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.

Each app (row) has values for catergory, rating, size, and more. Another dataset contains customer reviews of the android apps.

Explore and analyze the data to discover key factors responsible for app engagement and success.

```
 # We need to import some Libraries
import pandas as pd
import numpy as np
```

```
 # Mount drive and read csv file.To make sure we are using correct file path.
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
```

```
file_path ='/content/Play Store Data.csv'
df = pd.read_csv(file_path)
df
```

	App	Category	Rating	Reviews	Size	Installs	Type	P
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	
	U Launcher Lite –							

```
# To know if there is any missing value or non value in the given dataset.
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
```

Double-click (or enter) to edit

```
# defining a function which can be reuse
def printinfo():
  temp = pd.DataFrame(index= df.columns)
  temp ['data_type'] = df.dtypes
  temp ['null_count']= df.isnull().sum()
  temp ['unique_count']=df.nunique()
  return temp


# Let's call the function and see what it will returns
printinfo()
```

	data_type	null_count	unique_count
App	object	0	9660
Category	object	0	34
Rating	float64	1474	40
Reviews	object	0	6002
Size	object	0	462
Installs	object	0	22
Type	object	1	3
Price	object	0	93
Content Rating	object	1	6
Genres	object	0	120
Last Updated	object	0	1378
Current Ver	object	8	2832

```
# Now We will start the process of data cleaning, lets begins with the collumns.
df[df.Type.isnull()]
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Gen
Command					Varies					

```
# Now we can File the missing value with free
df['Type'].fillna("Free", inplace = True)


# After added the value we will be able check and see if that has been added correctly.
df.isnull().sum()
```

```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               0
Price              0
Content Rating     1
Genres             0
Last Updated       0
Current Ver        8
Android Ver        3
dtype: int64
```

```
# Now, we'll be able to change place of column content Rating
```

```
df[df['Content Rating'].isnull()]
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
	Life Made								

```
# We can clearly see that row 10472 has missing data for the categorycolumn
df.dropna(subset=['Content Rating'], inplace= True)


# We are having some of the unwanted columns
df.drop(['Current Ver','Last Updated','Android Ver'], axis=1 , inplace= True)


# We can fix the Rating column which contains a total of 1474 of missing values.
modeValueRating = df['Rating'].mode()


# Finally,after fixing all the missing values, we should have a look at our data frame
printinfo()
```

	data_type	null_count	unique_count
App	object	0	9659
Category	object	0	33
Rating	float64	1474	39
Reviews	object	0	6001
Size	object	0	461
Installs	object	0	21
Type	object	0	2
Price	object	0	92
Content Rating	object	0	6
Genres	object	0	119

```
# starting with the column Reviews, converting its type to int
df['Rating'] = df.Reviews.astype(int)


# We can see that the changes have taken its effect or not by calling our
printinfo()
```

	data_type	null_count	unique_count
App	object	0	9659
Category	object	0	33
Rating	int64	0	6001
Reviews	object	0	6001
Size	object	0	461
Installs	object	0	21
Type	object	0	2
Price	object	0	92

```
# Removing the + sign
# df['size'] = df.size.apply(lambda x: x.strip('+'))
# df['size'] = df.size.apply(lambda x: x.strip('+'))


#df = df.drop(labels=10472, axis=0)


# Removing the , symbol
#df['size']= df.size.apply(lambda x: x.replace(',', ''))
# Replacing the M and K symbol by multiplying the value with 100000
df['Size'] = df.Size.apply(lambda x: x.replace('M','e+6'))# For Converting the M to Mega
#df['size']= df.size.apply(lambda x: x.replace ('k', 'e+3'))# For converting the K to Kilo
# Replacing the Varies with device value with Nan
df['Size'] = df.Size.replace('Varies with device',np.NaN )
# Now, finally converting all these values to numeric type:
#df['size']= pd.to_numeric(df['size'])# Converting the string to Numeric type


# performing all of these operations
printinfo()
```

	data_type	null_count	unique_count
App	object	0	9659
Category	object	0	33
Reviews	object	0	6001
Type	object	0	2
Genres	object	0	119

```
# drop the rows of the column Size having nanvalues
df.dropna(subset = ['Size'],inplace=True)
```

```
# Let's remove the, symbol from the numbers
df[' Installs'] = df.Installs.apply(lambda x: x.replace(',',''))
```

```
# To convert this column from object to integer type
df['Installs'] = df.Installs.apply(lambda x: x.strip('+'))
```

```
for size in df['Size'].unique():
  print(size)
```

```
19e+6
14e+6
8.7e+6
25e+6
2.8e+6
5.6e+6
29e+6
33e+6
3.1e+6
28e+6
12e+6
20e+6
21e+6
37e+6
2.7e+6
5.5e+6
17e+6
39e+6
31e+6
4.2e+6
7.0e+6
23e+6
6.0e+6
6.1e+6
4.6e+6
9.2e+6
5.2e+6
11e+6
24th+6
9.4e+6
15e+6
10e+6
1.2e+6
```

```
26e+6
8.0e+6
7.9e+6
56e+6
57e+6
35e+6
54e+6
201k
3.6e+6
5.7e+6
8.6e+6
2.4e+6
27e+6
2.5e+6
16e+6
3.4e+6
8.9e+6
3.9e+6
2.9e+6
38e+6
32e+6
5.4e+6
18e+6
1.1e+6
2.2e+6
```

```
def fix_Size(s):
    try:
        if s.endswith('M'):
            return  float(s[:-1])

        elif s.endswith('K'):
            return float(s[:-1]) / 1024

        elif s== '1,000+':
            return 1.0

        else:
            return np.nan

    except:
        return np.nan



df['Size'] = df['Size'].apply(fix_Size)
df
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Con Ra
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	159	159	NaN	10,000	Free	0	Ever
1	Coloring book moana	ART_AND_DESIGN	967	967	NaN	500,000	Free	0	Ever
2	U Launcher Lite – FREE Live Cool	ART_AND_DESIGN	87510	87510	NaN	5,000,000	Free	0	Ever

```
for c in df['Category'].unique():
  print(c)
```

```
ART_AND_DESIGN
AUTO_AND_VEHICLES
BEAUTY
BOOKS_AND_REFERENCE
BUSINESS
COMICS
COMMUNICATION
DATING
EDUCATION
ENTERTAINMENT
EVENTS
FINANCE
FOOD_AND_DRINK
HEALTH_AND_FITNESS
HOUSE_AND_HOME
LIBRARIES_AND_DEMO
LIFESTYLE
GAME
FAMILY
MEDICAL
SOCIAL
SHOPPING
PHOTOGRAPHY
SPORTS
TRAVEL_AND_LOCAL
TOOLS
PERSONALIZATION
PRODUCTIVITY
PARENTING
WEATHER
VIDEO_PLAYERS
NEWS_AND_MAGAZINES
MAPS_AND_NAVIGATION
```

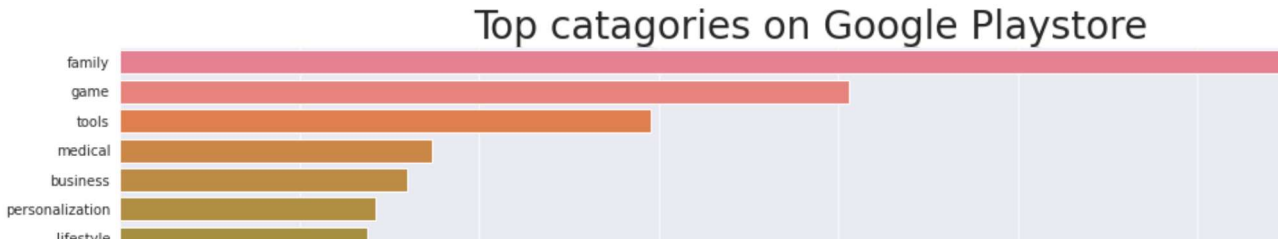
```
df['Category'] = df['Category'].apply(lambda c: c.lower())

for c in df['Category'].unique():
  print(c)
```

```
    art_and_design
    auto_and_vehicles
    beauty
    books_and_reference
    business
    comics
    communication
    dating
    education
    entertainment
    events
    finance
    food_and_drink
    health_and_fitness
    house_and_home
    libraries_and_demo
    lifestyle
    game
    family
    medical
    social
    shopping
    photography
    sports
    travel_and_local
    tools
    personalization
    productivity
    parenting
    weather
    video_players
    news_and_magazines
    maps_and_navigation
```

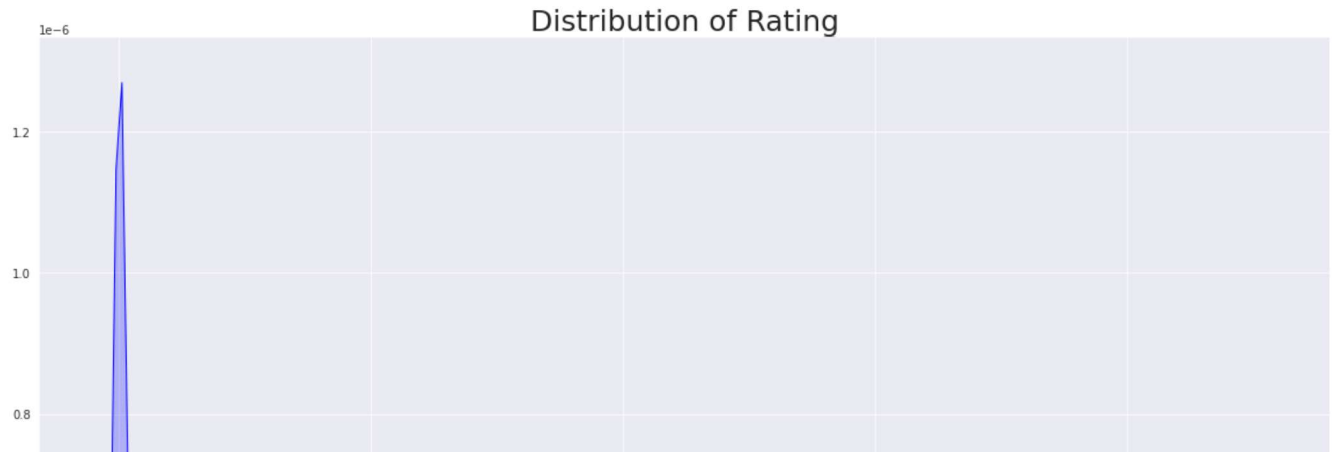
```
# Exploratory Analysis and Visualization
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 10
matplotlib.rcParams['figure.figsize'] = (8,4)
matplotlib.rcParams['figure.facecolor'] = '#00000000'


# Can we see what are the top categories in the play store,which contains the highest number
```

```
y = df['Category'].value_counts().index
x = df['Category'].value_counts()
xaxis =[]
yaxis =[]
for i in range(len(x)):
  xaxis.append(x[i])
  yaxis.append(y[i])

# We have defined our x and y axis.let us plot and see
plt.figure(figsize=(18,13))
plt.xlabel("Count")
plt.ylabel("catagory")

graph = sns.barplot(x = xaxis, y = yaxis, palette="husl")
graph.set_title("Top catagories on Google Playstore",fontsize =28)
```

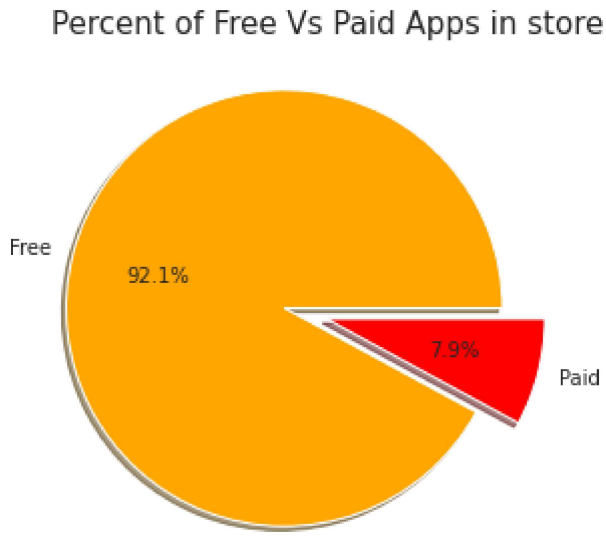
```
Text(0.5, 1.0, 'Top catagories on Google Playstore')
```




```
# Let's have a look at the distribution of the ratings of the data frame.
plt.figure(figsize=(20,15))
plt.xlabel("Rating")
plt.ylabel("Frequency")
graph = sns.kdeplot(df.Rating,color="b",shade =True)
plt.title('Distribution of Rating',size = 25);
```




```
# Let's plot a visualization graph to view what portion of the apps in the play store are pai
plt.figure(figsize=(5,5))
labels=df['Type'].value_counts(sort = True).index
sizes = df['Type'].value_counts(sort = True)
colors = ["Orange","Red"]
explode =(0.2,0)
plt.pie(sizes, explode=explode,  labels=labels, colors=colors, autopct='%1.1f%%',  shadow=Tru
plt.title('Percent of Free Vs Paid Apps in store',size =15)
plt.show()
```




```
# What are the count of Apps in different genres?
topAppsinGenres = df['Genres'].value_counts().head(50)


x3sis = []
y3sis = []

for i in range(len(topAppsinGenres)):
  x3sis.append(topAppsinGenres.index[i])
  y3sis.append(topAppsinGenres[i])
```

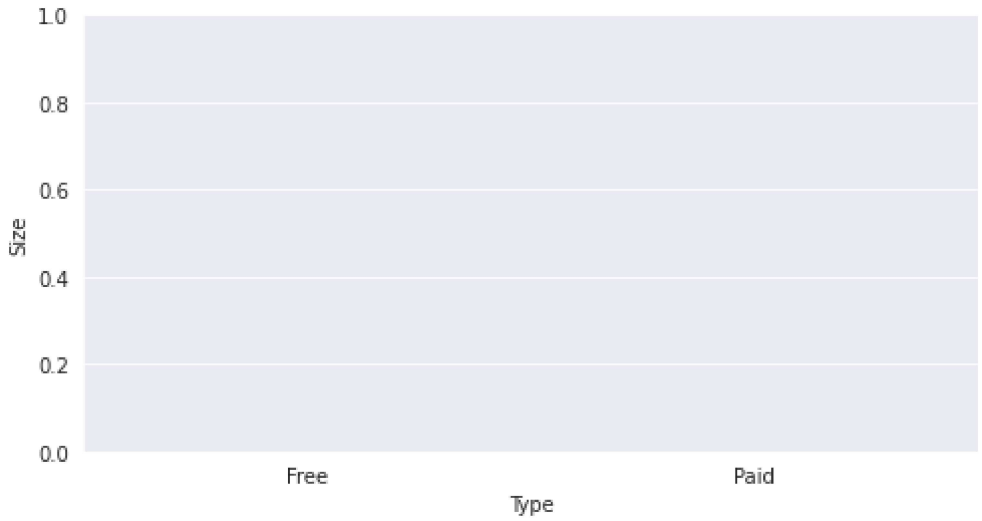
```
# a state to plot and gain an insight into our raised question.
plt.figure(figsize=(20,15))
plt.ylabel('Genres(App Count)')
plt.xlabel('Genres')
graph= sns.barplot(x=x3sis,y=y3sis,palette="deep")
graph.set_xticklabels(graph.get_xticklabels(),rotation=90,fontsize=14)
graph.set_title("Top Genres in the playstore", fontsize = 25);
```


```
# What is the average size of the free apps & paid apps.
df.groupby('Type').describe()['Size']
```

Type	count	mean	std	min	25%	50%	75%	max
Free	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Paid	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
sns.boxplot(x='Type', y='Size', data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9c7bbe4dd0>
```




```
# most reviewed apps.
x =df[df['Reviews']==df['Reviews'].max()]
x
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Ge
2989	GollerCepte	sports	9992	9992	NaN	1,000,000	Free	0	Everyone	S

```
x.hist('Reviews')
```

```
array([[]],
      dtype=object)
```