Analysing Google Play-Store

Project Report

by

Amit Gupta	**19ucs011**
Vipul Aggarwal	**19ucs042**
Rishi Raj Yadav	**19ucs045**
Tanay Makharia	**19ucs122**

Course Coordinator
Dr. Aloke Datta
Dr. Indra Deep Mastan
Dr. Sudheer Sharma

Department of Computer Science & Engineering
The LNM Institute of Information Technology, Jaipur

November 2020

INTRODUCTION

Our topic is data Analysis -> **Data Analysis** is the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data.

In our project we used python and its libraries to get fruitful information from the google play store dataset and showed it with the graph to get maximum output from it.

We implemented our code on Jupyter Notebook.

The Jupyter Notebook is an open-source web application that allows data scientists to create and share documents that integrate live code, equations, computational output, visualizations, and other multimedia resources, along with explanatory text in a single document.

I have downloaded dataset from:
https://www.kaggle.com/lava18/google-play-store-apps

OBJECTIVE

Main objective of our project is to understand the behaviour of people on google play-store and get the fruitful information from it to help developers to make apps according to it. Our project also helps businessmen to advertise their product on apps.

System Requirements:

1. Python3
2. Jupyter Notebook
3. Python libraries like numpy, pandas, scikit-learn, matplotlib, plotly, etc.
4. miniconda3 or anaconda

Dataset information:

The owner of this dataset scraped this data from the website of google play-store, On the other hand, Google Play Store uses sophisticated modern-day techniques (like dynamic page load) using jQuery, making scraping more challenging.

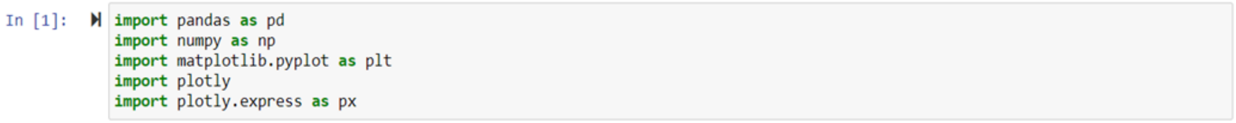
This Dataset contains information about apps like its rating, number of reviews etc.

There are a total of 13 columns and 10829 rows in the dataset.

Attributes of dataset:

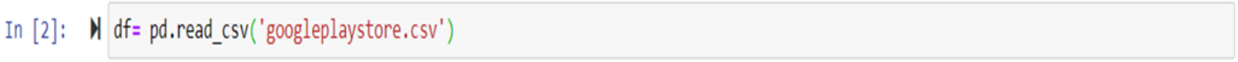
1. App
2. Category
3. Ratings
4. Reviews
5. Size
6. Installs
7. Type
8. Price
9. Content Ratings
10. Genres
11. Last Updated
12. Current Version
13. Android Version

Libraries:



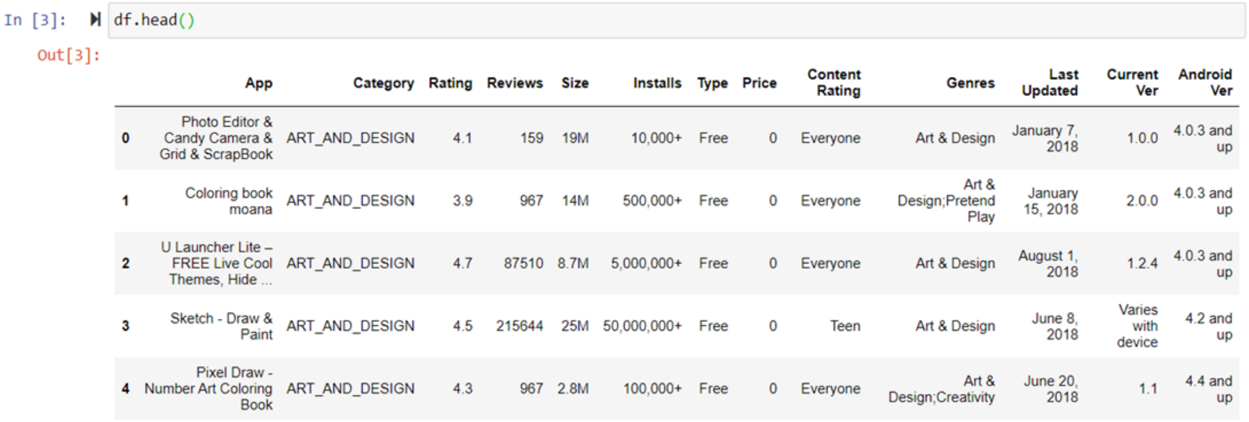
```
In [1]:  import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import plotly
         import plotly.express as px
```

Loading the Dataset:



```
In [2]:  df= pd.read_csv('googleplaystore.csv')
```

Dataset overview:



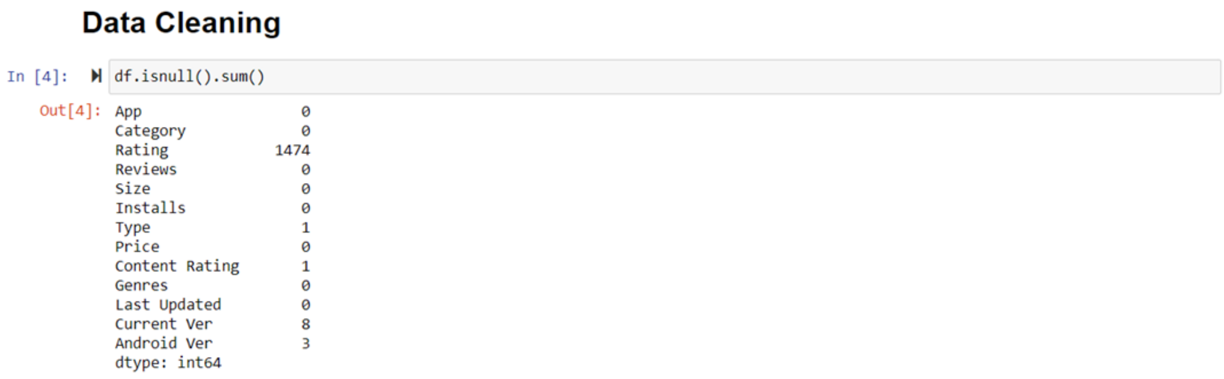
```
In [3]:  df.head()
```

Out[3]:

	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

There are so many inconsistencies in the data, we need to perform data cleaning.

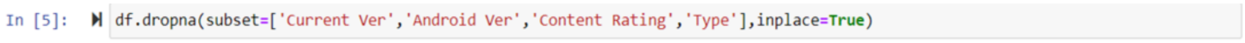
Data Cleaning:



Data Cleaning

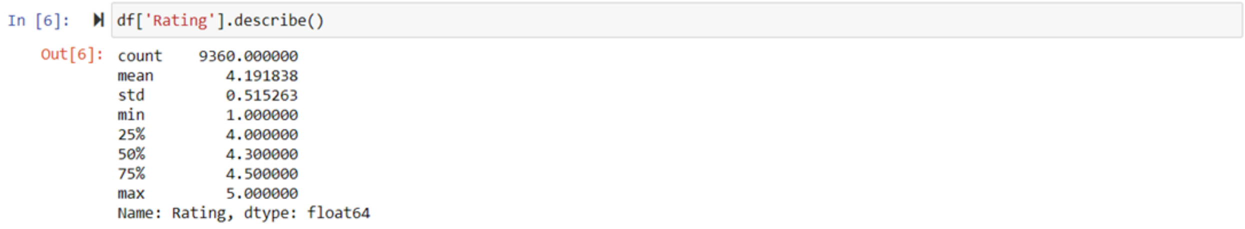
```
In [4]:  df.isnull().sum()

Out[4]:  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
```

As there are very few null values in Type, Content Rating, Current Ver and Android Ver thus we can drop the null values



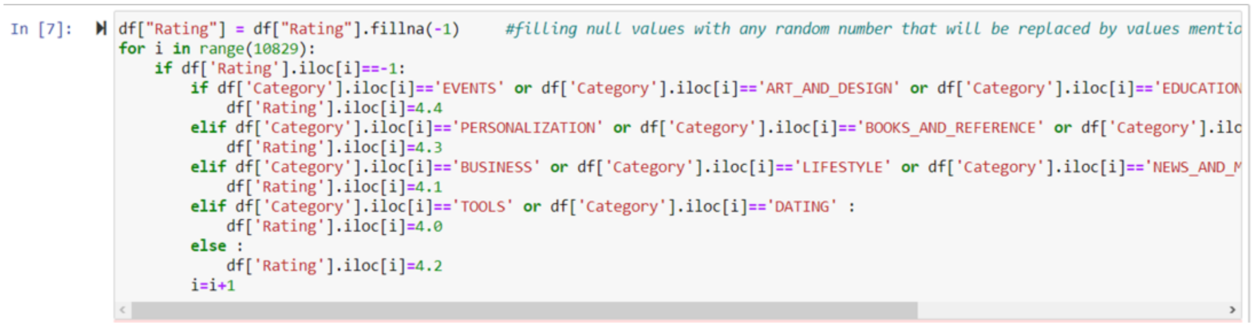
```
In [5]:  df.dropna(subset=['Current Ver','Android Ver','Content Rating','Type'],inplace=True)
```

In the ratings column there are 1474 null values so we can't remove all of them or replace all of them with a single value.



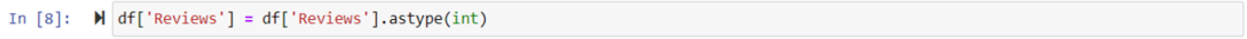
```
In [6]:  df['Rating'].describe()

Out[6]:  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
```

The average of Ratings columns is 4.2 but as the number of null values are high it won't be a good step to fill all the null values with a single value.
We will check the null values distribution under different categories of app and fill the null values with the average of the respective category.
We can find the distribution of null values under various categories using the excel pivot function.



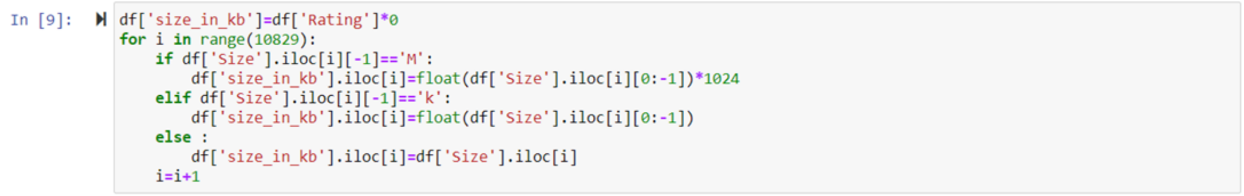
```
In [7]:  df["Rating"] = df["Rating"].fillna(-1)     #filling null values with any random number that will be replaced by values mentio
         for i in range(10829):
             if df['Rating'].iloc[i]==-1:
                 if df['Category'].iloc[i]=='EVENTS' or df['Category'].iloc[i]=='ART_AND_DESIGN' or df['Category'].iloc[i]=='EDUCATION
                     df['Rating'].iloc[i]=4.4
                 elif df['Category'].iloc[i]=='PERSONALIZATION' or df['Category'].iloc[i]=='BOOKS_AND_REFERENCE' or df['Category'].ilc
                     df['Rating'].iloc[i]=4.3
                 elif df['Category'].iloc[i]=='BUSINESS' or df['Category'].iloc[i]=='LIFESTYLE' or df['Category'].iloc[i]=='NEWS_AND_M
                     df['Rating'].iloc[i]=4.1
                 elif df['Category'].iloc[i]=='TOOLS' or df['Category'].iloc[i]=='DATING' :
                     df['Rating'].iloc[i]=4.0
                 else :
                     df['Rating'].iloc[i]=4.2
                 i=i+1
```

Convert Reviews column to int type to make further calculations and statistics easier.



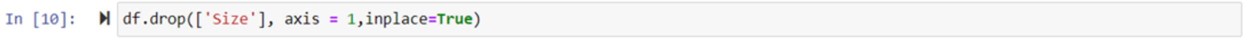
```
In [8]:  df['Reviews'] = df['Reviews'].astype(int)
```

In the “size” column some values are in kb and some are in mb so we can convert all values in kb. I have added a new column size_in_kb that will contain app size in kb.



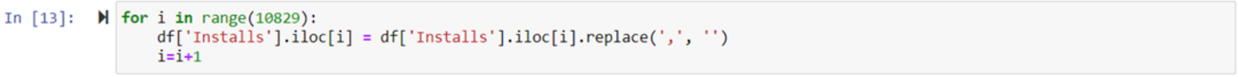
```
In [9]:  df['size_in_kb']=df['Rating']*0
         for i in range(10829):
             if df['Size'].iloc[i][-1]=='M':
                 df['size_in_kb'].iloc[i]=float(df['Size'].iloc[i][0:-1])*1024
             elif df['Size'].iloc[i][-1]=='k':
                 df['size_in_kb'].iloc[i]=float(df['Size'].iloc[i][0:-1])
             else :
                 df['size_in_kb'].iloc[i]=df['Size'].iloc[i]
             i=i+1
```

Now We can Drop column “Size”.

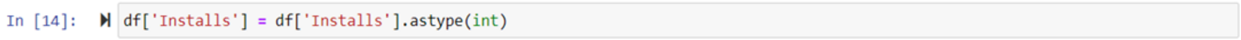


```
In [10]:  df.drop(['Size'], axis = 1,inplace=True)
```

In Installs column the values are in form of 10,000 or 500,000 so before converting it into integer we need to remove the commas in values

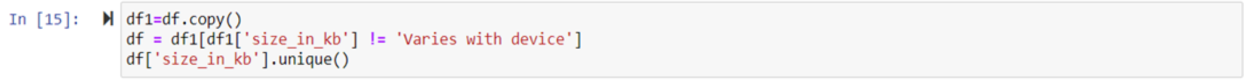


```
In [13]:  for i in range(10829):
              df['Installs'].iloc[i] = df['Installs'].iloc[i].replace(',', '')
              i=i+1
```



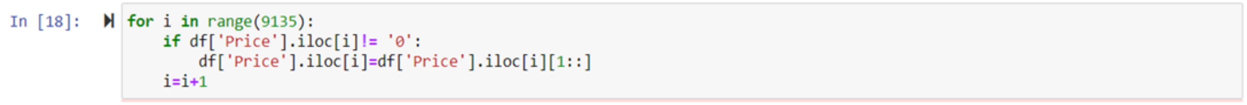
```
In [14]:  df['Installs'] = df['Installs'].astype(int)
```

In size_in_kb column there are some rows with the entry 'varies with device'. we can delete these rows.

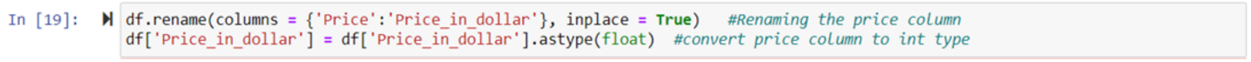


```
In [15]:  df1=df.copy()
          df = df1[df1['size_in_kb'] != 'Varies with device']
          df['size_in_kb'].unique()
```

Editing the price column:

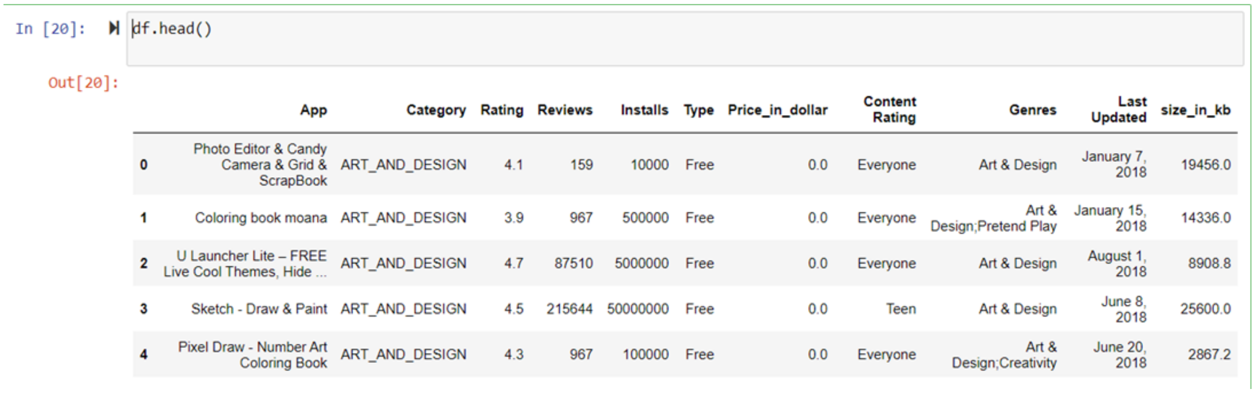


```
In [18]:  for i in range(9135):
              if df['Price'].iloc[i]!= '0':
                  df['Price'].iloc[i]=df['Price'].iloc[i][1::]
              i=i+1
```



```
In [19]:  df.rename(columns = {'Price':'Price_in_dollar'}, inplace = True)   #Renaming the price column
          df['Price_in_dollar'] = df['Price_in_dollar'].astype(float)  #convert price column to int type
```

Now we have cleaned up all the data and we check how the final dataset looks like:



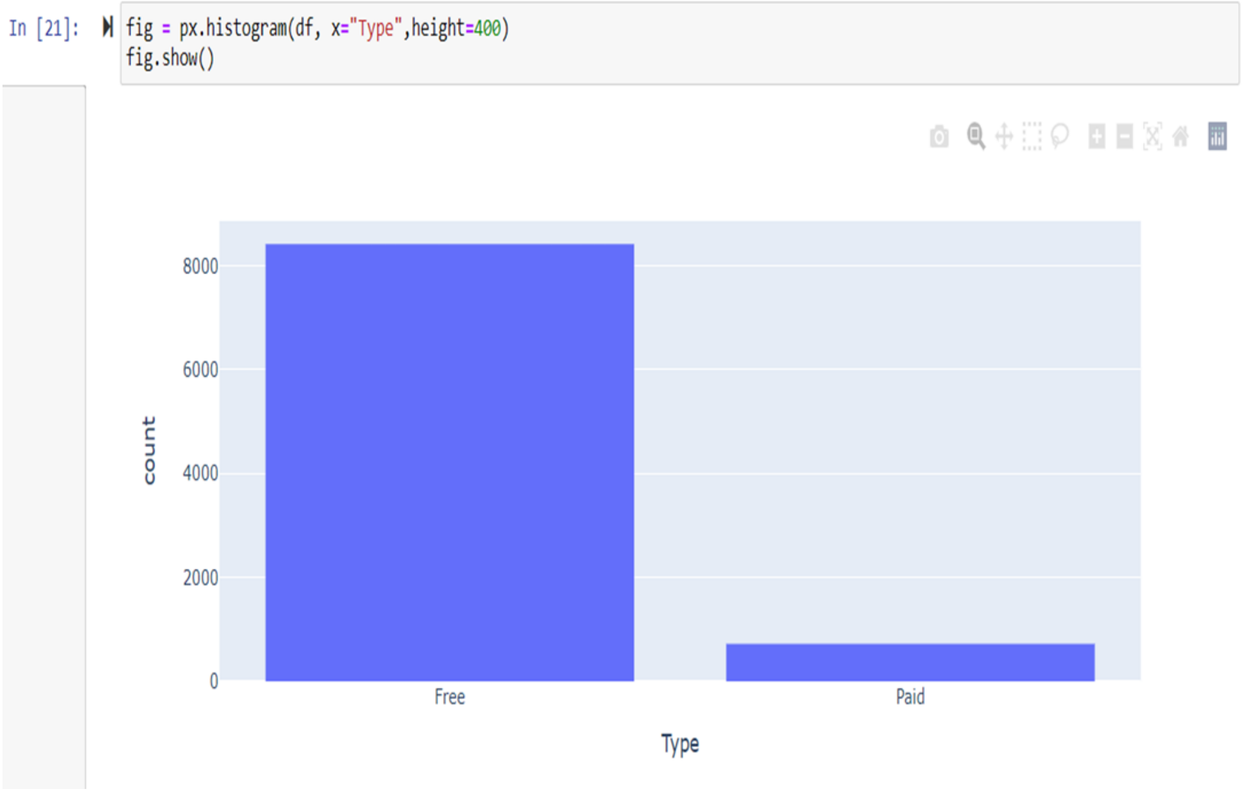
```
In [20]:  df.head()
```

Out[20]:

	App	Category	Rating	Reviews	Installs	Type	Price_in_dollar	Content Rating	Genres	Last Updated	size_in_kb
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	10000	Free	0.0	Everyone	Art & Design	January 7, 2018	19456.0
1	Coloring book moana	ART_AND_DESIGN	3.9	967	500000	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	14336.0
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	5000000	Free	0.0	Everyone	Art & Design	August 1, 2018	8908.8
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	50000000	Free	0.0	Teen	Art & Design	June 8, 2018	25600.0
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	100000	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	2867.2

Performing Statistics

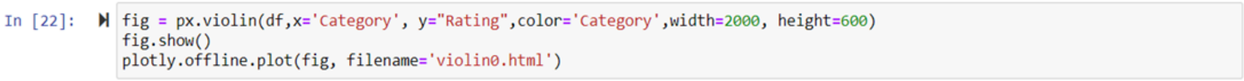
1. Number of Paid Vs Free Apps in dataset



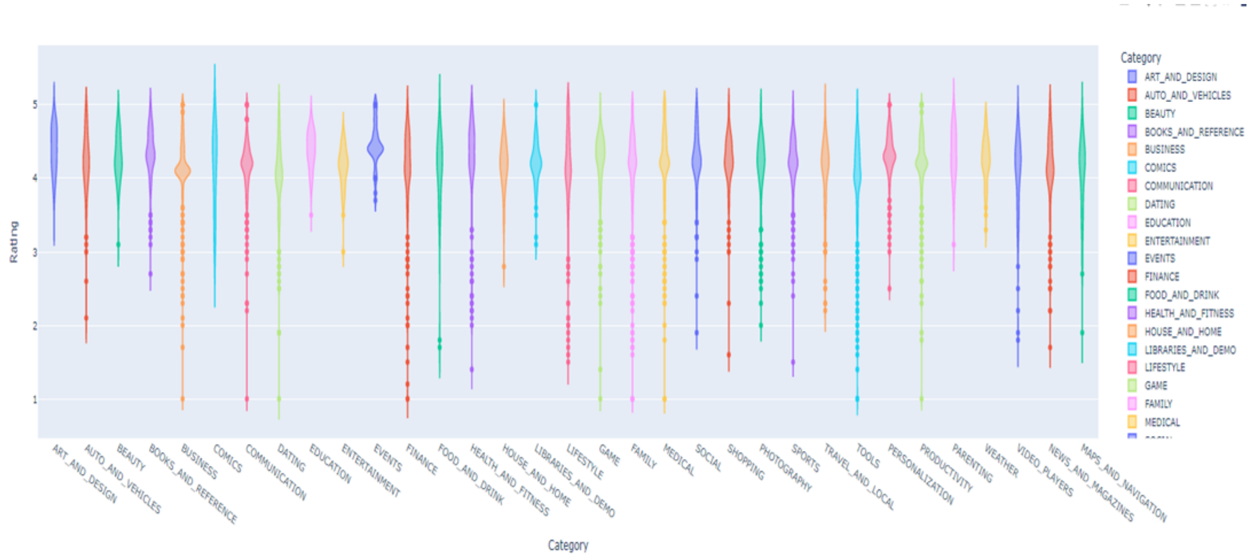
```
In [21]:  fig = px.histogram(df, x="Type",height=400)
          fig.show()
```

From the above graph we can conclude that free apps are more than Paid apps on play store.

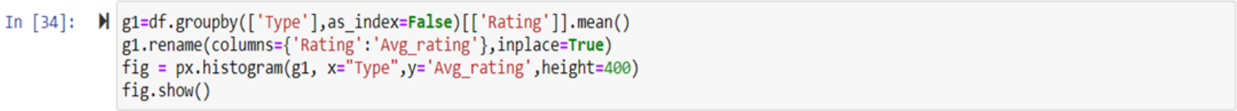
2. Distribution of Ratings in various categories



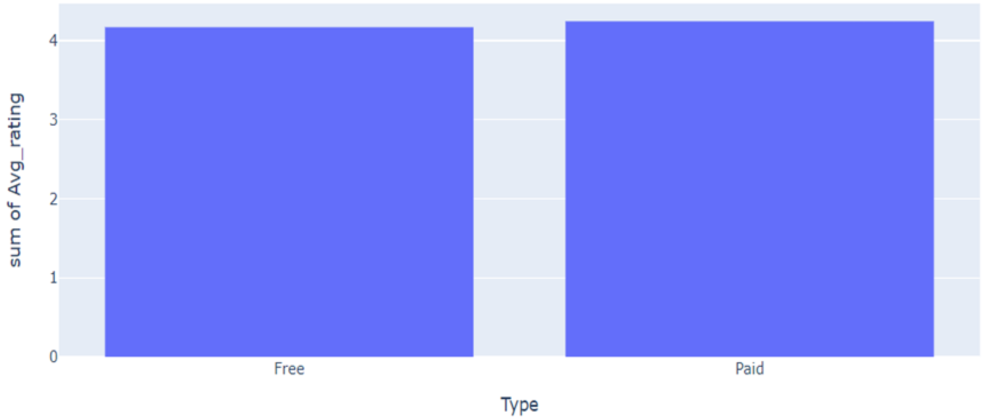
```
In [22]:  fig = px.violin(df,x='Category', y="Rating",color='Category',width=2000, height=600)
          fig.show()
          plotly.offline.plot(fig, filename='violin0.html')
```



3. Rating of Paid and free apps



```
In [34]:  g1=df.groupby(['Type'],as_index=False)[['Rating']].mean()
          g1.rename(columns={'Rating':'Avg_rating'},inplace=True)
          fig = px.histogram(g1, x="Type",y='Avg_rating',height=400)
          fig.show()
```

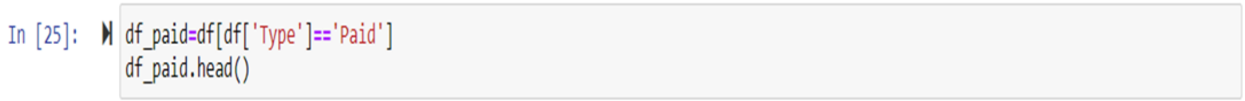


The above graph shows that the rating of paid apps is higher than free apps.

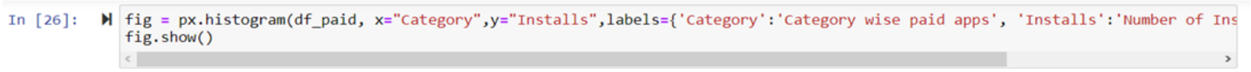
Rating of free app=4.1679

Rating of paid apps=4.2424

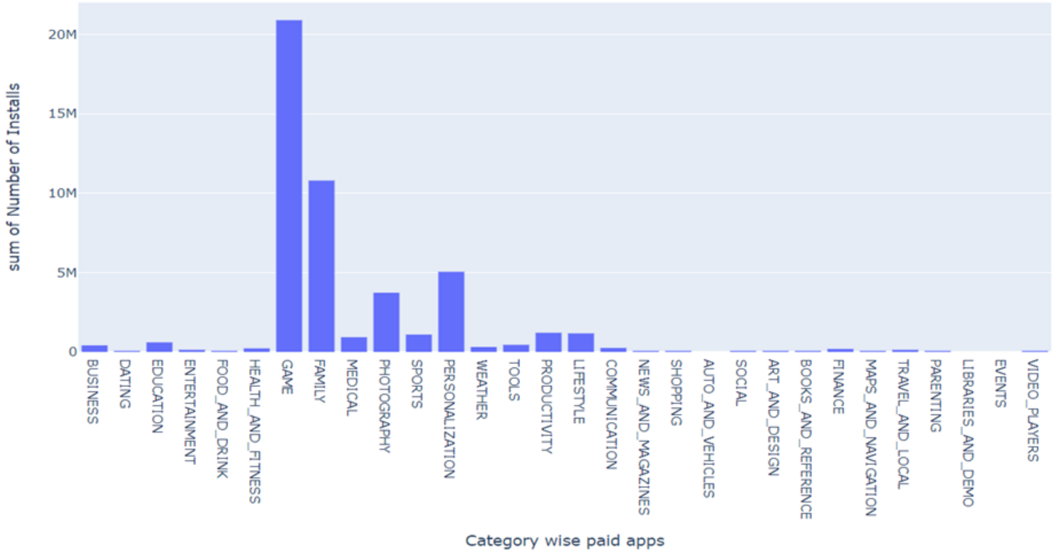
4. Analysing paid apps



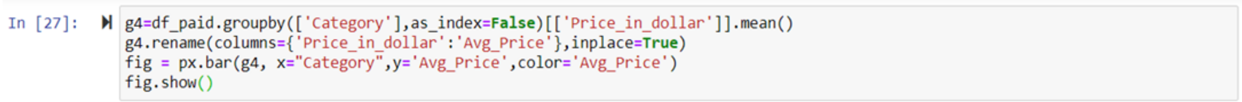
```
In [25]:  df_paid=df[df['Type']=='Paid']
          df_paid.head()
```



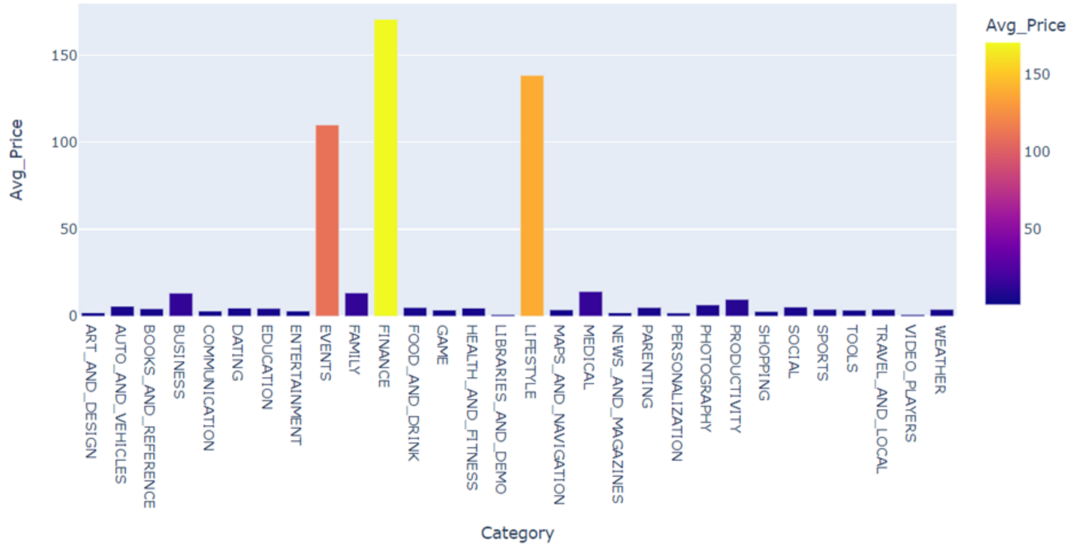
```
In [26]:  fig = px.histogram(df_paid, x="Category",y="Installs",labels={'Category':'Category wise paid apps', 'Installs':'Number of Ins
          fig.show()
```



From the above graph we can conclude that Games are installed more than other paid categories of apps from play-store.

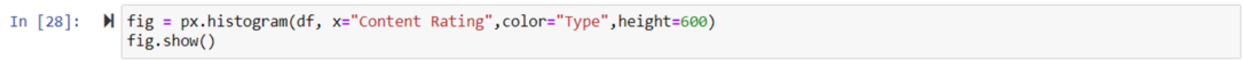


```
In [27]:  g4=df_paid.groupby(['Category'],as_index=False)[['Price_in_dollar']].mean()
          g4.rename(columns={'Price_in_dollar':'Avg_Price'},inplace=True)
          fig = px.bar(g4, x="Category",y='Avg_Price',color='Avg_Price')
          fig.show()
```

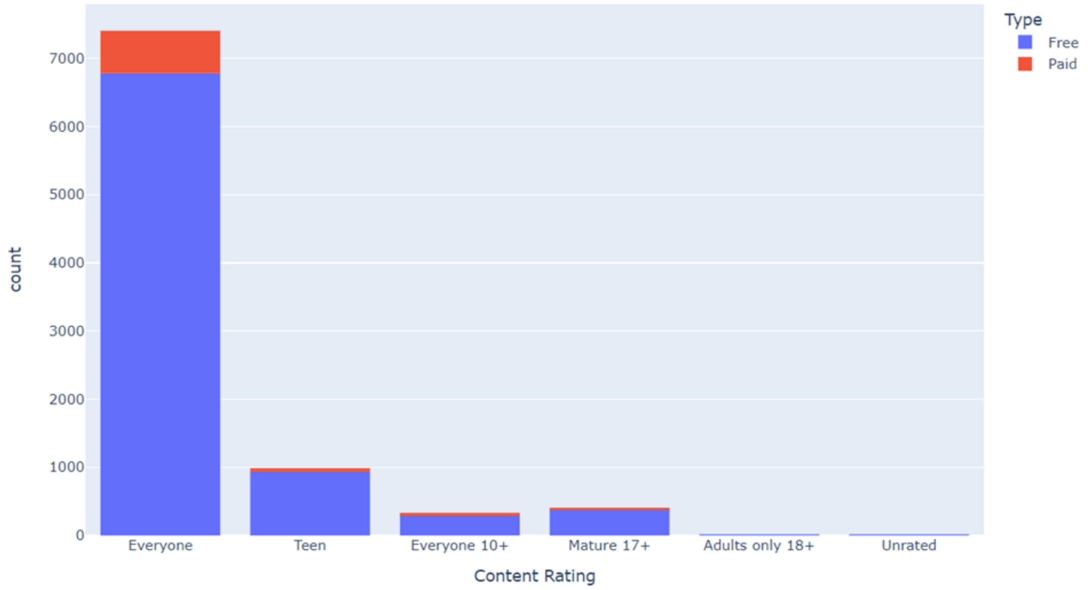


From the above bar plot we can conclude that finance apps have the highest average price.

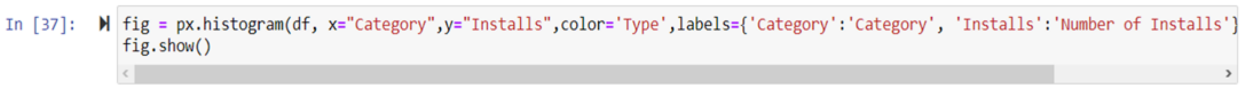
5. Comparing paid and free apps on the basis of content rating



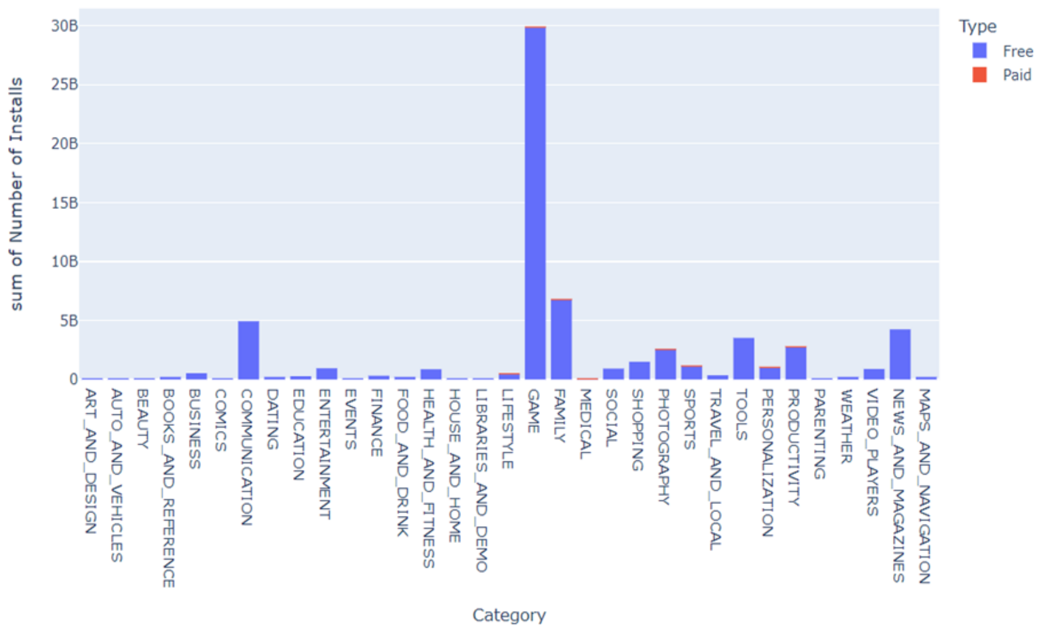
```
In [28]:  fig = px.histogram(df, x="Content Rating",color="Type",height=600)
          fig.show()
```



6. Determining number of instalments of various categories of app

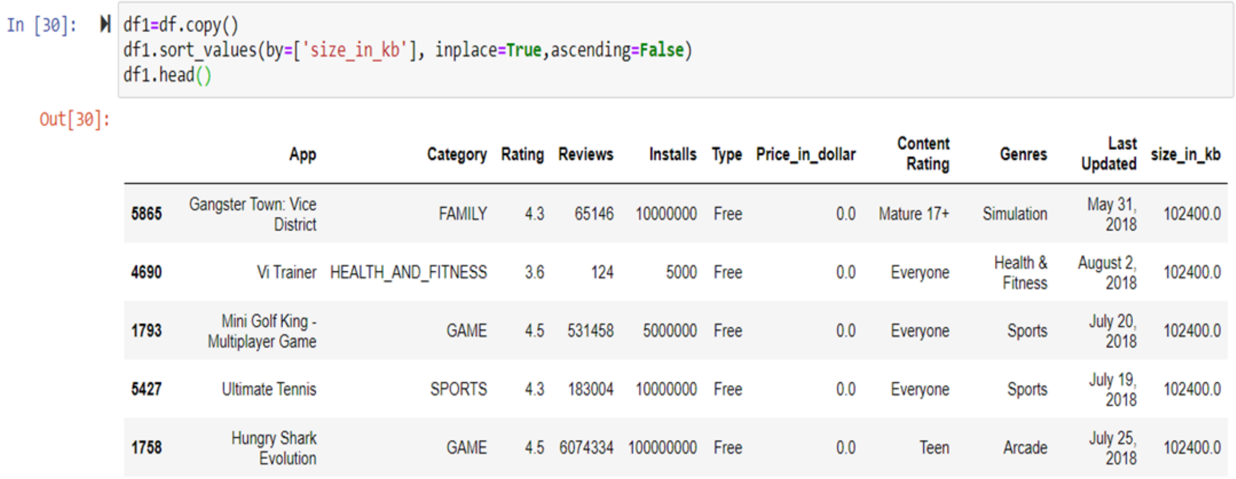


```
In [37]:  fig = px.histogram(df, x="Category",y="Installs",color='Type',labels={'Category':'Category', 'Installs':'Number of Installs'}
          fig.show()
```



From the plot we can say that the Game category has the highest number of installs so it is the most popular category.

7. Largest app in terms of size on play-store

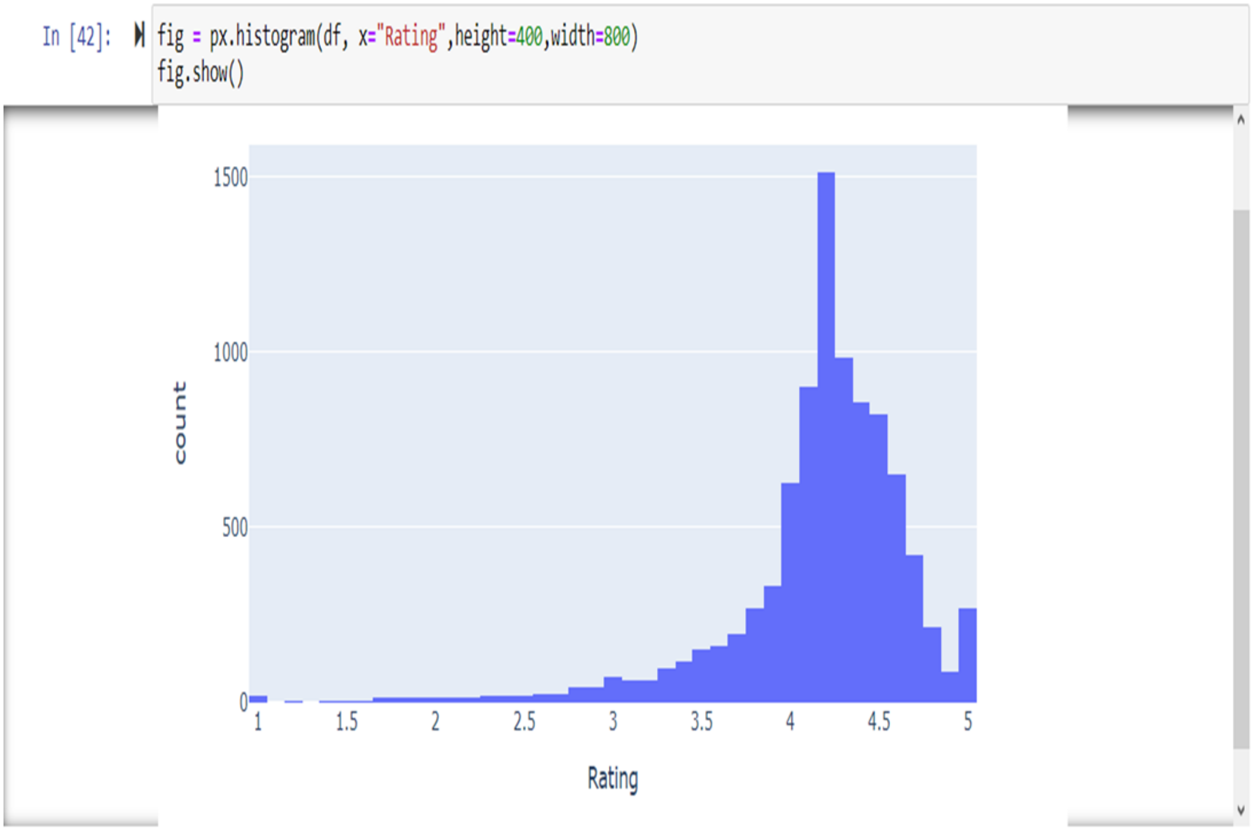


```
In [30]:  df1=df.copy()
          df1.sort_values(by=['size_in_kb'], inplace=True,ascending=False)
          df1.head()
```

Out[30]:

	App	Category	Rating	Reviews	Installs	Type	Price_in_dollar	Content Rating	Genres	Last Updated	size_in_kb
5865	Gangster Town: Vice District	FAMILY	4.3	65146	10000000	Free	0.0	Mature 17+	Simulation	May 31, 2018	102400.0
4690	Vi Trainer	HEALTH_AND_FITNESS	3.6	124	5000	Free	0.0	Everyone	Health & Fitness	August 2, 2018	102400.0
1793	Mini Golf King - Multiplayer Game	GAME	4.5	531458	5000000	Free	0.0	Everyone	Sports	July 20, 2018	102400.0
5427	Ultimate Tennis	SPORTS	4.3	183004	10000000	Free	0.0	Everyone	Sports	July 19, 2018	102400.0
1758	Hungry Shark Evolution	GAME	4.5	6074334	100000000	Free	0.0	Teen	Arcade	July 25, 2018	102400.0

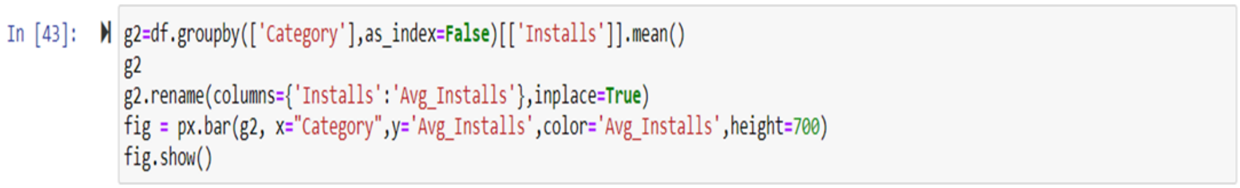
8. Distribution of Ratings given by people



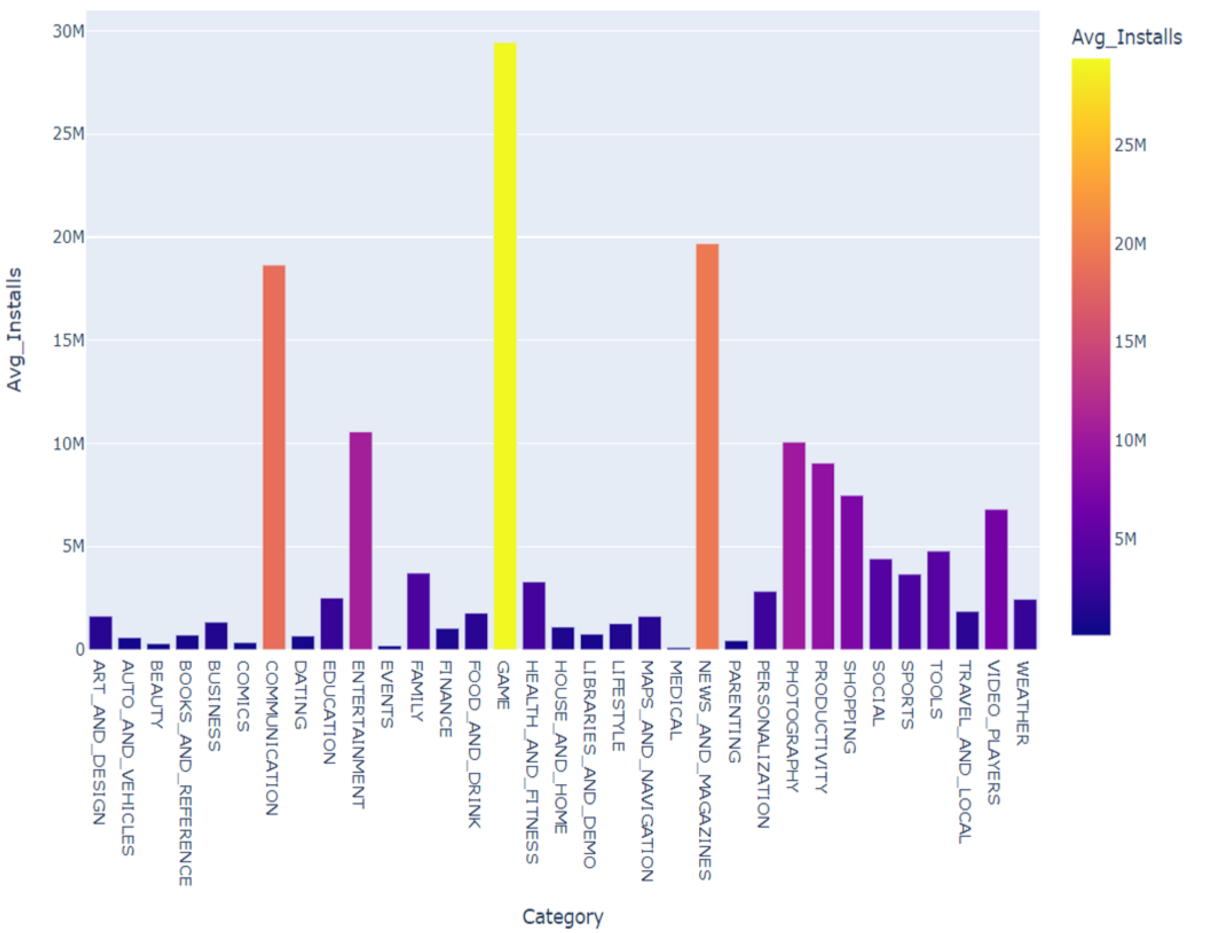
```
In [42]:  fig = px.histogram(df, x="Rating",height=400,width=800)
          fig.show()
```

The above graph shows that most people give rating between 4 and 5.

9. Average number of Installs in Various Categories

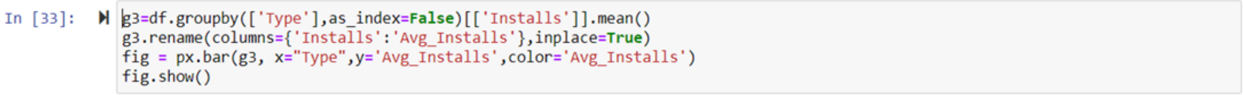


```
In [43]:  g2=df.groupby(['Category'],as_index=False)[['Installs']].mean()
          g2
          g2.rename(columns={'Installs':'Avg_Installs'},inplace=True)
          fig = px.bar(g2, x="Category",y='Avg_Installs',color='Avg_Installs',height=700)
          fig.show()
```

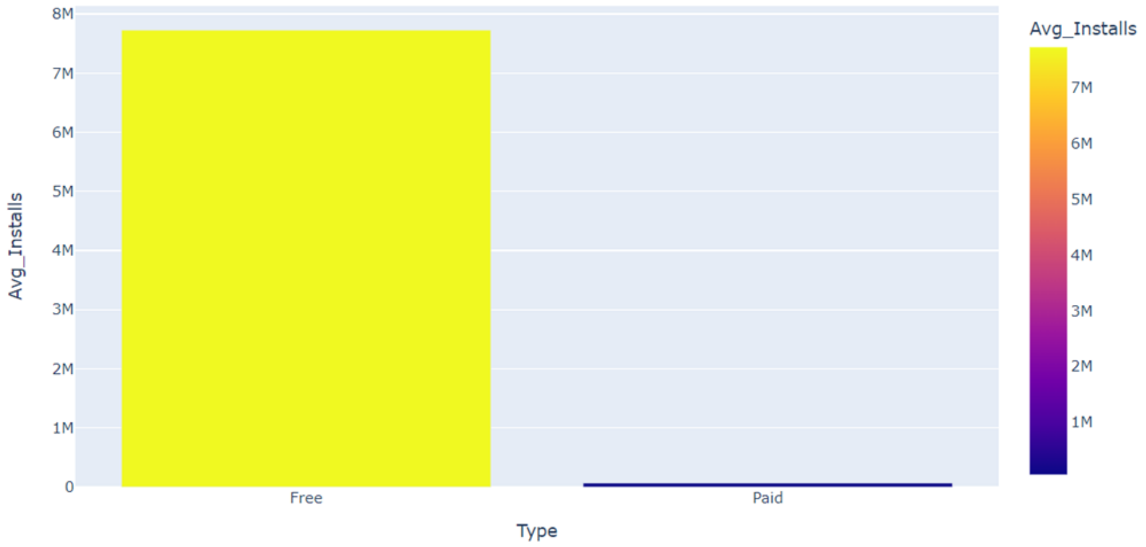


The above graph shows that Game apps are most installed on the play store.

10. Comparing paid and free apps on the basis of average number of instalments



```
In [33]:  g3=df.groupby(['Type'],as_index=False)[['Installs']].mean()
          g3.rename(columns={'Installs':'Avg_Installs'},inplace=True)
          fig = px.bar(g3, x="Type",y='Avg_Installs',color='Avg_Installs')
          fig.show()
```



From the bar plot we can say that the Average Installs for free apps is higher than that of paid apps.

Project Code Link .:

https://github.com/amitgupta20/PlayStore-DataAnalysis

REFERENCES

1.https://www.kaggle.com

2.https://www.coursera.org/learn/getting-started-in-google-analytics/home/welcome

3.https://www.coursera.org/learn/introduction-to-data-analytics/home/welcome

4.https://www.codecademy.com/paths/data-science/tracks/dscp-getting-started-with-data-science/modules/dscp-introduction-to-data-science/lessons/intro

5.https://matplotlib.org/

6. https://pandas.pydata.org/