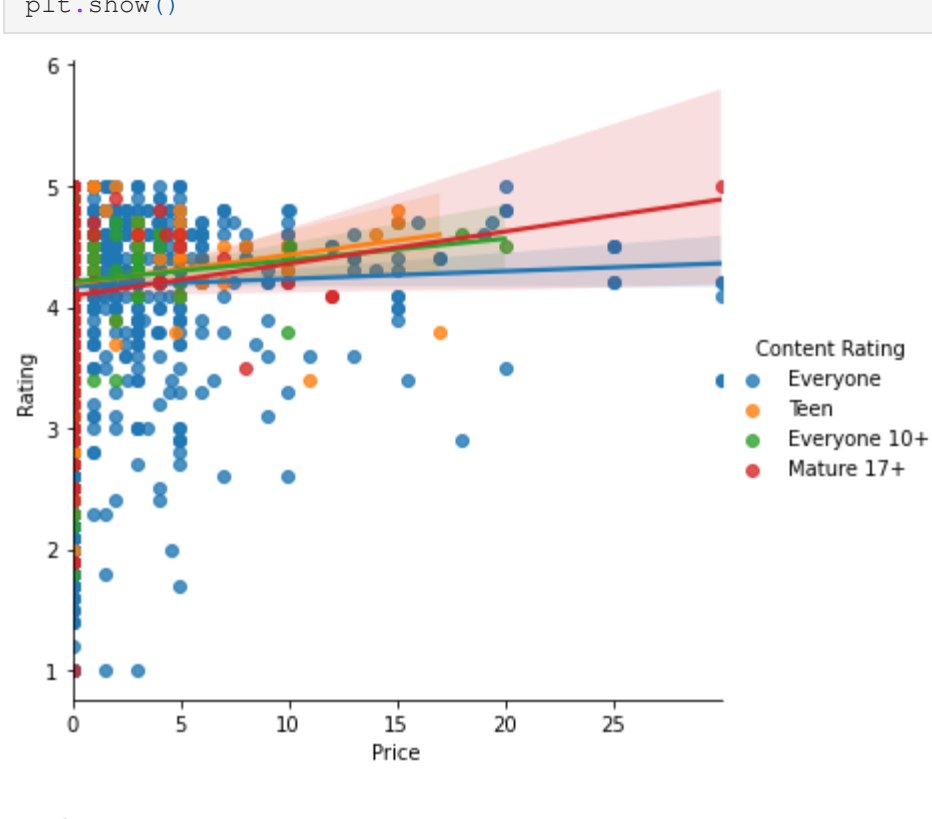
Rating has very less and positive relation with Price ,It means by increasing the price, Rating Increases

8d. Replot the data, this time with only records with price > 0

In [411…
```
Price = df1[df1['Price'] > 0]
sns.jointplot(x ="Price" , y = df1["Rating"] ,data = Price, kind = "reg" )
#plt.title("Joint Plot of Rating and Price", fontsize = 15)
plt.show()
```



In [412…
```
sns.lmplot(x='Price', y='Rating', hue ='Content Rating', data=df1)
plt.show()
```

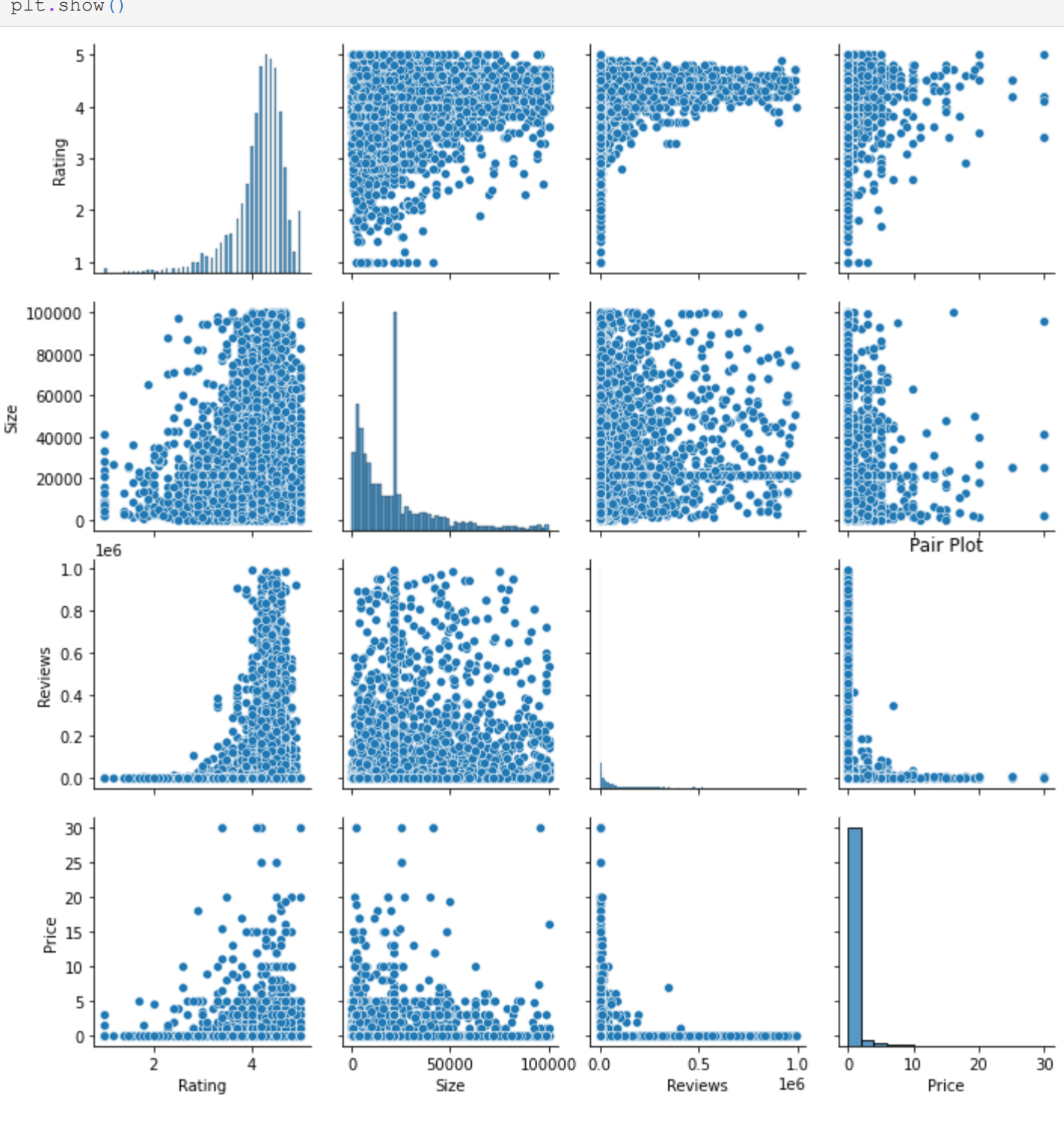


we found that Mature and Teen are more co-related in Content rating

9. Look at all the numeric interactions together

* a. Make a pairplort with the colulmns - 'Reviews', 'Size', 'Rating', 'Price</center>

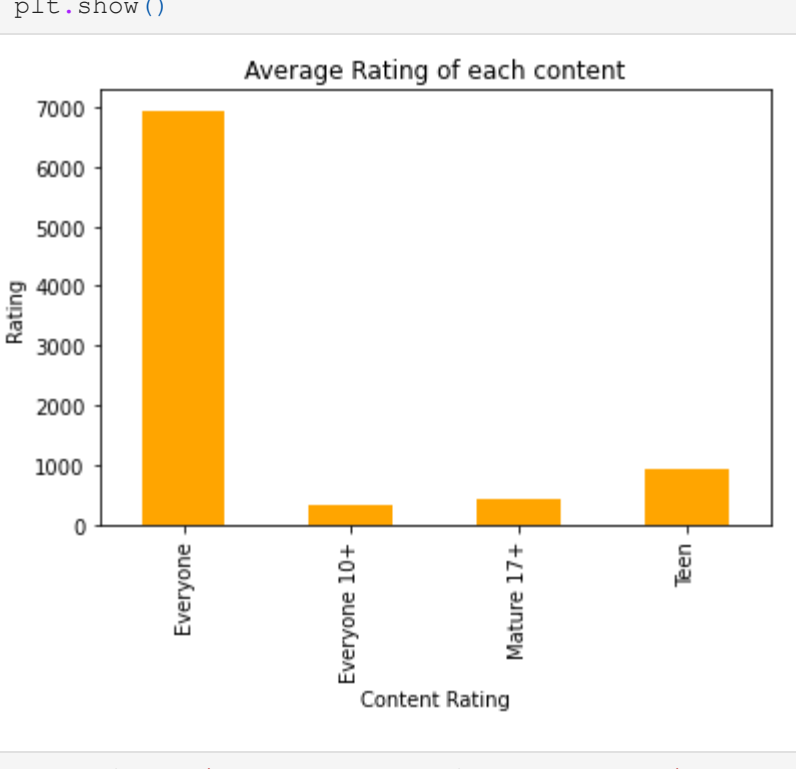
In [413…
```
sns.pairplot(df1,vars=['Rating','Size', 'Reviews', 'Price'])
plt.title("Pair Plot")
plt.show()
```



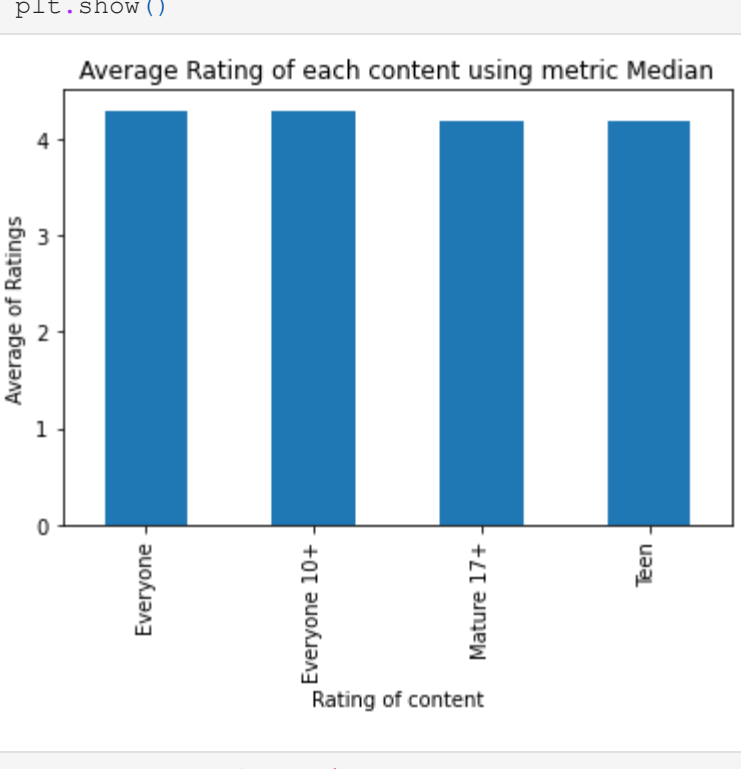
Part 10. Rating vs. content rating

- a. Make a bar plot displaying the rating for each content rating
- b. Which metric would you use? Mean? Median? Some other quantile?
- c. Choose the right metric and plot

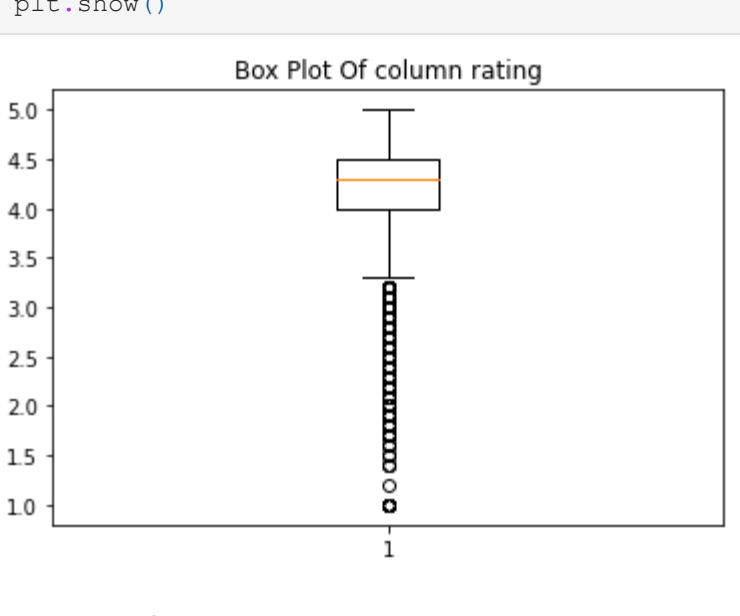
In [417…
```
df1.groupby(['Content Rating'])['Rating'].count().plot.bar(color="orange")
plt.ylabel('Rating')
plt.title("Average Rating of each content")
plt.show()
```



In [421…
```
a = df1['Rating'].groupby(df1['Content Rating']).median().plot(kind = 'bar')
plt.xlabel('Rating of content')
plt.ylabel('Average of Ratings')
plt.title("Average Rating of each content using metric Median")
plt.show()
```

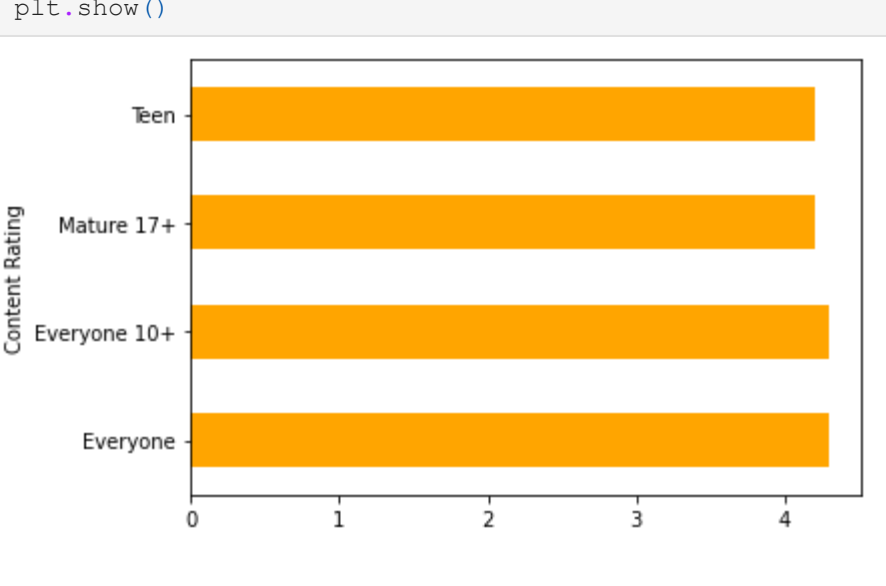


In [422…
```
plt.boxplot(df1['Rating'])
plt.title("Box Plot Of column rating")
plt.show()
```



Because of Outliers , median is the best measure of central tendency.

In [423…
```
df1.groupby(['Content Rating'])['Rating'].median().plot.barh(color="orange")
plt.show()
```

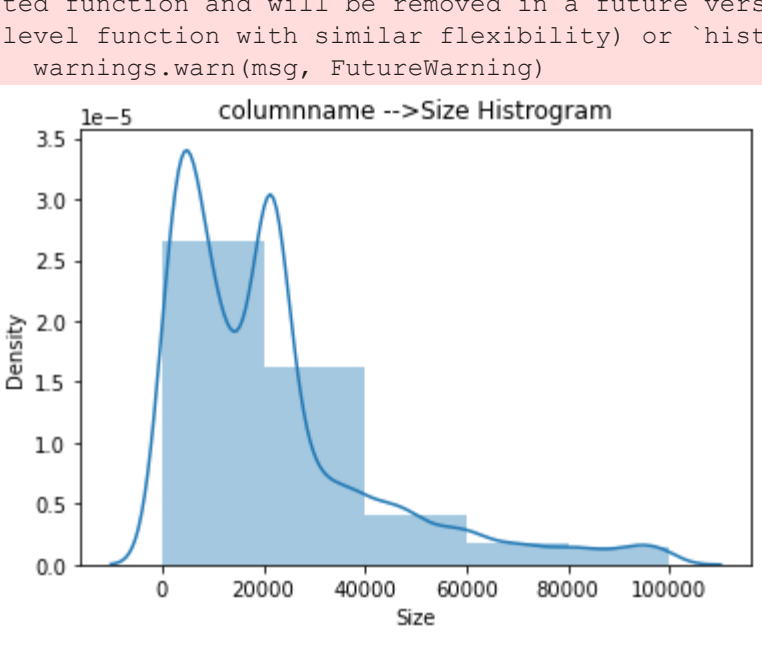


Part 11. Content rating vs. size vs. rating – 3 variables at a time

11.a)Create 5 buckets (20% records in each) based on Size

In [424…
```
sns.distplot(df1["Size"], bins=5)
plt.title("columnname -->Size Histrogram")
plt.show()
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



11.b). By Content Rating vs. Size buckets, get the rating (20th percentile) for each combination

In [425…
```
bins=[0, 20000, 40000, 60000, 80000, 100000]
df1['Bucket Size'] = pd.cut(df1['Size'], bins, labels=['0-20k','20k-40k','40k-60k','60k-80k','80k-100k'])
pd.pivot_table(df1, values='Rating', index='Bucket Size', columns='Content Rating')
```

Out[425…

Content Rating	Everyone	Everyone 10+	Mature 17+	Teen
Bucket Size				
0-20k	4.146001	4.247561	4.010582	4.182275
20k-40k	4.203324	4.166667	4.156291	4.173163
40k-60k	4.171457	4.265217	4.190476	4.242982
60k-80k	4.244221	4.280769	4.200000	4.273016
80k-100k	4.260127	4.304762	4.252632	4.269231

In [426…
```
df1.Size.quantile([0.2, 0.4,0.6,0.8])
```

Out[426…
```
0.2     4700.0
0.4    12000.0
0.6    21516.0
0.8    32000.0
Name: Size, dtype: float64
```

11.c). Make a heatmap of this

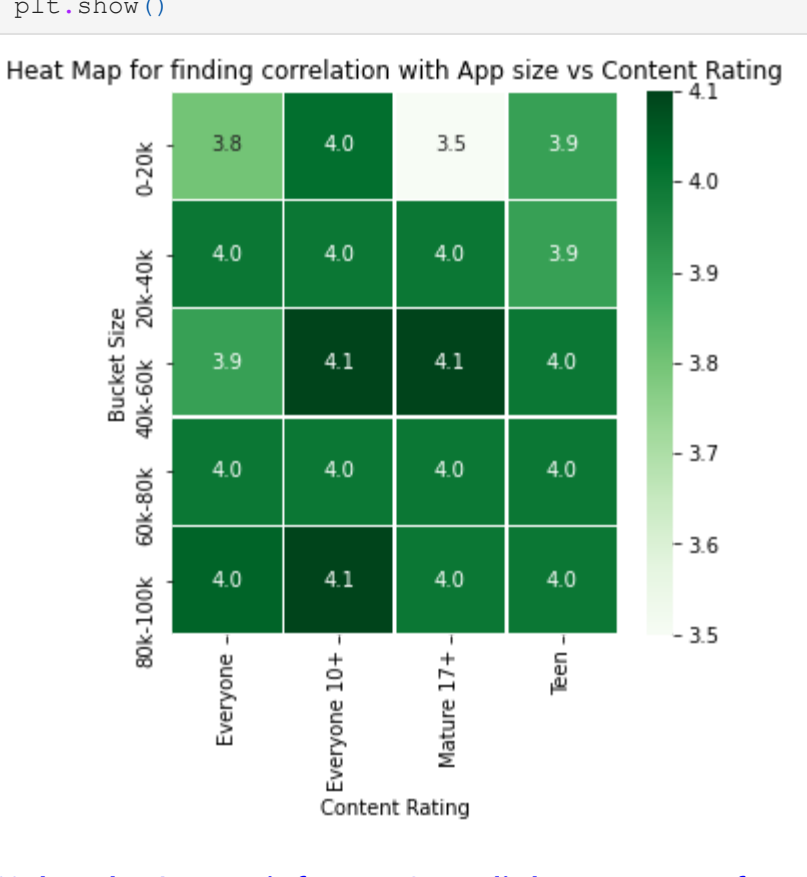
- i. Annotated
- ii. Greens color map

In [427…
```
temp3=pd.pivot_table(df1, values='Rating', index='Bucket Size', columns='Content Rating', aggfunc=lambda x:np.c
temp3
```

Out[427…

Content Rating	Everyone	Everyone 10+	Mature 17+	Teen
Bucket Size				
0-20k	3.80	4.02	3.5	3.9
20k-40k	4.00	4.00	4.0	3.9
40k-60k	3.90	4.10	4.1	4.0
60k-80k	4.00	4.00	4.0	4.0
80k-100k	4.04	4.10	4.0	4.0

In [428…
```
f,ax = plt.subplots(figsize=(5, 5))
sns.heatmap(temp3, annot=True, linewidths=.5, cmap='Greens',fmt='.1f',ax=ax)
plt.title("Heat Map for finding correlation with App size vs Content Rating")
plt.show()
```



11.d). What's your inference? Are lighter apps preferred in all categories? Heavier? Some

Based on above visualization , We can conclude that ,No Its not true that lighter apps are preferred in all categories.

Because apps with size 40k-60k and 80k-100k have got the highest rating in all cateegories.

So, in general we can conclude that heavier apps are preferred in all categories.

Thank You

In [ ]: