s

Project Report

on

Google Play Store Apps Analysis

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

in partial fulfilment of the requirements for the award of degree of

Master of Computer Applications

Submitted By

Name of Student: Pranav Mishra

Registration No. 12114762

Section: D2109

Supervised By

Name of Faculty: Mr. Kumar Vishal

Designation: Assistant. Professor

LOVELY FACULTY OF TECHNOLOGY & SCIENCES

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB

Nov-2022

Table of Content

Introduction and Objective

Introduction:

Google Play Store or formerly Android Market, is a digital distribution service developed and operated by Google. It is an official apps store that provides variety content such as apps, books, magazines, music, movies, and television programs. It serves an as platform to allow users with 'Google certified' Android operating system devices to download applications developed and published on the platform either with a charge or free of cost. With the rapidly growth of Android devices and apps, it would be interesting to perform data analysis on the data to obtain valuable insights.

The dataset that is going to be used is 'Google Play Store Apps' from Kaggle. It contains 10k of web scraped Play Store apps data for analysing the Android market.

Each app (row) has values for category, rating, reviews, size, installs, price, rated, last updated, and version.

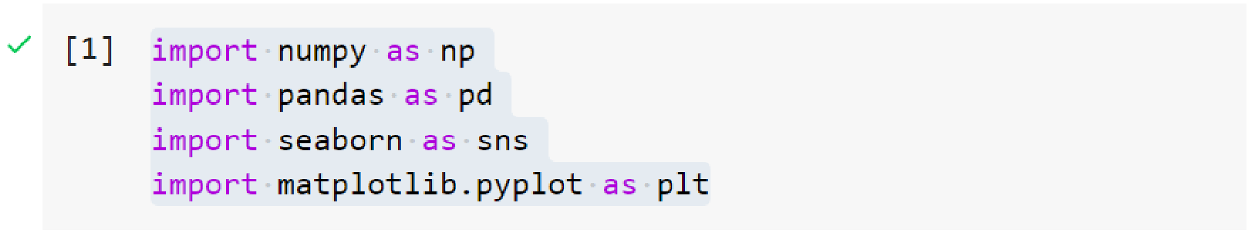
Objective:

1. Using the data to analyse consumer trends and determine which type of apps are the most popular and profitable.
2. Classifying applications based on their categories.
3. Presenting the growth of applications from 2016 to 2018.
4. Comparing different categories of applications based on the Android version.
5. Comparing the rates in different kinds of applications.
6. Assessing supported Android version with numbers of reviews based on different categories.

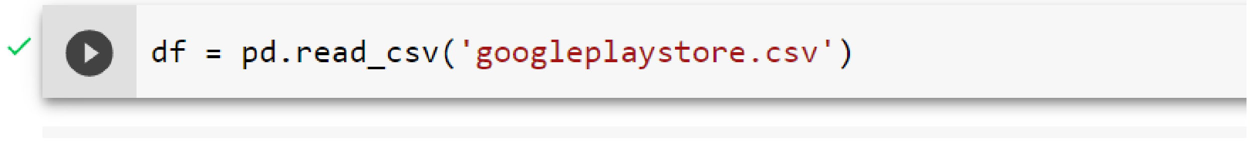
Screen Shots with coding:

Imports: Let us start by importing some of the required libraries with which we will be working on.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

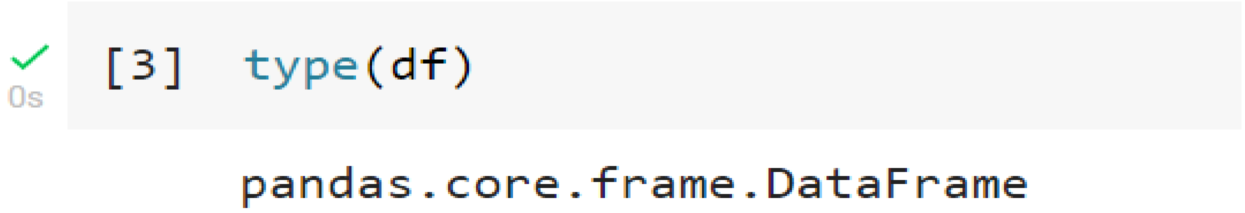


```
[1] import numpy as np
    import pandas as pd
    import seaborn as sns
    import matplotlib.pyplot as plt
```



```
df = pd.read_csv('googleplaystore.csv')
```

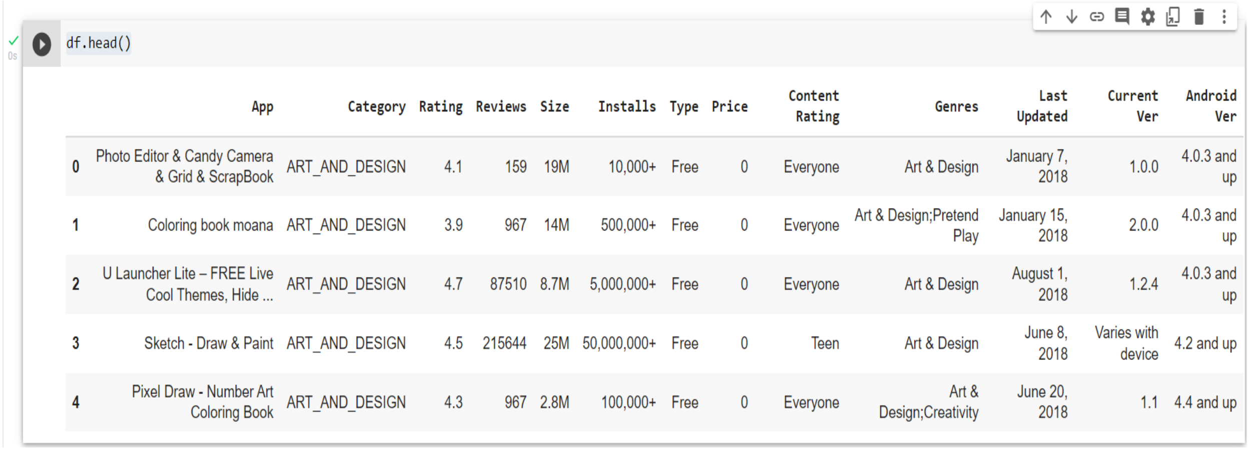
```
type(df)
```



```
[3] type(df)
0s

pandas.core.frame.DataFrame
```

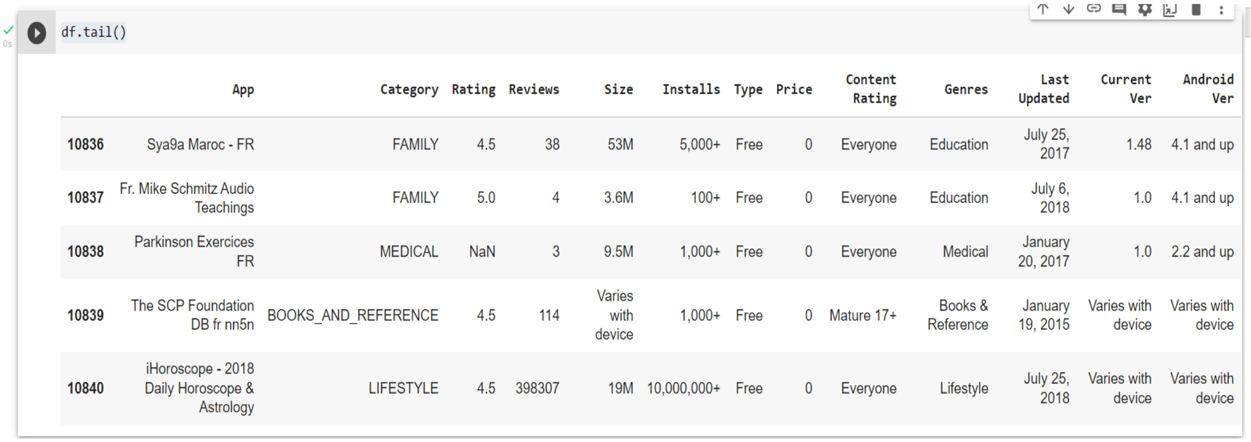
`df.head()` – The head() method returns a specified number of rows, string from the top.



```
df.head()
```

	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

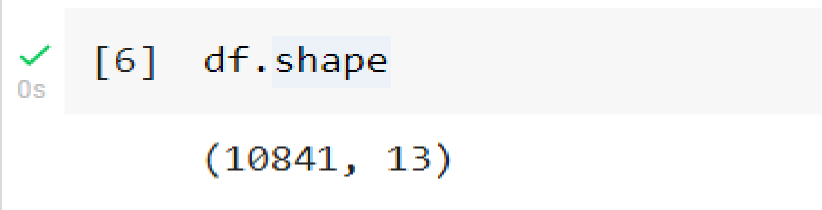
`df.tail()` – The tail() method returns a specified number of last rows.



```
df.tail()
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	July 25, 2017	1.48	4.1 and up
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	July 6, 2018	1.0	4.1 and up
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	Lifestyle	July 25, 2018	Varies with device	Varies with device

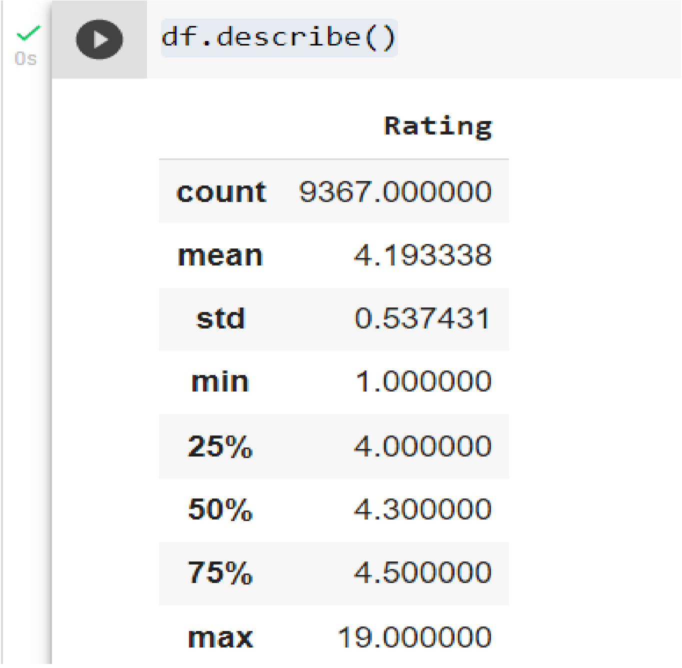
`df.shape` – The shape is the number of rows and columns of the DataFrame.



```
[6]  df.shape

     (10841, 13)
```

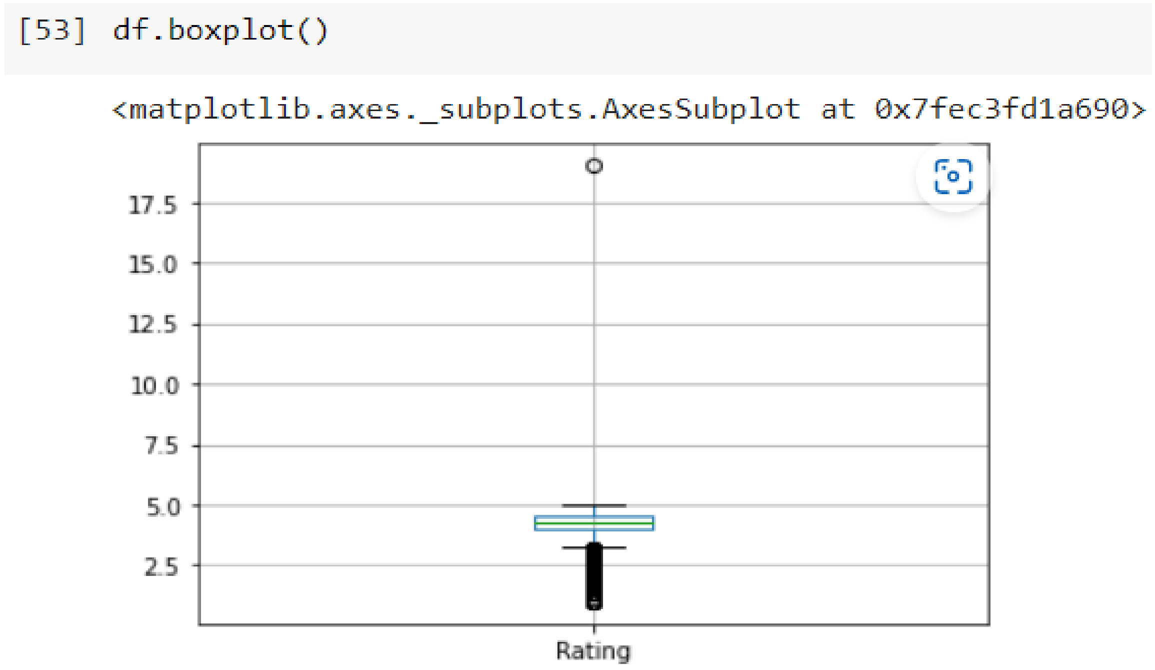
`df.describe()` – The describe() method returns description of the data in the DataFrame.



```
df.describe()
```

	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

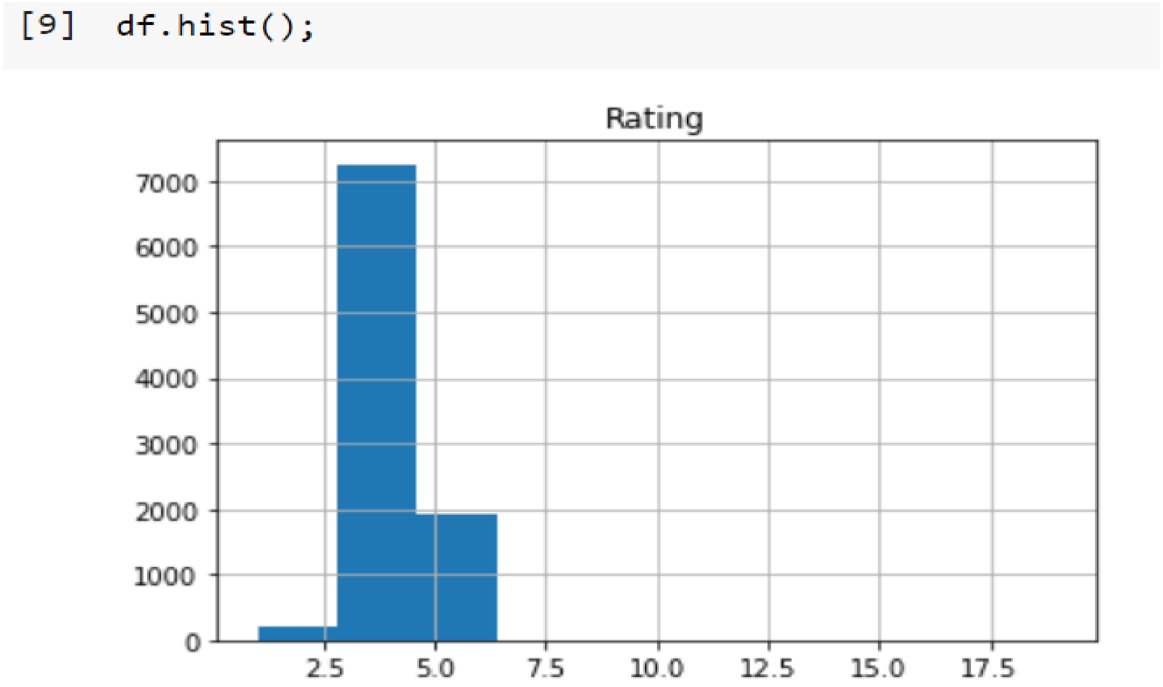
`df.boxplot()`- Box plot is also called a Whisker plot which provides a summary of a set of data that includes minimum, first-quartile, median, third quartile, and maximum value.



```
[53] df.boxplot()

     <matplotlib.axes._subplots.AxesSubplot at 0x7fec3fd1a690>
```

`df.hist()`- The hist () function is defined as a quick way to understand the distribution of certain numerical variables from the dataset**.**

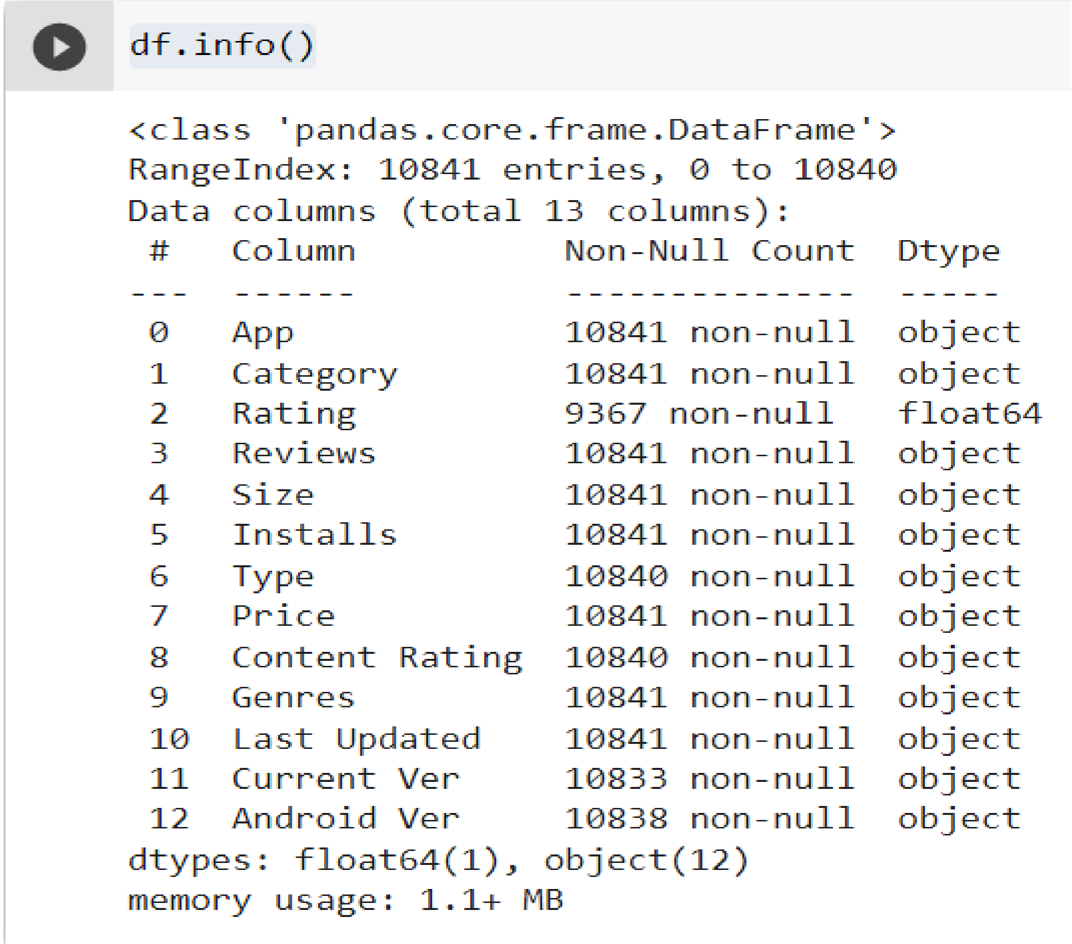


```
[9]  df.hist();
```

Data Cleaning:

Data cleansing or data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

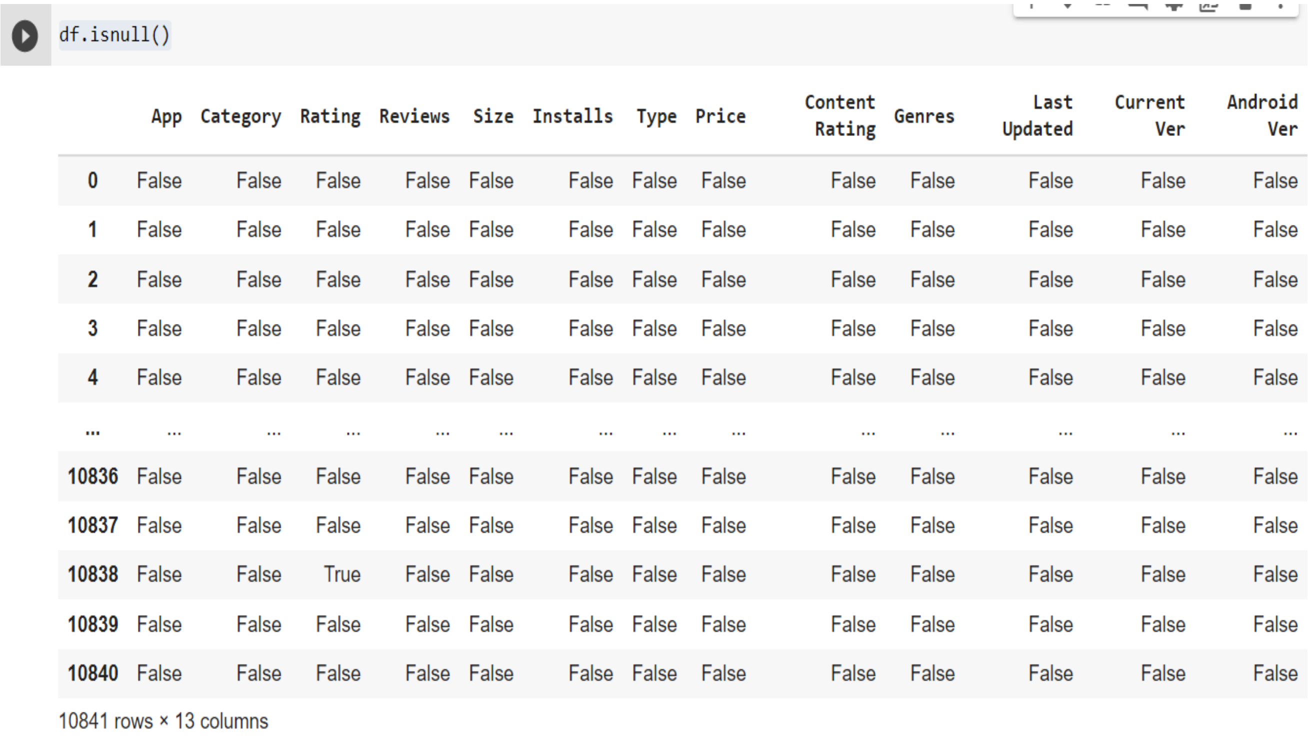
df.info() - Use info() function to print full summary of the data frame.



```
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  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
```

`df.isnull()` – Pandas isnull() function detect missing values in the given object. It returns a Boolean same-sized object indicating if the values are NA. Missing values gets mapped to True and non-missing value gets mapped to False.

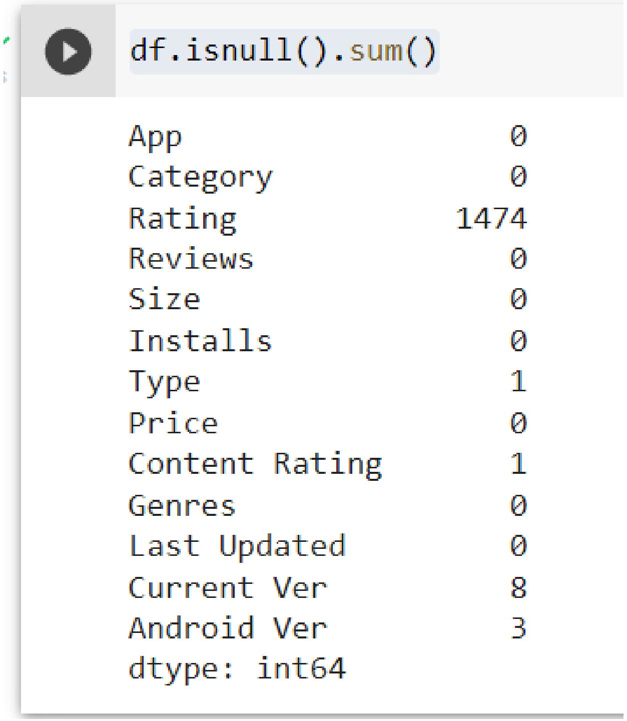


```
df.isnull()
```

	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

`df.isnull().sum()` – It gives you pandas series of column names along with the sum of missing values in each column.

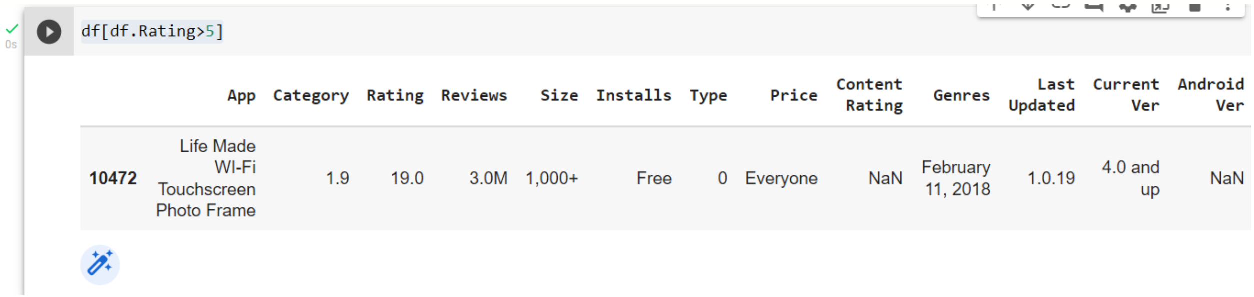


```
df.isnull().sum()
```

```
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
```

Checking how many outliers are there:

df[df.Rating>5]

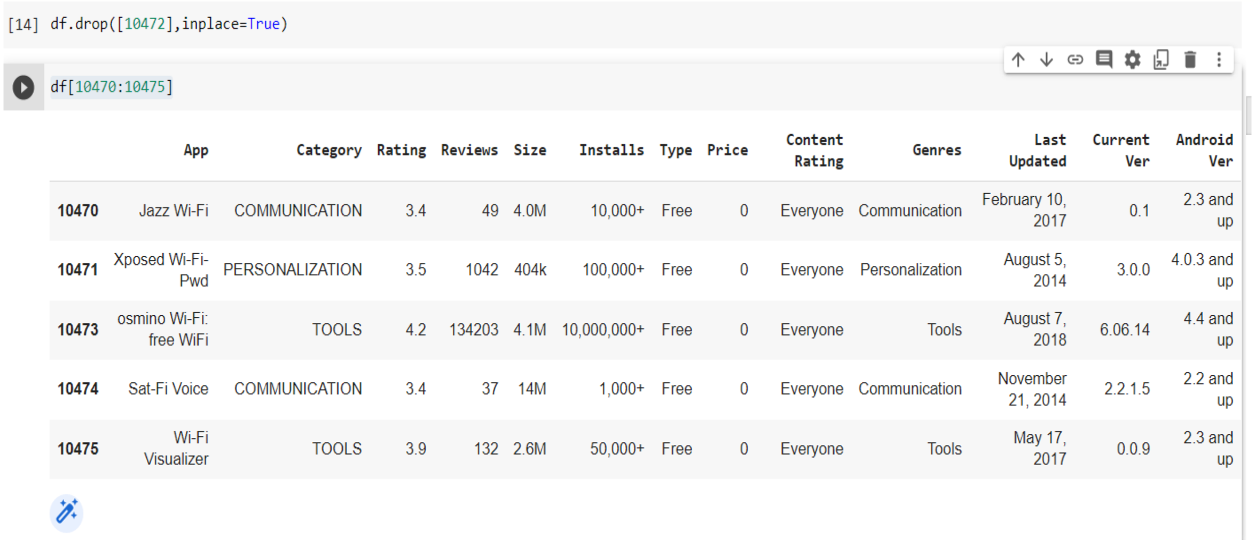


```
df[df.Rating>5]
```

	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

df.drop([10472],inplace=True) - The drop() method removes the specified row or column.

df[10470:10475] – Check the drop row and column.

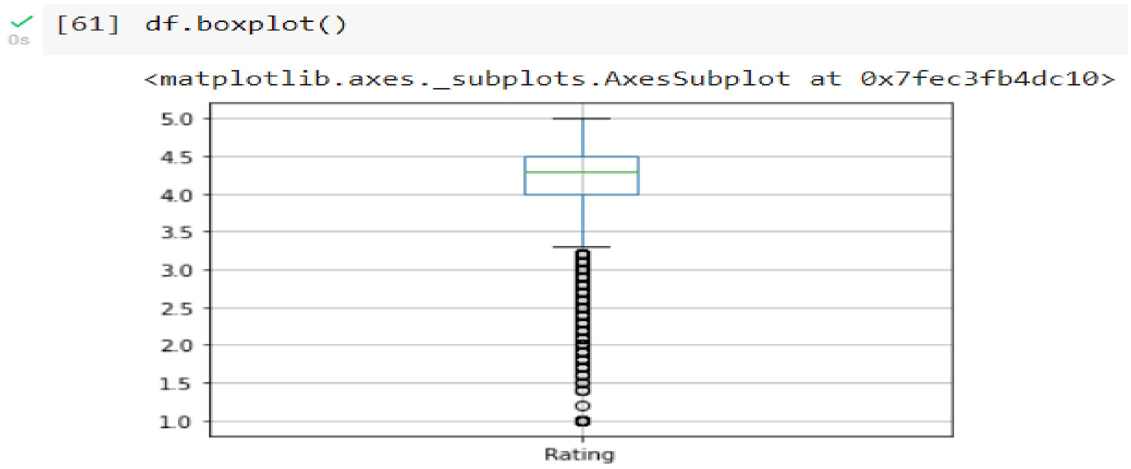


```
[14] df.drop([10472],inplace=True)
```

```
df[10470:10475]
```

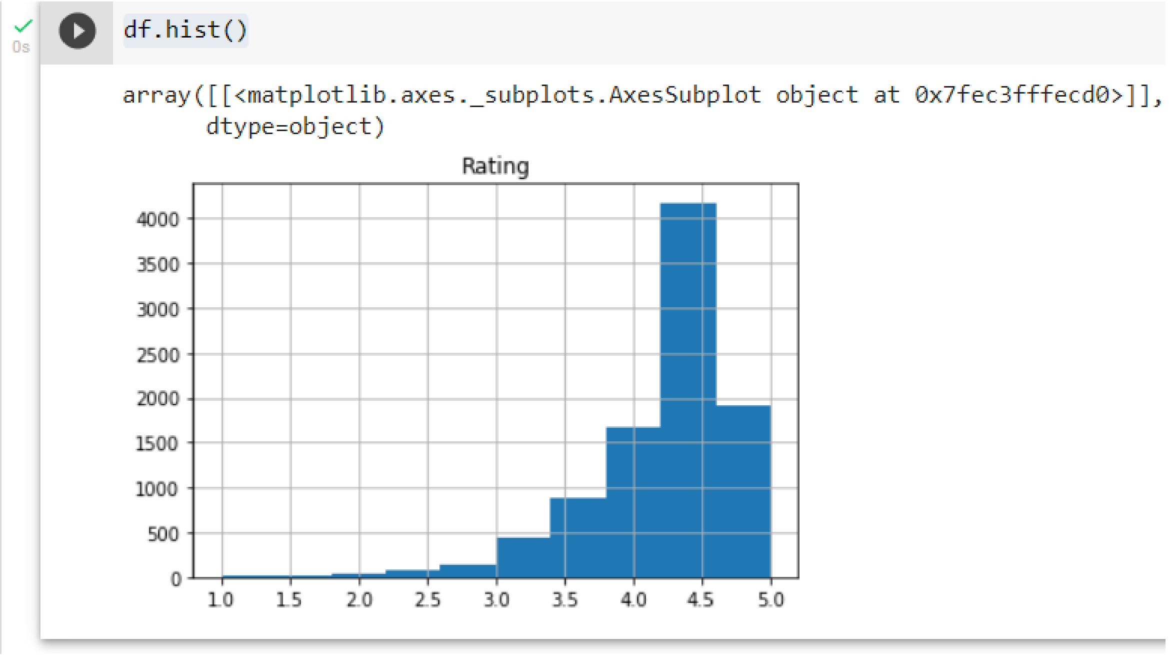
	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10470	Jazz Wi-Fi	COMMUNICATION	3.4	49	4.0M	10,000+	Free	0	Everyone	Communication	February 10, 2017	0.1	2.3 and up
10471	Xposed Wi-Fi-Pwd	PERSONALIZATION	3.5	1042	404k	100,000+	Free	0	Everyone	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

df.boxplot();



```
[61] df.boxplot()
<matplotlib.axes._subplots.AxesSubplot at 0x7fec3fb4dc10>
```

```
df.hist()
```



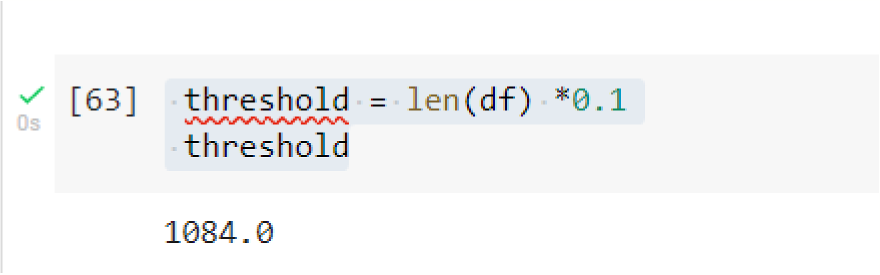
```
df.hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fec3fffecd0>]],
      dtype=object)
```

Remove column that are 90% empty:

threshold = len(df) *0.1
threshold

A threshold value is passed as parameter and all values in series that are more than the threshold values become equal to it.

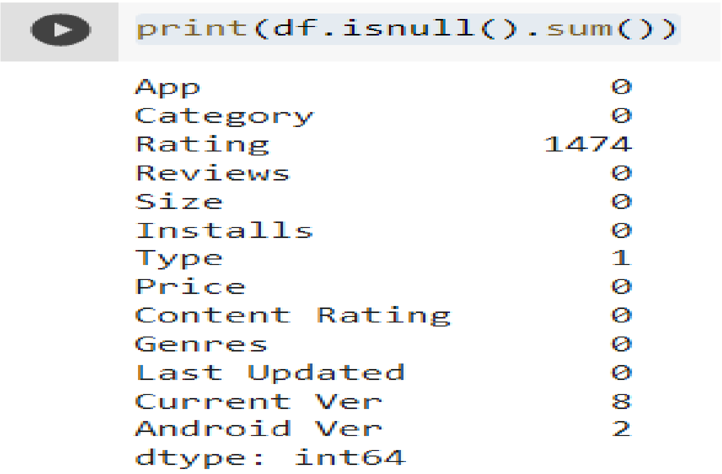


```
[63] threshold = len(df) *0.1
     threshold
```

```
1084.0
```

```
df.dropna(thresh = threshold,axis=1,inplace=True)
print(df.isnull().sum())
```

Pandas *dropna()* method allows the user to analyse and drop Rows/Columns with Null values in different ways.



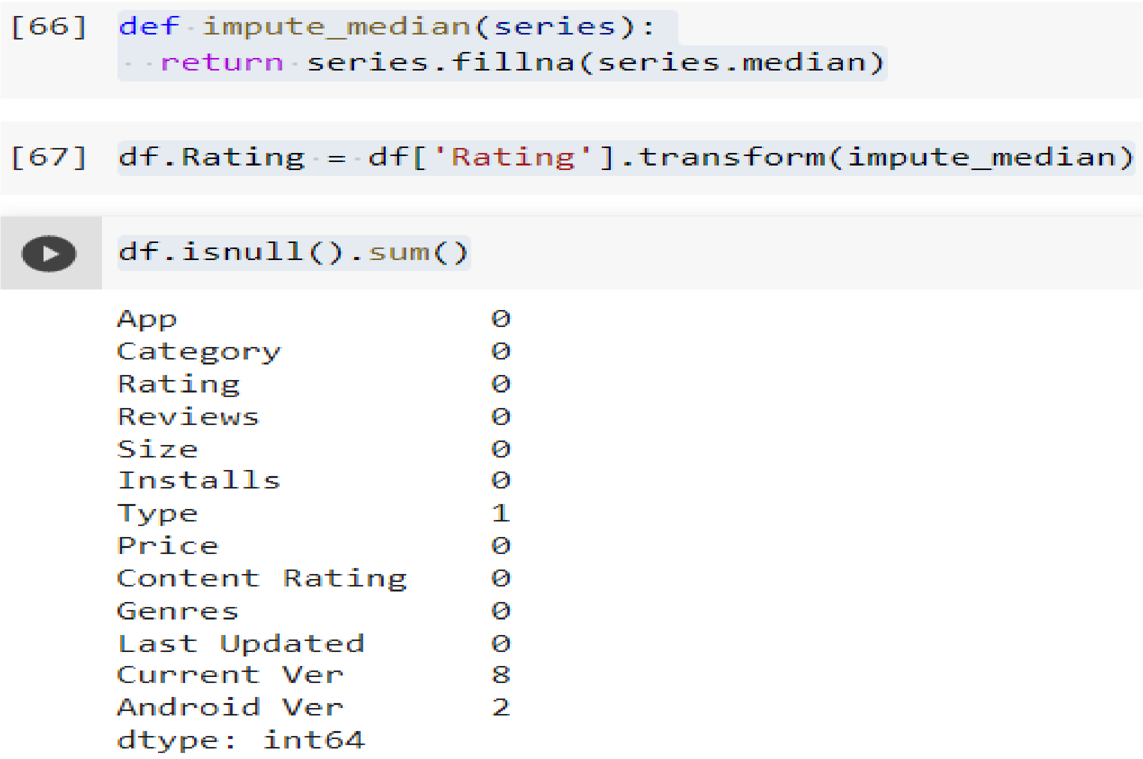
```
print(df.isnull().sum())
```

```
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
```

Data Manipulation:

```
def impute_median(series):
  return series.fillna(series.median)
df.Rating = df['Rating'].transform(impute_median)

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



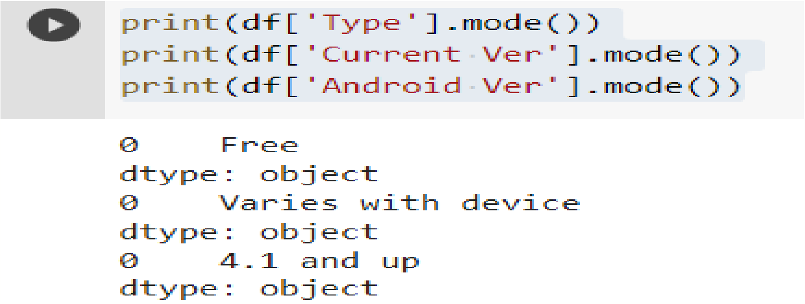
```
[66] def impute_median(series):
       return series.fillna(series.median)
```

```
[67] df.Rating = df['Rating'].transform(impute_median)
```

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

```
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
```

```
print(df['Type'].mode())
print(df['Current Ver'].mode())
print(df['Android Ver'].mode())
```

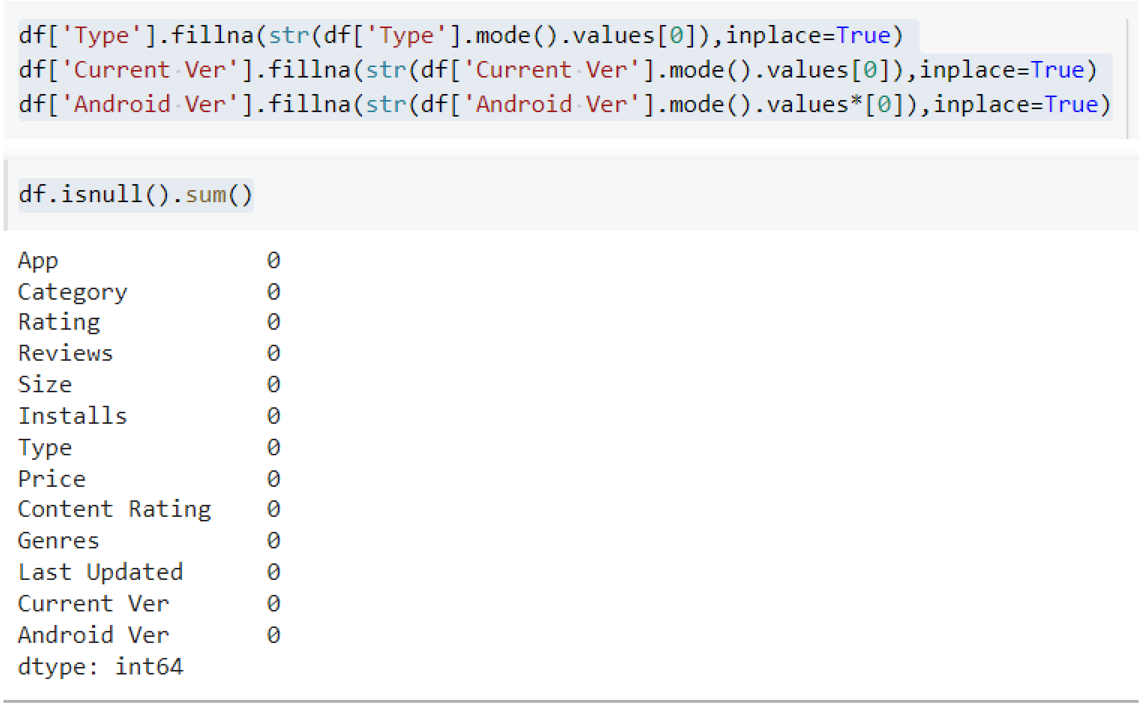


```
print(df['Type'].mode())
print(df['Current Ver'].mode())
print(df['Android Ver'].mode())
```

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

```
df['Type'].fillna(str(df['Type'].mode().values[0]),inplace=True)
df['Current Ver'].fillna(str(df['Current Ver'].mode().values[0]),inplace=True)
df['Android Ver'].fillna(str(df['Android Ver'].mode().values*[0]),inplace=True)

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



```
df['Type'].fillna(str(df['Type'].mode().values[0]),inplace=True)
df['Current Ver'].fillna(str(df['Current Ver'].mode().values[0]),inplace=True)
df['Android Ver'].fillna(str(df['Android Ver'].mode().values*[0]),inplace=True)
```

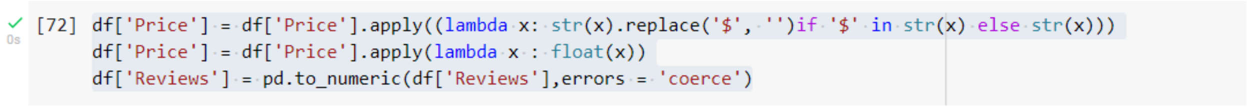
```
df.isnull().sum()
```

```
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
```

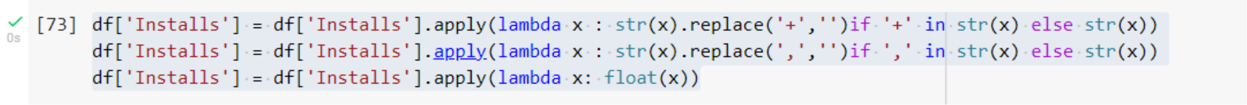
```
df['Price'] = df['Price'].apply((lambda x: str(x).replace('$', '')if '$' in str(x) else str(x)))
df['Price'] = df['Price'].apply(lambda x : float(x))
df['Reviews'] = pd.to_numeric(df['Reviews'],errors = 'coerce')


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

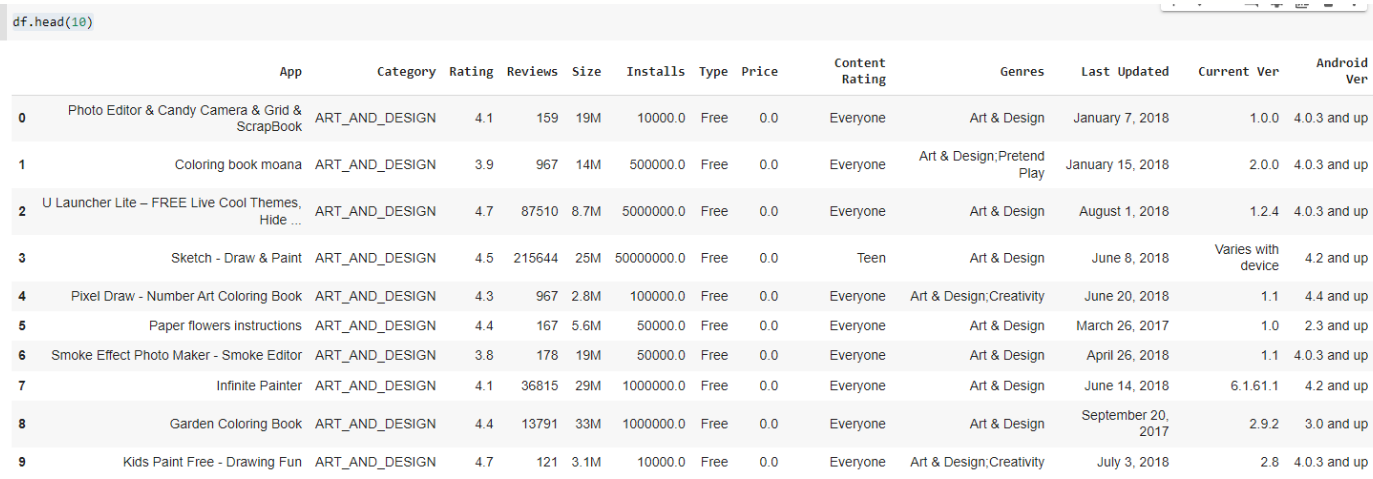
```
df.head(10)
```



```
[72] df['Price'] = df['Price'].apply((lambda x: str(x).replace('$', '')if '$' in str(x) else str(x)))
     df['Price'] = df['Price'].apply(lambda x : float(x))
     df['Reviews'] = pd.to_numeric(df['Reviews'],errors = 'coerce')
```



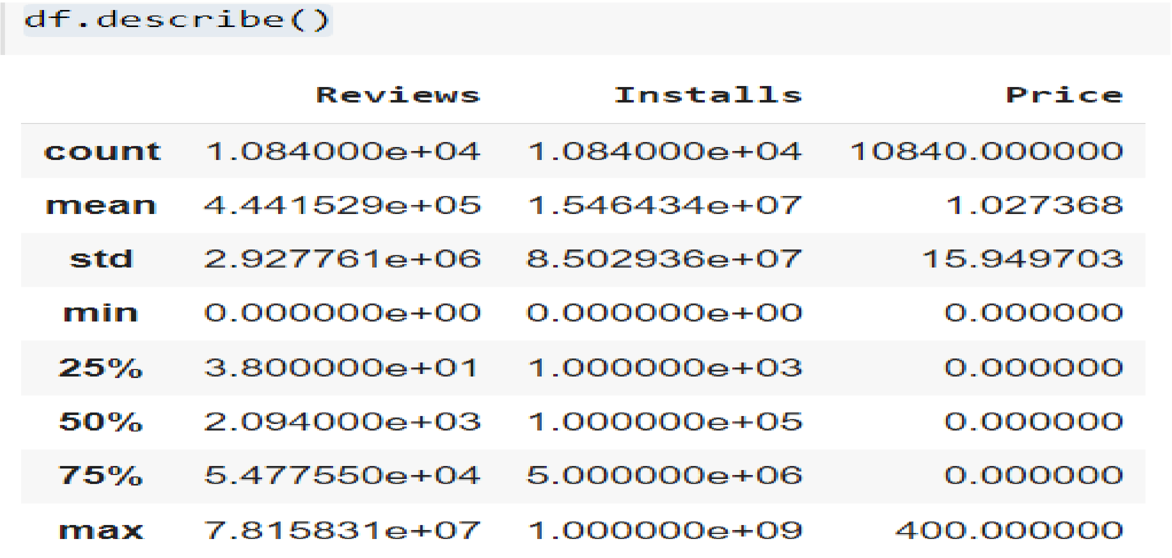
```
[73] df['Installs'] = df['Installs'].apply(lambda x : str(x).replace('+','')if '+' in str(x) else str(x))
     df['Installs'] = df['Installs'].apply(lambda x : str(x).replace(',','')if ',' in str(x) else str(x))
     df['Installs'] = df['Installs'].apply(lambda x: float(x))
```



```
df.head(10)
```

	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	10000.0	Free	0.0	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	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	1.0	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

```
df.describe()
```



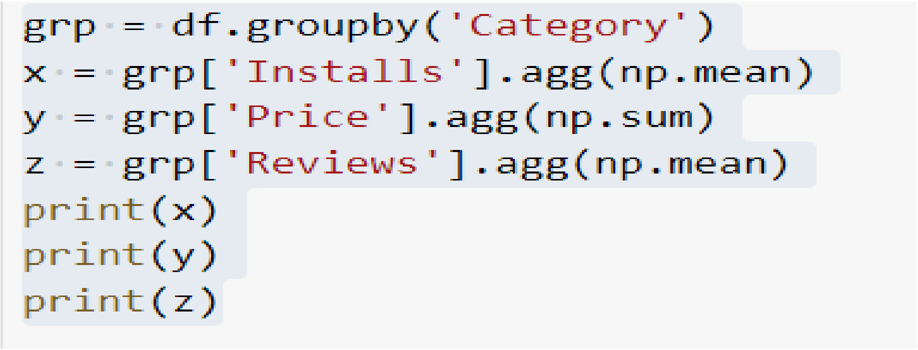
```
df.describe()
```

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

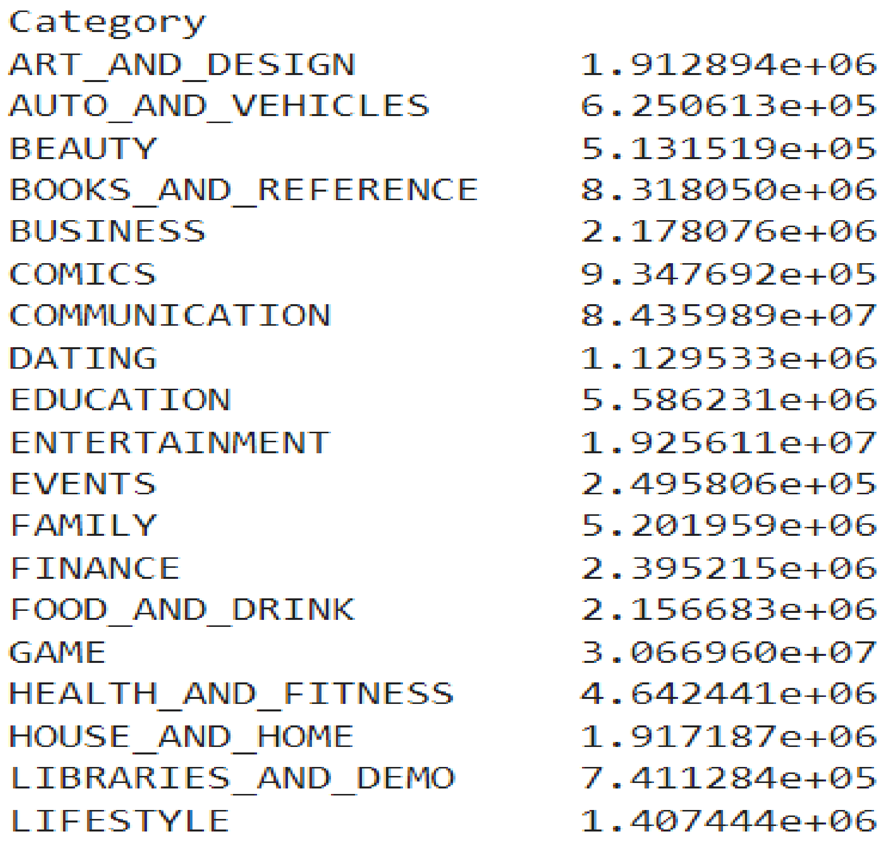
Data Visualization: Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations.

```
grp = df.groupby('Category')
x = grp['Installs'].agg(np.mean)
y = grp['Price'].agg(np.sum)
z = grp['Reviews'].agg(np.mean)
print(x)
print(y)
print(z)
```

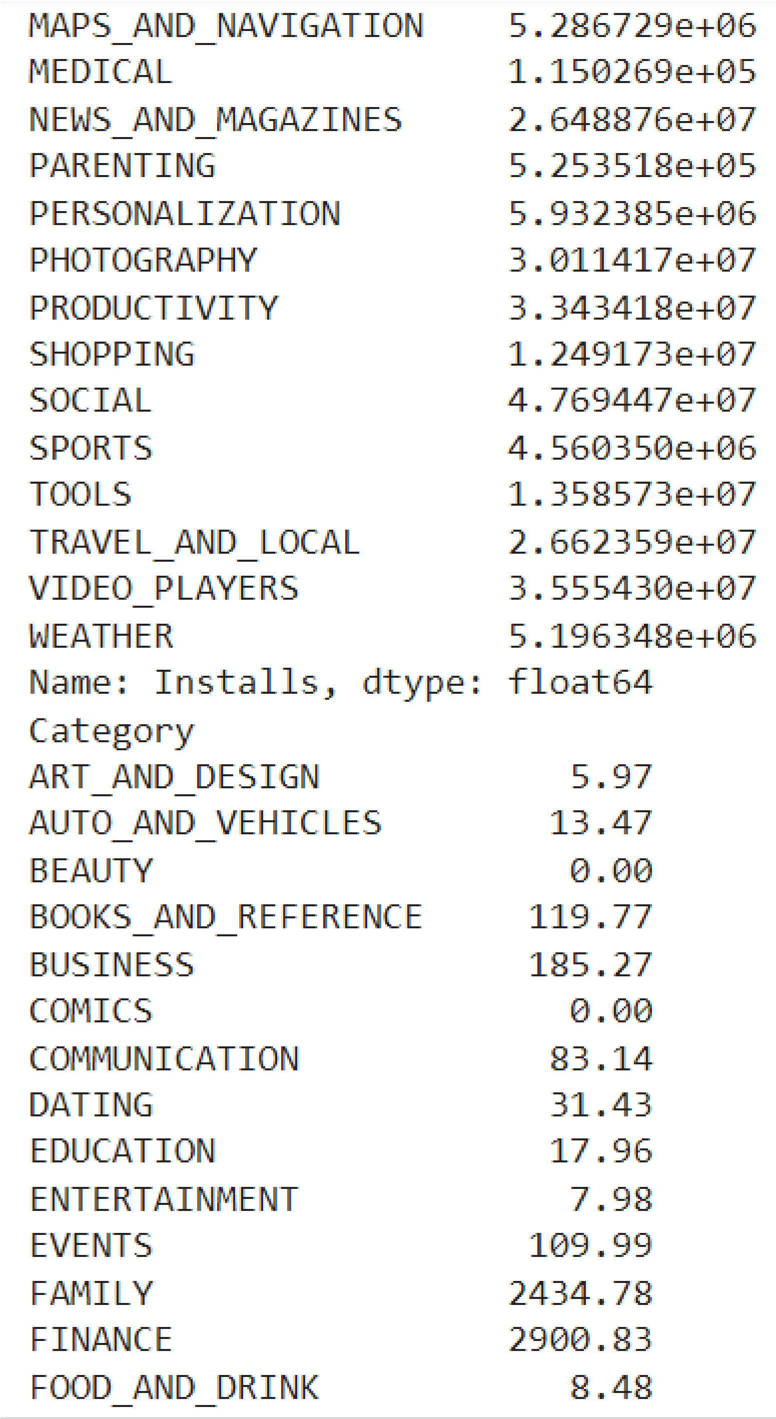
Pandas groupby is used for grouping the data according to the categories and apply a function to the categories.



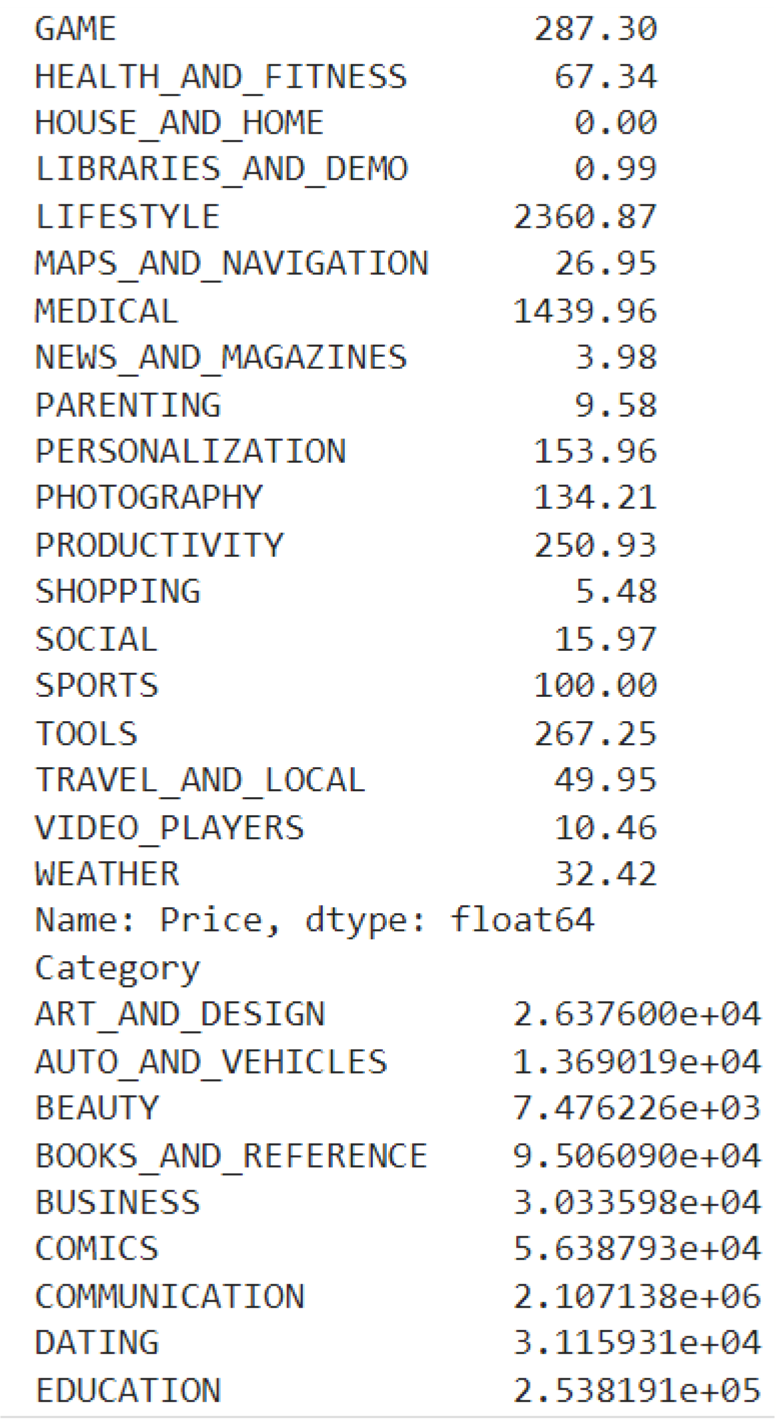
```
grp = df.groupby('Category')
x = grp['Installs'].agg(np.mean)
y = grp['Price'].agg(np.sum)
z = grp['Reviews'].agg(np.mean)
print(x)
print(y)
print(z)
```



```
Category
ART_AND_DESIGN           1.912894e+06
AUTO_AND_VEHICLES        6.250613e+05
BEAUTY                   5.131519e+05
BOOKS_AND_REFERENCE      8.318050e+06
BUSINESS                 2.178076e+06
COMICS                   9.347692e+05
COMMUNICATION            8.435989e+07
DATING                   1.129533e+06
EDUCATION                5.586231e+06
ENTERTAINMENT            1.925611e+07
EVENTS                   2.495806e+05
FAMILY                   5.201959e+06
FINANCE                  2.395215e+06
FOOD_AND_DRINK           2.156683e+06
GAME                     3.066960e+07
HEALTH_AND_FITNESS       4.642441e+06
HOUSE_AND_HOME           1.917187e+06
LIBRARIES_AND_DEMO       7.411284e+05
LIFESTYLE                1.407444e+06
```



```
MAPS_AND_NAVIGATION    5.286729e+06
MEDICAL                1.150269e+05
NEWS_AND_MAGAZINES     2.648876e+07
PARENTING              5.253518e+05
PERSONALIZATION        5.932385e+06
PHOTOGRAPHY            3.011417e+07
PRODUCTIVITY           3.343418e+07
SHOPPING               1.249173e+07
SOCIAL                 4.769447e+07
SPORTS                 4.560350e+06
TOOLS                  1.358573e+07
TRAVEL_AND_LOCAL       2.662359e+07
VIDEO_PLAYERS          3.555430e+07
WEATHER                5.196348e+06
Name: Installs, 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
```

```
plt.figure(figsize=(16,5))
plt.plot(x , 'ro' , color='b')
plt.xticks(rotation=90)
plt.title('Category Vs Installs')
plt.xlabel('Category -------->')
plt.ylabel('Installs -------->')
plt.show()
```

plt.figure

- The **figure() function** in pyplot module of matplotlib library is used to create a new figure.

plt.plot

- The pyplot.plot () or plt.plot () is a **method of matplotlib pyplot module use to plot the line**.

plt.xticks

- The **annotate() function** in pyplot module of matplotlib library is used to get and set the current tick locations and labels of the x-axis.

plt.title

- The title() method in matplotlib module is used to specify title of the visualization depicted and displays the title using various attributes.

plt.xlabel

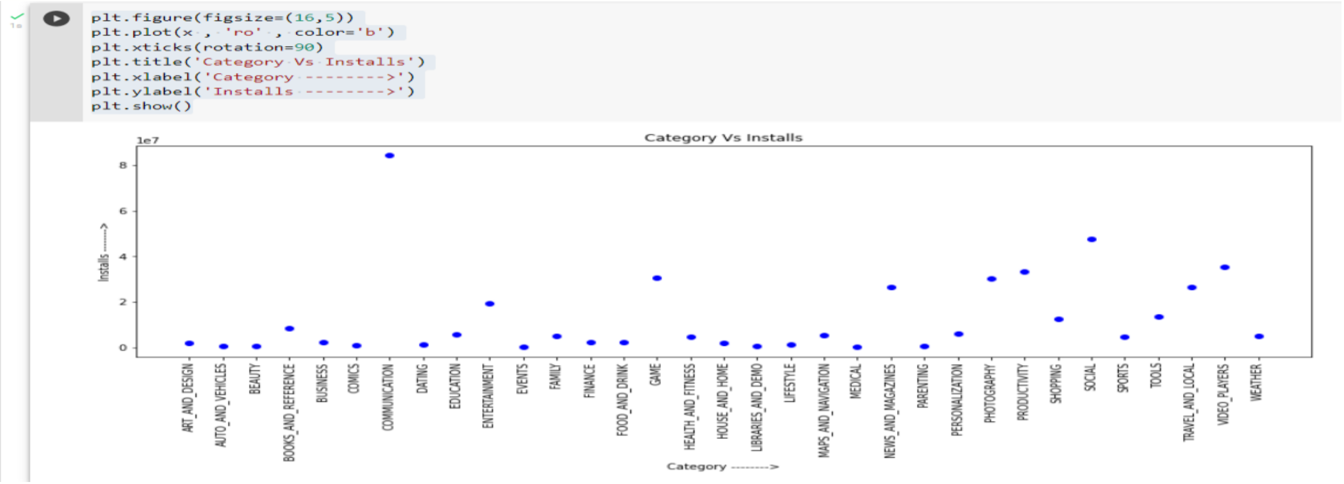
- The **xlabel() function** in pyplot module of matplotlib library is used to set the label for the x-axis.

plt.ylabel

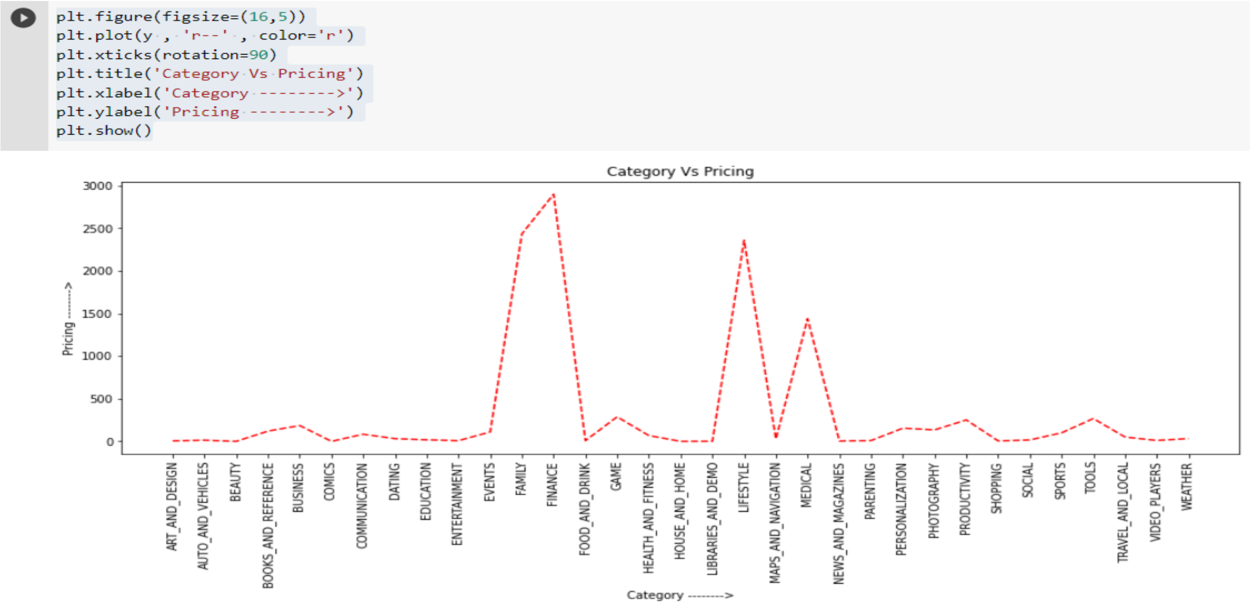
- This function sets the label for the y-axis of the plot.

plt.show()

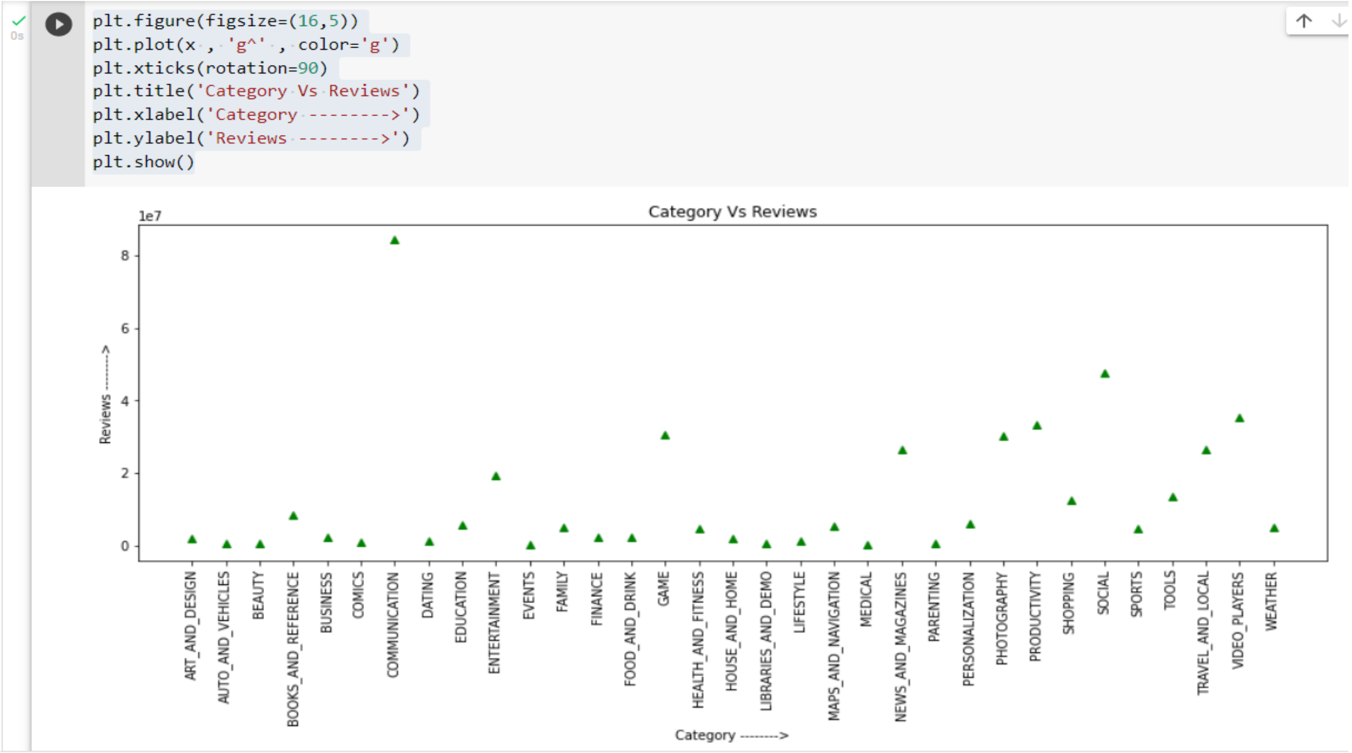
- The **show() function** in pyplot module of matplotlib library is used to display all figures.



```
plt.figure(figsize=(16,5))
plt.plot(x , 'ro' , color='b')
plt.xticks(rotation=90)
plt.title('Category Vs Installs')
plt.xlabel('Category -------->')
plt.ylabel('Installs -------->')
plt.show()
plt.figure(figsize=(16,5))
plt.plot(y , 'r--' , color='r')
plt.xticks(rotation=90)
plt.title('Category Vs Pricing')
plt.xlabel('Category -------->')
plt.ylabel('Pricing -------->')
plt.show()
```



```
plt.figure(figsize=(16,5))
plt.plot(y , 'r--' , color='r')
plt.xticks(rotation=90)
plt.title('Category Vs Pricing')
plt.xlabel('Category -------->')
plt.ylabel('Pricing -------->')
plt.show()
plt.figure(figsize=(16,5))
plt.plot(x , 'g^' , color='g')
plt.xticks(rotation=90)
plt.title('Category Vs Reviews')
plt.xlabel('Category -------->')
plt.ylabel('Reviews -------->')
plt.show()
```



```
plt.figure(figsize=(16,5))
plt.plot(x , 'g^' , color='g')
plt.xticks(rotation=90)
plt.title('Category Vs Reviews')
plt.xlabel('Category -------->')
plt.ylabel('Reviews -------->')
plt.show()
```

Validation Checks:
Validation checks, also called *edit checks*, are programs designed to identify flawed data, or *discrepancies*.

Testing (Testing techniques and Testing strategies):

Chi-Square Test:
A **chi-squared test** (symbolically represented as χ^2) is basically a data analysis on the basis of observations of a random set of variables. Usually, it is a comparison of two statistical data sets.

Properties

The following are the important properties of the chi-square test:

- Two times the number of degrees of freedom is equal to the variance.
- The number of degrees of freedom is equal to the mean distribution
- The chi-square distribution curve approaches the normal distribution when the degree of freedom increases.

Formula

The chi-squared test is done to check if there is any difference between the observed value and expected value.

$$\chi^2 = \sum(O_i - E_i)^2/E_i$$

Future scope of the project:

- Prediction of the number of reviews and installs by using the regression model.
- Identifying the categories and stats of the most installed apps.
- Exploring the correlation between the size of the app, the version of Android, etc on the number of installs.

Conclusion:

The Google Play Store Apps report provides some useful insights regarding the trending of the apps in the play store. As per the graphs visualizations shown above, most of the trending apps (in terms of users' installs) are from the categories like GAME, COMMUNICATION, and TOOL even though the number of available apps from these categories are twice as much lesser than the category FAMILY. The trending of these apps is most probably due to their nature of being able to entertain or assist the user. Besides, it also shows a good trend where we can see that developers from these categories are focusing on the quality instead of the quantity of the apps.

Other than that, the charts shown above actually implies that most of the apps having good ratings of above 4.0 are mostly confirmed to have high number of reviews and user installs. There are some spikes in term of size and price but it should not reflect that apps with high rating are mostly big in size and pricy as by looking at the graphs they are most probably are due to some minority. Furthermore, most of the apps that are having high number of reviews are from the categories of SOCIAL, COMMUNICATION and GAME like Facebook, WhatsApp Messenger, Instagram, Messenger – Text and Video Chat for Free, Clash of Clans etc.

Even though apps from the categories like GAME, SOCIAL, COMMUNICATION and TOOL of having the highest number of installs, rating and reviews are reflecting the current trend of Android users, they are not even appearing as category in the top 5 most expensive apps in the store (which are mostly from FINANCE and LIFESTYLE). As a conclusion, we learnt that the current trend in the Android market is mostly from these categories which either assisting, communicating or entertaining apps.