GROUP 10

Assignment -1

Winter Semester 2021-22	**CSE3050**	**Data Visualization and Presentation**	**ELA**	**VL2021220501732**

Name	**Registration No.**
AMOGH AM	20BDS0172
JOSHITA REDDY G	20BDS0178
HARI PRIYA A	20BDS0352
KUDITHIPATI BINDHU SREE	20BDS0356

WEEK 1 TASK

1) What is python environment?

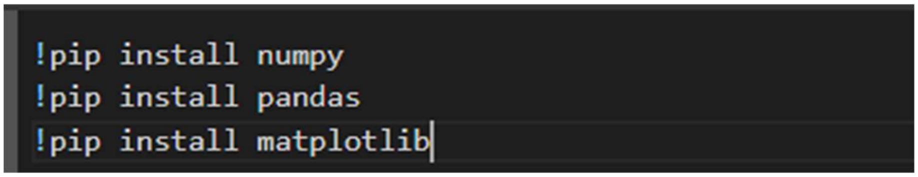
At its core, the main purpose of Python virtual environments is to create an isolated environment for python projects. This means that each project can have its own dependencies, regardless of what dependencies every other project has.

In our little example above, we'd just need to create a separate virtual environment for both *ProjectA* and *ProjectB*, and we'd be good to go. Each environment, in turn, would be able to depend on whatever version of *ProjectC* they choose, independent of the other.

2) How to create and use it?

- Open a terminal
- Setup the pip package manager
- Install the virtualenv package
- Create the virtual environment
- Activate the virtual environment
- Deactivate the virtual environment
- Optional: Make the virtual environment your default Python
- More: Python virtualenv documentation

3) Installing packages in an environment



```
!pip install numpy
!pip install pandas
!pip install matplotlib
```

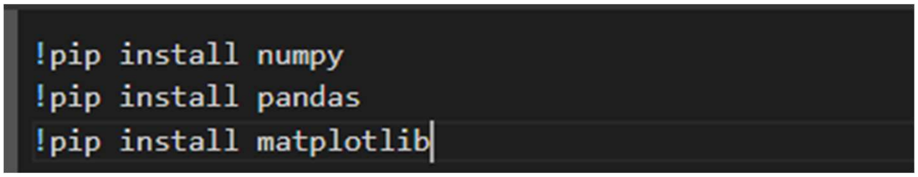
4) Using a particular python interpreter for that environment?

Yes, we are using Jupyter Notebook.

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality."

5) How to use the settings for installing packages?

We used default settings of jupyter notebook.We used these commands to install packages..



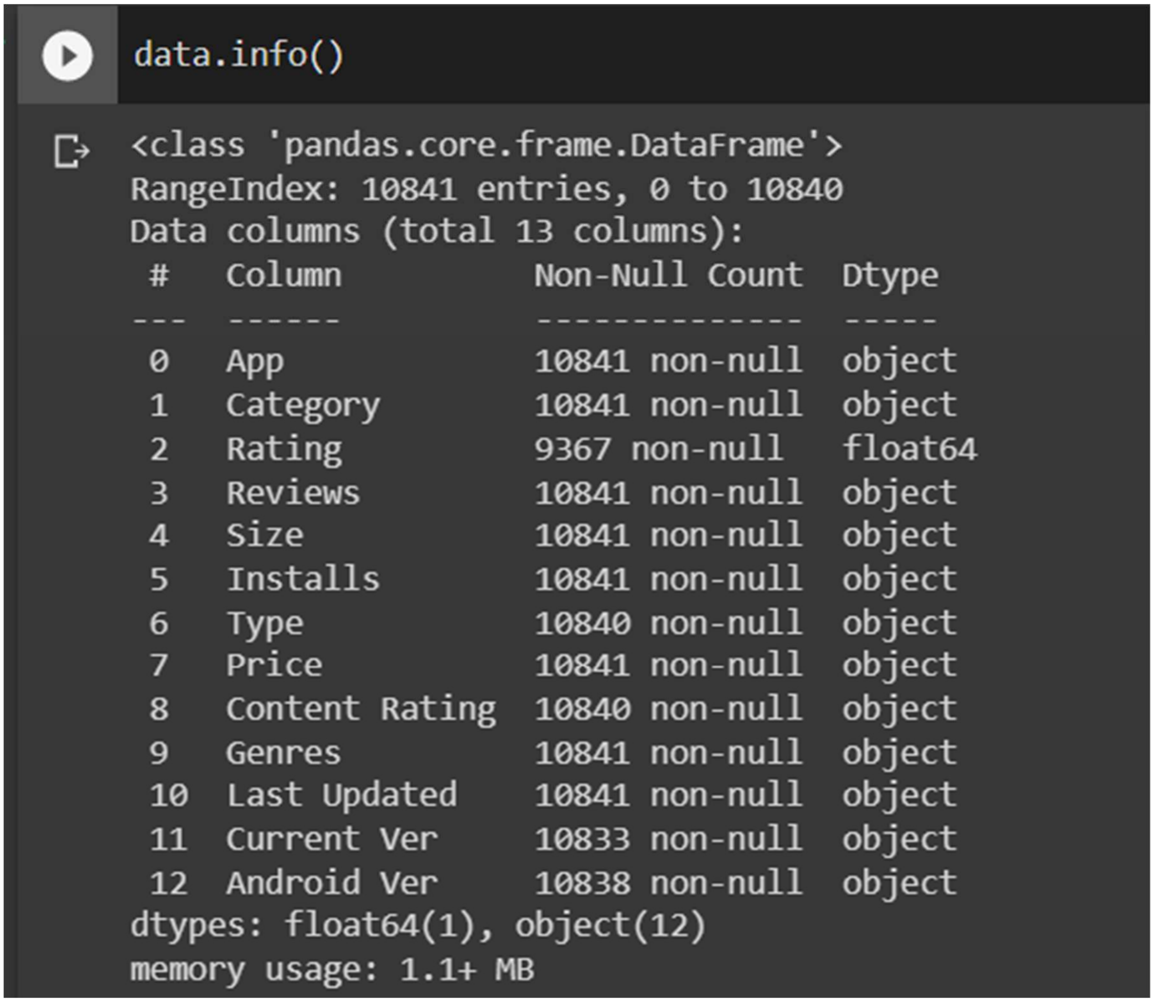
```
!pip install numpy
!pip install pandas
!pip install matplotlib
```

6) Team members 4, identify the application for the team

The Play Store apps data has enormous potential to drive app-making businesses to success. Actionable insights can be drawn for developers to work on and capture the Android market!

WEEK 2 TASK

1) For the identified project topic, explore the dataset associated with the topic. List the attributes and their data types associated as discussed in the theory.



```
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
```

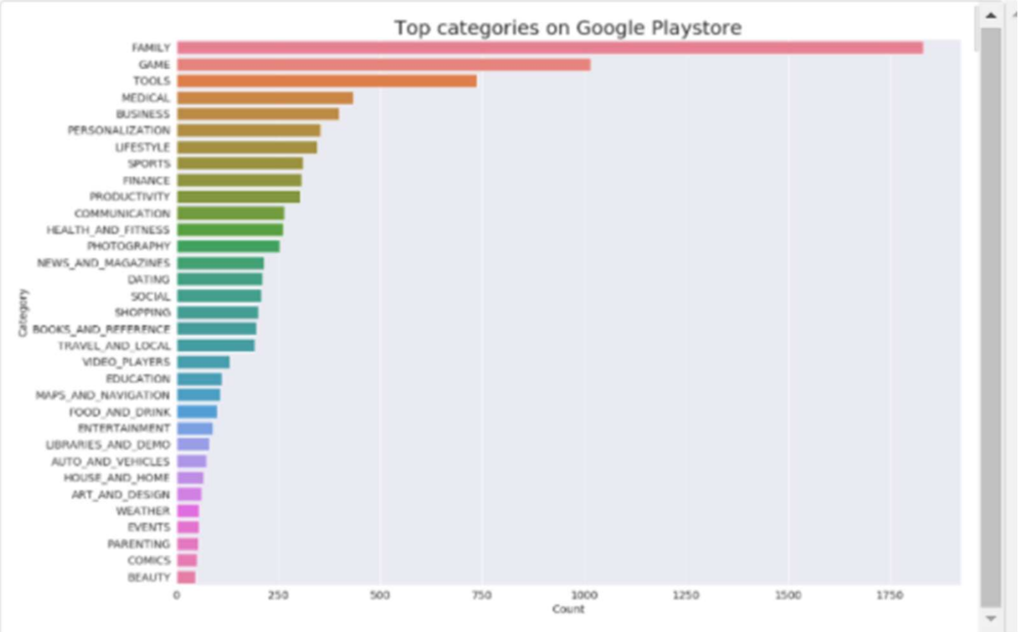
GROUP 10

2) Each member should apply one visualization technique for the comparative task and composition task respectively.

1. BAR GRAPH :

CODE :

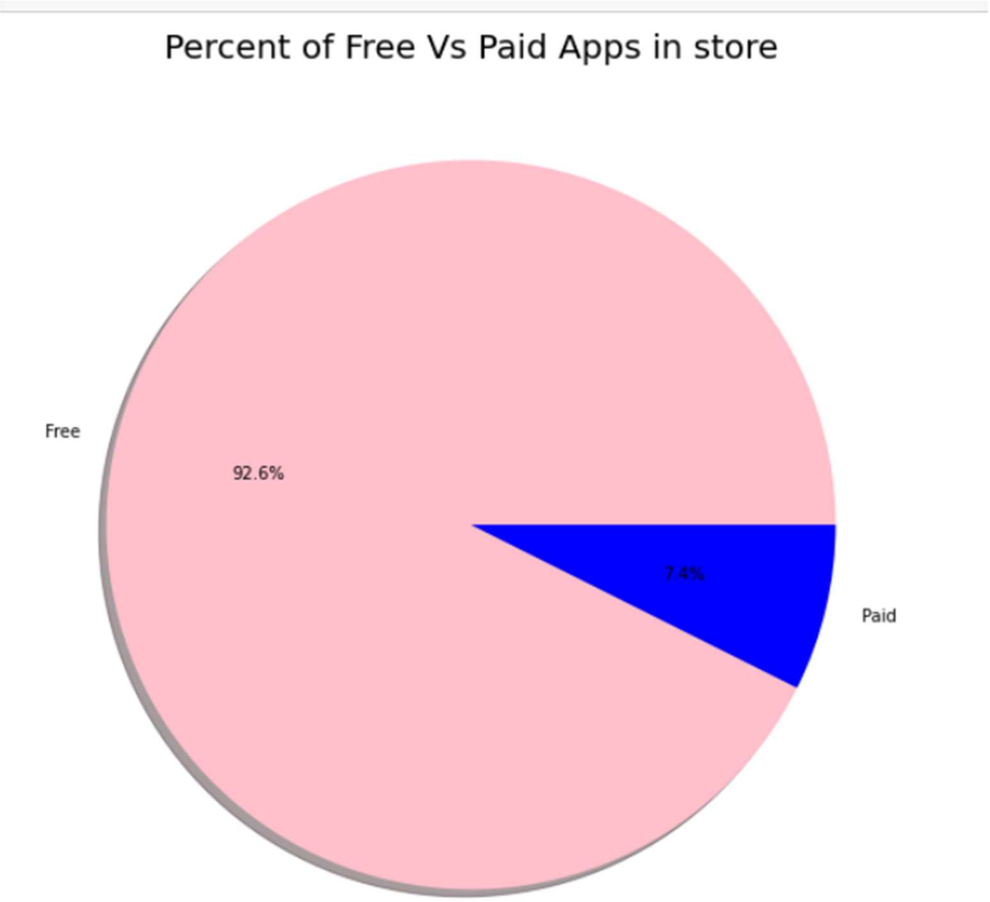
```
y = df['Category'].value_counts().index

x = df['Category'].value_counts()

xsis = []

ysis = []

for i in range(len(x)):

    xsis.append(x[i])

    ysis.append(y[i])

plt.figure(figsize=(18,13))

plt.xlabel("Count")

plt.ylabel("Category")

graph = sns.barplot(x = xsis, y = ysis, palette= "husl")

graph.set_title("Top categories on Google Playstore", fontsize = 25);
```



2)PIE CHART :

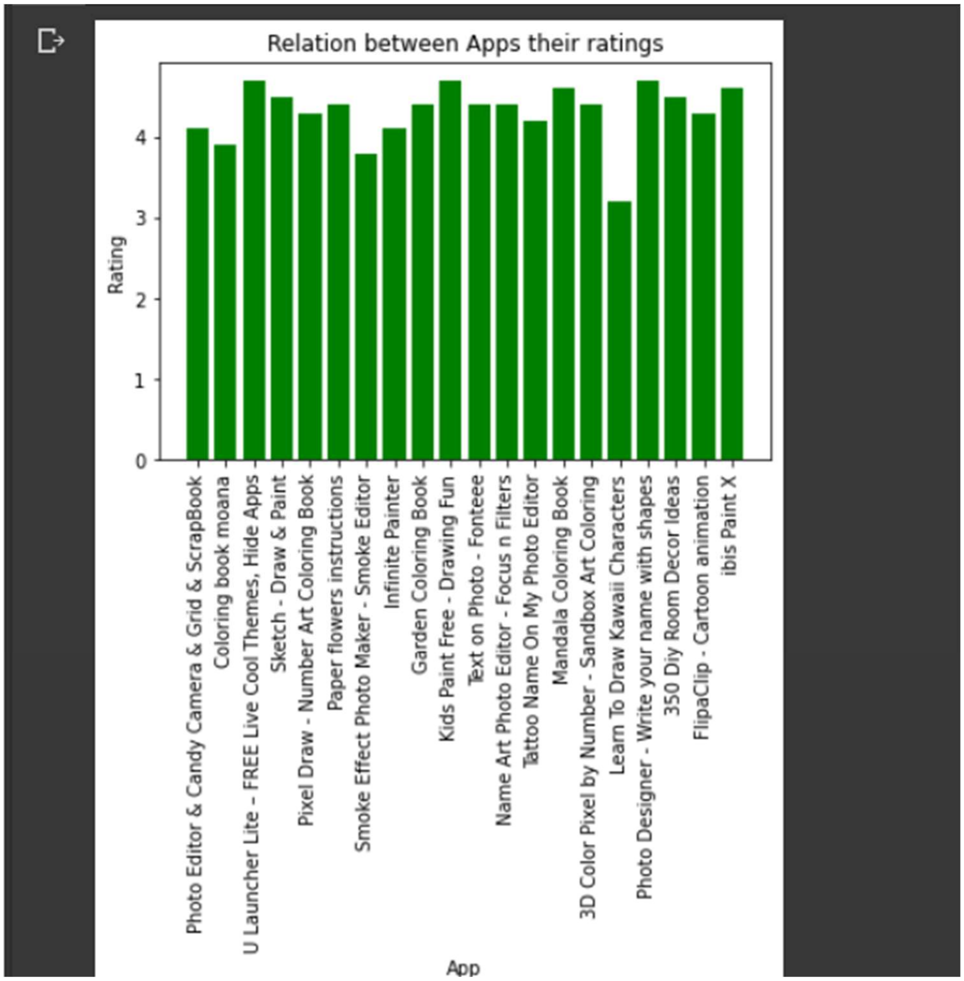
CODE :

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

labels = df['Type'].value_counts(sort = True).index

sizes = df['Type'].value_counts(sort = True)

colors = ["pink","blue"]

explode = (0.2,0)

plt.pie(sizes, labels=labels, colors=colors,autopct='%1.1f%%', shadow=True,startangle=0 )

plt.title('Percent of Free Vs Paid Apps in store',size = 20)

plt.show()
```



3. Bar Graph-2

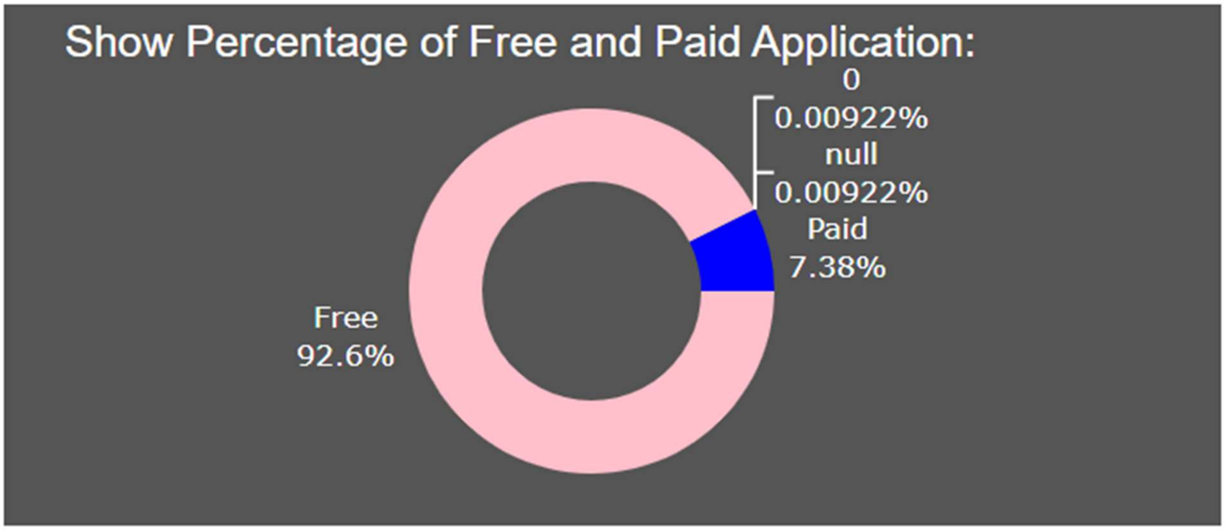
```
plt. bar(d[ 'App' ], d[ 'Rating' ], color='green")

plt. title('Relation between Apps their ratings' )

plt.xticks(rotation = 90)

plt.xlabel( 'App")

plt.ylabel( 'Rating' )

plt. show( )
```



The app ‘Learn to draw kawaii characters’ has the lowest rating

4. Percentage (Donut) Chart:

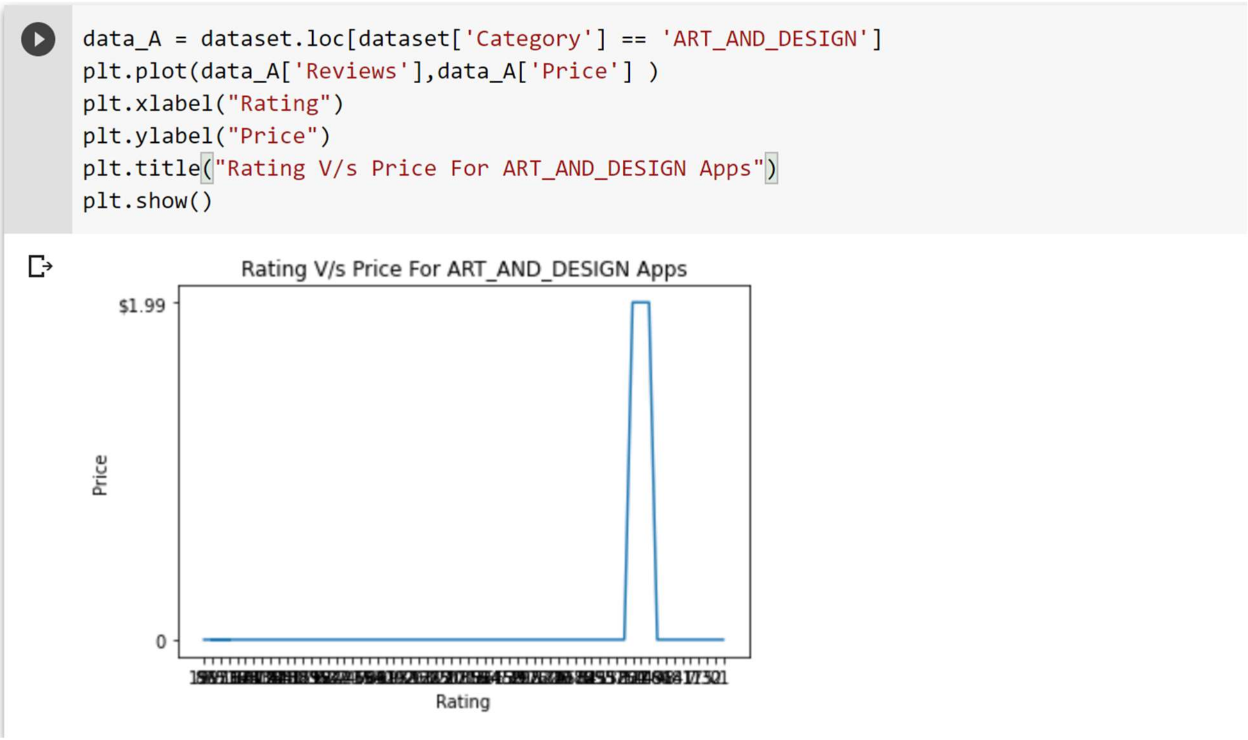
```
    fig= px.pie(x, names='Type', height=300, width=700, hole=0.6,
title='Show Percentage of Free and Paid Application:',
color_discrete_sequence=['#FFC0CB', '#0000FF'])
fig.update_traces(hovertemplate=None, textposition='outside',
textinfo='percent+label', rotation=90)
fig.update_layout(margin=dict(t=60, b=30, l=0, r=0), showlegend=False,
plot_bgcolor='#555', paper_bgcolor='#555',
title_font=dict(size=25, color='#FFFFFF', family="Lato, sans-serif"),
font=dict(size=17, color='#FFFFFF'),
hoverlabel=dict(bgcolor="#444", font_size=13,
font_family="Lato, sans-serif"))
```



There are more number of free apps on the google play store

5. Line Chart

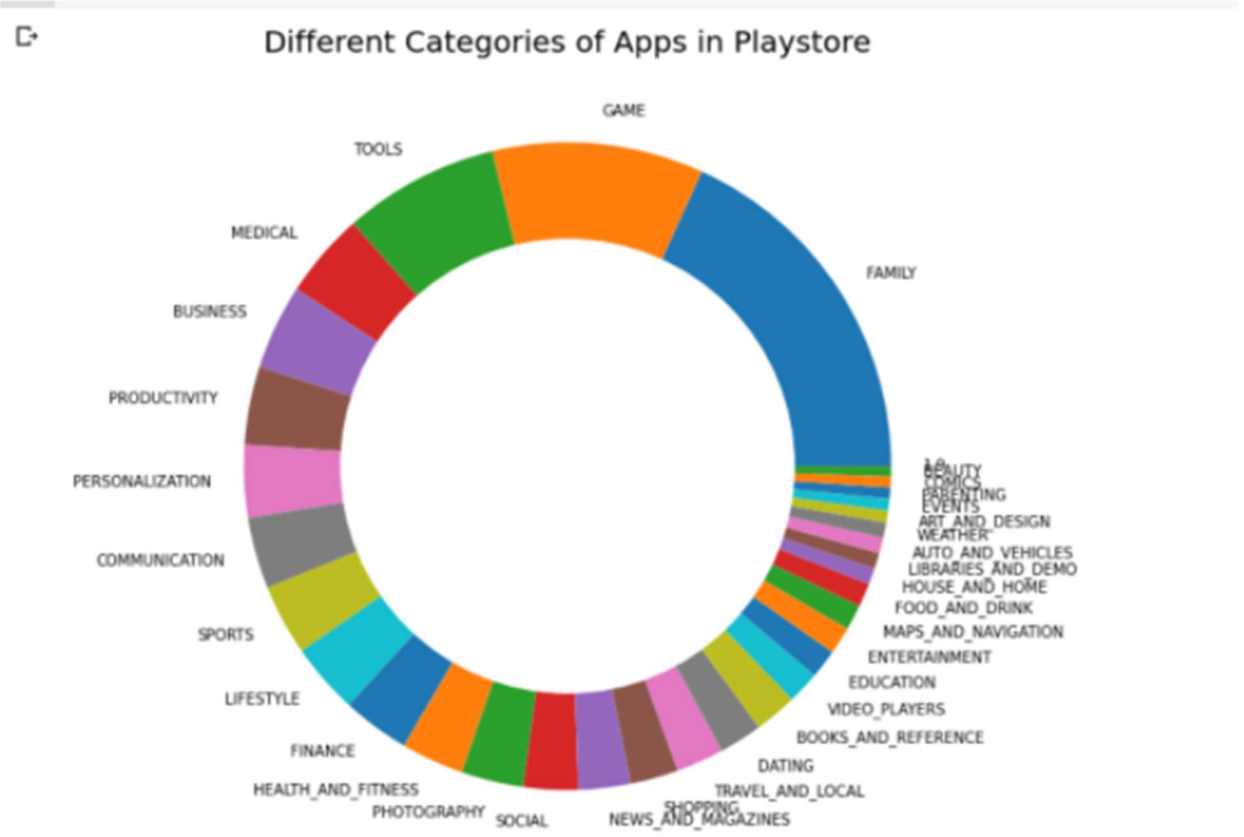
```
data_A = dataset.loc[dataset['Category'] == 'ART_AND_DESIGN']

plt.plot(data_A['Reviews'],data_A['Price'] )

plt.xlabel("Rating")

plt.ylabel("Price")

plt.title("Rating V/s Price For ART_AND_DESIGN Apps")

plt.show()
```



```
data_A = dataset.loc[dataset['Category'] == 'ART_AND_DESIGN']
plt.plot(data_A['Reviews'],data_A['Price'] )
plt.xlabel("Rating")
plt.ylabel("Price")
plt.title("Rating V/s Price For ART_AND_DESIGN Apps")
plt.show()
```

6. Donut Chart

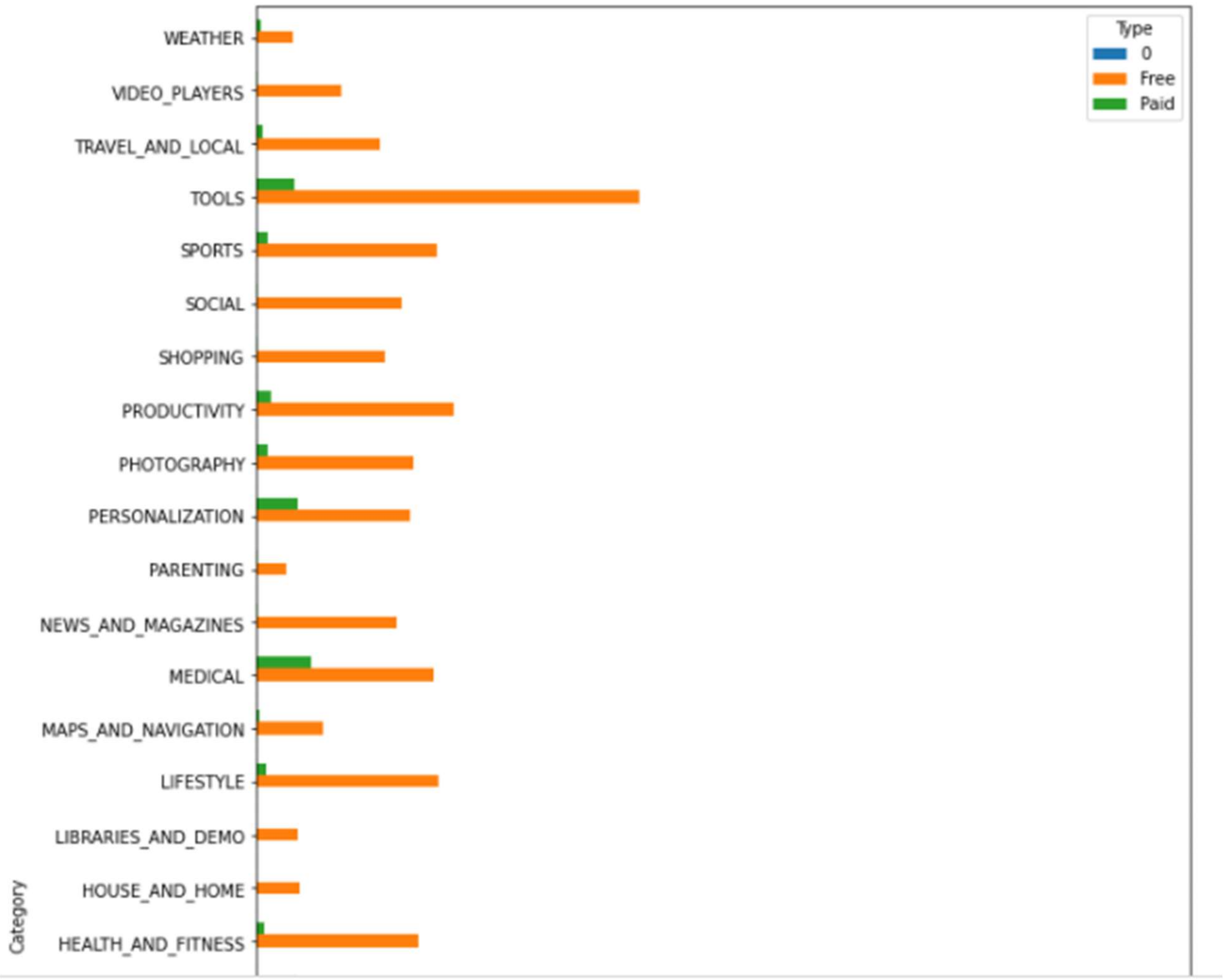
```
plt.figure(figsize=(10,10))

labels = dataset['Category'].value_counts(sort = True).index

sizes = dataset['Category'].value_counts(sort = True)

my_circle=plt.Circle( (0,0), 0.7, color='white')

p=plt.gcf()

p.gca().add_artist(my_circle)

plt.pie(sizes, labels=labels )

plt.title('Different Categories of Apps in Playstore',size = 20)

plt.show()
```



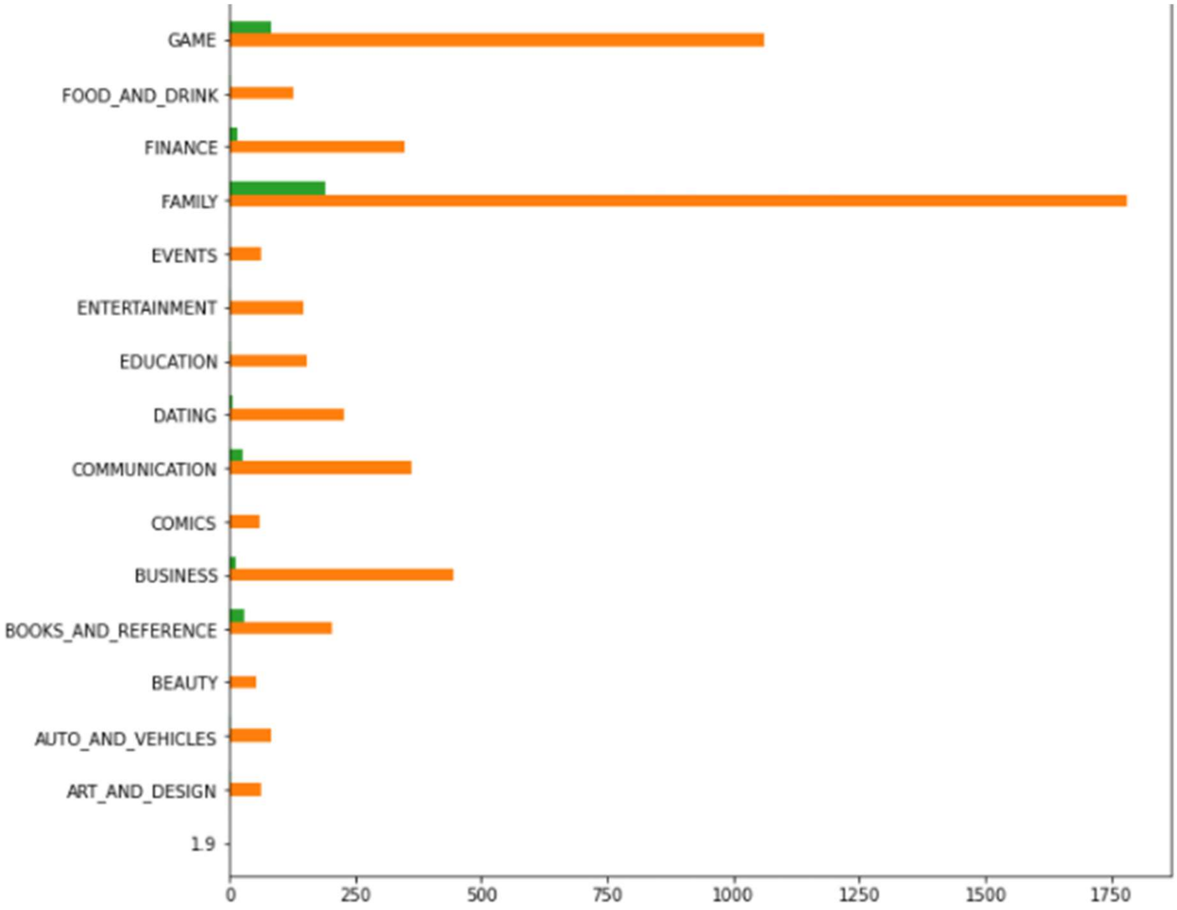
The beauty category has the least number of app

7.Double bar graph:

```
x['Type'].replace(to_replace=['0'], value=['Free'],inplace=True)

x['Type'].fillna('Free', inplace=True)

print(x.groupby('Category')['Type'].value_counts())

Type_cat = x.groupby('Category')['Type'].value_counts().unstack().plot.barh(figsize=(10,20), width=0.7)

plt.show()
```



GROUP 10

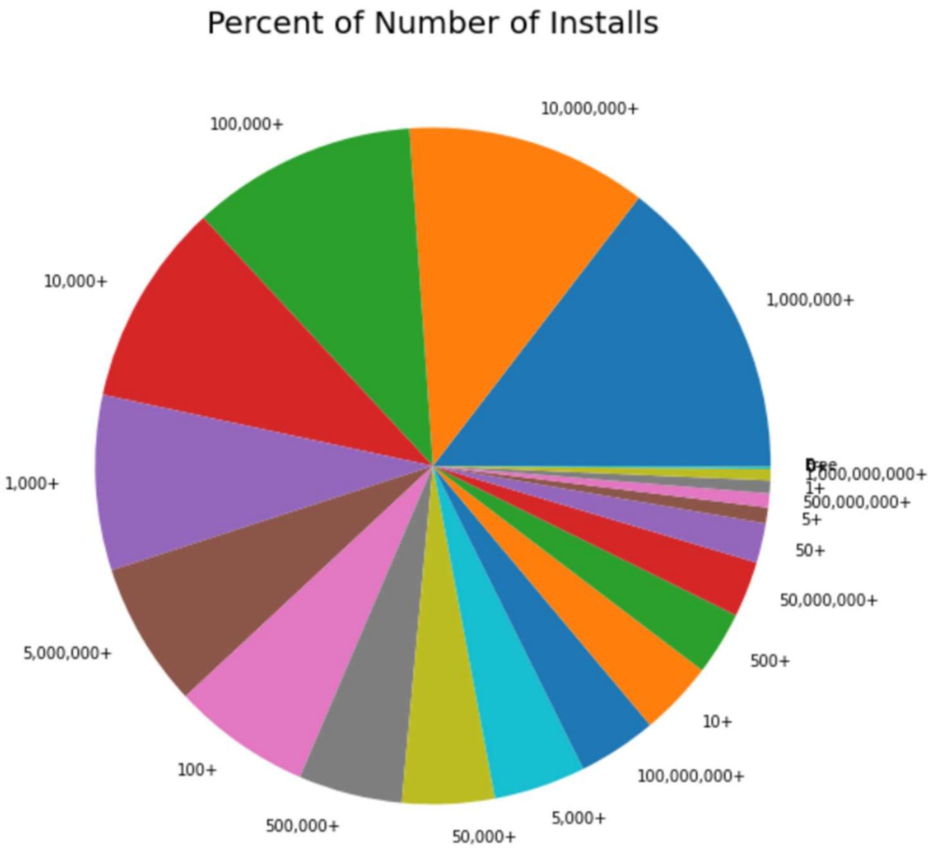


There are more number of paid apps in the finance category

The Family category has the most number of apps

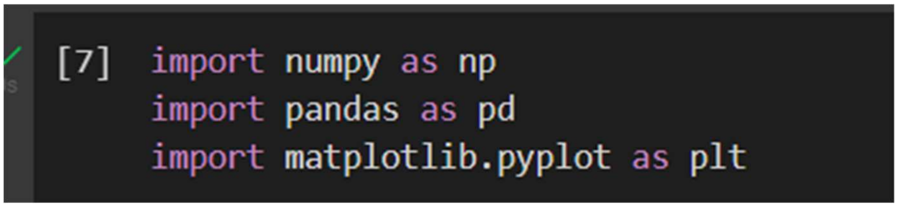
8.) Pie Chart

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

labels = dataset['Installs'].value_counts(sort = True).index

sizes = dataset['Installs'].value_counts(sort = True)

plt.pie(sizes, labels=labels,startangle=0 )

plt.title('Percent of Number of Installs',size = 20)

plt.show()
```



There are more number of apps with 1,000,000+ installs where the apps with 1,000,000,000+ installs are less.

3) Python library to be used: pandas, numpy, matplotlib.pyplot



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

4) Deduce the insights obtained from visualization of the data and list a minimum of two insights

- Apps falling in category ‘FAMILY’, ‘GAME’, ‘TOOLS’ have the maximum count in google play store. This tends to show that businesses tend to invest more in these kinds of apps.
- The high percentage of free apps shows that users tend to prefer not using paid apps.
- ‘U-Launcher Lite’ & ‘Photo Designer’ have high ratings.
- As the app rating increase, the cost of the app also increases.
- In each category, the percentage of paid apps is very less compared to free apps.

GROUP 10

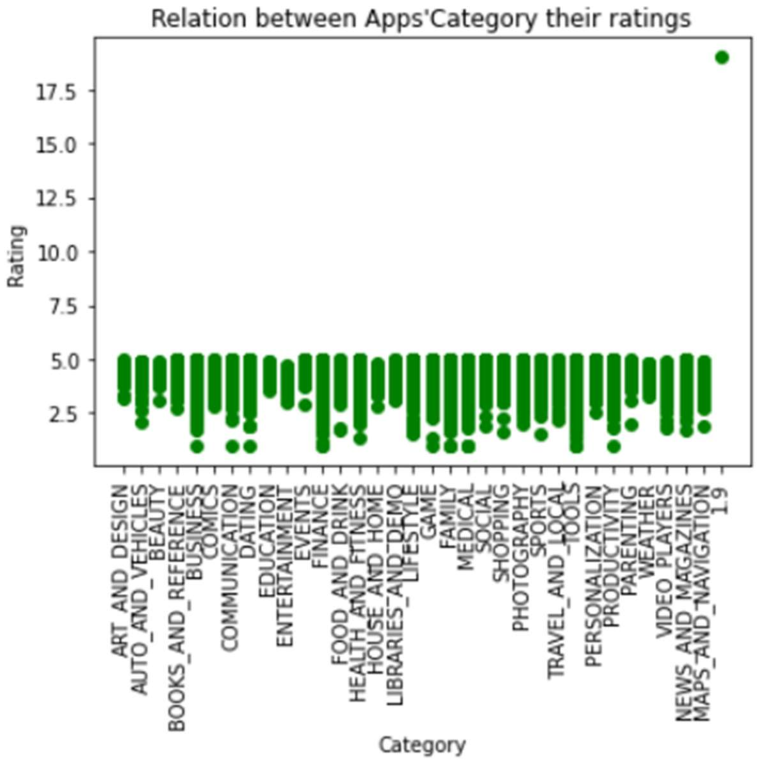
WEEK 3 TASK

1) Apply techniques to understand the distribution and relational characteristics of the attributes associated with the considered database.

RELATIONAL TECHNIQUES:

SCATTER PLOT:

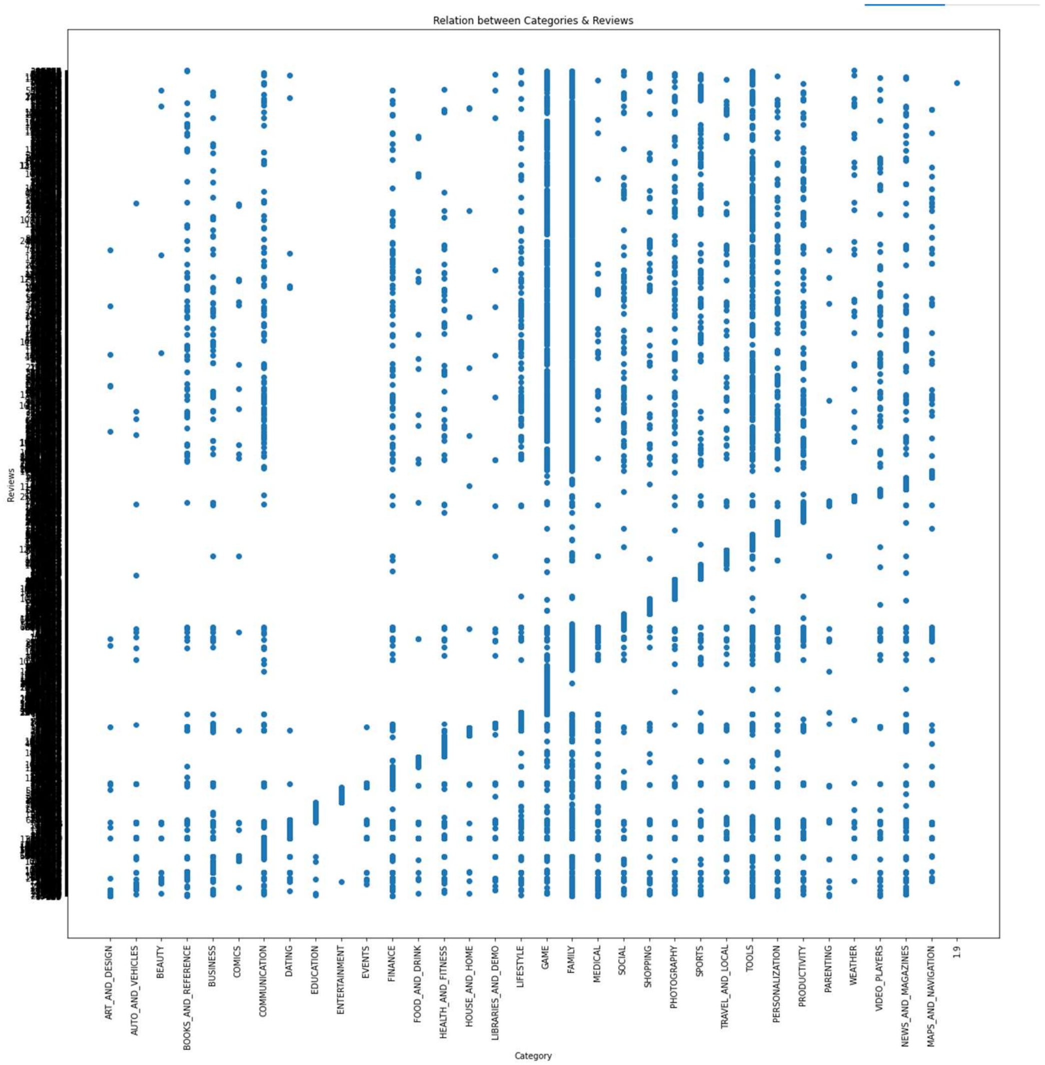
```
plt.scatter(data['Category'],data['Rating'],color='green')

plt.title("Relation between Apps'Category their ratings")

plt.xticks(rotation = 90)

plt.xlabel('Category')

plt.ylabel('Rating')

plt.show()
```



The category with the smallest strip has the highest ratings

CODE:

```
plt.figure(figsize=(20,20))

plt.scatter(dataset['Category'],dataset['Reviews'],color='green')

plt.title("Relation between Categories & Reviews")

plt.xticks(rotation = 90)

plt.xlabel('Category')

plt.ylabel('Reviews')

plt.show()
```



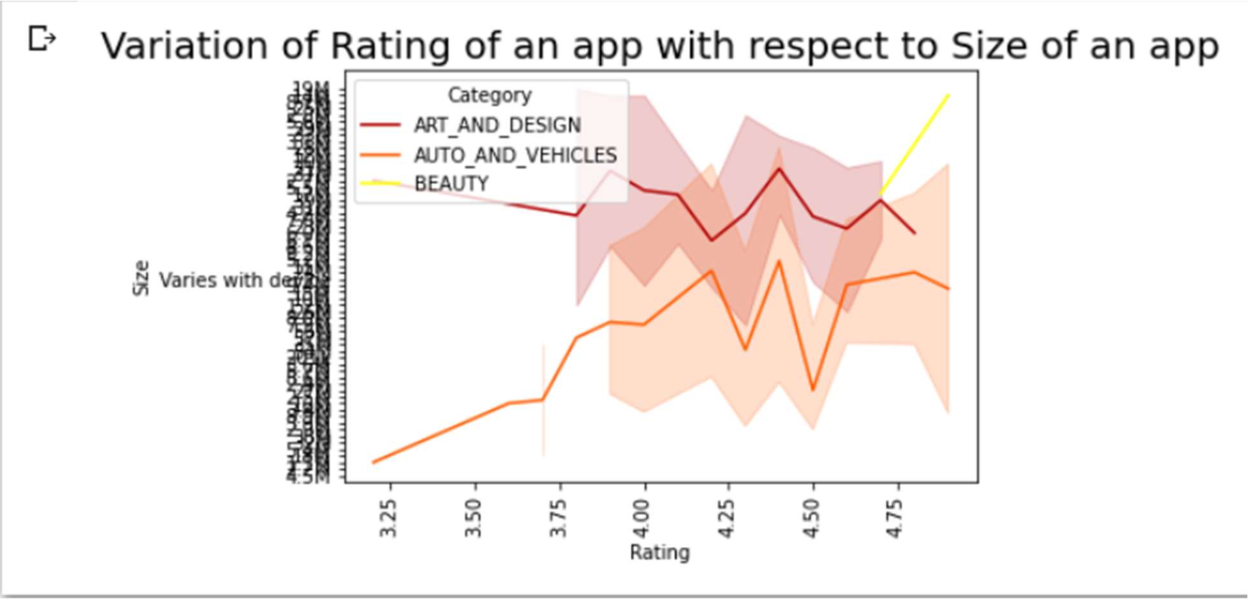
‘FAMILY’ Category has the most reviews from the plot.

DISTRIBUTION TECHNIQUES:

LINE PLOT:

CODE:

```
d=data.head(100)

sns.lineplot(x="Rating",y=”Size”,data=d,hue="Category",palette="hot",da
shes=False,markers=["o","<"],legend="brief")

plt.xticks(rotation = 90)

plt.title("Variation of Rating of an app with respect to Size of an
app",fontsize=20)

plt.show()
```

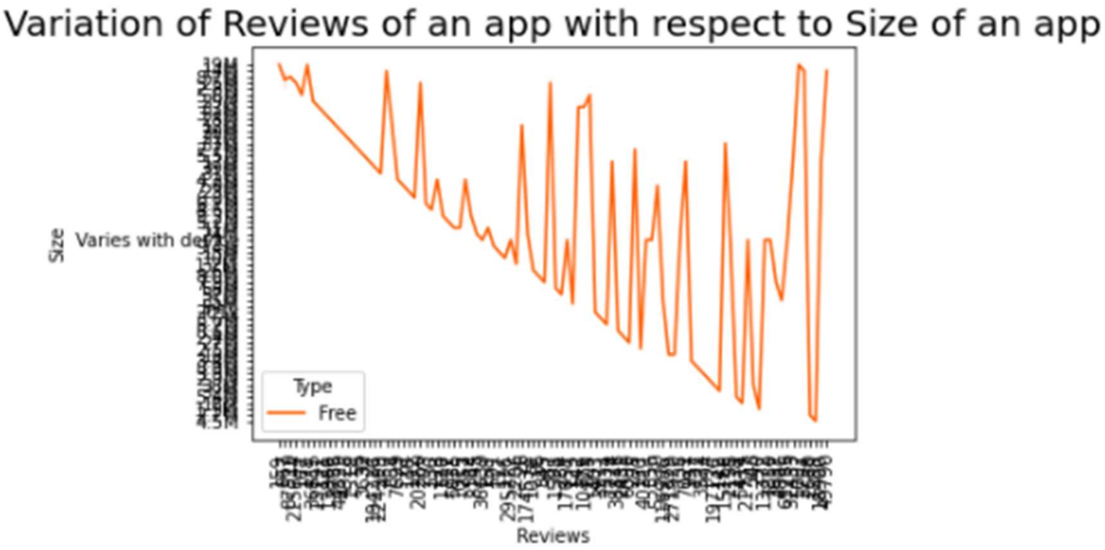


Most apps in categories ‘ART_AND_DESIGN’, ‘AUTO_AND_VEHICLES’,’ BEAUTY’ have higher size have ratings above 4.

3.

CODE:

```
d=x.head(100)

sns.lineplot(x="Reviews",y="Size",data=d,hue="Type",palette="hot",dashes=False,markers=["o","<"],legend="brief")

plt.xticks(rotation =90)

plt.title("Variation of Reviews of an app with respect to Size of an app",fontsize=20)

plt.show()
```

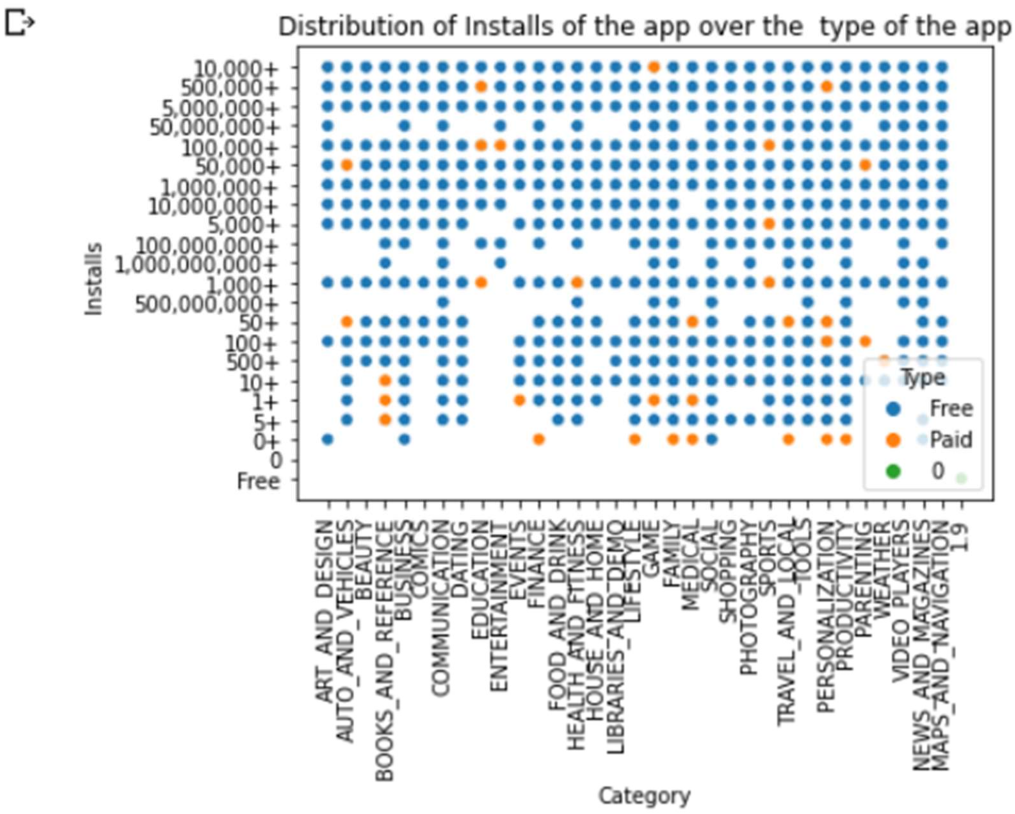


GROUP 10

SCATTER PLOT

CODE:

```
import seaborn as sns

x=pd.DataFrame(data)

g=sns.scatterplot(x="Category", y="Installs",hue='Type',data=data);

plt.xticks(rotation = 90)

plt.title("Distribution of Installs of the app over the  type of the
app")

plt.show()
```



- Free apps have more installs compared to paid apps. This indicates that people prefer free apps over paid apps.

2) Include title, xlabel, ylabel, legends, colours, font size

Done Above

3) Each member has to contribute one graph for each task

Done Above

4) Describe the observations and state the inference for each graph with respect to the considered problem.

- Free apps have more installs compared to paid apps. This indicates that people prefer free apps over paid apps.
- Most apps in categories ‘ART_AND_DESIGN’, ‘AUTO_AND_VEHICLES’,’ BEAUTY’ have higher size and have ratings above 4.
- ‘FAMILY’ Category has the most reviews from the plot.